MISP User Guide

MISP Intelligence Sharing Platform

The MISP Project & Contributors

Contents

1 Introduction . . . 19
1.1 Acknowledgement . . . 19
1.2 Contributing . . . 19
1.3 Format . . . 20
1.4 License . . . 20

I Getting Started

2 Book Convention . . . 23

3 Convention Used in This Book . . . 25
3.1 The MISP name . . . 25
3.2 Language . . . 25
3.3 Code of Conduct . . . 25
3.4 Example install . . . 25
3.5 MISP Instance . . . 25
3.6 Example Organisations . . . 25
3.7 Example IOCs . . . 26

4 General Concepts . . . 27
4.0.1 Admins and Site Admins . . . 27
4.0.2 Background Jobs . . . 27
4.0.3 Correlation . . . 27
4.0.4 Distribution levels . . . 27
4.0.5 MISP Instance . . . 27
4.0.6 Pivot path . . . 27
4.0.7 Pivoting . . . 28
4.0.8 Proposals . . . 28
4.0.9 Publishing . . . 28
4.0.10 Pull . . . 28
4.0.11 Push . . . 28
4.0.12 Scheduled Tasks . . . 28
4.0.13 Sync User . . . 28
4.0.14 Synchronisation . . . 28
4.0.15 Tagging . . . 28
4.0.16 Templating . . . 29

5 Requirements . . . 31

6 MISP Instance requirements . . . 33
6.1 Intro . . . 33

6.2 **Software requirements** . . . 33
6.3 **The biggie** . . . 33

7 **Sizing your MISP instance** . . . 35
7.1 **Database** . . . 35
7.2 **Feed caching** . . . 35

8 **Get Your Own MISP Instance** . . . 37
8.1 **Choosing a deployment method** . . . 37
8.2 **Evaluation virtual machine** . . . 37
8.2.1 Download . . . 37
8.2.2 Import the appliance . . . 37
8.2.3 Credentials . . . 38
8.2.4 Networking . . . 38
8.2.5 Setting the base URL . . . 39

9 **Quick Start** . . . 41

10 **Quick Start** . . . 43
10.1 **Login into MISP** . . . 43
10.2 **I forgot my admin password :(** . . . 43
10.3 **Tasks to do after first Start** . . . 44
10.4 **Password Policy** . . . 44
10.5 **tl;dr** . . . 45
10.5.1 Create an Event . . . 45
10.5.2 Browse Past Events . . . 46
10.5.3 Export Events for logsearches . . . 47
10.6 **Create an Event** . . . 48
10.7 **Describe Event** . . . 49
10.7.1 Free-Text Import Tool . . . 49
10.7.2 Tags and Taglist . . . 51
10.7.3 Local tags . . . 53
10.7.4 Suggestions . . . 56
10.8 **Browsing Events** . . . 56
10.9 **Export Events for Log Search** . . . 56
10.10 **Enable a Taxonomy** . . . 57
10.11 **Enable and fetch a feed** . . . 58

11 **General Layout** . . . 59

12 **General Layout** . . . 61
12.1 **The top bar** . . . 61
12.1.1 Simple User . . . 61
12.1.2 Admin User . . . 61
12.1.3 Content of each of the top menu bar drop-down menus . . . 62
12.2 **The left bar** . . . 73

II Using MISP

13 User Management and Global Actions . . . 77
13.1 First run of the system . . . 77
13.2 Your profile . . . 77
13.3 API keys . . . 78
13.4 Multi-factor authentication . . . 78
13.5 Personal settings . . . 78
13.6 User administration and roles . . . 78

14 Using the System . . . 79
14.1 The event lifecycle . . . 79
14.2 Creating an event . . . 79
14.3 Populating an event with data . . . 80
14.3.1 Add Attribute . . . 80
14.3.2 Add Object . . . 81
14.3.3 Add Attachment . . . 81
14.3.4 Populate from... . . . 82
14.4 Adding context to your data . . . 83
14.4.1 Tagging: taxonomies, galaxies and tag collections . . . 83
14.4.2 Analyst data: notes, opinions and relationships . . . 84
14.4.3 Enrichment . . . 84
14.4.4 Decaying scores . . . 84
14.5 The event view . . . 84
14.5.1 General event information . . . 85
14.5.2 Content toggles . . . 86
14.5.3 The attribute table . . . 86
14.5.4 Pivoting and related events . . . 87
14.5.5 Event discussion . . . 87
14.6 Collaborating on events . . . 87
14.6.1 Proposing changes to another organisation's event . . . 87
14.6.2 Sightings . . . 87
14.6.3 Contacting the reporter . . . 87
14.6.4 Delegating an event . . . 88
14.7 Publishing an event . . . 88
14.8 Extending events . . . 89
14.9 Browsing and searching . . . 89
14.9.1 Listing events . . . 89
14.9.2 Listing and searching attributes . . . 89
14.10 Editing events and attributes . . . 90
14.11 Create and manage Sharing Groups . . . 90
14.12 Exporting data . . . 93
14.13 Connecting to other instances . . . 93
14.14 REST API . . . 93
14.15 Multi-factor authentication (TOTP/HOTP) . . . 93
14.15.1 Generating TOTP/HOTP tokens . . . 93
14.15.2 Logging in with TOTP/HOTP . . . 95

14.15.3 Deleting and regenerating tokens . . . 96
14.15.4 Combining multiple factors . . . 96

15 MISP Extended Events . . . 97
15.1 When to use an extended event . . . 97
15.2 Creating an extended event . . . 97
15.3 Viewing extended events . . . 97
15.4 Finding extended events . . . 97
15.5 Further reading . . . 98

16 Create an Event Based on a Report . . . 99
16.0.1 Adding an event . . . 99
16.0.2 Adding Attributes . . . 103
16.0.3 Freetext Import Tool . . . 109
16.0.4 Batch Import . . . 112
16.0.5 Modify the event . . . 113

17 Event Reports . . . 115
17.1 Writing a report . . . 115
17.1.1 Referencing MISP elements inline . . . 115
17.2 Creating a report . . . 115
17.2.1 Import from URL . . . 116
17.2.2 Generate report from Event . . . 116
17.3 Extracting indicators from a report . . . 116
17.4 Template variables . . . 116
17.5 Exporting a report . . . 116
17.6 Distribution, visibility and sync . . . 117
17.7 AI-assisted extraction (experimental) . . . 117
17.8 Permissions and API . . . 117

18 Analyst Data . . . 119
18.1 The three types . . . 119
18.1.1 Notes . . . 119
18.1.2 Opinions . . . 119
18.1.3 Relationships . . . 120
18.2 What can be annotated . . . 120
18.3 Working with analyst data in the UI . . . 120
18.4 Distribution and sharing . . . 120
18.5 Collaboration and synchronisation . . . 121
18.6 Analyst Data Blocklist . . . 121
18.7 API and permissions . . . 121

19 Sightings . . . 123
19.0.1 Explanation . . . 123
19.0.2 Using sightings on an event (GUI) . . . 123
19.0.3 Sighting policies and privacy . . . 126
19.0.4 SightingDB . . . 126
19.0.5 Real-time sharing and blocklists . . . 126
19.0.6 Using sightings on an event (API) . . . 126

20 Decaying Models . . . 127
20.1 How scoring works . . . 127
20.1.1 The base score . . . 127
20.2 Model parameters . . . 127
20.3 Decay formulas . . . 128
20.4 Mapping models to attribute types . . . 128
20.5 Simulating and previewing a model . . . 128
20.6 Using decay scores in queries and the event view . . . 128
20.7 Ownership, sharing and default models . . . 129
20.7.1 Import, export and updating defaults . . . 129

21 Delegation . . . 131
21.0.1 Send a delegation request . . . 131
21.0.2 Answer a delegation request . . . 136

22 Best Practices . . . 139
22.1 Tagging . . . 139
22.1.1 Tagging at event level versus tagging at attribute level . . . 139
22.1.2 Minimal subset of tags to use for each event . . . 140
22.2 Setting distribution . . . 140
22.3 Use objects instead of combined attributes . . . 144
22.4 Use proposals when possible to request author to make changes to an event . . . 144
22.5 Make use of the 2.5 collaboration and lifecycle features . . . 144

III Data Model & Context

23 Categories and Types . . . 149
23.0.1 Attribute Categories vs. Types . . . 149
23.0.2 Categories . . . 177
23.0.3 Types . . . 178

24 Taxonomies . . . 183
24.1 Contributing to Taxonomy . . . 185
24.2 Reserved Taxonomy . . . 186
24.3 Adding Taxonomy in MISP . . . 186
24.4 Adding a private taxonomy . . . 187
24.5 How to use Taxonomy in MISP . . . 187
24.5.1 Filtering the distribution of events among MISP instances . . . 187
24.5.2 MISP Taxonomies - tools . . . 187
24.5.3 Other use cases using MISP taxonomies . . . 188
24.6 More options to configure taxonomies. . . . 188
24.6.1 Setting a taxonomy as “Highlighted” . . . 188
24.6.2 Setting a taxonomy as “Required” . . . 189
24.6.3 Exclusive taxonomies . . . 189
24.6.4 Bulk actions and normalising custom tags . . . 190
24.7 Future functionalities related to MISP taxonomies . . . 190

25 Galaxies . . . 191
25.1 Managing galaxies in the UI . . . 191
25.1.1 Enabling, disabling and updating galaxies . . . 194
25.1.2 Matrix galaxies . . . 194
25.2 Using galaxies in events and attributes . . . 195
25.3 Creating your own galaxy clusters (Galaxy 2.0) . . . 196
25.3.1 Publishing custom clusters . . . 197
25.4 Forking a cluster . . . 197
25.5 Cluster relationships . . . 197
25.6 Blocklisting galaxy clusters . . . 197
25.7 Authoring galaxies as JSON . . . 197
25.8 Available galaxies . . . 198

26 MISP Objects . . . 199
26.1 Using objects . . . 199
26.2 Object references and relationships . . . 203
26.3 Editing objects . . . 203
26.4 Object templates . . . 203
26.4.1 Creating object templates . . . 203

27 Warning Lists . . . 205
27.1 Warning Lists and False Positives . . . 205
27.2 Usage . . . 205
27.2.1 Warninglists and data export . . . 205
27.2.2 Check individual values for warning list hits . . . 206
27.2.3 Updating warninglists . . . 206
27.2.4 How matching works: types and categories . . . 206
27.2.5 Managing warninglists from the GUI . . . 206
27.2.6 Creating a custom warninglist on disk . . . 207

28 Notice Lists . . . 209
28.1 How notices are triggered . . . 209
28.2 Managing notice lists . . . 209

IV Sharing & Synchronisation

29 Synchronization and Sharing . . . 213

30 Sharing / Synchronisation . . . 215

31 Synchronisation . . . 217
31.1 Concept . . . 217
31.2 Pull vs Push synchronisation . . . 218
31.2.1 Push . . . 218
31.2.2 Pull . . . 219
31.2.3 Benefits of using push and pull in conjunction . . . 219
31.3 Adding a server . . . 219

31.3.1 Synchronising analyst data and collections (2.5) . . . 220
31.4 Test connection . . . 221
31.5 Rules . . . 221
31.6 Troubleshooting . . . 222
31.7 Sharing and distribution . . . 222
31.7.1 Distribution settings . . . 222
31.7.2 Community . . . 223
31.7.3 Sharing-groups . . . 223
31.7.4 Distribution mechanisms . . . 223
31.8 Collaboration . . . 226
31.8.1 Proposals . . . 226
31.8.2 Forums / Threats . . . 227
31.8.3 Comments to events . . . 228
31.8.4 Contact a reporter . . . 228
31.8.5 Receive alerts . . . 229
31.9 Cerebrate . . . 230
31.10 Sharing group blueprints . . . 230
31.11 TAXII 2.x servers . . . 231
31.12 Recommendation . . . 231
31.12.1 MISP Staging System . . . 231
31.12.2 MISP SECOps System . . . 231

32 Managing Feeds . . . 233

33 Feeds . . . 235
33.1 Managing feeds . . . 235
33.1.1 Default feeds . . . 236
33.1.2 Caching feeds . . . 237
33.1.3 Fetching feeds . . . 237
33.1.4 Search feed caches . . . 237
33.1.5 Adding feeds . . . 237
33.1.6 Importing feeds . . . 244
33.1.7 Analyze feed overlap . . . 244
33.1.8 Exporting feeds . . . 245
33.2 Feed correlation . . . 245

34 Collections . . . 247
34.1 What a collection contains . . . 247
34.2 Creating and populating a collection . . . 247
34.3 Distribution and sharing . . . 248
34.4 Synchronising collections between instances . . . 248
34.5 API . . . 248

35 Cryptographic Keys & Event Signing . . . 251
35.1 How it works . . . 251
35.1.1 The instance signing key . . . 251
35.2 Protecting an event . . . 251
35.3 Verification on synchronisation . . . 252

35.4 Permissions and the server-signing endpoint . . . 252
35.5 API . . . 252

V Automation & Integration

36 Automation and MISP API . . . 257
36.1 The MISP REST API . . . 257
36.2 General . . . 257
36.2.1 Automation URL . . . 257
36.2.2 Automation key . . . 257
36.2.3 Accept and Content-Type headers . . . 260
36.3 Automation using PyMISP . . . 260
36.4 Status Codes . . . 260
36.5 Error Handling . . . 261
36.5.1 Wrong endpoint chosen . . . 261
36.6 Search . . . 261
36.7 Events management . . . 262
36.7.1 /events . . . 262
36.7.2 GET /events . . . 262
36.7.3 POST /events . . . 263
36.7.4 DELETE /events . . . 263
36.7.5 GET /events/index . . . 263
36.7.6 POST /events/AddTag . . . 263
36.7.7 POST /events/removeTag . . . 264
36.7.8 GET /events/pushEventToZMQ/ . . . 264
36.7.9 GET /events/nids NIDS rules export . . . 265
36.7.10 GET /events/hids Hash - HIDS database export . . . 265
36.7.11 GET /events/stix STIX export . . . 266
36.8 Tag management . . . 267
36.8.1 POST /tags/add . . . 267
36.8.2 POST /tags/attachTagToObject . . . 267
36.8.3 POST /tags/removeTagFromObject . . . 268
36.8.4 GET /tags/tagStatistics/ . . . 269
36.9 Attribute management . . . 269
36.9.1 POST /attributes/add/ . . . 269
36.9.2 GET /attributes . . . 270
36.9.3 POST /attributes/delete/ . . . 270
36.9.4 POST /attributes/addTag . . . 271
36.9.5 POST /attributes/removeTag . . . 271
36.9.6 GET /attributes/attributeStatistics . . . 272
36.9.7 GET /attributes/describeTypes Describe types API . . . 272
36.9.8 POST /attributes/restSearch . . . 272
36.10 Objects management . . . 277
36.10.1 POST /objects/delete/[object_id]/[hard_delete] . . . 277
36.11 Server management . . . 278
36.11.1 GET /servers/getVersion . . . 278
36.12 Sightings . . . 278
36.12.1 POST /sightings/add/ . . . 278
36.13 User management . . . 278
36.13.1 POST /admin/users/add . . . 279

36.13.2 POST admin/users/edit/ . . . 279
36.13.3 POST admin/users/delete/ . . . 279
36.13.4 GET admin/users . . . 280
36.13.5 GET admin/users/view/ . . . 280
36.13.6 POST admin/users/add/ . . . 281
36.14 Discussion API . . . 281
36.15 Organisation management . . . 281
36.16 Special Cases . . . 283
36.16.1 XML Export . . . 283
36.16.2 CSV export . . . 284
36.17 RPZ export . . . 285
36.18 Text export . . . 286
36.19 RESTful searches with XML result export . . . 287
36.20 Export attributes of event with specified type as XML . . . 288
36.21 Filtering event metadata . . . 289
36.22 Download attachment or malware sample . . . 289
36.23 Download malware sample by hash . . . 289
36.24 Upload malware samples using the "Upload Sample" API . . . 290
36.25 Proposals API . . . 290
36.26 Sharing groups . . . 291
36.27 Enable, disable and fetching feeds via the API . . . 291
36.28 Sightings API . . . 292
36.29 Warninglists API . . . 295
36.29.1 GET warninglists/index . . . 295
36.29.2 GET warninglists/view/1 . . . 296

37 Attribute statistics API . . . 297

38 Additional statistics . . . 299

39 MISP modules . . . 301
39.1 Description . . . 301
39.1.1 GET /modules/ . . . 301
39.1.2 POST /modules/queryEnrichment . . . 302

40 PyMISP - Python Library to Access MISP . . . 305
40.0.1 Capabilities . . . 305
40.0.2 Installation . . . 305
40.0.3 Getting started . . . 305
40.0.4 Using PyMISP . . . 306
40.0.5 Existing examples . . . 306
40.0.6 Going further . . . 308

41 ZeroMQ - MISP Publish/Subscribe . . . 311
41.1 MISP ZeroMQ . . . 311
41.1.1 MISP ZeroMQ configuration . . . 311
41.1.2 MISP ZeroMQ debugging and testing . . . 312
41.1.3 Testing with sub.py tool . . . 313

41.1.4 Notification Schemas . . . 314
41.1.5 Tips for Building a Subscriber . . . 321

42 Modules . . . 323
42.1 Module families . . . 323
42.2 Running the module server . . . 323
42.3 Enabling and configuring modules in MISP . . . 323
42.4 Using modules . . . 323
42.5 Module endpoints . . . 324

43 External Connectors . . . 325
43.1 How MISP connects to other tools . . . 325
43.2 Notable integrations . . . 325
43.3 Worked example: MISP to Microsoft Graph Security . . . 326
43.3.1 Prerequisites . . . 326
43.3.2 Getting started . . . 326
43.3.3 Key configuration options (`config.py`) . . . 326
43.3.4 Reading and troubleshooting . . . 326

VI Administration

44 Administration . . . 329
44.1 Users . . . 329
44.1.1 Adding a new user . . . 329
44.1.2 Listing all users . . . 331
44.1.3 Contacting a user . . . 333
44.2 Organisations . . . 334
44.2.1 Adding a new organisation . . . 334
44.2.2 Listing all organisations . . . 335
44.2.3 Merge organisations . . . 338
44.3 Roles . . . 338
44.3.1 Adding a new role . . . 340
44.3.2 Listing roles . . . 340
44.4 Tools . . . 341
44.5 Server settings and maintenance . . . 341
44.5.1 Worker types . . . 343
44.5.2 Restarting or starting the workers . . . 343
44.6 Blocklists and block rules . . . 343
44.6.1 Event blocklist . . . 344
44.6.2 Event block rules . . . 346
44.6.3 Organisation blocklist . . . 346
44.6.4 Sighting blocklist . . . 347
44.6.5 Galaxy cluster blocklist . . . 348
44.6.6 Analyst data blocklist . . . 348
44.7 Import Regexp . . . 348
44.7.1 The purpose of Import Regexp entries . . . 348
44.7.2 Adding and modifying entries . . . 348
44.8 Managing the Signature allowedlist . . . 348
44.8.1 Allowlisting an address . . . 348

44.8.2 Managing the list . . . 349
44.9 Managing correlations . . . 349
44.9.1 Correlation exclusions . . . 349
44.9.2 Correlation rules . . . 349
44.9.3 Over-correlating values . . . 349
44.9.4 Top correlations . . . 350
44.9.5 Correlation engine . . . 350
44.10 Logs . . . 350
44.10.1 Audit logs . . . 350
44.10.2 Access logs . . . 350
44.11 Background Processing . . . 351
44.11.1 Command line tools for the background workers . . . 351
44.11.2 Monitoring the Background Processes . . . 351
44.11.3 Scheduling recurring tasks . . . 352
44.12 Other 2.5 administration features . . . 353
44.12.1 Advanced authentication keys . . . 353
44.12.2 Decaying models . . . 353
44.12.3 Workflows . . . 353
44.12.4 Cerebrate . . . 353
44.12.5 TAXII servers . . . 353
44.12.6 Dashboards . . . 353
44.12.7 Benchmarking . . . 353
44.13 Various administration tips & tricks . . . 354
44.13.1 Setting a Publish Alert Filter . . . 354
44.13.2 Default sharing level . . . 355
44.13.3 Adding organisation logos . . . 356
44.13.4 The scheduler worker is not starting . . . 356
44.13.5 How to redirect HTTP to HTTPS . . . 356
44.13.6 Increase max size of Samples / other files . . . 356
44.13.7 Support & feature requests . . . 357
44.13.8 More information in the notification emails about new events . . . 357
44.13.9 Get top API users . . . 357
44.13.10 MISP Logs . . . 358
44.13.11 Logging of failed authentication attempts . . . 358
44.13.12 Clearing expired sessions . . . 358
44.13.13 Troubleshooting MISP not connecting to redis but redis-cli working . . . 359
44.13.14 Errors about fields or tables . . . 359
44.14 Jobs . . . 359
44.14.1 Scheduled Tasks . . . 360
44.15 MISP Backup . . . 360
44.16 MISP Restore . . . 360
44.17 Command line interface (CLI) commands . . . 360
44.17.1 Administering MISP via the CLI . . . 360
44.17.2 Automating certain console tasks . . . 363
44.17.3 Managing the background workers . . . 363
44.18 Administration of TOTP/HOTP . . . 364
44.18.1 Mandating TOTP/HOTP usage . . . 365
44.18.2 Transitioning from e-mail OTP to TOTP/HOTP . . . 365
44.18.3 How to use TOTP/HOTP . . . 365

45 Workflows & Blueprints . . . 367
45.1 Concepts . . . 367

45.1.1 Blocking vs non-blocking . . . 367
45.2 Enabling workflows . . . 368
45.3 The workflow editor . . . 368
45.3.1 Modules must be enabled . . . 368
45.4 Ad-hoc workflows . . . 368
45.5 Workflow Blueprints . . . 369
45.6 Debugging and logs . . . 369
45.7 Permissions and execution context . . . 369
45.8 API and CLI . . . 369

46 Updating Python Dependencies . . . 371
46.1 Installation . . . 371
46.2 Updating MISP and its dependencies . . . 371
46.2.1 Updating MISP core . . . 371
46.2.2 Updating the python dependencies . . . 372

VII Reference & FAQ

47 Keyboard Shortcuts . . . 377
47.1 Default shortcuts . . . 377
47.2 Customising the shortcuts . . . 378

48 Translations - i18n & l10n . . . 379

49 MISP and internationalization (i18n) . . . 381
49.1 Translating MISP (for translators) . . . 381
49.1.1 Style . . . 381
49.2 Enabling a language on your instance . . . 381
49.3 Making MISP translatable (for developers) . . . 382
49.3.1 Use named placeholders, not `%s` . . . 382
49.3.2 Keep HTML out of translatable strings . . . 382
49.3.3 Wrap button labels and form labels . . . 382
49.4 Issues . . . 382
49.5 Translating the website or this book . . . 383
49.6 Reach out to the community . . . 383

50 FAQ . . . 385

51 Frequently Asked Questions . . . 387
51.1 General questions . . . 387
51.1.1 Where can I get support? . . . 387
51.1.2 What are the hardware requirements? . . . 387
51.1.3 How to monitor MISP? . . . 387
51.2 Specific questions . . . 387
51.2.1 Can I configure MISP encrypted notification emails to contain more information in the subject? . . . 387
51.2.2 Why are there malicious files in my MISP directory? . . . 388

51.2.3 How can I restart the workers? . . . 388
51.2.4 How can I redirect HTTP to HTTPs? . . . 388
51.2.5 When I try to access my new installation, I am redirected to localhost:8443 and get an error. . . . 389
51.2.6 How can I define the default sharing level? . . . 389
51.2.7 How can I add an organisation logo and/or foter logo? . . . 389
51.2.8 How can I import data directly from PDF reports? . . . 389
51.2.9 I have many failed jobs when doing email notification. What should I do? . . . 389
51.2.10 I have issues with pushing events . . . 390
51.2.11 I have many users or API accesses, what's the best PHP session handler? . . . 390
51.2.12 Is there TAXII support? . . . 390
51.2.13 Wipe MISP data - Remove all data . . . 390
51.2.14 Constantly acknowledging my self-signed certificate drives me nuts . . . 390
51.2.15 How can I change the theme? . . . 391
51.2.16 How can I deal with a MISP instance that has pulled in feeds over and over into new events, generating hundreds of GBs of junk correlations, rendering the instance unusable? . . . 391
51.2.17 I have a long list of events that I want to delete via the API, do I really have to loop through each and issue a delete to /events/delete? . . . 391
51.2.18 I can no longer log in. How do I reset the admin password? . . . 391
51.2.19 How do I set the baseurl from the command line? . . . 392
51.3 Usage questions . . . 392
51.3.1 How can I see all the deleted events in a MISP instance? . . . 392
51.4 Permission issues . . . 392
51.4.1 RHEL/CentOS . . . 393
51.4.2 Redis Connection problems . . . 393
51.5 RHEL/CentOS SELinux debug . . . 393
51.5.1 Clearing the audit logs . . . 394
51.6 When to update MISP? . . . 394
51.6.1 How to switch from tagged releases and back? . . . 394
51.7 OS Upgrades . . . 394
51.7.1 PHP . . . 395
51.7.2 Python3 . . . 395
51.7.3 Apache . . . 395
51.7.4 init/systemd . . . 395
51.7.5 MariaDB/MySQL/redis . . . 395
51.7.6 git . . . 395
51.8 Hardening . . . 395
51.8.1 How do I harden my MISP instance? . . . 395
51.9 Maintenance mode . . . 395
51.9.1 Is there a MISP maintenance mode? . . . 395
51.10 Update MISP fails . . . 396
51.10.1 What can go wrong if I update MISP? . . . 398
51.10.2 error: pathspec 'app/composer.json' did not match any file(s) known to git . . . 398
51.10.3 MISP modules "Connection refused" . . . 399
51.11 Uninstalling MISP . . . 399
51.12 Updating PyMISP to incorporate newer versions of the MISP object templates . . . 399
51.13 How to disable freetext/custom/user-created tags and only allow certain tags . . . 400
51.14 How to enable the csv import module? . . . 400

51.15 Why do I see 'The request has been black-holed' when I submit forms? . . 400
51.16 Importing large feeds creates PHP Fatal error . . . 400
51.17 I deleted the admin user by mistake . . . 400
51.18 config.php is not writeable . . . 401
51.19 How to debug misp-dashboard . . . 401
51.20 How to update object templates? . . . 401
51.21 What to do if my REST client is throwing SSL errors when trying to query my MISP instance? . . . 401
51.22 What to do if my REST client cannot reach the host, despite me being able to issue requests using Curl / Postman / etc.? . . . 401
51.23 How would one set up a sharing group with a remote org, where we only share a mutual community instance (i.e. we both have sync users on that instance). On our local instance, they exist as a remote org (from events that have synced from their instance via our shared community instance). . . . 402
51.24 Is it possible to propose objects to an event? . . . 402
51.25 How to use the enforceWarninglist parameter in REST search? . . . 402
51.26 Column not found issue . . . 402
51.27 WatchList Customization . . . 403
51.27.1 How to create a customized WatchList. . . . 403
51.28 How to add a galaxy to an event via PyMISP . . . 403
51.29 How to block attachment (and malware-sample) to be ingested . . . 403
51.30 What are the required steps after a MISP installation to have a properly running instance? . . . 403

52 Developer FAQ . . . 405
52.1 Main Developer Resources . . . 405
52.2 About the codebase . . . 405

53 User Stories . . . 407

54 User Personas . . . 417
54.1 Primary personas . . . 417
54.1.1 Farrah . . . 417
54.1.2 Adam . . . 418
54.2 Secondary personas . . . 419
54.2.1 Tina . . . 419
54.2.2 Henry . . . 420
54.2.3 Jacob . . . 421
54.2.4 Jay . . . 422
54.2.5 Sarah . . . 422
54.2.6 Priya . . . 423
54.2.7 Marco . . . 424
54.3 Other personas . . . 425
54.3.1 Malcolm . . . 425

Appendices . . . 427

A Appendices . . . 429
B Appendix A: External Authentication . . . 431
C Appendix B: ACL descriptors . . . 435
D Appendix C: Official MISP developments . . . 443
E Appendix D: Third-party development . . . 445
F Appendix E: Other Threat Intel Ressources . . . 449
G Appendix F: LDAP Authentication . . . 451
H Appendix G: SimpleBackgroundJobs Migration guide . . . 455
H.1 Install requirements . . . 455
H.2 MISP Config . . . 456
H.2.1 Notes . . . 456
I Glossary . . . 457
I.1 API . . . 457
I.2 RESTful . . . 457
I.3 Sharing groups . . . 457
I.4 Site admin . . . 457
I.5 Indicators . . . 457
I.6 IOC . . . 457
I.7 NIDS . . . 458
I.8 MISP . . . 458
I.9 MISP Attribute . . . 458
I.10 Observable . . . 458
I.11 MISP Event . . . 458
I.12 MISP Extended Events . . . 458
I.13 MISP feeds . . . 458
I.14 MISP format . . . 458
I.15 MISP Galaxy Cluster . . . 458
I.16 MISP modules . . . 459
I.17 MISP Instance . . . 459
I.18 MISP Objects . . . 459
I.19 MISP GnuPG Key . . . 459
I.20 MISP Sightings . . . 459
I.21 MISP Taxonomies . . . 459
I.22 Local tags . . . 459
I.23 MISP warninglists . . . 460
I.24 MISP noticelist . . . 460
I.25 Org Admin . . . 460

I.26 OSINT **460**
I.27 Pivot path **460**
I.28 Pivoting **460**
I.29 Proposals **460**
I.30 Publishing **460**
I.31 Pull **461**
I.32 Push **461**
I.33 Roles **461**
I.34 Scheduled Tasks **461**
I.35 Standard MISP Install **461**
I.36 Sync User **461**
I.37 Synchronisation **461**
I.38 Tagging **462**
I.39 Templating **462**
I.40 Analyst Data **462**
I.41 Event Report **462**
I.42 Decaying Model **462**
I.43 Workflow **462**
I.44 Cerebrate **462**
I.45 Collection **462**
I.46 Community **462**

1. Introduction



Figure 1.1: MISP logo

This guide introduces MISP, the MISP Intelligence Sharing Platform — an open source platform for sharing threat intelligence. It is intended for ICT professionals such as security analysts, incident responders, and malware reverse engineers who share threat intelligence with MISP or integrate MISP into other security monitoring tools. The guide covers day-to-day use of the MISP graphical user interface together with its automated interfaces (API) so that teams can integrate MISP into their security environment and operate one or more MISP instances. It is maintained against the current **MISP 2.5.x** release series; where behaviour differs from earlier versions this is called out in the relevant chapter.

1.1 Acknowledgement

The MISP user guide is a collaborative effort involving contributors to MISP, including:

- Belgian Ministry of Defence (CERT)
- CIRCL Computer Incident Response Center Luxembourg
- Iklody IT Solutions
- NATO NCIRC
- Cthulhu Solutions
- CERT-EU

and many other contributors, especially those who participated in the MISP hackathons.

1.2 Contributing

We welcome contributions to the MISP book. If you want to contribute, see our contributing guide.

1.3 Format

The MISP book is available in HTML, PDF, and ePub.

1.4 License

The MISP user guide is dual-licensed under GNU Affero General Public License version 3 and CC-BY-SA 4.0 international.

- Copyright (C) 2012 Christophe Vandeplas
- Copyright (C) 2012 Belgian Defence
- Copyright (C) 2012 NATO / NCIRC
- Copyright (C) 2013-2026 Andras Iklody
- Copyright (C) 2015-2026 Alexandre Dulaunoy
- Copyright (C) 2014-2026 CIRCL - Computer Incident Response Center Luxembourg
- Copyright (C) 2018 Camille Schneider
- Copyright (C) 2018-2026 Steve Clement
- Copyright (C) 2021 Jeroen Pinoy

I Getting Started

2 Book Convention 23

3 Convention Used in This Book 25
3.1 The MISP name 25
3.2 Language 25
3.3 Code of Conduct 25
3.4 Example install 25
3.5 MISP Instance 25
3.6 Example Organisations 25
3.7 Example IOCs 26

4 General Concepts 27

5 Requirements 31

6 MISP Instance requirements 33
6.1 Intro 33
6.2 Software requirements 33
6.3 The biggie 33

7 Sizing your MISP instance 35
7.1 Database 35
7.2 Feed caching 35

8 Get Your Own MISP Instance 37
8.1 Choosing a deployment method 37
8.2 Evaluation virtual machine 37

9 Quick Start 41

10 Quick Start 43
10.1 Login into MISP 43
10.2 I forgot my admin password :(. 43
10.3 Tasks to do after first Start 44
10.4 Password Policy 44
10.5 tl;dr 45
10.6 Create an Event 48
10.7 Describe Event 49
10.8 Browsing Events 56
10.9 Export Events for Log Search 56
10.10 Enable a Taxonomy 57
10.11 Enable and fetch a feed 58

11 General Layout 59

12 General Layout 61
12.1 The top bar 61
12.2 The left bar 73

2. Book Convention

3. Convention Used in This Book

`code block or value`

- Used for variable, function or menu names in MISP.

3.1 The MISP name

MISP is a recursive acronym: it stands for the **MISP Intelligence Sharing Platform**. The project was first named the *Malware Information Sharing Platform*, but as it grew to cover every kind of threat intelligence — not just malware — the name was redefined recursively, in the spirit of GNU ("GNU's Not Unix"). Throughout this book, *MISP* refers to the MISP Intelligence Sharing Platform.

3.2 Language

The language in this book is American English. All the screenshots and examples are in English.

3.3 Code of Conduct

The same code of conduct applies to this book as to the main MISP project. Because documentation often reflects the spirit of a software project, please treat the book with the same care and consideration. You can read the `Code of Conduct` here.

3.4 Example install

The examples and screenshots provided in this book have been created with the MISP Autogenerated VM.

To get a copy of the latest VM click here

3.5 MISP Instance

In general when talking about a network of inter-connected MISP servers, each server is a MISP instance. Whilst we have no strong feelings towards anyones naming schemes, as a rule of thumb try to have a scheme that makes everyday use easy when analysts need to talk about remote MISP instances.

The hostname used for the instance in this book is `misp.local` and we will henceforth refer to it either by name or as `local MISP instance`.

3.6 Example Organisations

As MISP is a platform to support information sharing, example organisations are often used within this book.

A set of users and organisations are used in the different examples.

The following two organisations are regularly used as example:

- Setec Astronomy with UUID `58d38339-7b24-4386-b4b4-4c0f950d210f`
- Acme Finance with UUID `58d38326-eda8-443a-9fa8-4e12950d210f`

Starting from MISP 2.4.71, the example organisations with the above mentioned UUID are **blocklisted** to avoid large distribution of sample events while testing a MISP instance. If you want to test your distribution, the sample organisation blocklisting can be removed in `Administration/Manage Org blocklists` (this menu is only available when the `MISP.enableOrgBlocklisting` setting is turned on).

3.7 Example IOCs

As with the example organisations, we want to make this book as useful as possible by using real life examples.

The following IOC examples have been used:

- Sirefef (aka ZeroAccess) Sample Event ID: #31337
- WannaCry Sample Event ID: #42
- Dridex Sample Event ID: #23

4. General Concepts

4.0.1 Admins and Site Admins

There are two types of admins in MISP: Admins (also referred to as org admins) and Site Admins. Whilst the former can only do some limited administration of users of his/her own organisation, site admins have access to all of the features and data of the system. They are in charge of making sure that the system runs correctly and the maintenance of MISP.

4.0.2 Background Jobs

A lot of the heavier tasks are a burden to users, in that their actions can cause long delays (and in some cases timeouts) while the application logic is executing. To alleviate this, long processes have been (if enabled) moved to background jobs, meaning that their execution happens asynchronously in the background, allowing the user to freely interact with the platform whilst the request is being processed. In MISP 2.5 this is the standard execution model, provided by the SimpleBackgroundJobs system (Supervisor + Redis).

4.0.3 Correlation

Correlation is the automatic linking of events that share the same piece of information. Whenever an attribute value that a user is allowed to see also appears on another event, MISP creates a correlation between the two, surfacing the relationship in the event view (and as a graph). This is what turns a collection of individual events into a navigable web of related intelligence and is the basis for pivoting. Correlation can be disabled per attribute (the "disable correlation" flag), suppressed for noisy values through correlation exclusions and correlation rules, and tuned by the correlation engine — see the Administration chapter.

4.0.4 Distribution levels

Every event, attribute, and object carries a distribution setting that controls how far it may spread when instances synchronise. The levels are: **Your organisation only** (never leaves your instance/organisation), **This community only** (stays on this instance and its direct connections but is not re-shared onwards), **Connected communities** (may be shared one hop further, to the communities your community connects to), **All communities** (freely shareable) and **Sharing group** (visible only to an explicitly enumerated set of organisations/instances). Attributes and objects can additionally **inherit** the distribution of their parent event. Distribution degrades by one step on each synchronisation hop, so information cannot travel further than its originator intended. Distribution is covered in depth in the Sharing chapter.

4.0.5 MISP Instance

A MISP instance is an installation of the MISP software and the connected database. All the data visible to the users is stored locally in the database and data that is shareable (based on the distribution settings) can be synchronised with other instances via the Sync actions. The instance that you are reading this manual on will be referred to as "this instance" or "your instance". The instances that your instance synchronises with will be referred to as "remote instances".

4.0.6 Pivot path

The (branching) path taken by a user from event to event while following correlation links. This is represented by the branching graph in the event view.

4.0.7 Pivoting

The act of navigating from event to event through correlation links.

4.0.8 Proposals

Each event can only be directly edited by users of the original creator organisation (and site admins). However, if another organisation would like to amend an event with extra information on an event, or if they'd like to correct a mistake in an attribute, they can create a Proposal. These proposals could then be accepted by the original creator organisation. These proposals can be pulled to another server, allowing users on connected instances to propose changes which then could be accepted by the original creators on another instance (and subsequently pushed back).

4.0.9 Publishing

When an event is first created by a user, it is visible to everyone on the instance based on the access rights ("Your organisation only" events will not be visible to users of other organisations), but they will not be synchronised and they won't be exportable. For this, a user with publishing permission of the organisation that created the event has to publish the event. The system will then inform all the users of the instance that are subscribing to e-mail notifications and who have access to view the published event via an e-mail.

4.0.10 Pull

Pulling is the process of using the configured sync user on a remote instance to REST GET all of the accessible data (based on the distribution rights) to your instance and store it.

4.0.11 Push

Pushing is the process of using a configured instance link to send an event or all accessible events (limited by the distribution rights) through the REST interface to a remote instance.

4.0.12 Scheduled Tasks

Certain common tasks can be scheduled for a later execution or for regular recurring executions. These tasks currently include caching all of the export formats, pulling from all eligible instances and pushing to all eligible instances.

4.0.13 Sync User

A user of a role that grants sync permissions, these users (and their authentication keys) are used to serve as the points of connection between instances. Events pushed to an instance are pushed to a sync user, who then creates the events on the remote instance. Events pulled are added by the sync user that is used to connect the remote instance to your instance. As an administrator, keep in mind that a sync user needs auth key and publish permissions, has to have undergone the mandatory password change and has to have accepted the Terms of Use in order for the sync to work. Please make sure that all of these steps are taken before attempting to push or pull.

4.0.14 Synchronisation

What we call synchronisation is an exchange of data between two (or more) MISP instances through our pull and push mechanisms.

4.0.15 Tagging

Users with tagging rights can assign various dynamically created tags to events, allowing an arbitrary link between events to be created. It is possible to filter events based on these tags and they can also be used to filter events for the automation.

4.0.16 Templating

Users with templating rights can create easy to fill forms that help with the event creation process.

5. Requirements

6. MISP Instance requirements

6.1 Intro

There are various ways you can run a MISP instance.

- Virtualized with docker/ansible/packer etc
- VMware/Virtualbox/Xen etc
- Dedicated hardware
- Road warrior setups
- Air-gapped setups

Whilst there is never an ultimate answer to what specifications a system needs, we try to give an approximate answer depending on your use case.

6.2 Software requirements

MISP is a LAMP-stack application. The core software requirements for MISP 2.5 are:

- **Operating system**: a current, supported Linux. The official installer scripts target **Ubuntu 24.04 LTS**, **Debian 12 / 13**, and **RHEL 9.4**.
- **PHP 8.x** (`>=8.1, <9.0`; the installers use **PHP 8.3**). The extensions `json`, `mbstring`, `xml`, `dom`, `simplexml`, `pcre` and `curl` are required, and `gd`, `openssl`, `redis`, `zip`, `intl`, `bcmath`, `ssdeep`, `zstd`, `brotli` and `apcu` are recommended.
- **MariaDB** (or MySQL) for the main database — SSD-backed storage is strongly recommended (see Database).
- **Redis** — required for caching (feeds, warninglists, exports) and for the background-job system.
- **Supervisor** — MISP 2.5 runs its background workers via **SimpleBackgroundJobs** on Supervisor (Redis-backed); the legacy CakeResque worker system has been removed.
- **Python 3** with a virtualenv for PyMISP and the helper scripts (`pymisp==2.5.34.1` is the pinned/recommended version), plus optionally **misp-modules** for enrichment/import/export.

The web server (Apache or nginx) and the exact package list are handled by the installer scripts; see the installation guides for the per-OS specifics.

6.3 The biggie

Having millions of events with millions of attributes (indicators) will eventually result in sub-par performance. Ideally you have millions of attributes and thousands of events. But this also depends on how you ingest the data. With millions of attributes a bottleneck could be the correlation engine. Especially if you have many duplicates in your events. (Use the feed matrix to see if feeds are massively overlapping)

7. Sizing your MISP instance

Sizing a MISP instance highly depends on how the instance will be used. The number of users, data ingested, data points used, number of events, number of correlations and API usage are all parameters which should be considered while sizing your instance.

From a hardware perspective, MISP's requirements are quite humble, a web server with 2+ cores and 8-16 GB of memory should be plenty, though more is always better, of course. A lot of it depends on the data set and the number of users you are dealing with.

Some considerations for what might affect your requirements:

- How highly correlating your data is (correlations are generally memory and computation intensive), if you have a high correlation ratio, consider either lowering this with better management of the data (correlate flag on attributes) or by increasing the memory and CPU available;
- Number of samples and attachments directly affect the disk usage;
- Concurrent user counts affect the memory usage and CPU utilisation, especially if you have a list of API users querying MISP frequently;
- Number of remote feeds and servers cached and kept in memory will also increase the memory requirements of the system;
- The amount of logging / activity / longevity of the server can increase the disk requirements both on the database as well as the local log file stash;

To give some indications of some of the operational servers:

- 16GB memory and 2 vcpus are quite common for smaller sharing hubs and end-point MISPs;
- large sharing communities (such as the CIRCL private sector community) use 128 GB of memory with 32 physical CPU cores on modern Xeon CPUs;
- A mid-sized sharing community can comfortably run on 8GB of memory with 4 vcpus while serving over a thousand users;
- The training instances we use, run on a meager 2GB of memory and a single vcpu (though we would not recommend using this for anything besides trainings / experimentation);

7.1 Database

The main database of MISP relies on MariaDB. Using SSDs is highly recommended to ensure a low latency on the I/O and ensure an efficient access to the database.

The type of storage used by MariaDB can also have an impact of the latency and disk space used.

7.2 Feed caching

Feed caching using RAM to store elements from the feeds enabled and cached. As an example, if you use the default available feeds, you can use up to 1.2Gb of memory if all feeds are enabled.

8. Get Your Own MISP Instance

This chapter helps you get your own MISP instance up and running. There are several ways to deploy MISP depending on whether you want a quick evaluation or a production instance; the authoritative, always-current instructions live on the MISP project download page and in the MISP repository, so this chapter points you at the right option rather than reproducing install steps that change between releases.

8.1 Choosing a deployment method

- **Official installer** — the recommended way to run a production MISP on your own host. MISP provides an installer script and step-by-step guides for the supported operating systems (recent Ubuntu, Debian and RHEL-based distributions). Start from the download page.
- **Docker** — the misp-docker project provides official container images and a `docker compose` setup, which is a convenient way to run MISP (and misp-modules) in containers.
- **Cloud images** — MISP can be deployed from cloud/marketplace images and with infrastructure-as-code; see the download page for the current options.
- **Evaluation virtual machine** — for a quick try-out or a training session, CIRCL maintains a ready-to-run virtual machine image (see below). It is well suited to evaluation and training, but a fresh install or container deployment is preferred for production.

Whichever method you choose, keep the instance updated (see the FAQ and updating the Python libraries), and review the requirements chapter for the supported software versions.

8.2 Evaluation virtual machine

CIRCL maintains an up-to-date MISP virtual machine image, regenerated regularly from the MISP repository. It is an easy out-of-the-box option, well suited to product evaluation and to training sessions.

8.2.1 Download

The best place to get the latest MISP virtual machine image, along with the available training materials, is the MISP training materials page on the CIRCL website. Download the image and validate its SHA512 fingerprint before using it.

8.2.2 Import the appliance

In VirtualBox, use **Import Appliance...** to import the virtual machine.

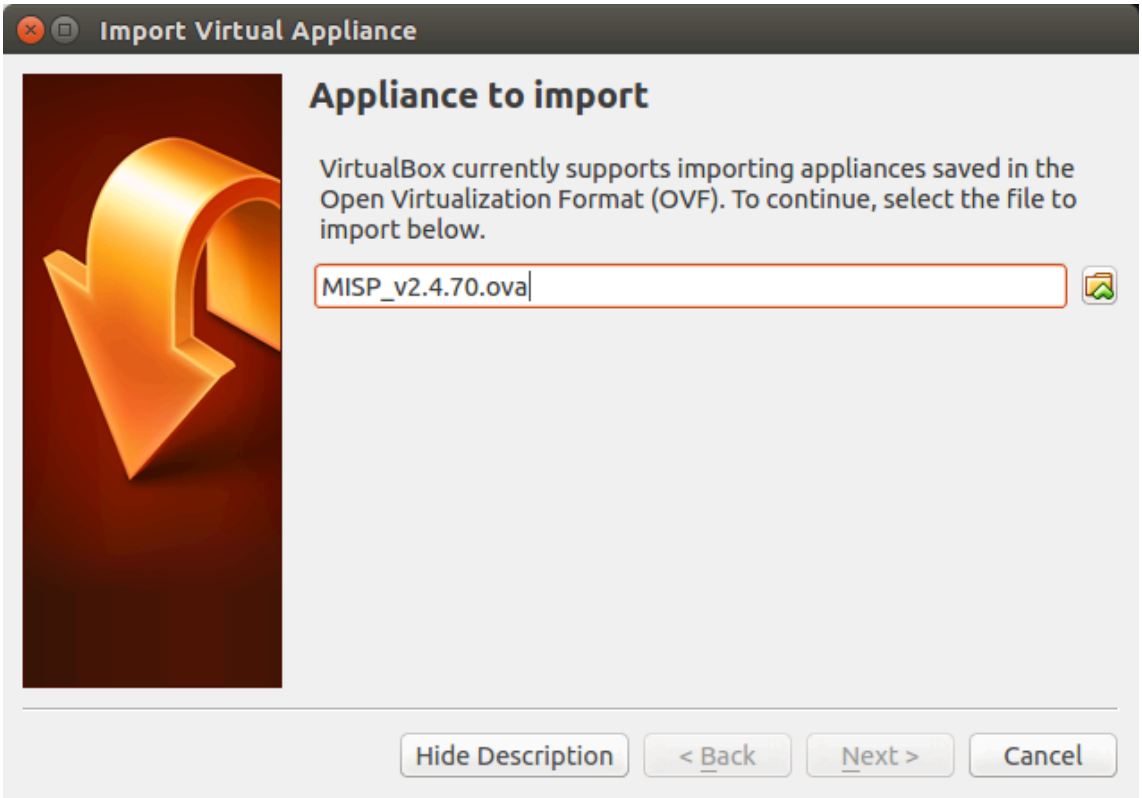


Figure 8.1: Import Appliance…

These instructions cover VirtualBox; for other virtualisation solutions (such as VMware) see the notes on the training materials page. ESXi servers have also been tested, though you may need to adjust the ATA bus manually.

8.2.3 Credentials

- **GUI admin:** `admin@admin.test` / `admin` — the site-admin account with full rights (create your own users and change this immediately).
- **Shell/SSH:** `misp` / `Password1234`.
- **MySQL:** generated when the VM is built; the details are in `~misp/mysql.txt`.

8.2.4 Networking

By default the VM is reachable via SSH on host port **2222** and the web UI on **http://localhost:8080/**. VirtualBox forwards a set of host ports to the guest:

- **ssh** — 2222 on host ⟶ 22 on guest
- **http** — 8080 on host ⟶ 80 on guest (main web UI)
- **https** — 8443 on host ⟶ 443 on guest (not in use by default)
- **MISP dashboard** — 8001 on host ⟶ 8001 on guest
- **misp-modules** — 1666 on host ⟶ 6666 on guest

To change the forwarding, select the running VM and go to **Settings ⟶ Network ⟶ Advanced ⟶ Port forwarding**.

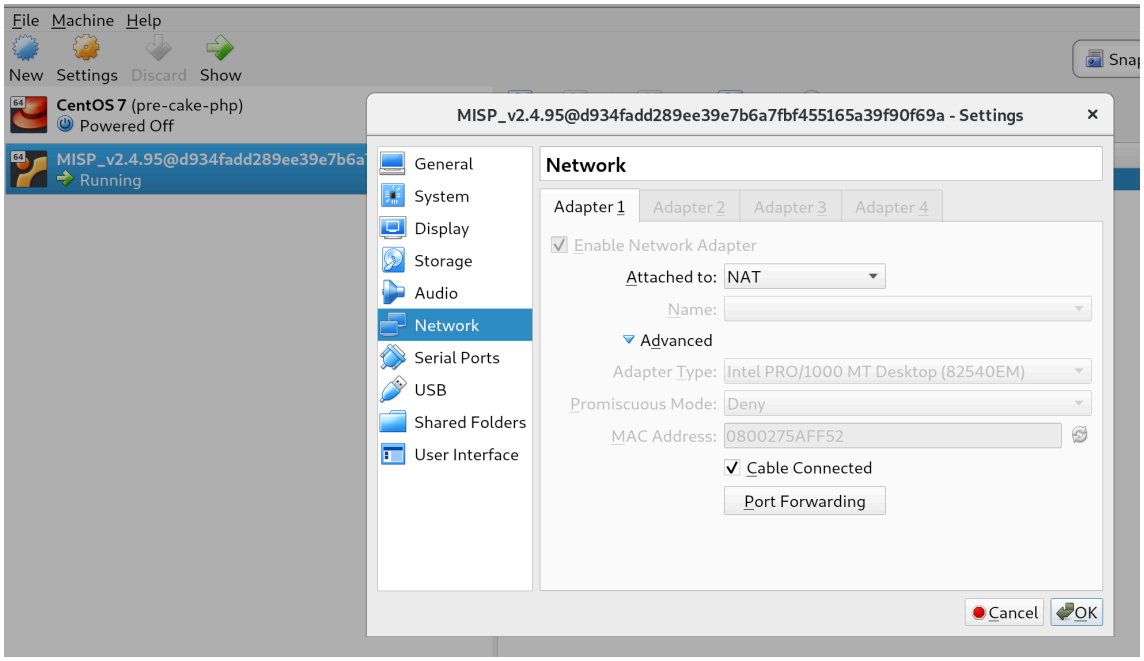


Figure 8.2: Overview of Network settings

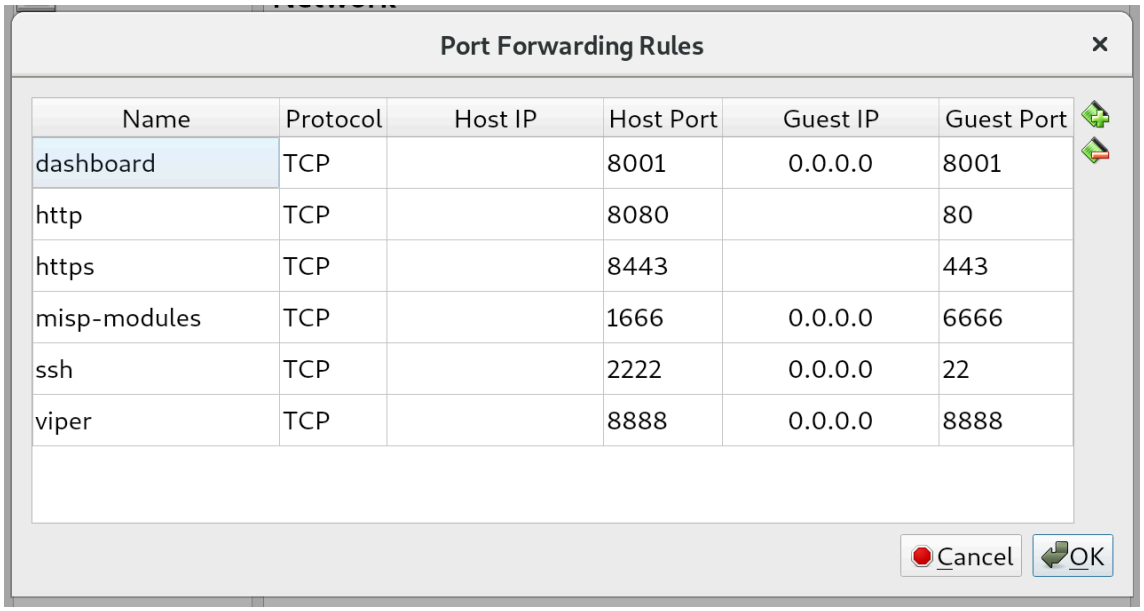


Figure 8.3: Overview of forwarded ports

> **Note**
>
> VMware users connect to whatever IP the VM has on the host — there is no port forwarding as with VirtualBox.

8.2.5 Setting the base URL

If you access the VM from another host, set the base URL in the MISP settings. From the command line:

```
sudo -u www-data /var/www/MISP/app/Console/cake Baseurl [baseurl]
```

You can confirm it in the config:

```
grep baseurl /var/www/MISP/app/Config/config.php
```

An external base URL can be set later, from the GUI or the command line (changing it is never required just to reach the GUI):

```
sudo -u www-data /var/www/MISP/app/Console/cake Admin setSetting MISP.external_baseurl
[external_baseurl]
```

9. Quick Start

10. Quick Start



MISP (the MISP Intelligence Sharing Platform) software facilitates the exchange and sharing of threat intelligence, Indicators of Compromise (IoCs) about targeted malware and attacks, financial fraud or any intelligence within your community of trusted members. MISP sharing is a distributed model containing technical and non-technical information which can be shared within closed, semi-private or open communities. Exchanging such information should result in faster detection of targeted attacks and improve the detection ratio, whilst also reducing the number of false positives.

With the focus on automation and standards, MISP provides you with a powerful ReST API, extensibility (via misp-modules) or additional libraries such as PyMISP, jump ahead to these chapters to get started.

10.1 Login into MISP

MISP default credentials: Username: admin@admin.test Password: admin

10.2 I forgot my admin password :(

You can quickly reset it via the command line. You need to know the Admins E-Mail address. Replace www-data with whoever runs the webserver.

```
sudo -u www-data /var/www/MISP/app/Console/cake Password admin@admin.test Password1234
```

10.3 Tasks to do after first Start

1. Change site admin password
2. Activate Feeds
3. Setup your User 3.1 Designate a Site Admin and an Org Admin 3.2 Add some contributing users and assign the corresponding Roles
4. MISP Administration 4.1 Edit your first organisations' name

10.4 Password Policy

- `[12]`: Ensure that the password is at least 12 characters long
- `[A-Z]`: contains at least one upper-case
- `[0-9| ]`: includes a digit or a special character
- `[a-z]`: at least one lower-case character.

If you need a password generator use: - Ubuntu / Debian: pwgen - Built-in generator in KeePass and other password managers - Built-in generator in various web browsers

All Generator tools are only possibilities without any guarantee!

10.5 tl;dr

10.5.1 Create an Event



Figure 10.1: Overview create an event in MISP

10.5.2 Browse Past Events

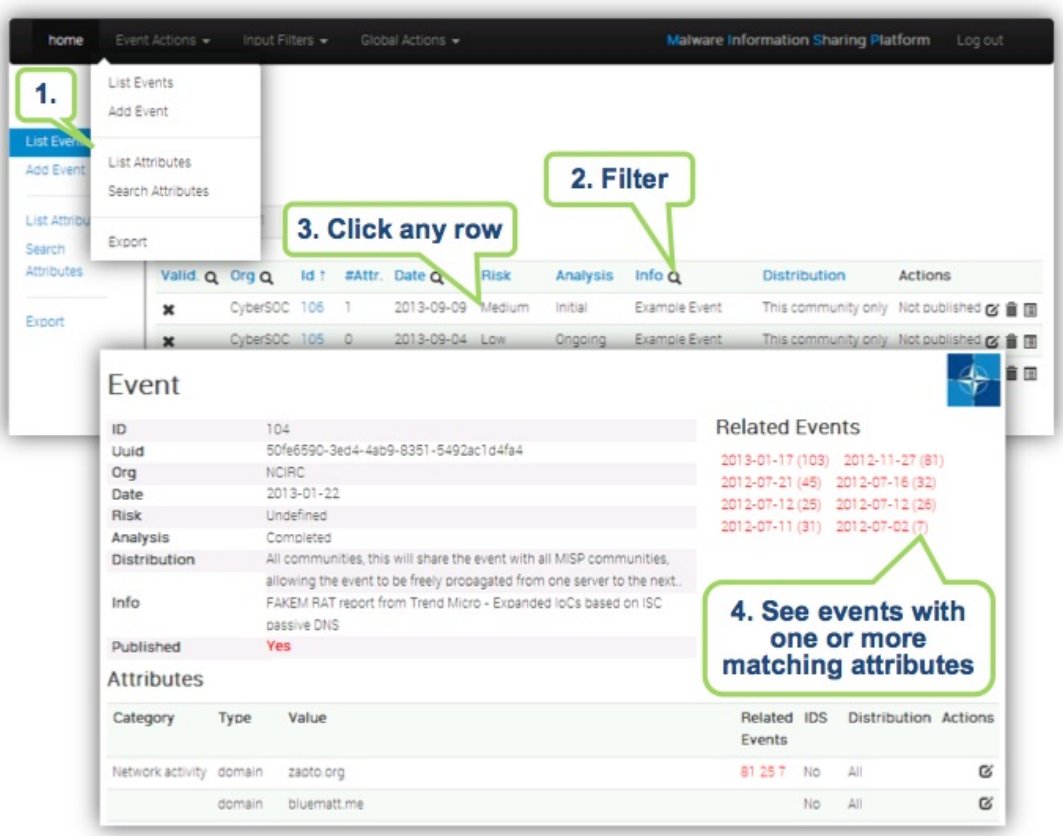


Figure 10.2: Overview browse past evente in MISP

10.5.3 Export Events for logsearches



Figure 10.3: Overview export events for logsearches

10.6 Create an Event

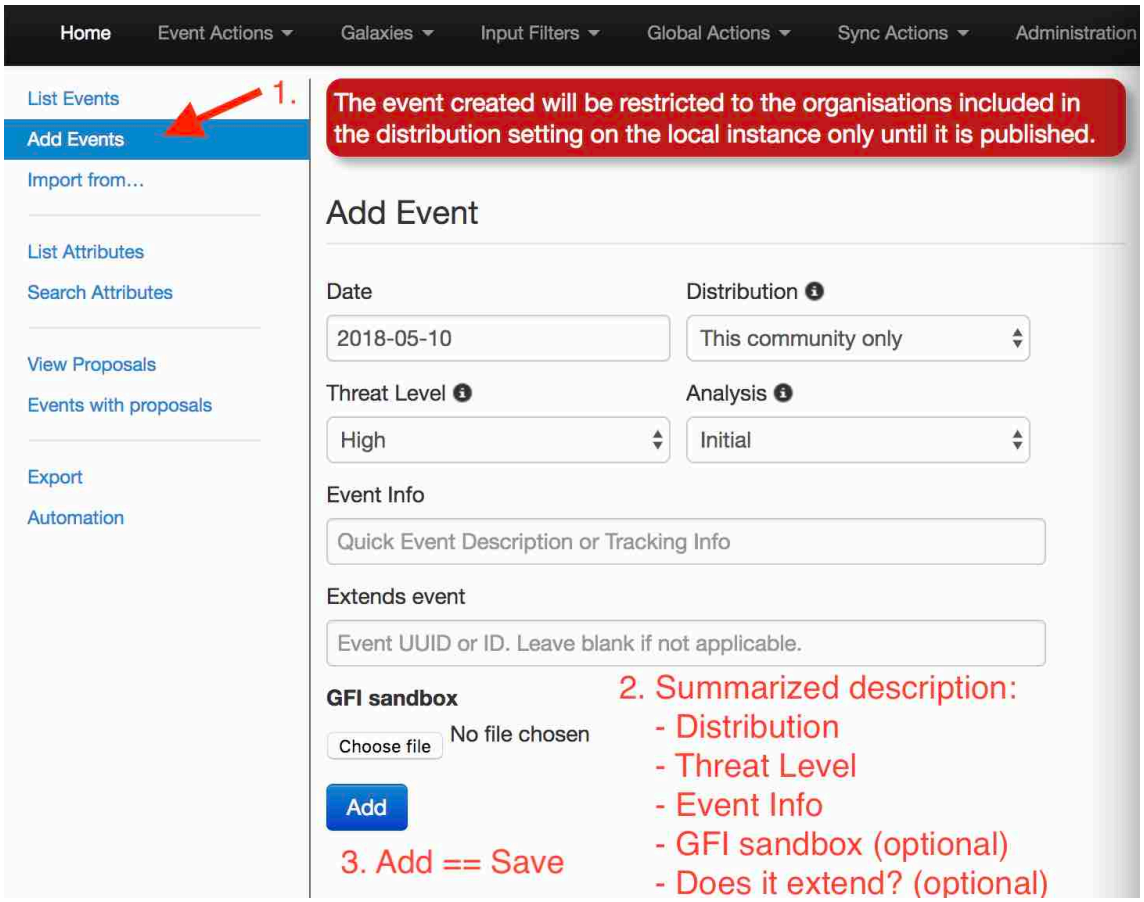


Figure 10.4: Create an Event in MISP

You only have to add a few pieces of information to register your Event. Further details will be specified after the Event has been added.

10.7 Describe Event

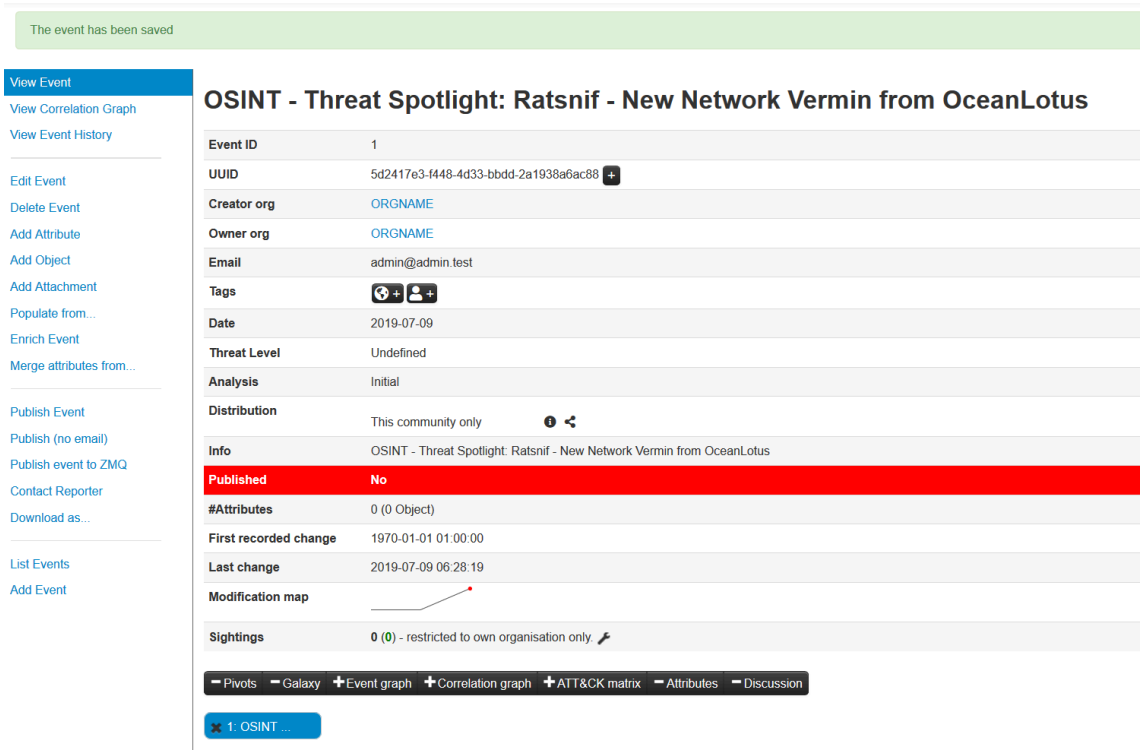


Figure 10.5: Describe Event

Now you can specify the information for your Event (you will need to scroll the window).

10.7.1 Free-Text Import Tool

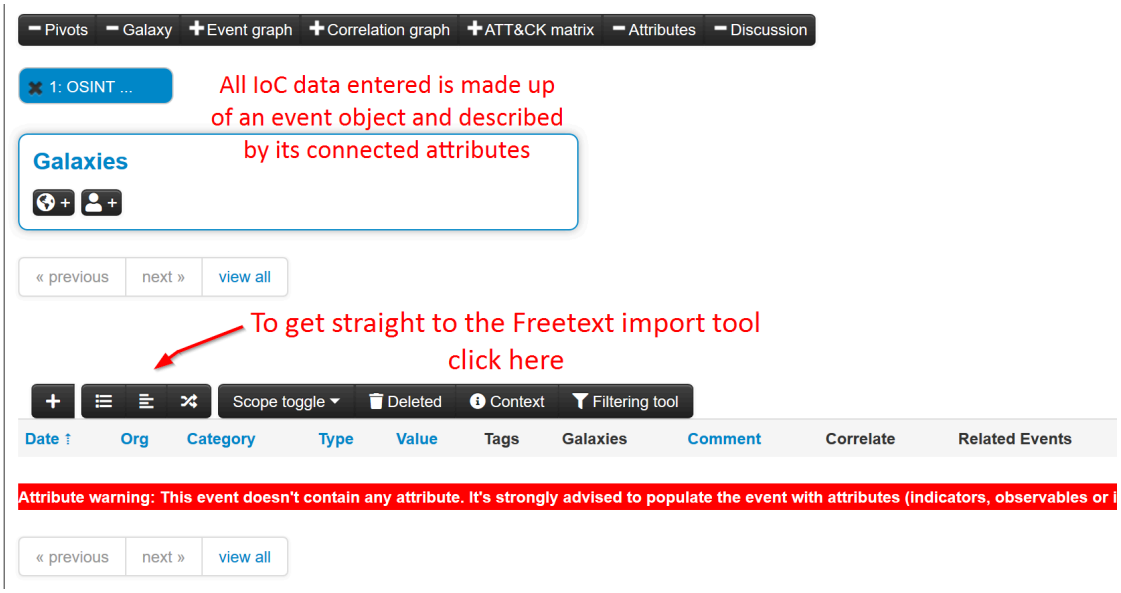


Figure 10.6: Use Freetext import

The following will pop-up.

If you have a list of indicators from which you would like to quickly generate attributes then the **Free-text import tool** is just what you need. Simply paste your list of indicators (separated by line-breaks) into this tool.

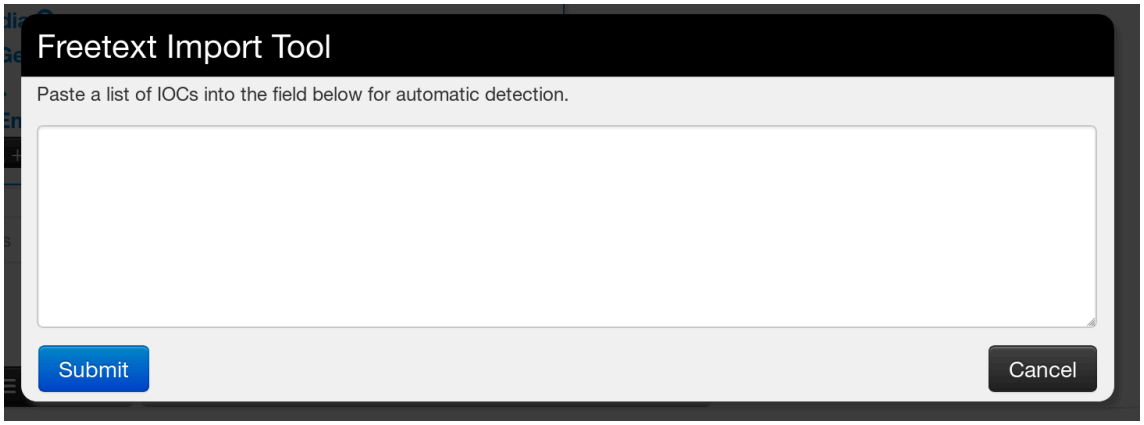


Figure 10.7: Freetext import pop-up

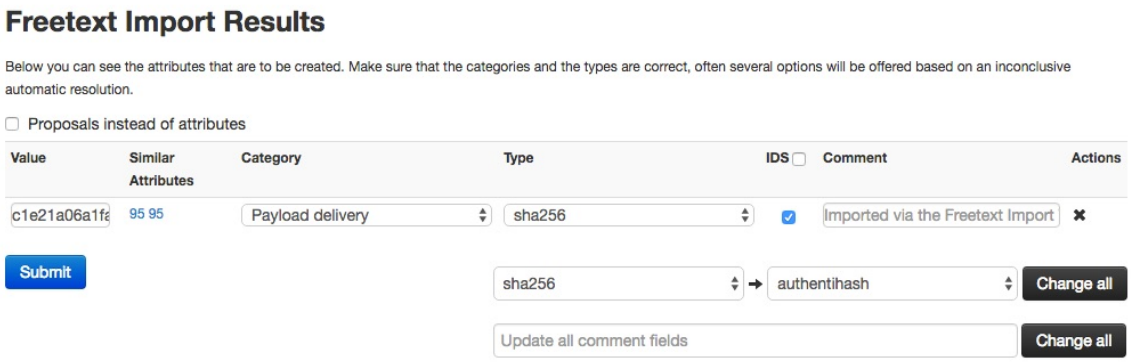


Figure 10.8: FreeText Import result

The tool will help you to find similarities between your import and other issues already registered in MISP.

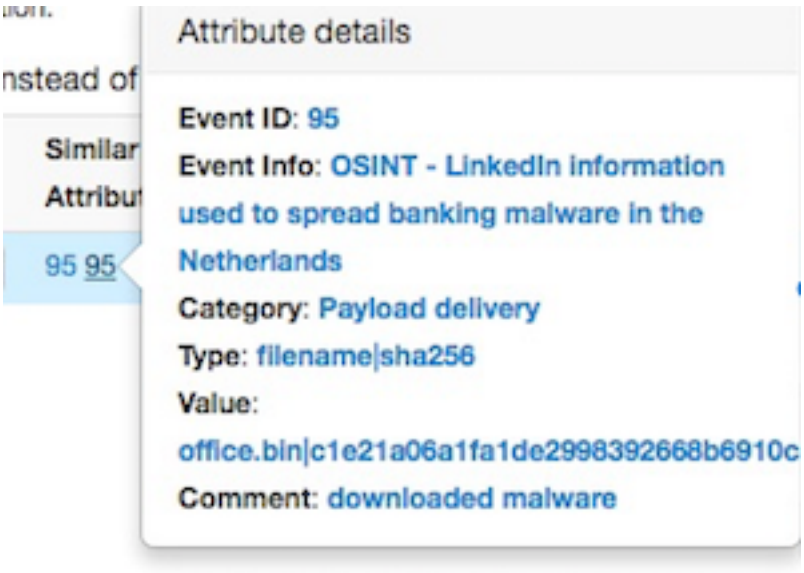


Figure 10.9: FreeText Suggest

For example, you can see the ID of all related Events and view their information.

Alternative to import

An alternative route to reach the Freetext import tool is shown below.

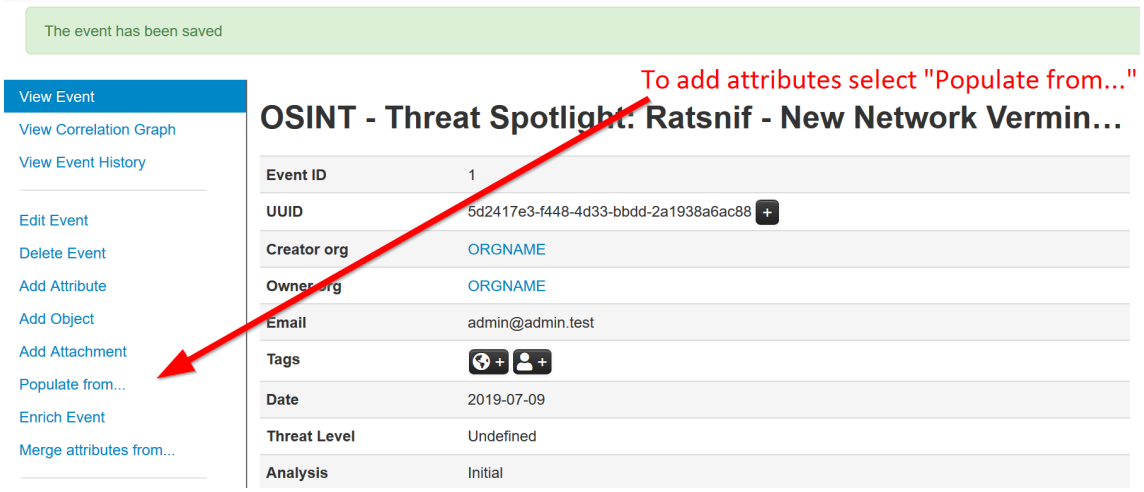


Figure 10.10: Import From

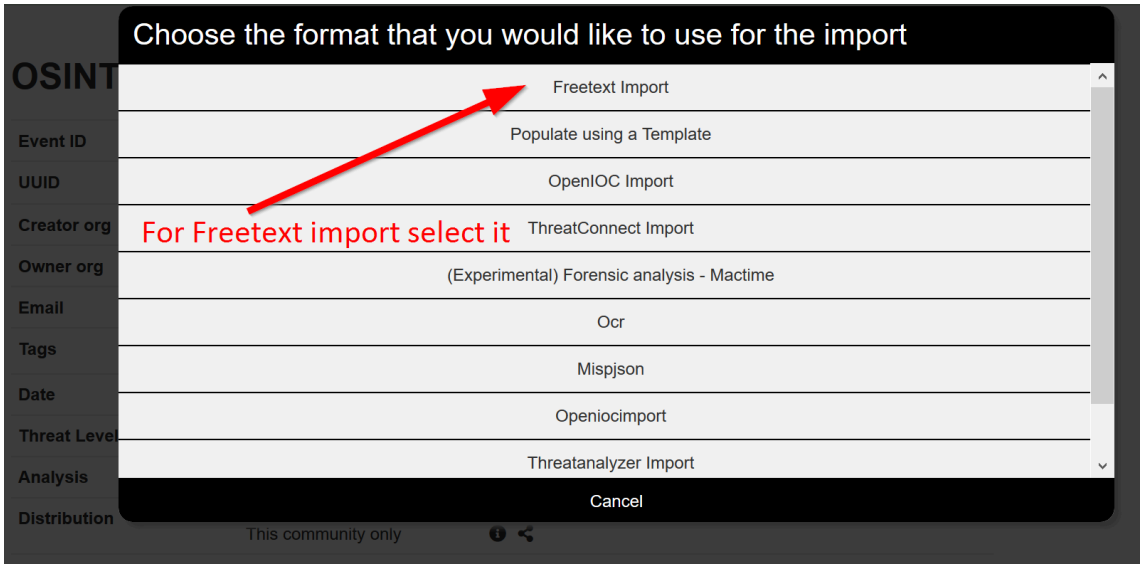


Figure 10.11: Import From pop-up

10.7.2 Tags and Taglist

10.7.2.1 Using existing Data

Another easy way to add information is to use Tags. You can see the result of adding existing Tags (circl:incident-classification=XSS ans circl:incident-classification=“information-leak).

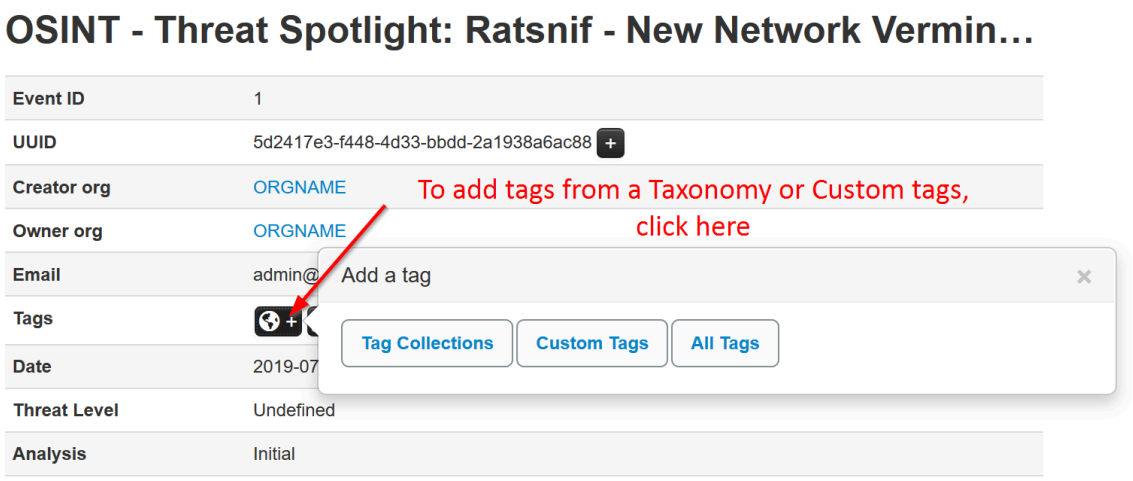


Figure 10.12: Add Tag

By clicking the button, you can add more tags from an existing Taglist.

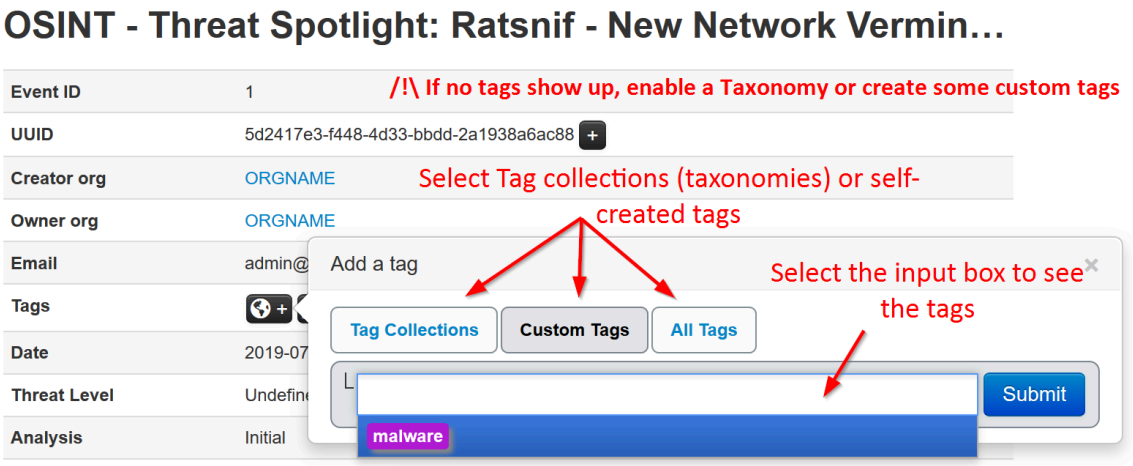


Figure 10.13: Taglist

In particular the “Taxonomy Library: circl” Taglist is very complete.
Once you added the tag(s) it will show in you main event window and in the list event view.

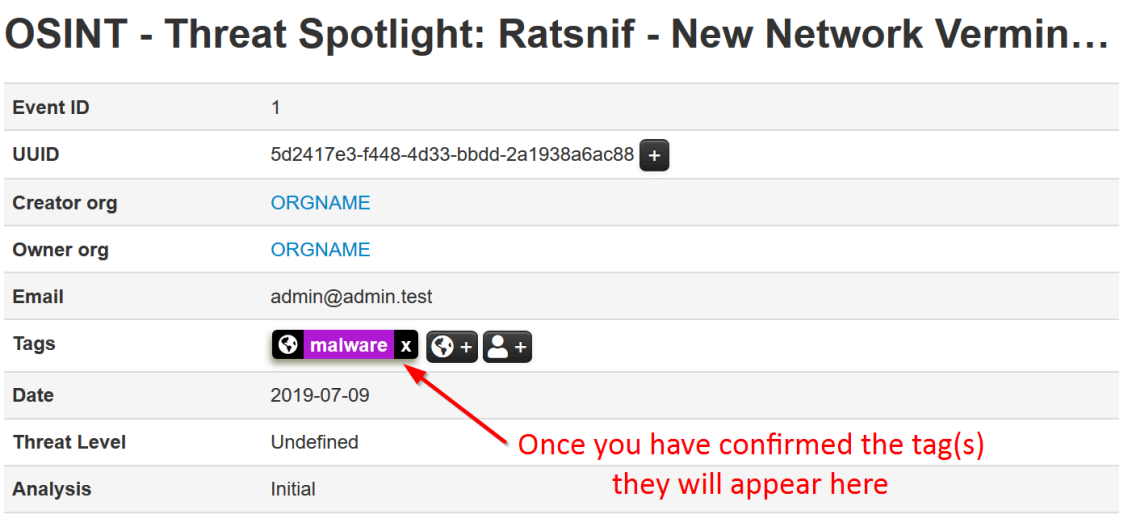


Figure 10.14: Tag Added

10.7.3 Local tags

Local tags can be added in a similar fashion.

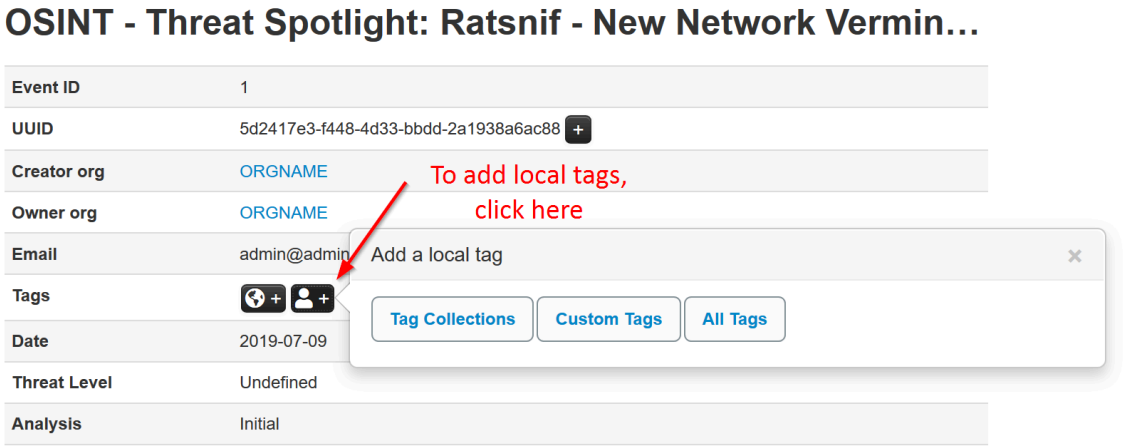


Figure 10.15: Add Local Tag

They will be identified by a corresponding icon.

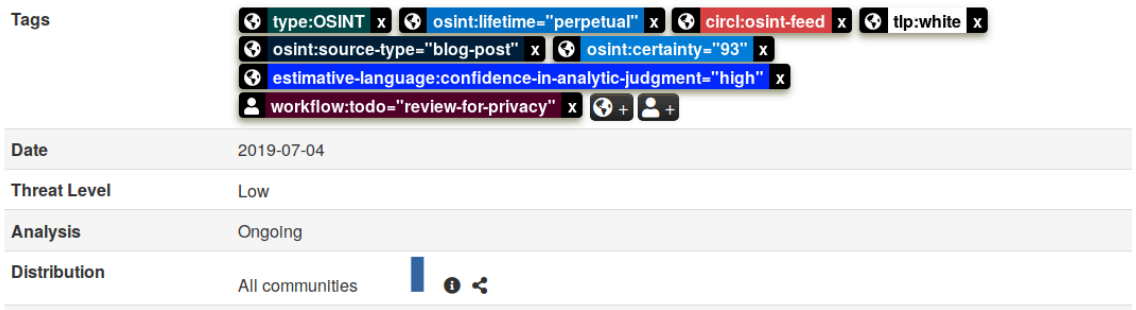


Figure 10.16: Local Tags Mix

10.7.3.1 No tags in list

In case you get the below. You need to either enable an existing Taxonomy or add some custom tags.

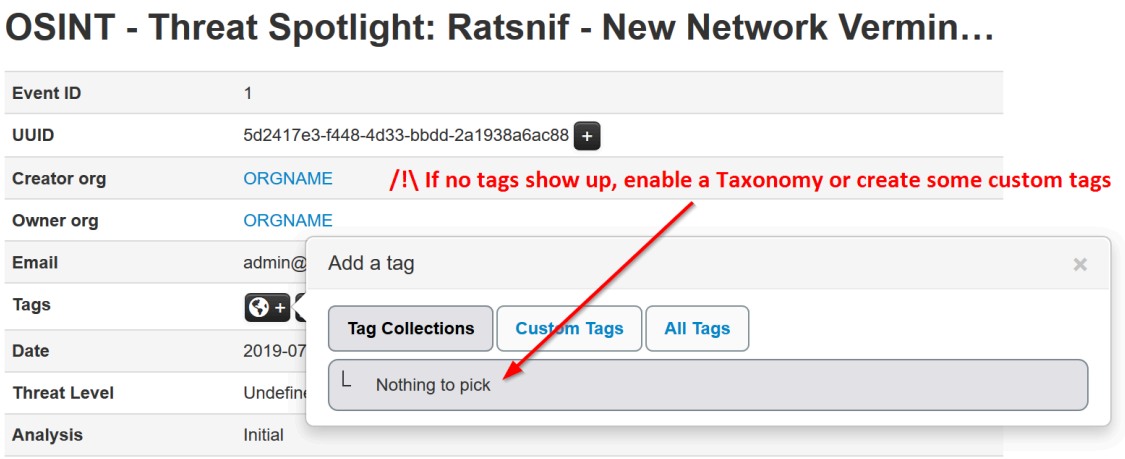


Figure 10.17: No tags

10.7.3.2 Make your own Taglist

If you want make your own Taglist, select Add Tag.

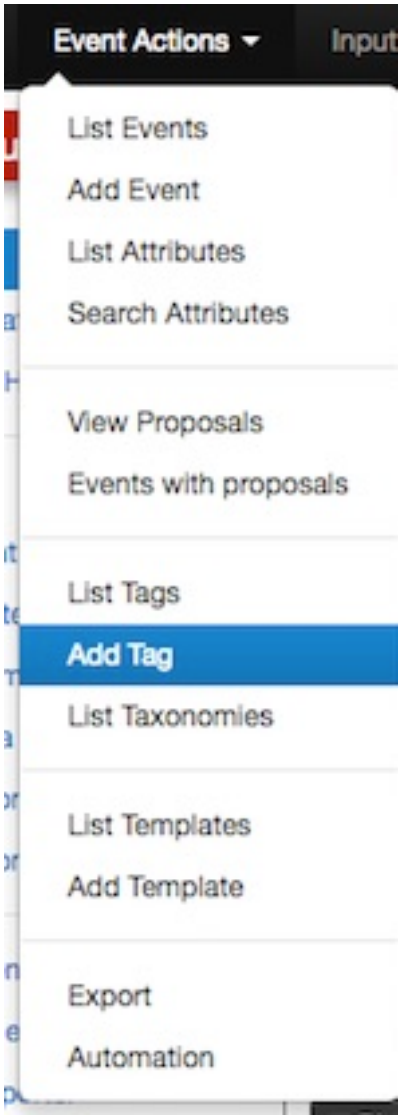


Figure 10.18: Select Add New Tag

You will see the following window:

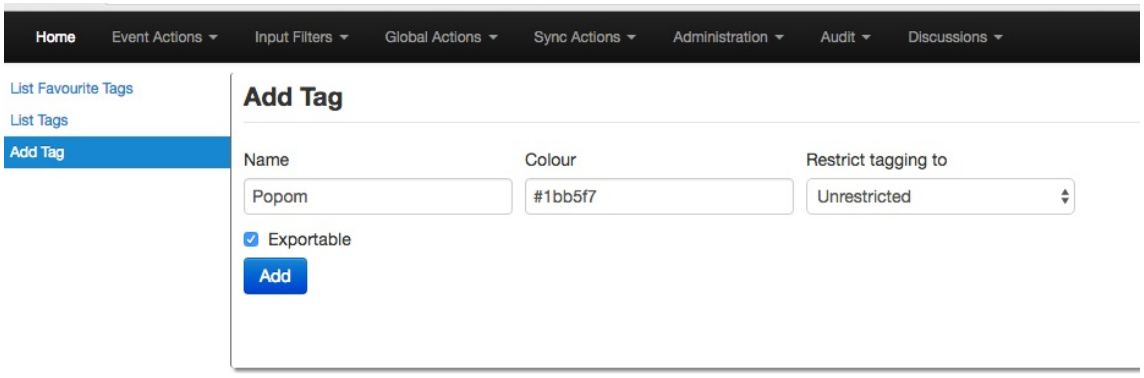


Figure 10.19: Define Tag

Then, when you add the new tag it will appear in the Custom Taglist.

10.7.4 Suggestions

The following attribute types should be added for each Event: - ip-src: source IP of attacker - email-src: email used to send malware - md5/sha1/sha256: checksum - Hostname: full host/dnsname of attacker - Domain: domain name used in malware

10.8 Browsing Events

To see your Event, select List Events from the menu Events Action. You can click any row and select a filter.

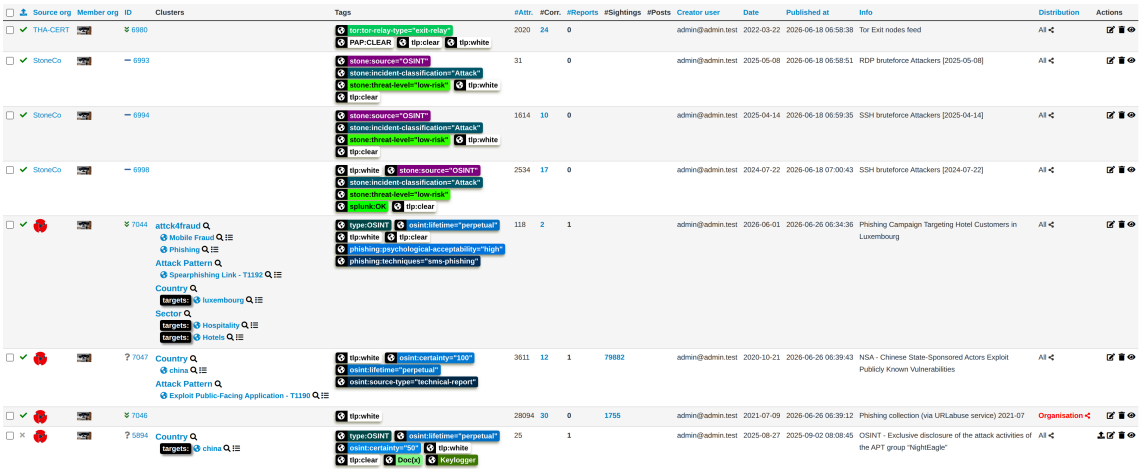


Figure 10.20: Browsing Events

If you click on your Event's number, you can see all the information related to your Event.

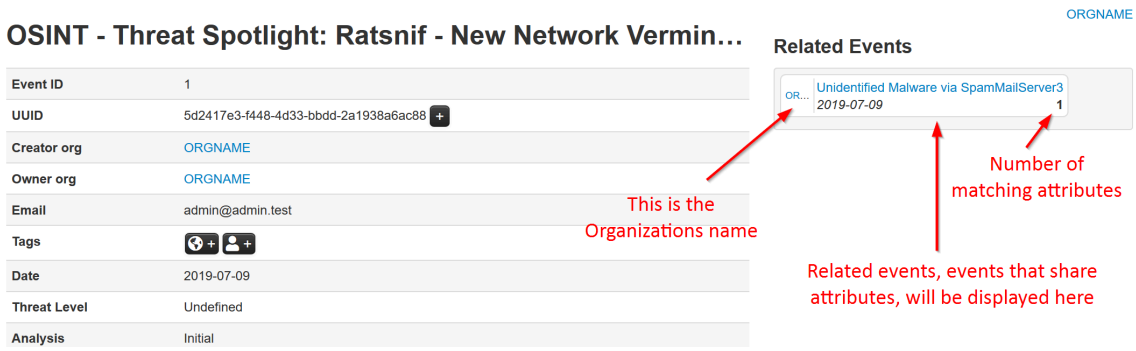


Figure 10.21: See Event

10.9 Export Events for Log Search

MISP can export your data in many formats for use in other tools — from plain MISP JSON/XML to detection rules (Suricata, Snort, Bro/Zeek), host and hash lists, CSV and STIX. Attributes flagged **for Intrusion Detection System** (the `to_ids` flag) are the ones included in the detection-rule exports; attributes without it are treated as context only. Use the export options to download the format you need — see the automation chapter for the full list of formats and the API.

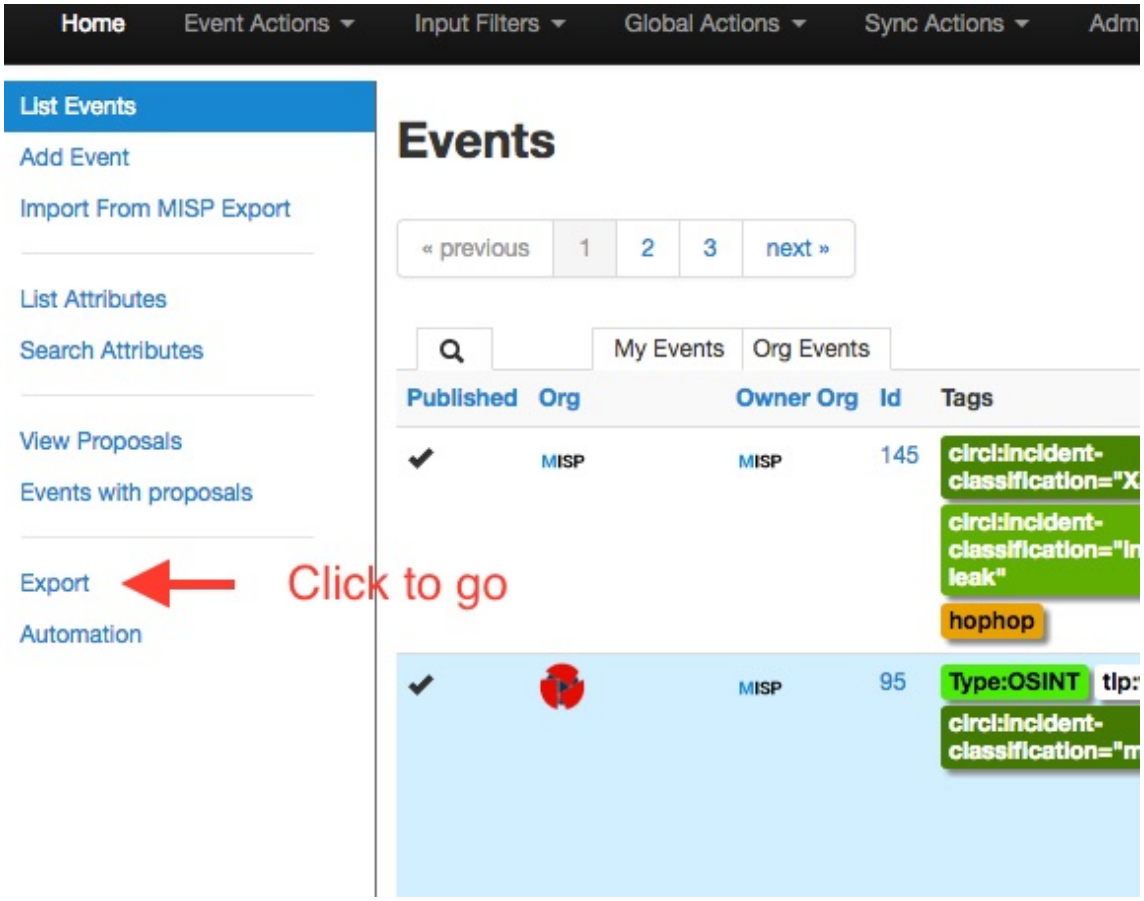


Figure 10.22: Quick Export

Simply click on any of the following buttons to download the appropriate data for log correlation.

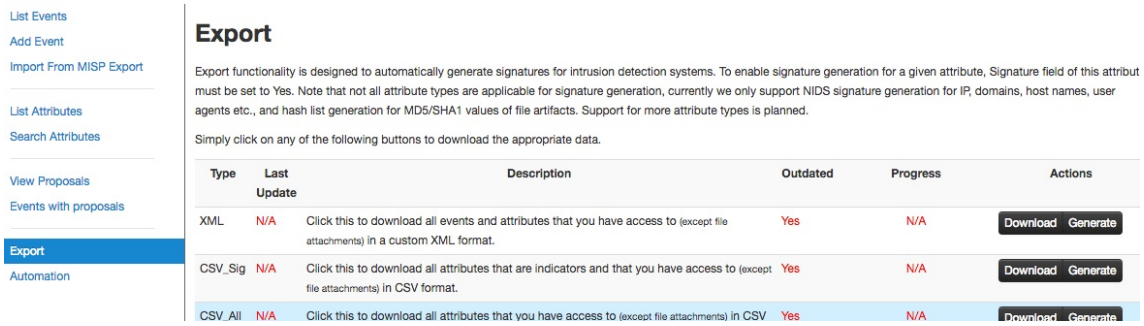


Figure 10.23: Select Format

10.10 Enable a Taxonomy

Most taxonomies are disabled out of the box. To start tagging with one, go to **Event Actions → List Taxonomies**, find the taxonomy you want (for example `tlp`), and enable it — you can also enable all of its tags at once. Its tags then appear in the tag picker on events and attributes. See the Taxonomies chapter for more.

10.11 Enable and fetch a feed

MISP ships with a set of public OSINT feeds. To use one, go to **Sync Actions → Feeds**, enable a feed (the CIRCL OSINT feed is a good starting point), and then either **cache** it — for fast correlation against your data without importing — or **fetch** it to import its events into your instance. See the Managing feeds chapter for the details.

11. General Layout

12. General Layout

12.1 The top bar

This menu contains all of the main functions of the site as a series of dropdown menus. These menus contain all (from the current user's perspective) accessible functions sorted into several groups.

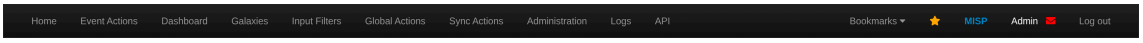
12.1.1 Simple User



Figure 12.1: This is the main menu that will be accessible from all of the views. In some instances, some additional buttons that will appear on top of these when a view provides it.

- **Home button:** This button will return you to the start screen of the application, which is the event index page or the page the user set as custom home page using the star in the top bar.
- **Event Actions:** All the malware data entered into MISP is made up of an event object that is described by its connected attributes. The Event actions menu gives access to all the functionality that relates to the creation, modification, deletion, publishing, searching and listing of events and attributes.
- **Dashboard:** Allows you to create a custom dashboard using widgets.
- **Galaxies:** Shortcut to the list of MISP Galaxies on the MISP instance.
- **Input Filters:** Input filters alter what and how data can be entered into this instance. Apart from the basic validation of attribute entry by type, it is possible for the site administrators to define regular expression replacements and blocklists for certain values in addition to blocking certain values from being exportable. Users can view these replacement and blocklist rules here while an administrator can alter them.
- **Global Actions:** This menu gives you access to information about MISP and this instance. You can view and edit your own profile, view the manual, read the news or the terms of use again, see a list of the active organisations on this instance and a histogram of their contributions by attribute type.
- **API:** Links to the OpenAPI/Swagger reference for this instance's API and to the built-in REST client.
- **MISP:** Simple link to your BASEURL
- **Name:** Name (Auto generated from Mail address) of current logged in user
- **Envelope:** Link to User Dashboard where you can consult some of your notifications and changes since last visit. Like some of the proposals received for your organisation.
- **Log out:** The Log out button to end your session immediately.

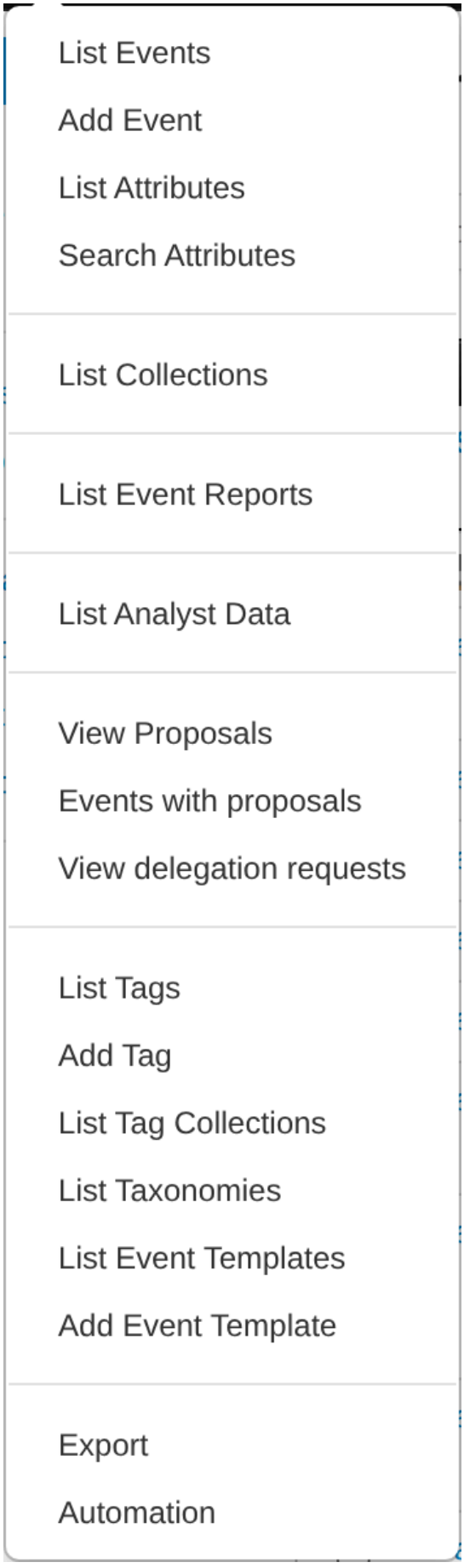
12.1.2 Admin User



* **Home button:** See description given earlier for user. * **Event Actions:** See description given earlier for user. * **Dashboard:** See description given earlier for user. * **Galaxies:** You can additionally update the Galaxies. * **Input Filters:** See description given earlier for user. * **Global Actions:** See description given earlier for user. * **Sync Actions:** With administrator access rights, shows a list of the connected instances and allows the initiation of a push and a pull. See sharing and synchronisation. * **Administration:** Administrators can add, edit or remove user accounts and user roles. Roles define the access rights to certain features such as publishing of events, usage of the REST interface or synchronization of any user belonging to the given role. Site administrators can also access a contact form, through which it is possible to reset the passwords of users, or to just get in touch with them via encrypted e-mails. * **Logs:** If you have audit permissions, you can browse the instance's logs here (this menu was formerly called *Audit*). See the Logs section below. * **API:** See description given earlier for user. * **MISP:** See description given earlier for user. * **Admin:** User role. * **Envelope:** See description given earlier for user. * **Log out:** See description given earlier for user.

12.1.3 Content of each of the top menu bar drop-down menus

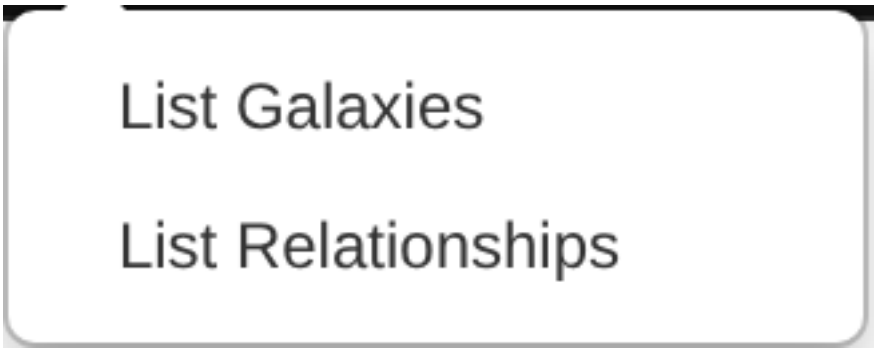
12.1.3.1 Event actions



* **List Events:** Lists all the events in the system that

are not private or belong to your organisation. You can add, modify, delete, publish or view individual events from this view. * **Add Event:** Allows you to fill out an event creation form and create the event object, which you can start adding attributes. * **List Attributes:** Lists all the attributes in the system that are not private or belong to your organisation. You can modify, delete or view each individual attribute from this view. * **Search Attributes:** You can set search terms for a filtered attribute index view here. * **List Collections:** List collections — named bundles of events and other elements that can be managed and synchronised together. * **List Event Reports:** List the rich-text event reports on this instance. * **List Analyst Data:** List analyst data (notes, opinions and relationships) on this instance. * **View Proposals:** Shows a list of all proposals that you are eligible to see. * **Events with proposals:** Shows all of the events created by your organisation that has pending proposals. * **List Tags:** List all the tags that have been created by users with tag creation rights on this instance. * **List Tag Collections:** List all the tag collections that have been created by users with tag creation rights on this instance. Tag collections allow you to assign a collection of tags to an event or attribute in one action. * **Add Tag:** Create a new tag. * **List Taxonomies:** List all of the taxonomies installed on the MISP instance. This is also the place to activate the taxonomies as a Org Admin/Site Admin. * **List Event Templates:** List all of the event templates created by users with template creation rights on this instance. * **Add Event Template:** Create a new event template. * **Export:** Export the data accessible to you in various formats. * **Automation:** If you have authentication key access, you can view how to use your key to use the REST interface for automation here.

12.1.3.2 Galaxies



* **List Galaxies:** Index of MISP Galaxies on the MISP instance. * **List Relationships:** Shows a list of galaxy cluster relationships.

12.1.3.3 Input filters



Figure 12.2: Input filters

- **Import Regexp:** You can view the Regular Expression rules, which modify the data that can be entered into the system. This can and should be used to help filter out personal information from automatic imports (such as removing the username from windows file paths), having unified representation for certain common values for easier correlation or simply standardizing certain input. It is also possible to block certain values from being inserted. As a site administrator or a user with regex permission, you can also edit these rules.
- **Signature Allowlist:** You can view the allowlist rules, which contains the values that are blocked from being used for exports and automation on this instance. Site administrators have access to editing this list.
- **List Warninglists:** MISP warninglists are lists of well-known indicators that can be associated to potential false positives, errors or mistakes. The warning lists are integrated in MISP to display an info/ warning box at the event and attribute level.
- **List Noticelists:** MISP noticelists inform MISP users of the legal, privacy, policy or even technical implications of using specific attributes, categories or objects.
- **List Correlation Exclusions:** Allows for the exclusion of certain values from the correlation engine.

12.1.3.4 Global Actions

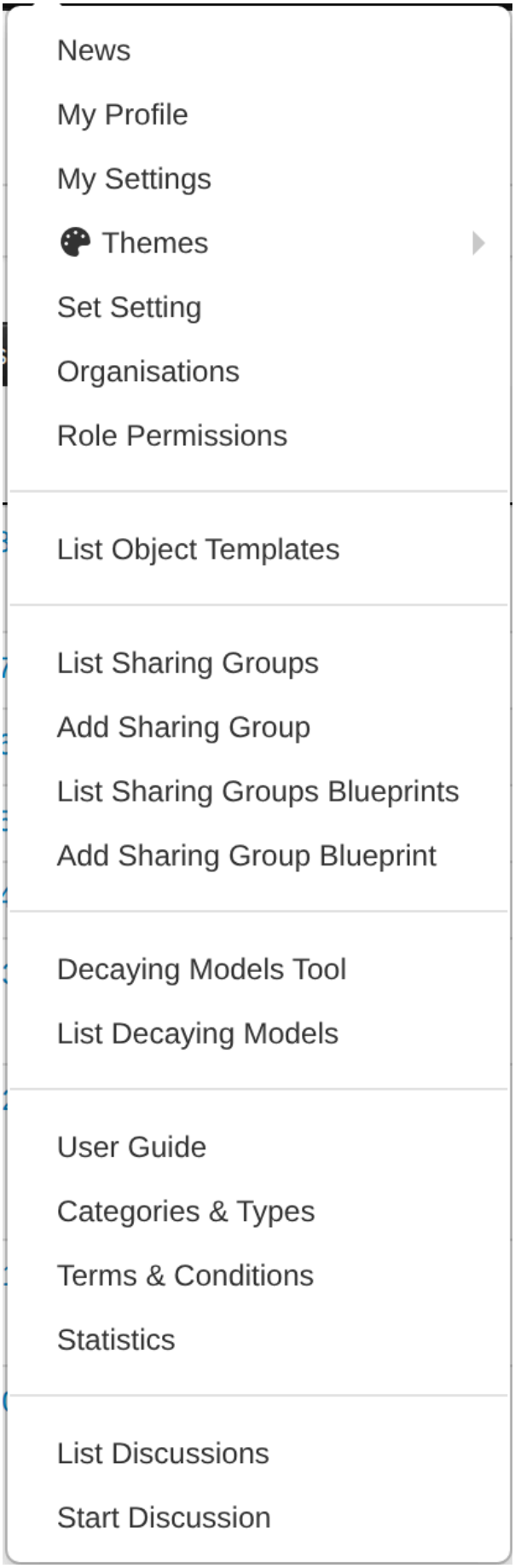
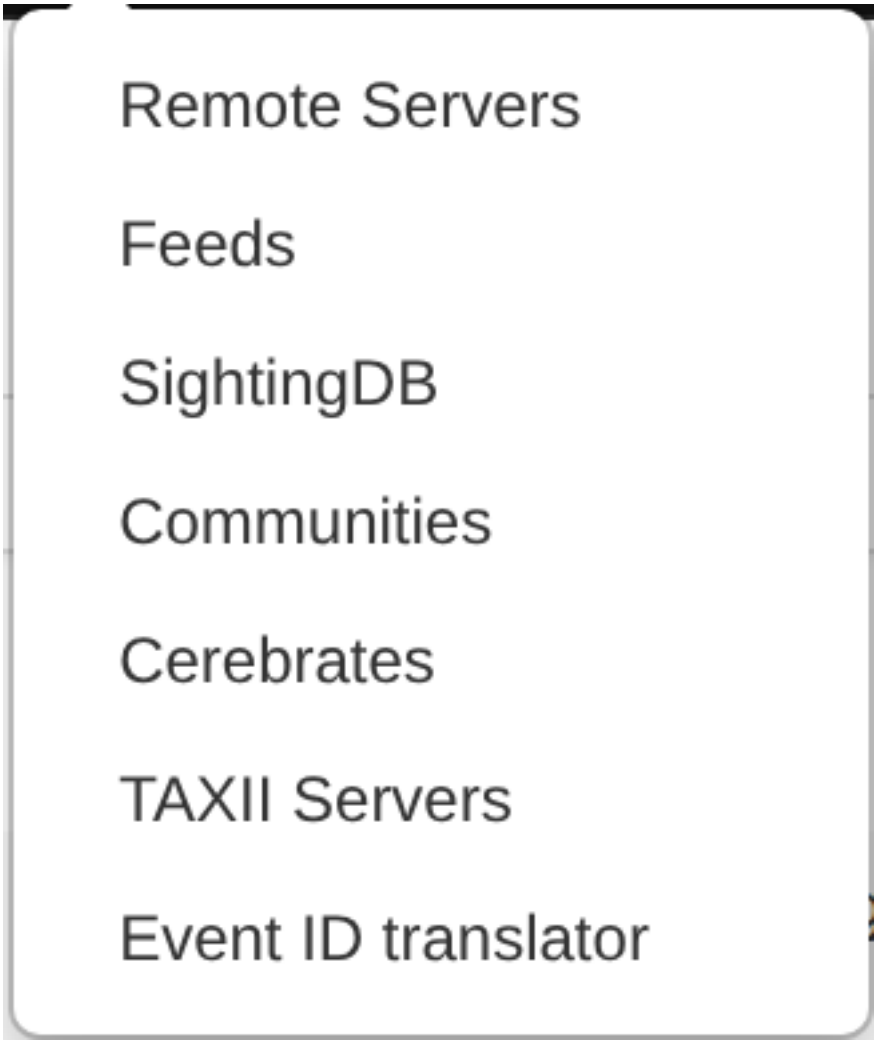


Figure 12.3: Global Actions

- **News:** Read about the latest news regarding the MISP system
- **My Profile:** Manage your user account.
- **My Settings:** View your user specific settings.
- **Set Setting:** Set user specific settings.
- **Organisations:** View the organisations having a presence on this instance.
- **Role Permissions:** You can view the role permissions here.
- **List Sharing Groups:** You can view the list of existing Sharing Groups to which you or your organisation have access.
- **Add Sharing Group:** Create a sharing group.
- **Decaying Models Tool:** Allows you to create decaying models.
- **List Decaying Models:** Manage existing decaying models.
- **User Guide:** A link to this user guide.
- **Categories & Types:** Quick overview of Attribute Categories and Types. e.g: md5 -> Payload delivery, Artifacts dropped, Payload installation, External analysis
- **Terms & Conditions:** General terms and conditions which can be configured in Administration -> Server Settings -> MISP Settings: MISP.terms_file . From the UI: "The filename of the terms and conditions file. Make sure that the file is located in your MISP/app/files/terms directory"
- **Statistics:** View a series of statistics about the users and the data on this instance.
- **List Discussions:** List threads of discussions created on the MISP instance by the organisations connected to this local community.
- **Start Discussion:** Create a new discussion thread.

12.1.3.5 Sync Actions



* **Import Server Settings:** Import sync server configuration. * **List Servers:** Connect your MISP instance to other instances, or view and modify the currently established connections. * **List Feeds:** Follow the RSS feeds of other

organisation or CERTs worldwide. * **Search Feed Caches:** Search for values potentially contained in the cached feeds and servers. * **List SightingDB Connections:** Allows you to manage existing SightingDB connections. SightingDB is an alternate sighting database that MISP interconnects with. * **Add SightingDB Connection:** Create a SightingDB connection. * **List Communities:** A list of communities that chose to advertise their existence to the general MISP user-base. * **Cerebrates:** Connect your MISP to one or several Cerebrate instances to act as lookup directories for organisation and sharing group information. * **TAXII Servers:** Configure TAXII 2.x server endpoints to push MISP data to. * **Event ID translator:** Allows to translate a local ID into the corresponding event ID on sync servers configured.

12.1.3.6 Administration

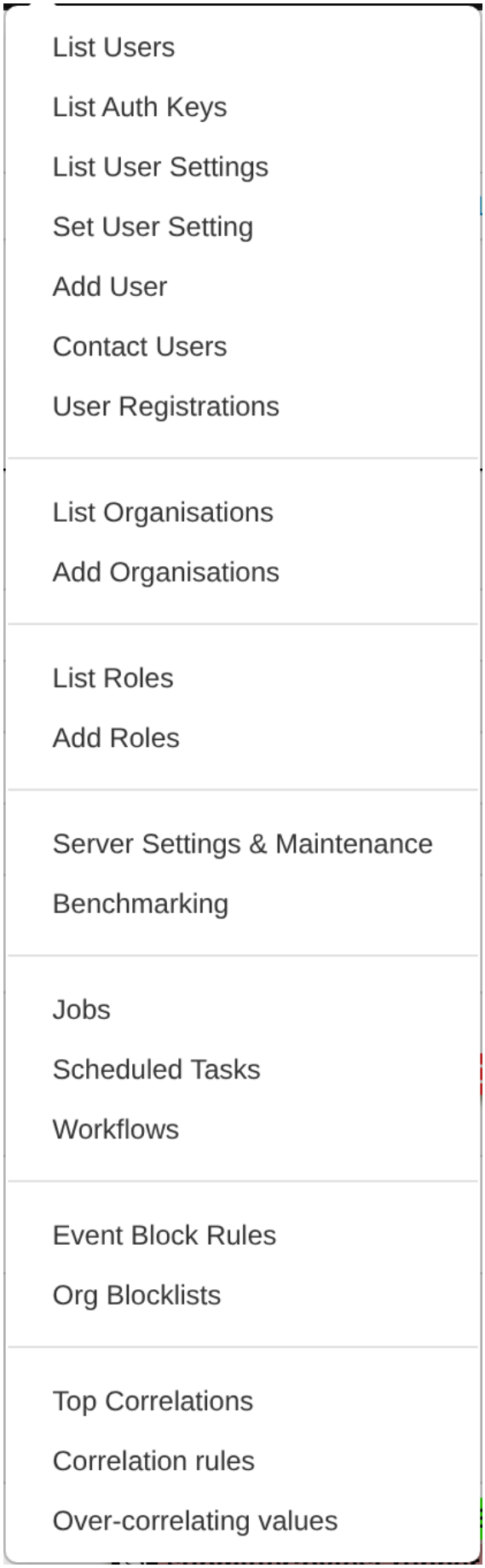


Figure 12.4: Administration

- **List Users:** View, modify or delete the currently registered users.
- **List Auth Keys:** Shows a list of the auth keys on this instance (auth keys from the advanced auth key system) with their comments.
- **List User Setting:** Lists user setting for the users on this instance.
- **Set User Setting:** Set user specific settings for the users on this instance.
- **Add User:** Create an account for a new user for your organisation. Site administrators can create users for any organisation.
- **Contact Users:** You can use this view to send messages to your current or future users or send them a temporary password.

When adding a new user to the system, or when you want to manually reset the password for a user, just use the "Send credentials automatically" setting.

After selecting the action, choose who the target of the e-mails should be (all users, a single user or a user not yet in the system).

You can then specify (if eligible) what the e-mail address of the target is (for existing users you can choose from a dropdown menu).

In the case of a new user, you can specify the future user's GnuPG key, to send his/her new key in an encrypted e-mail.

The system will automatically generate a message for you, but it is also possible to write a custom message if you tick the check-box, don't worry about assigning a temporary password manually. The system will do that for you, it will be in the mail after your custom message.

- **User Registrations:** You can find messages sent to this instance in this view.
- **List Organisations:** View the organisations having a presence on this instance, with some useful information.
- **Add Organisations:**
- **List Roles:** List, modify or delete currently existing roles.
- **Add Role:** Create a new role group for the users of this instance, controlling their privileges to create, modify, delete and to publish events and to access certain features such as the logs or automation.
- **Server Settings & Maintenance:** Various tools, upgrade scripts that can help a site-admin run the instance & Set up and diagnose your MISP installation.
- **Jobs:** View the background jobs and their progress
- **Scheduled Tasks:** Schedule the pre-defined tasks for your instance (this currently includes export caching, server pull and server push).
- **Event Block Rules:** Set event block rules. Event block rules allow you to add a simple tag filter to block events from being added or synced. See administration.
- **Blocklist Event:** Link to form where you can quickly add an event to a blocklist with it's UUID. For more information, see administration.
- **Manage Event Blocklists:** List of blocklisted events on MISP instance.
- **blocklist Organisation:** Link to for where you can quickly add an organisation to a blocklist with it's UUID. For more information, see administration.
- **Manage Org blocklists:** List of blocklisted Organisations on this instance.
- **Workflows:** Manage event-driven automation workflows (site admins, when enabled).
- **Top Correlations / Correlation rules / Over-correlating values:** Tools to inspect and tune the correlation engine.
- **Benchmarking:** Per-user and per-endpoint performance metrics (site admins, when enabled).

The **Sharing Group Blueprints**, **List Object Templates** and **Decaying Models** tools are reached from the *Global Actions* menu.

12.1.3.7 Logs

The *Audit* menu has been renamed **Logs** and now groups four views (see the administration chapter):

- **Application Logs:** The legacy application log — logins/logouts, failed authentication and other application events.
- **Audit Logs:** The detailed audit trail of data changes (shown when the audit log is enabled).
- **Access Logs:** The per-request HTTP access log (site admins only).
- **Search Logs:** Search the application log.

12.1.3.8 API

- **OpenAPI:** The browsable OpenAPI/Swagger reference for this instance's API.

- **REST client:** The built-in interactive REST client for building, sending and inspecting API calls.

12.2 The left bar

The menu items in this menu bar are different depending on the current page you are on. The blue highlight shows you what page you are on.

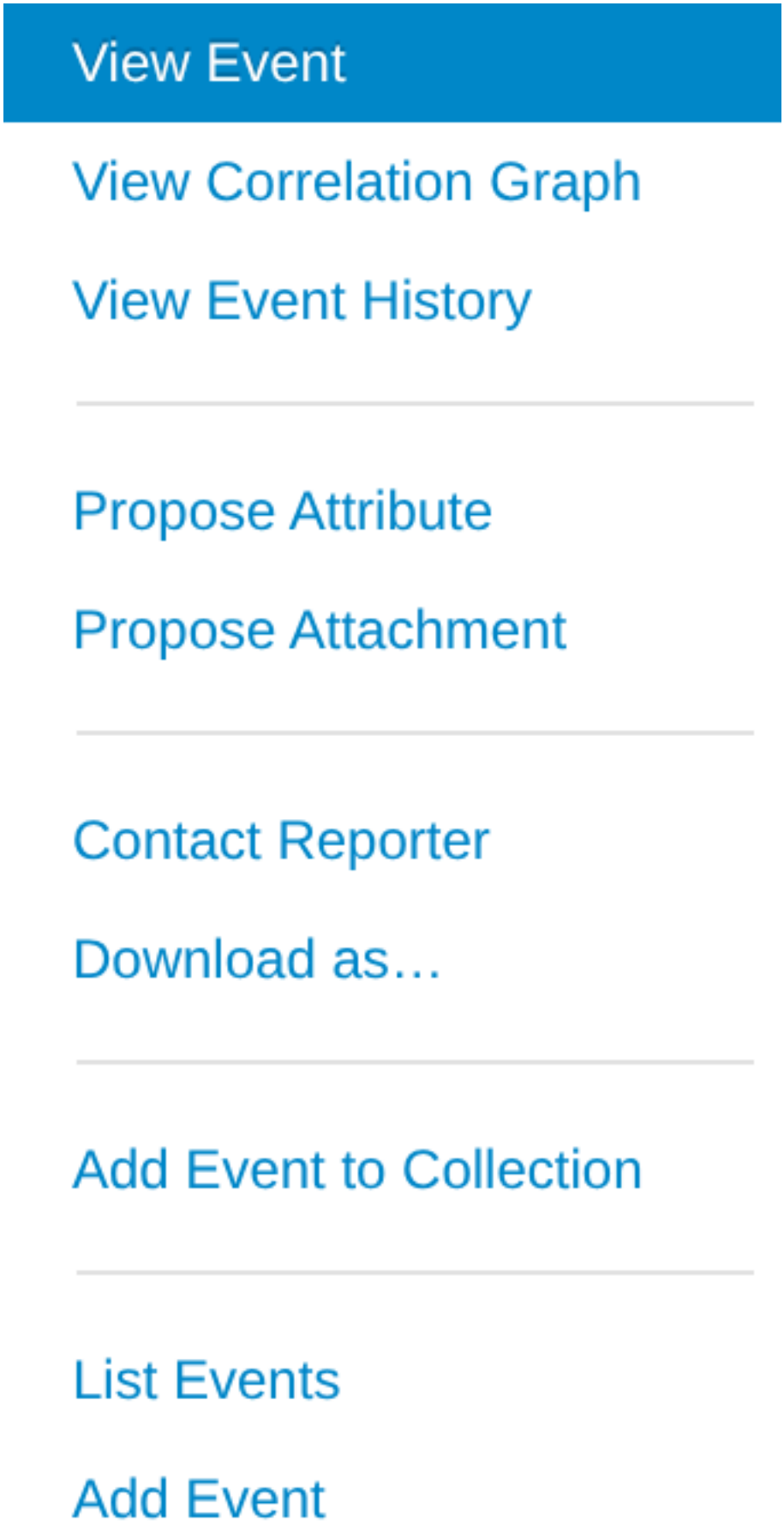


Figure 12.5: Screenshot of left menu bar

II Using MISP

14.4 Adding context to your data . . . 83
14.5 The event view . . . 84
14.6 Collaborating on events . . . 87
14.7 Publishing an event . . . 88
14.8 Extending events . . . 89
14.9 Browsing and searching . . . 89
14.10 Editing events and attributes . . . 90
14.11 Create and manage Sharing Groups . . . 92
14.12 Exporting data . . . 93
14.13 Connecting to other instances . . . 93
14.14 REST API . . . 93
14.15 Multi-factor authentication (TOTP/HOTP) . . . 93

15 MISP Extended Events . . . 97
15.1 When to use an extended event . . . 97
15.2 Creating an extended event . . . 97
15.3 Viewing extended events . . . 97
15.4 Finding extended events . . . 97
15.5 Further reading . . . 98

16 Create an Event Based on a Report . 99

17 Event Reports . . . 115
17.1 Writing a report . . . 115
17.2 Creating a report . . . 115
17.3 Extracting indicators from a report . . . 116
17.4 Template variables . . . 116
17.5 Exporting a report . . . 116
17.6 Distribution, visibility and sync . . . 117
17.7 AI-assisted extraction (experimental) . . . 117
17.8 Permissions and API . . . 117

18 Analyst Data . . . 119
18.1 The three types . . . 119
18.2 What can be annotated . . . 120
18.3 Working with analyst data in the UI . . . 120
18.4 Distribution and sharing . . . 120
18.5 Collaboration and synchronisation . . . 121
18.6 Analyst Data Blocklist . . . 121
18.7 API and permissions . . . 121

19 Sightings . . . 123

20 Decaying Models . . . 127
20.1 How scoring works . . . 127
20.2 Model parameters . . . 127
20.3 Decay formulas . . . 128
20.4 Mapping models to attribute types . . . 128
20.5 Simulating and previewing a model . . . 128
20.6 Using decay scores in queries and the event view . . . 128
20.7 Ownership, sharing and default models . . . 129

21 Delegation . . . 131

22 Best Practices . . . 139
22.1 Tagging . . . 139
22.2 Setting distribution . . . 140
22.3 Use objects instead of combined attributes . . . 144
22.4 Use proposals when possible to request author to make changes to an event . . . 144
22.5 Make use of the 2.5 collaboration and lifecycle features . . . 144

13. User Management and Global Actions

This chapter covers managing your own MISP account — first login, your profile, notification preferences, API keys, multi-factor authentication and personal settings. Creating and administering *other* users, and the role/permission model, are covered in the Administration chapter.

13.1 First run of the system

When you first log in with the credentials provided by your administrator, there are a few things to do before you start using the system:

- **Accept the terms of use** — shown immediately after your first login. Read the page and click **Accept Terms** at the bottom.
- **Change your password** — you are then prompted to set your own password. It must satisfy the MISP password policy. Enter it in both fields and submit.

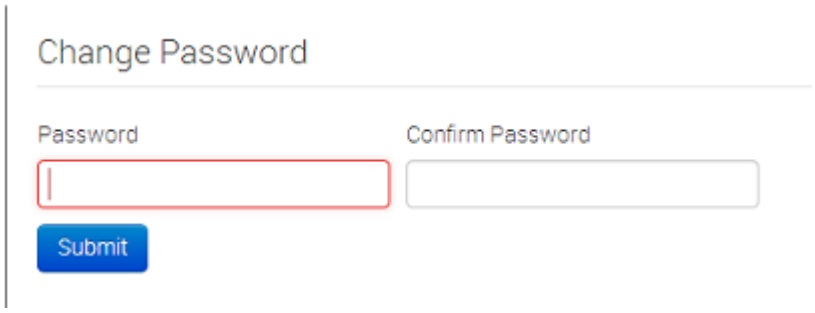


Figure 13.1: Changing the password

- **Set up your PGP/GnuPG key** — so MISP can encrypt the emails it sends you. Go to **My Profile → Edit profile** and paste your public key into the *GnuPG Key* field.
- **Subscribe to notifications** — in the same Edit profile view, tick the relevant boxes (see below).

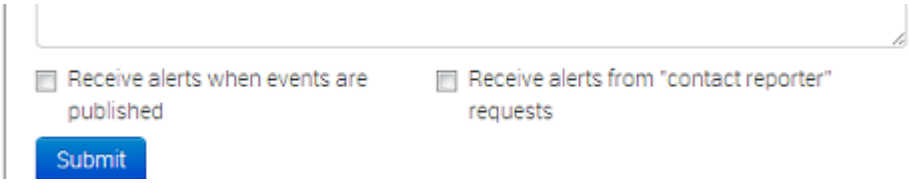


Figure 13.2: Use these checkboxes to subscribe to auto-alerts and contact-reporter emails.

13.2 Your profile

My Profile (`Global Actions → My Profile`) shows your account details and, via **Edit profile**, lets you manage:

- your display name and PGP/GnuPG (and optional S/MIME) key for encrypted mail;
- **Receive alerts when events are published** (auto-alert) — email whenever a new event you are eligible to see is published;
- **Receive alerts from "contact reporter" requests** — email when another user contacts your organisation about one of its events. (You may still be contacted directly as an event's creator even with this off.)

13.3 API keys

To use the REST API or PyMISP, create an authentication key under **My Profile → Auth Keys**. Modern MISP uses *advanced* auth keys: each key is stored hashed, shown only once at creation, and can carry a comment, an expiry, an IP allowlist and a read-only flag. See the automation chapter for the full workflow.

13.4 Multi-factor authentication

If your instance enables it, you can secure your account with TOTP/HOTP multi-factor authentication from **My Profile**. See Multi-factor authentication for the user-facing walkthrough, and the administration chapter for enabling it instance-wide.

13.5 Personal settings

My Settings (`Global Actions → My Settings`) holds per-user preferences, including your interface **theme**, dashboard configuration, and the **publish alert filter** — a JSON filter that narrows which published-event emails you receive (by tag, creator organisation or threat level). See Setting a Publish Alert Filter in the administration chapter for the filter syntax.

13.6 User administration and roles

Administrators (site and organisation admins) create, edit, disable and contact users, and assign them roles. A role sets a user's base permission level and a set of feature permissions. All of this is covered in the Administration → Users and Roles sections.

14. Using the System

This chapter walks through the day-to-day work of an analyst in the MISP web interface: creating events, populating them with attributes and objects, adding context (tags, galaxies, analyst data), collaborating with other organisations, and finally publishing and sharing your findings. Individual subsystems have their own dedicated chapters — this chapter shows how they come together in the normal event workflow and links out for the details.

14.1 The event lifecycle

In MISP, an **event** is a container that groups a set of related indicators and their context — typically everything you know about a single incident, campaign, report or collection. A typical event goes through the following stages:

1. **Create** the event with a short set of metadata (date, distribution, threat level, analysis stage, a descriptive title).
2. **Populate** it with data — attributes (individual indicators), objects (structured groups of attributes), and attachments.
3. **Contextualise** the data with tags, taxonomies, galaxy clusters and analyst data so consumers understand what it means and how much to trust it.
4. **Collaborate** — receive proposals and sightings, discuss, and optionally delegate the event to another organisation.
5. **Publish** the event, which makes it available to the eligible exports, notifies subscribers, and pushes it to the instances you synchronise with.

Editing an event (or any of its data) after publication automatically un-publishes it; you republish once you are done.

14.2 Creating an event

During this first step you create a skeleton event that holds only general information — you populate it with attributes and objects in the next stage. To start, click the **Add Event** button in the left-hand menu and fill out the form.

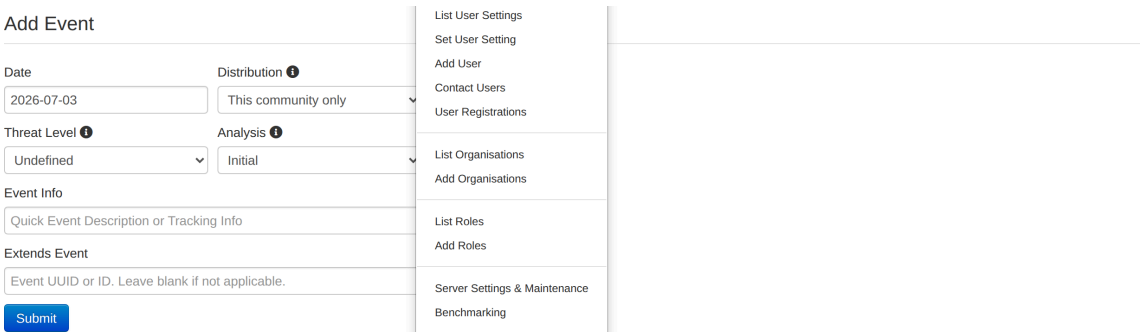


Figure 14.1: Fill this form out to create a skeleton event, before proceeding to populate it with attributes and objects.

The form contains the following fields (in order):

- **Date:** The date the incident took place (not to be confused with the event's creation date). Click the field to open a date-picker; it defaults to today.

- **Distribution:** Controls who will be able to see this event once it is published, and whether — and how — it is synchronised to other servers. The distribution is inherited by the event's data: the most restrictive setting of the event and the individual attribute always wins. The options are:
  - **Your organisation only:** Only members of your organisation can see the event. It will not be synchronised. *(Upon push: not pushed. Upon pull: pulled.)*
  - **This community only:** Everyone in your MISP community — your own organisation, other organisations on this server, and organisations on servers that synchronise with this one — can see it. *(Upon push: not pushed. Upon pull: pulled and downgraded to "Your organisation only".)*
  - **Connected communities:** As above, plus the organisations one further hop away (the communities of the servers your server connects to). *(Upon push: downgraded to "This community only" and pushed. Upon pull: pulled and downgraded to "This community only".)*
  - **All communities:** The event is shared freely and can propagate from one server to the next without restriction. *(Upon push: pushed. Upon pull: pulled.)*
  - **Sharing group:** The event is shared only with the organisations (and instances) defined in a sharing group, giving you fine-grained control. Selecting this reveals the **Sharing Group** dropdown. See Create and manage Sharing Groups.

  The pre-selected value follows your instance's `MISP.default_event_distribution` setting; review it before saving, as the out-of-the-box default is the most permissive level.
- **Threat Level:** The risk level of the event — **High** (sophisticated APTs and 0-day attacks), **Medium** (Advanced Persistent Threats), **Low** (general/mass malware), or **Undefined**. The field defaults to *Undefined* and can be refined later.
- **Analysis:** The current stage of your analysis — **Initial**, **Ongoing** or **Completed**.
- **Event Info:** A short, concise description of the event, ideally starting with your internal reference. Keep the detail for the attributes — this is a title, not a report. Note that the instance may automatically rewrite strings that match a regular-expression replacement configured by your administrator.
- **Extends Event:** Optionally turn this event into an extension of an existing event by entering the parent event's UUID or ID. A live preview of the parent event is shown as you type. Leave blank if not applicable.

> Tip
>
> If your instance has event templates configured, the Add Event page offers a **Create event via template instead** shortcut. Templates give you a guided, pre-filled walkthrough (with attached objects and mandatory-field checks) instead of the blank form — handy for recurring report types.

14.3 Populating an event with data

Once the skeleton event exists you can add data to it. Everything is driven from the event view's left-hand menu; the main entry points are **Add Attribute**, **Add Object**, **Add Attachment** and **Populate from...** (bulk import). You can also let MISP extract indicators for you with enrichment or from an event report.

14.3.1 Add Attribute

An **attribute** is a single indicator or piece of context — an IP address, a hash, a filename, a comment, and so on. Click **Add Attribute** to open the form.

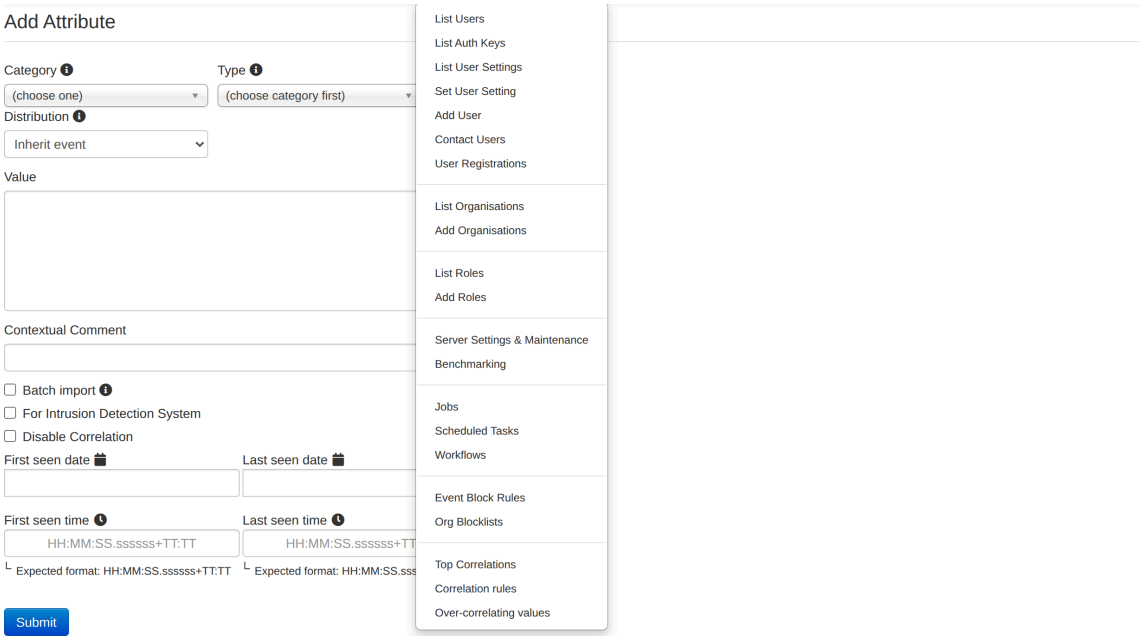


Figure 14.2: The Add Attribute form.

- **Category:** What aspect of the incident the attribute describes (for example *Payload delivery* or *Network activity*).
- **Type:** How that aspect is expressed (for example `ip-src`, `sha256`, `filename`). The type list is filtered to those valid for the chosen category. For the full category/type matrix see Categories and Types.
- **Distribution:** Who can see this attribute. In addition to the five event distribution levels, attributes offer **Inherit event** — which keeps the attribute's distribution aligned with the event and is the recommended default (and the pre-selected value).
- **Value:** The indicator itself, in a format valid for the chosen type (for example `11.11.11.11` for `ip-src`). The value is searched for administrator-configured regular-expression replacements on entry, and is what the correlation engine matches on.
- **Contextual Comment:** Free-text notes about the attribute. Comments are not used for correlation.
- **Batch import:** Tick this to paste several values of the same type into the value field (one per line) and have MISP create one attribute per line.
- **For Intrusion Detection System** (`to_ids`): Marks the attribute as an actionable indicator of compromise, so it is included in the IDS/detection exports (NIDS rules, etc.) unless overridden by a warninglist. Leave it off for attributes that are contextual only.
- **Disable Correlation:** Excludes this individual attribute from the correlation engine — useful for values that would otherwise create noisy, meaningless matches.
- **First seen / Last seen:** Optionally record the observation window for the indicator using the date-and-time slider. These feed the event timeline and are useful for decaying and time-based analysis.

14.3.2 Add Object

Objects group several related attributes into a single structured unit — a `file` object bundling a filename with its hashes and size, a `domain-ip` object, an email object, and so on — using community-maintained templates. Click **Add Object**, pick a template, and fill in its fields. Objects can also reference one another (for example, a network-connection object referencing the file that initiated it). See the MISP objects chapter for the full workflow, including object references and relationships.

14.3.3 Add Attachment

Use **Add Attachment** to upload files — malware samples, external analysis reports, or artefacts dropped by the malware.

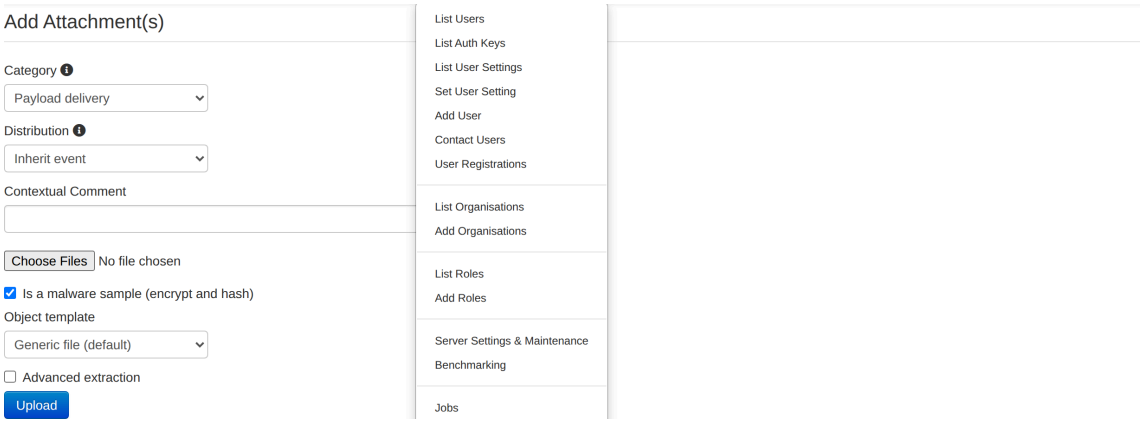


Figure 14.3: Point the uploader at the file you want to attach. Mark it as a malware sample so it is neutralised on storage.

- **Category:** What the file describes (defaults to *Payload delivery*).
- **Distribution:** Who can see the attachment; the same options as attributes.
- **Contextual Comment:** Optional notes.
- **Values:** The file(s) to upload — you can select several at once.
- **Is a malware sample (encrypt and hash):** Checked by default. When set, the file is zipped and password-protected (password `infected`) so users cannot accidentally execute it, and it is stored as a `malware-sample` object with its hashes computed automatically. Untick it only for benign files, which are stored as plain `attachment` attributes.
- **Object template / Advanced extraction:** Choose the object template used to represent the upload; when advanced extraction is installed, MISP can pull additional attributes out of the file (for example from a PE binary).

14.3.4 Populate from...

When you already have a list of indicators or a structured export, use **Populate from...** to create many attributes at once. The button opens a chooser offering:

- **MISP JSON** — merge event content from a MISP JSON file into this event.
- **Freetext Import** — paste any block of text (a report, an email, a list of IOCs) and let MISP find and classify the indicators.
- **Populate using a Template** — fill a guided, sectioned form to generate attributes.
- **OpenIOC Import** — import indicators from a `.ioc` file.
- **ThreatConnect Import** — import from a ThreatConnect CSV export.
- **(Experimental) Forensic analysis – Mactime** — import a mactime timeline.
- **Any enabled import module** — additional formats (for example CSV, STIX, various vendor reports) provided by misp-modules appear here automatically.

The most commonly used option is **Freetext Import**: paste a line-break separated list of indicators and MISP suggests the most likely category/type for each one.

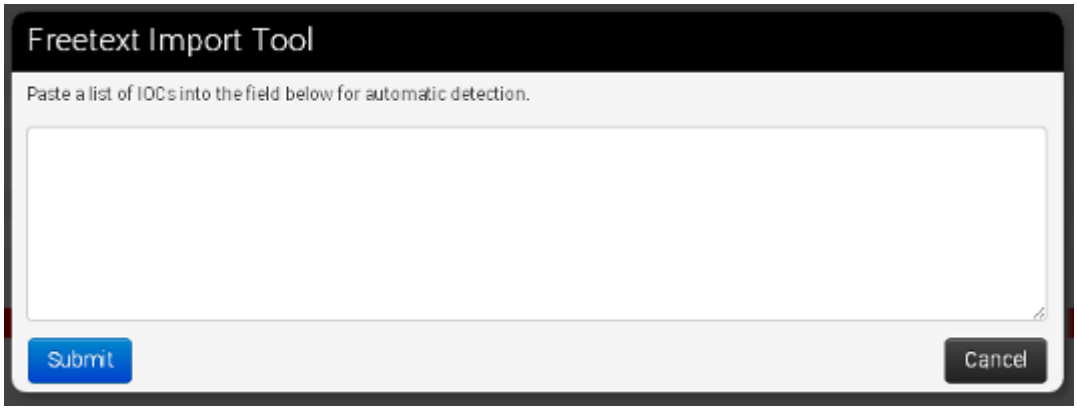


Figure 14.4: Paste a line-break separated list of indicators into the freetext import tool.

Because many values are valid for several category/type combinations, MISP proposes the most common option and lets you adjust the category, type and IDS flag per row before saving. Values that already exist elsewhere are flagged with their correlations, and values that hit a warninglist are highlighted so you can decide whether to keep them.

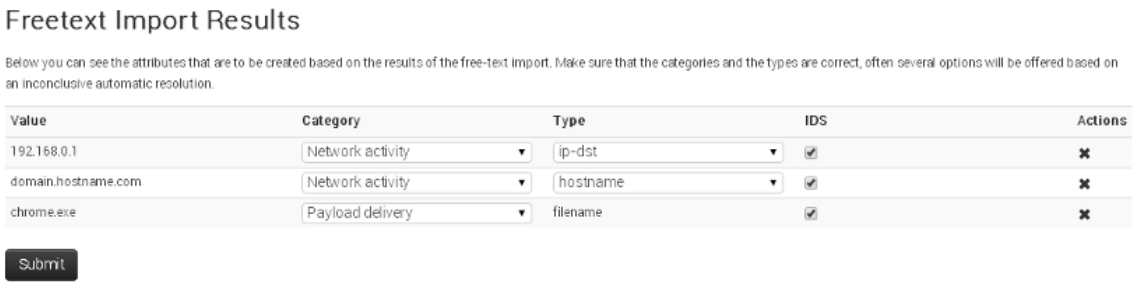


Figure 14.5: Review and adjust the proposed attributes — including warninglist hits and existing correlations — before saving.

14.4 Adding context to your data

Raw indicators are only half the story. MISP offers several complementary ways to say what your data *means* and how much to trust it.

14.4.1 Tagging: taxonomies, galaxies and tag collections

Tags classify events and attributes. Next to the **Tags** field on an event (and on each attribute) you will find two add buttons:

- **Add a tag** (globe icon) attaches a **global** tag that is shared and synchronised along with the event.
- **Add a local tag** (user icon) attaches a **local** tag that stays on your instance only — it is never exported or synchronised. Local tags can only be applied by users of the instance's host organisation.

Clicking either button opens a two-step picker. The first step lets you choose from your **Favourite Tags**, **Tag Collections**, **Custom Tags**, **All Tags**, or a specific **Taxonomy Library** (one entry per enabled taxonomy). The second step lets you pick the actual tag/predicate value. Taxonomies are curated, machine-readable tag vocabularies (TLP, PAP, admiralty-scale, and many more) — see the Taxonomies chapter.

Galaxy clusters are attached through a separate control (the **Galaxies** field, with its own *Add new cluster* / *Add new local cluster* buttons). While they are stored as tags under the hood, galaxies represent richer knowledge objects — threat actors, malware families, MITRE ATT&CK techniques, sectors, and so on — with descriptions, synonyms and relationships. See the Galaxies chapter.

Tag collections let you apply a curated bundle of tags in one action — pick the *Tag Collections* entry in the tag picker and the collection's member tags are expanded onto the target.

14.4.2 Analyst data: notes, opinions and relationships

Beyond tags, MISP lets analysts annotate almost any element — events, attributes, objects, galaxy clusters, event reports, and even other annotations — with **analyst data**:

- **Notes** — free-text commentary in Markdown.
- **Opinions** — an explicit confidence/agreement score (0–100) with an optional comment, so consumers can see whether the community trusts a piece of data.
- **Relationships** — typed links from one element to another (including elements in other events), building a graph of related intelligence.

Analyst data is created from the context icon shown next to an element's UUID and is shared and synchronised alongside the data it annotates.

14.4.3 Enrichment

MISP can automatically look up extra information for your data through misp-modules:

- **Enrich Event** (left-hand menu) runs the applicable expansion modules against the whole event as a background job and appends the results.
- On an individual attribute, the **Add enrichment** action (asterisk icon) queries expansion modules that support that attribute's type; **Query Cortex** does the same via Cortex analysers where configured.
- If hover enrichment is enabled on your instance, hovering over (or clicking the magnifier on) a value shows on-the-fly lookups without leaving the page.

14.4.4 Decaying scores

If your administrator has configured decaying models, attributes can carry a **decaying score** column showing, per model, how relevant the indicator still is over time (green while valid, red once it has decayed). Clicking a score opens the decay simulation tool. Decaying lets consumers automatically age out stale indicators instead of trusting them forever.

14.5 The event view

The event view is where you spend most of your time. It shows the event's metadata, a set of collapsible content panels, and the full attribute/object table with inline tools.

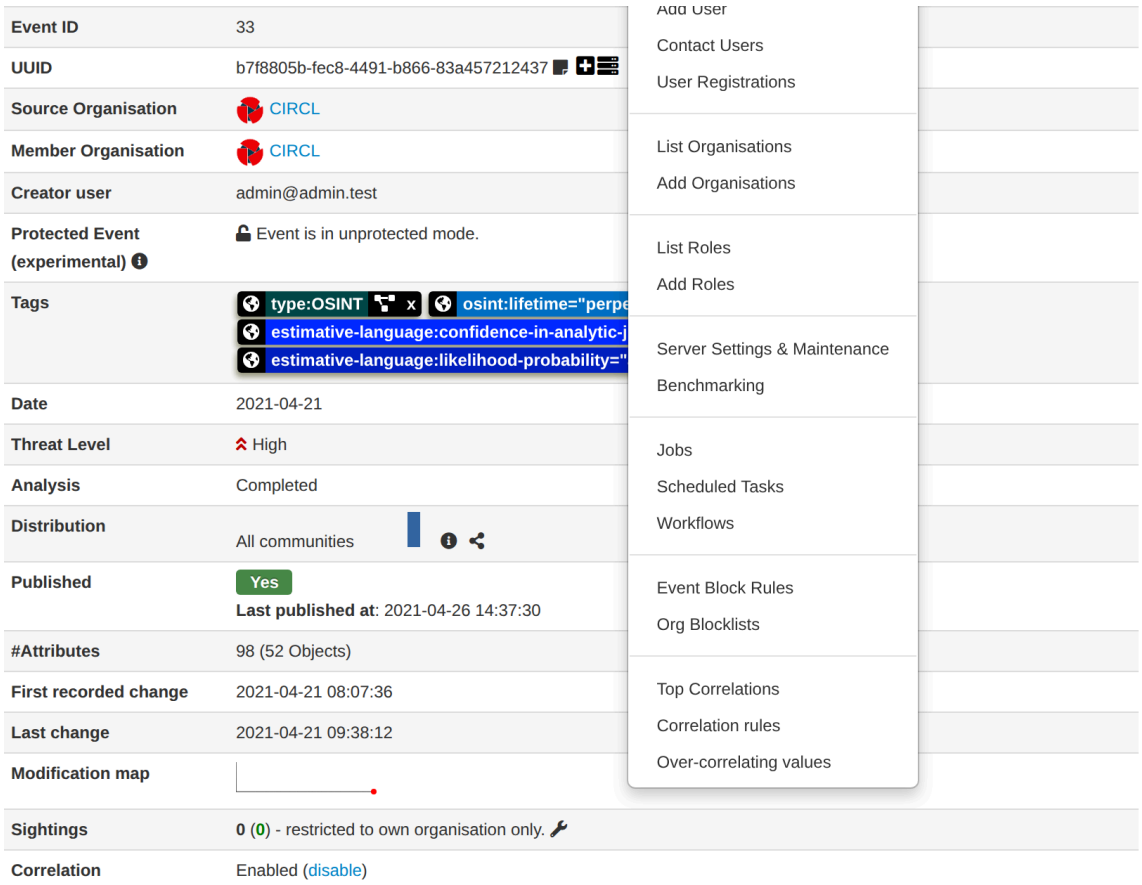


Figure 14.6: The event view: general information, collapsible content panels, and the attribute table with inline tools.

14.5.1 General event information

The information block at the top of the view shows (depending on your permissions and the event's state):

- **Event ID** and **UUID** — the UUID row also offers quick actions to *extend this event* and to *check this event on other servers*.
- **Creator org** / **Owner org** — the organisation that authored the event versus the organisation that owns the local copy (the latter is visible to site admins).
- **Contributors** — organisations that contributed via proposals; click a logo to see a filtered history.
- **Protected Event** — when present, the event is cryptographically signed so recipients can verify it has not been tampered with in transit.
- **Tags**, **Date**, **Threat Level**, **Analysis**, **Distribution** — the metadata set on creation, editable inline.
- **Warnings** — data-quality or security warnings about the event.
- **Published** — whether the event is published, together with its first- and last-published timestamps.
- **#Attributes** — the attribute count, with the number of objects in parentheses.
- **First recorded change** / **Last change** / **Modification map** — when the event changed, with a sparkline of edit activity.
- **Extends** / **Extended by** — links to the parent or child events when this event is part of an extended event.
- **Sightings** / **Activity** — how many times the event's data has been seen (with an activity sparkline) and a shortcut to advanced sightings.
- **Correlation** — whether correlation is enabled for the event, with a toggle to turn it off (useful for large, low-value events).

The right-hand column adds side panels for **related events**, **related feeds** and **related servers** where applicable.

14.5.2 Content toggles

A row of toggle buttons controls which panels are shown in the main content area. You can collapse and expand each independently:



Figure 14.7: The content-toggle buttons above the event view.

1. **Pivots** — the pivot thread graph of events you have traversed (see below).
2. **Galaxy** — the galaxy clusters attached to the event.
3. **Event graph** — an interactive graph of the event's objects, attributes and their references/relationships.
4. **Event timeline** — a time-based view built from the attributes' first/last seen values.
5. **Correlation graph** — a graph of the events this one correlates with.
6. **Galaxy matrix** — the ATT&CK-style matrix highlighting the techniques present in the event.
7. **Event reports** — the rich-text event reports attached to the event (this panel opens automatically when reports exist).
8. **Attributes** — the attribute and object table (see below).
9. **Discussion** — the event's discussion thread (shown when you have access to discussions).

When you open the **Galaxy matrix** panel on an event carrying ATT&CK galaxy clusters, MISP highlights the techniques present in the event on the matrix:

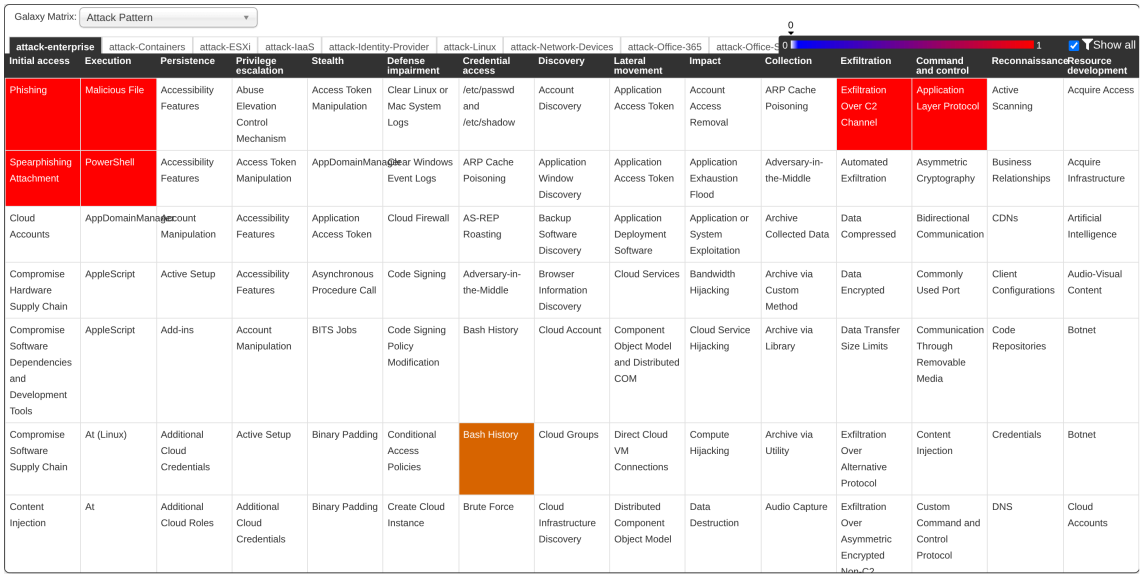


Figure 14.8: The Galaxy matrix panel highlighting the event's ATT&CK techniques.

14.5.3 The attribute table

The **Attributes** panel lists every attribute and object in the event. The columns include the date, a context icon (analyst notes/opinions/relationships), category, type, value, tags, galaxies, comment, correlation and related events, IDS flag, distribution, sightings and activity, and — when enabled — SightingDB and decaying-score columns.

Each row offers inline tools (subject to your permissions):

- **Inline edit** — click a category, type, value, comment or distribution cell to edit it in place.
- **Sightings** — thumbs-up to add a sighting, thumbs-down to mark a false positive, and a wrench for advanced sighting options. See Sightings.
- **Enrichment** — the asterisk (expansion modules) and eye (Cortex) icons, as described under Enrichment.

- **Edit / Delete** — standard actions; deletion is a soft delete on published events (so it can propagate) and can be restored, or a permanent delete on events that were never published.
- **Tag / Galaxy** — add tags or galaxy clusters directly on the row.

Attributes that belong to an object are grouped under that object; proposals appear inline with the attribute they affect (see Proposing changes).

14.5.4 Pivoting and related events

Two events are *related* when they share one or more attribute values (for example both contain `11.11.11.11`). Related events are listed in the right-hand panel as links. Moving from one event to another through these links is called **pivoting**, and MISP records the path in the **Pivots** panel: each visited event is a bubble connected to the one you came from, letting you branch, jump back, and prune the graph. The pivot thread resets when you navigate away from the event view.

14.5.5 Event discussion

Every event has its own discussion thread where eligible users can talk about it. Posters are shown by organisation and an anonymised user identifier (their email is only visible to members of their own organisation). Use the quick-response field at the bottom of the thread to post without reloading the page, and the `[quote]`, `[event]` and `[thread]` tags to quote a post or link to another event or thread.

14.6 Collaborating on events

14.6.1 Proposing changes to another organisation's event

If you can see an event created by another organisation but cannot edit it directly, you can suggest changes as **proposals** (shadow attributes). Use the propose buttons that replace the add/edit controls to propose a new attribute or a change to an existing one; the creating organisation can then accept or discard your proposal.



Figure 14.9: An attribute with a proposal attached is highlighted; a standalone grey proposal is a suggested new attribute.

When the creating organisation is on another instance, they receive your proposal on their next pull, and republishing the event sends the accepted change back to you. For richer, non-destructive collaboration (commentary, confidence, typed links) consider analyst data instead.

14.6.2 Sightings

A **sighting** records that you actually observed an indicator (or that it is a false positive). Sightings are a lightweight, high-signal form of feedback that feeds decaying and lets producers see which indicators are still active. Add them from the thumbs-up/thumbs-down icons on each attribute row, or in bulk. See the Sightings chapter for sighting policies, anonymisation and SightingDB.

14.6.3 Contacting the reporter

To reach the organisation that reported an event, open the event and click **Contact Reporter** (available when the creating organisation is local to your instance). Your message is emailed together with the event's details.

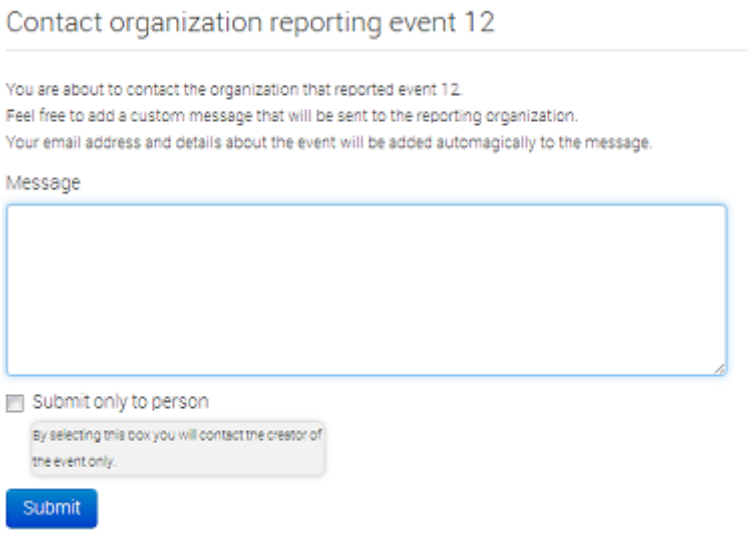


Figure 14.10: Enter your message and choose whether the whole reporting organisation or only the reporter receives it.

By default every member of the reporting organisation is emailed; tick the checkbox to reach only the individual reporter.

14.6.4 Delegating an event

You can ask another organisation to (re)publish an event on your behalf — for example to publish anonymously through a trusted partner. Use **Delegate Publishing** from the event menu; the recipient sees the request under *View delegation requests* and can accept or discard it. See the Delegation chapter for the full flow.

14.7 Publishing an event

Once the event holds everything you want to share, finalise it by publishing.

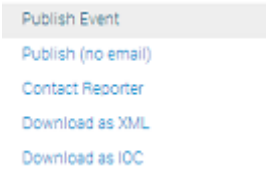


Figure 14.11: Use “Publish (no email)” for minor changes such as fixing a typo.

Publishing an event:

- makes its data available to all eligible exports (and readies IDS-flagged attributes for detection-rule generation);
- emails subscribed, eligible users (those with auto-alert enabled and the right to see the event); and
- pushes the event to the instances you synchronise with, from where it propagates further according to the distribution rules.

The event menu offers several variants:

- **Publish** — the normal publish, with notification emails.
- **Publish (no email)** — publish without alerting anyone; use it for minor edits.
- **Unpublish** — retract a published event.
- **Publish Sightings** — push sighting updates without republishing the whole event.

- **Delegate Publishing** — hand publication to another organisation (see above).
- **Publish to ZMQ / Kafka** — emit the event to the real-time ZeroMQ or Kafka feeds, when those plugins are enabled.

If your instance has background jobs enabled, publication is queued and may not complete instantly.

14.8 Extending events

An **extended event** lets you build on an existing event without editing it — ideal for adding your own analysis to a report you received, or for maintaining a long-running campaign event as a series of contributions. Create one either by entering the parent's UUID in the **Extends Event** field when creating an event, or with **Extend this event** from an existing event's UUID row. The event view can then show the combined (extended) picture, and search filters let you include or isolate extensions. See the Extended events chapter.

14.9 Browsing and searching

14.9.1 Listing events

List Events (left-hand menu) shows the most recent events. The list includes, among other columns, the published state, creator and owner organisations, tags and galaxies, the attribute count, the event date, threat level, analysis stage, the event info, distribution, and per-row action buttons (view, edit, publish, delete — subject to your permissions).

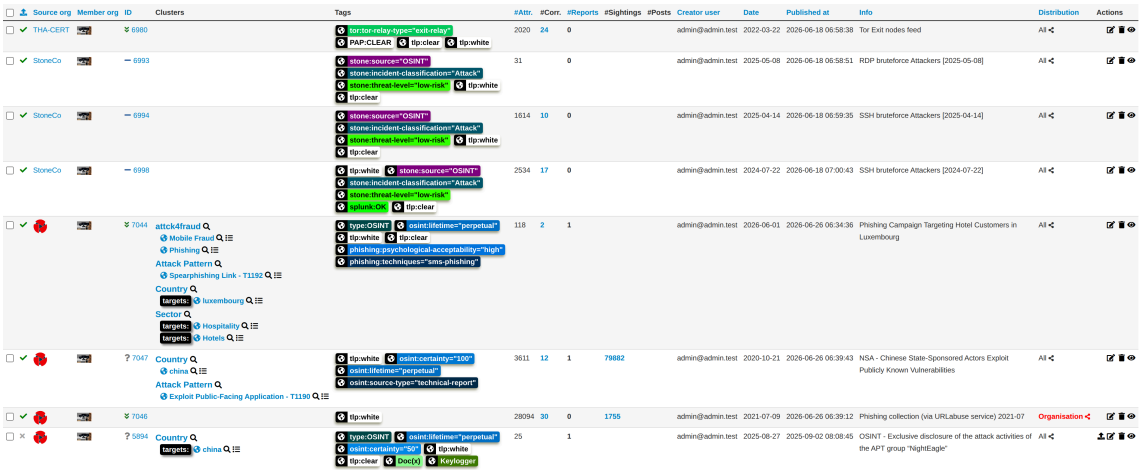


Figure 14.12: The event index. Use the controls to filter, view, or act on events.

Use the filter/query controls at the top of the index to narrow the list by any combination of criteria, and the scope tabs to switch between your organisation's events and everything you can see.

14.9.2 Listing and searching attributes

List Attributes produces a comparable list at the attribute level.

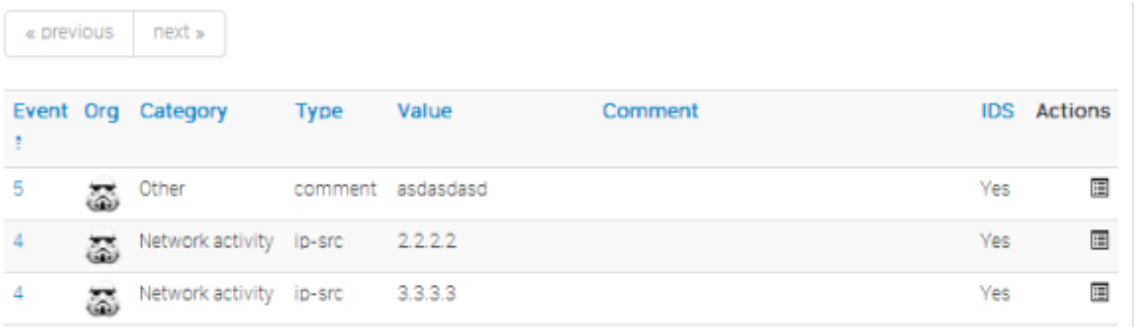


Figure 14.13: Use the buttons on the right to view the owning event or to edit/delete the attribute.

To find specific data, use the attribute search, which is backed by the same powerful query engine as the REST API: search by value, type, category, tag, organisation, date range and more, and export the result set directly. Because the search UI and the API share the same filters, anything you can find interactively you can also automate.

14.10 Editing events and attributes

Every event and attribute can be edited. Find it (by browsing or searching), then use the edit control — for attributes you can also edit most fields inline in the attribute table, or double-click an event in a list to jump straight into edit mode. The edit form is the same as the corresponding creation form, pre-filled with the current values. Remember that editing an event (directly, or indirectly by changing one of its attributes) un-publishes it, so you will need to publish again once you are done.

14.11 Create and manage Sharing Groups

Sharing groups are re-usable, fine-grained distribution lists. They let you combine organisations from your own instance (local organisations) with organisations from directly or indirectly connected instances (external organisations), and control exactly which instances the data may travel to. Sharing groups can be created by any user with the *sharing group editor* permission, and edited by members of the creating organisation or of any organisation marked as an **extender** — the mechanism for delegating membership management to a trusted partner (for example a sector representative who knows who should be included).

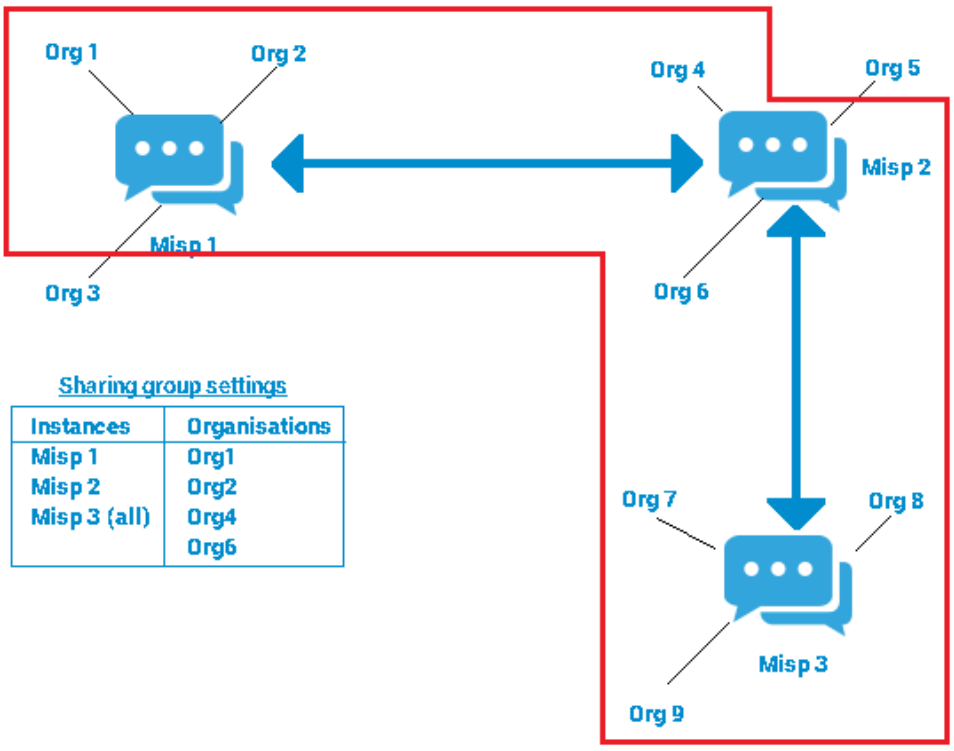


Figure 14.14: A sample sharing group involving three instances, showing the ways to include or exclude organisations.

Use sharing groups when the standard distribution levels cannot express the sharing you need; otherwise prefer the simpler distribution levels, and prefer the *Inherit event* distribution on attributes whenever their sharing group would match the event's. Sharing groups can add complexity for users and some processing overhead on the data marked with them.

Create and edit sharing groups from **Global Actions → List Sharing Groups / Add Sharing Group**. The editor is a wizard with four tabs:

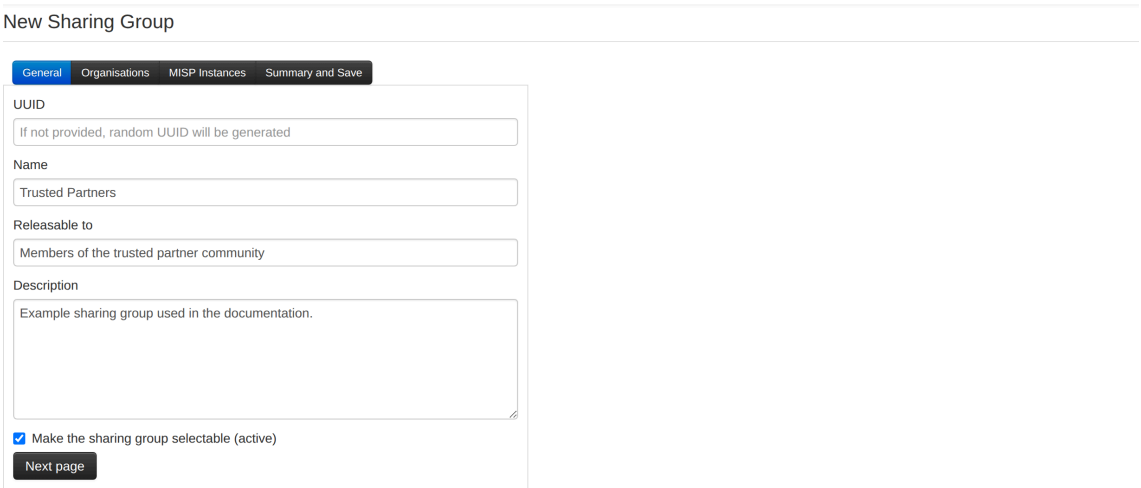


Figure 14.15: The General tab of the sharing group editor.

- **General:** Metadata describing the group.
  - **Name:** The unique name of the sharing group.
  - **Releasable to:** A human-readable description of who the data may be shared with. This is informational only (aimed at extender organisations); MISP does not enforce it.
  - **Description:** A natural-language statement of the group's intent.
  - **Make the sharing group selectable (active):** Uncheck to retire a group — existing data keeps honouring it, but it is no longer offered when setting distribution, reducing clutter.

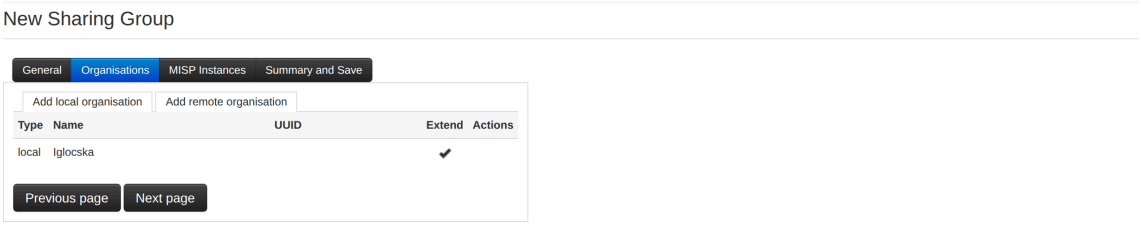


Figure 14.16: The Organisations tab of the sharing group editor.

- **Organisations:** The list of organisations named in the group, split into **local** and **remote/external** tabs. (Local organisations have at least one local user; remote organisations are created automatically when synchronising. A remote organisation can be converted to local if one of its users later joins your instance.) Tick **extend** to let an organisation edit the sharing group — the creating organisation is always an extender.



Figure 14.17: The MISP Instances tab of the sharing group editor.

- **MISP Instances:** The instances the data may be synchronised to. Any user who can see the event on a given instance may pull it, but the organisation distribution list still applies.
  - **Enable roaming mode:** Ignore the instance list and rely purely on the organisation list — an instance becomes eligible whenever its host organisation is in the list. This removes the need to know specific instance URLs, at the cost of relying on correctly configured host-org settings.
  - **Add instance:** Add an instance from those configured under *Sync Actions → List Servers*.
  - **All orgs:** Automatically include every organisation on the selected instance, without naming them individually — useful for exchanging with a whole community, and for privacy-sensitive communities that prefer not to enumerate members.

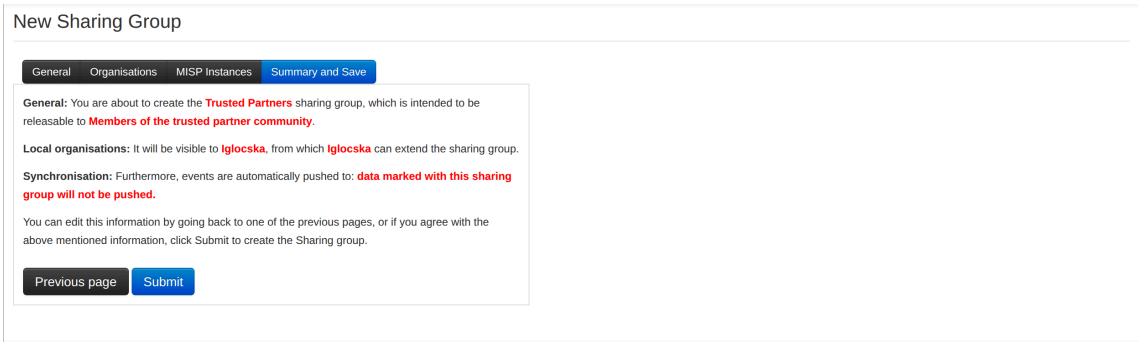


Figure 14.18: The Summary and Save tab of the sharing group editor.

- **Summary and Save:** A highlighted summary of the whole configuration. Review it carefully before saving — mistakes here can expose data to the wrong organisations or grant unwanted editing rights. Note that a sharing group has no effect until it is selected as the distribution of an event or attribute.

14.12 Exporting data

The recommended way to get data out of MISP is the search/export engine (`restSearch`), which is shared by the web UI, the built-in REST client and the automation API.

For a single event, use **Download as...** in the event menu to export it in any of the supported formats — MISP JSON/XML, OpenIOC, CSV (with or without context), STIX 1 and 2, RPZ, Suricata/Snort/Bro rules, a plain-text value list, and any enabled export module. Each option is generated on demand from the current data.

For bulk and automated export, drive `/events/restSearch` and `/attributes/restSearch` directly (or through the REST client), which accept the same filters as the search UI and return the same wide range of formats. See the automation chapter for the full parameter and format reference.

> **Note**
>
> The older two-page cached-export page (previously reached via *Export*) is disabled by default on current instances in favour of `restSearch`; you may not see it at all.

14.13 Connecting to other instances

Synchronising with other MISP servers — adding sync connections, configuring push and pull rules, and running feeds — is covered in the Sharing & synchronisation chapter (server administration) and the Managing feeds chapter. Setting up sync also depends on having an appropriately configured sync user and authentication key, described in Administration.

14.14 REST API

MISP exposes a full REST API for programmatic access to events, attributes and almost every other object, using JSON (or XML) over HTTP with an API key. The canonical reference — authentication, the `restSearch` filters, the complete list of return formats, and the interactive OpenAPI documentation and built-in REST client — lives in the Automation & API chapter. For Python, see PyMISP.

14.15 Multi-factor authentication (TOTP/HOTP)

MISP supports time-based and counter-based one-time-password multi-factor authentication (TOTP/HOTP). For how to enable and configure it instance-wide, see the Administration chapter.

14.15.1 Generating TOTP/HOTP tokens

From the top menu bar, go to **Global Actions → My Profile** and click the **TOTP Generate** button.

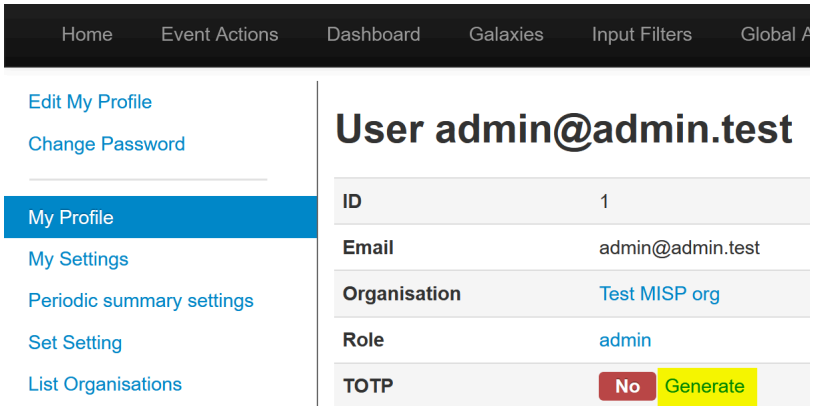


Figure 14.19: My Profile view with the TOTP generate button highlighted.

On the next screen, add the secret to your preferred authenticator application (for example a mobile authenticator or a desktop tool) and confirm the setup by entering a verification code.

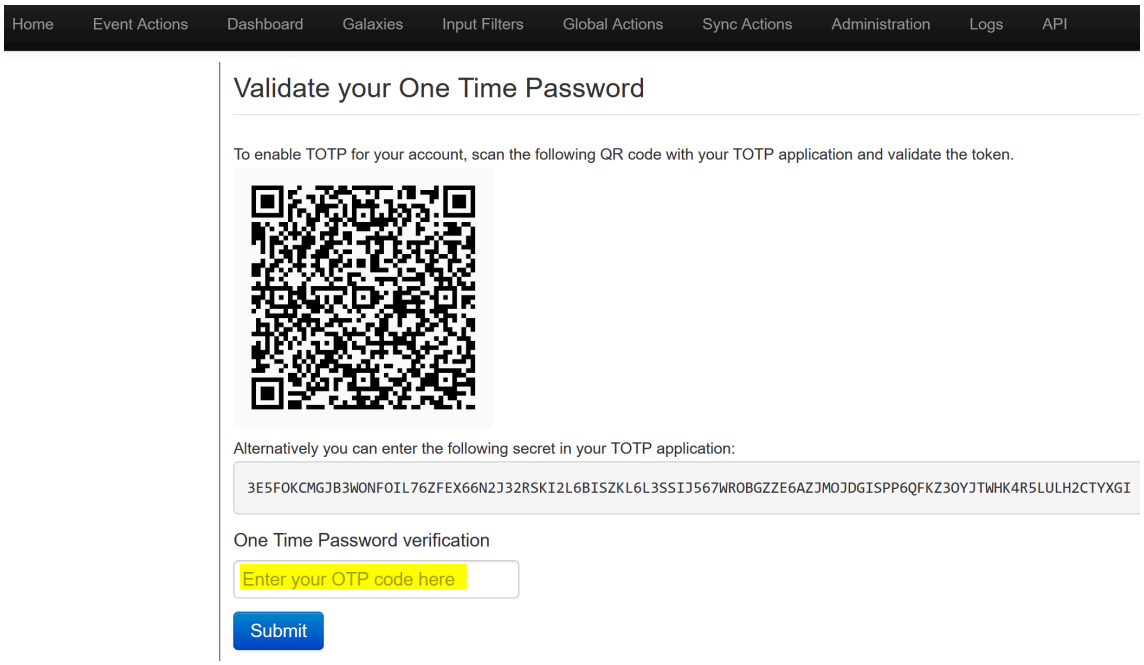


Figure 14.20: The TOTP setup and validation screen.

Once validated, you are shown a set of 50 single-use HOTP (paper) tokens as a fallback for when your authenticator is unavailable.

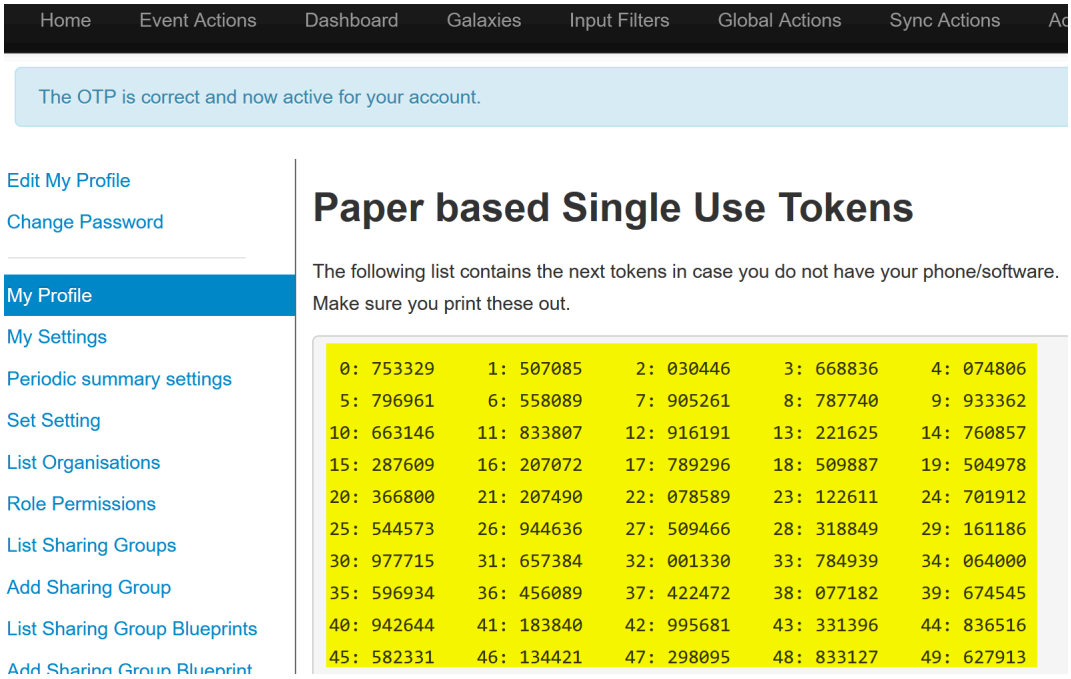


Figure 14.21: The single-use paper token page.

You can view these tokens again later from your profile via the **View paper tokens** button.

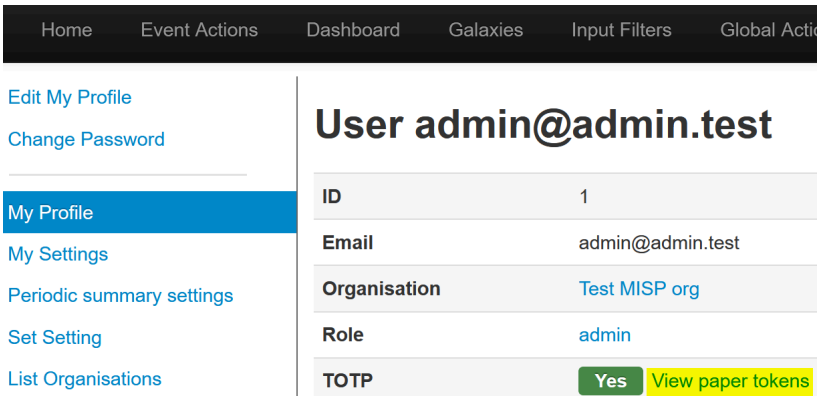


Figure 14.22: My Profile with the View paper tokens button highlighted.

14.15.2 Logging in with TOTP/HOTP

After setting up TOTP/HOTP, you are prompted for a one-time password on subsequent logins.

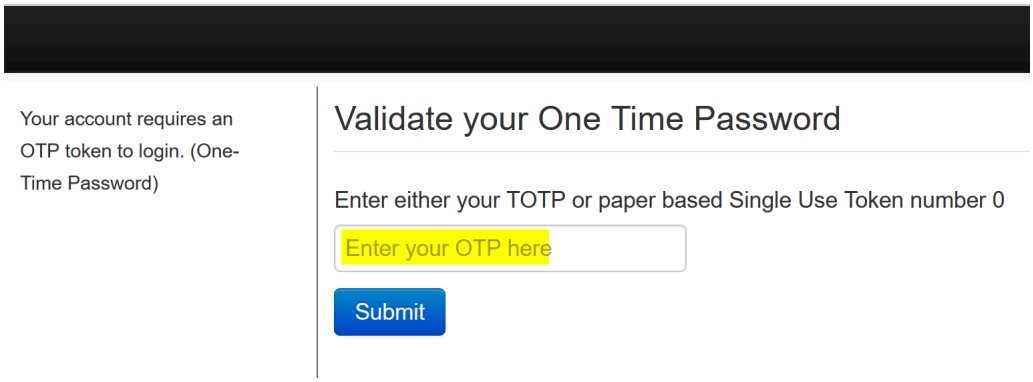


Figure 14.23: The OTP prompt shown after entering your credentials.

Enter either a code from your authenticator or one of the numbered paper tokens.

14.15.3 Deleting and regenerating tokens

Removing a user's TOTP/HOTP setup can only be done by site or organisation admins. Contact your organisation admin (preferred) or a site admin if you need to reset your tokens.

14.15.4 Combining multiple factors

You cannot combine multiple forms of multi-factor authentication at once: once TOTP/HOTP is assigned to your user, email OTP is not also used for it. If your instance has email OTP configured, it applies again once your TOTP/HOTP setup is removed.

15. MISP Extended Events

An **extended event** lets you build on top of an existing event without modifying it. Instead of editing someone else's event (which you often cannot do) or attaching proposals to it, you create your *own* event that declares itself an extension of the original. MISP can then present a **combined view** that merges the original event with all the events that extend it.

15.1 When to use an extended event

Extending is the right tool when you want to add your own analysis, indicators or context to an event while keeping a clear separation between the original and your contribution:

- **Adding to intelligence you received** — you pull a report from a partner or a feed and want to enrich it with your own findings, without altering the source event (so it can still be updated cleanly on the next sync).
- **Collaborative, long-running events** — several organisations each maintain their own extension of a shared "base" event (for example an ongoing campaign), keeping ownership of their own data.
- **Layering** — separating a raw, machine-generated base event from a curated, human-analysed layer on top.

Compare this with the alternatives: **editing** changes the event in place (and requires ownership); **proposals** suggest changes to the original organisation for them to accept; **extending** creates an independent, separately-owned event that references the original.

15.2 Creating an extended event

There are two ways to create one:

- **From the event you want to extend** — open the original event and use **Extend this event** (the extend button on the event's UUID row). This starts a new event with the parent already filled in.
- **From the Add Event form** — enter the parent event's UUID (or ID) in the **Extends Event** field when creating an event. A live preview of the parent is shown as you type.

Your extension is a normal event in every other respect: it has its own distribution, its own attributes and objects, and its own owner. It simply records, via its `extends_uuid`, which event it builds upon. An event can extend another event on the same instance or one received from a remote instance.

15.3 Viewing extended events

The single-event view understands the extension relationship:

- The **general information** block shows **Extends** (the parent this event builds on) and **Extended by** (the events that extend this one), as links.
- You can switch between viewing an event on its own and the **Extended view**, which merges the parent and all of its extensions into one combined picture — useful for seeing the full intelligence across all contributors at once. The page title indicates when you are in *Extended view* (looking at a parent with its extensions merged in) or *Extending view* (looking at an extension in the context of its parent).

15.4 Finding extended events

When browsing or searching events you can filter on the extension relationship. The event index can show an **Is extension** column, and the search/REST API accepts:

- **`extended`** — return the combined (extended) form of the matching events.
- **`is_extended`** — restrict to events that are extended by at least one other event.
- **`is_extension`** — restrict to events that are themselves extensions of another event.

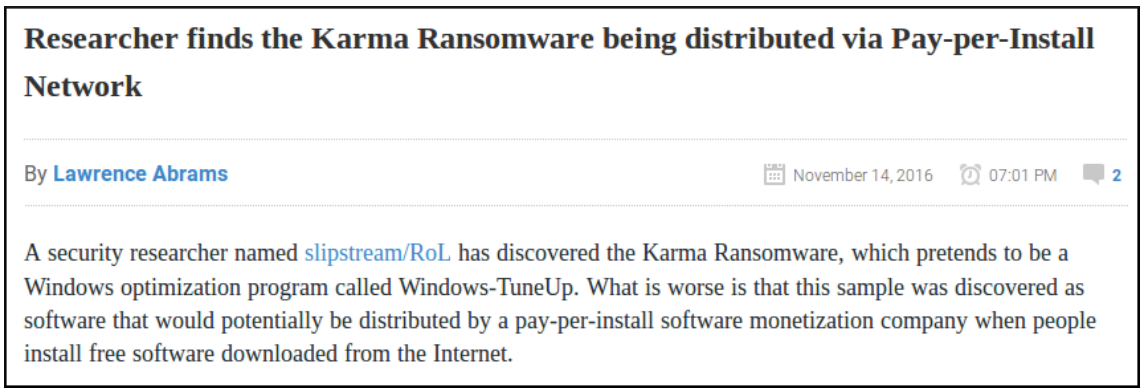
15.5 Further reading

For the original design write-up and worked examples, see the MISP project blog post Extended Events.

16. Create an Event Based on a Report

> ⚠ Warning
>
> A specific permission is required to create an event.

For this example, we will use a report published on Bleeping Computer, which we will treat as OSINT.



Researcher finds the Karma Ransomware being distributed via Pay-per-Install Network

By **Lawrence Abrams** November 14, 2016 07:01 PM 2

A security researcher named slipstream/RoL has discovered the Karma Ransomware, which pretends to be a Windows optimization program called Windows-TuneUp. What is worse is that this sample was discovered as software that would potentially be distributed by a pay-per-install software monetization company when people install free software downloaded from the Internet.

Figure 16.1: Report title

16.0.1 Adding an event

First, create a new event by clicking **Add Event** in the Events list view.

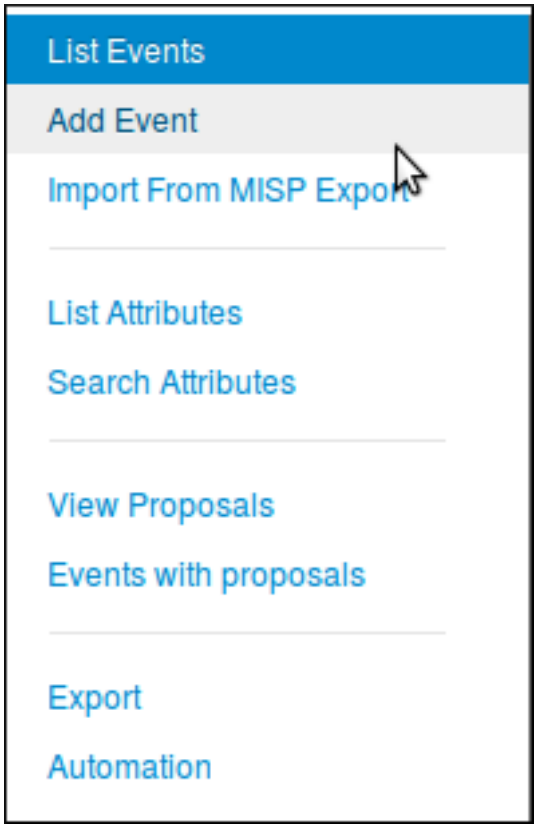


Figure 16.2: Add Event Option

This opens the add event form.

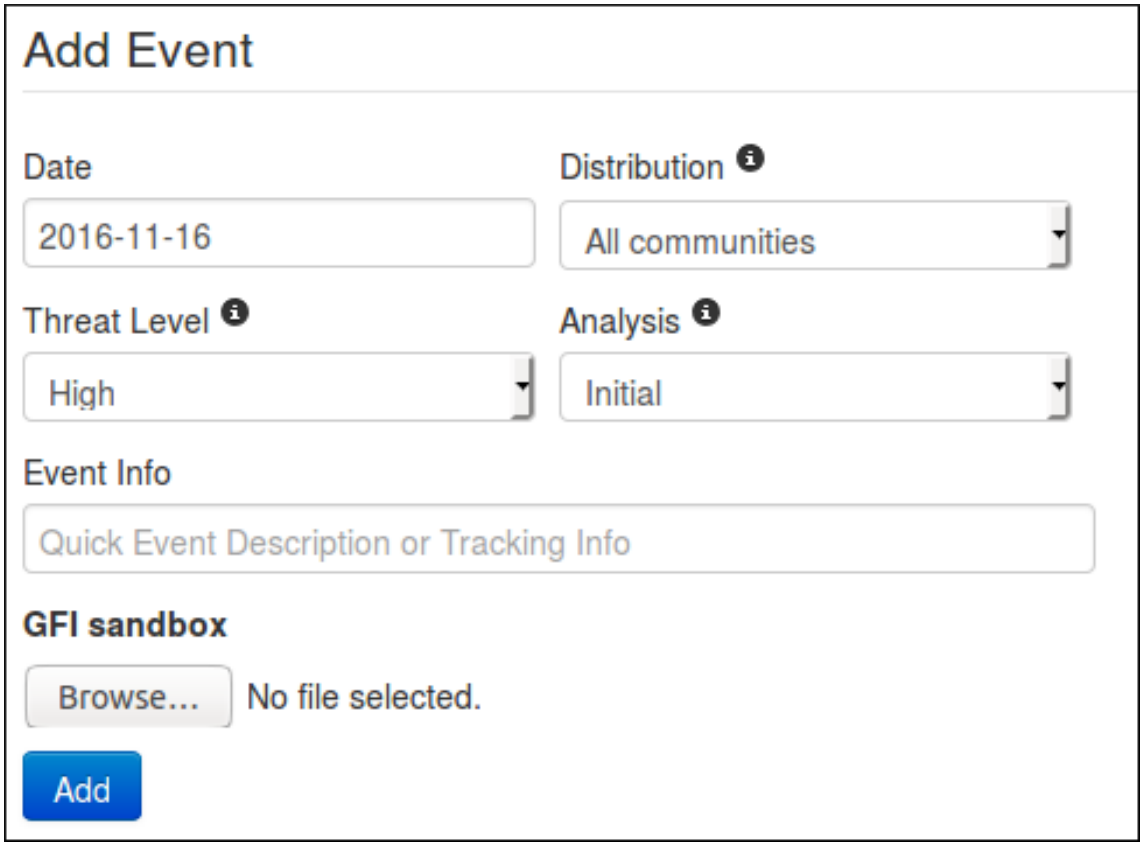


Figure 16.3: Add Event form

Let's populate it with the information we already have: * Date: Here we will put the date of the report, so 2016-11-14 * Distribution: Depending on the event, we might want it to be more or less spread across the MISP instances. For this one, since it is a public report, there is no reason to limit the diffusion so "All communities". * Threat Level: This is self-explanatory. Since the ransomware described in the report does not appear to use a major exploit, we can choose **Low** or **Undefined**. We will use **Undefined** because it can be refined later. * Analysis: Give the current stage of the analysis. Since the report is published, we can assume that the analysis is completed. * Event Info: This field is effectively the event title, so it makes sense to use the report title here. Because the source is public, we also prefix it with `OSINT`.

> **Note**
>
> The add-event form no longer has the legacy *GFI sandbox* field. Depending on your version it also offers a few optional fields — most notably **Extends event** (to make this event an extension of another), and, when you set **Distribution** to *Sharing group*, a selector for which sharing group to use. See the Using the system chapter for a full reference to the current add-event form.

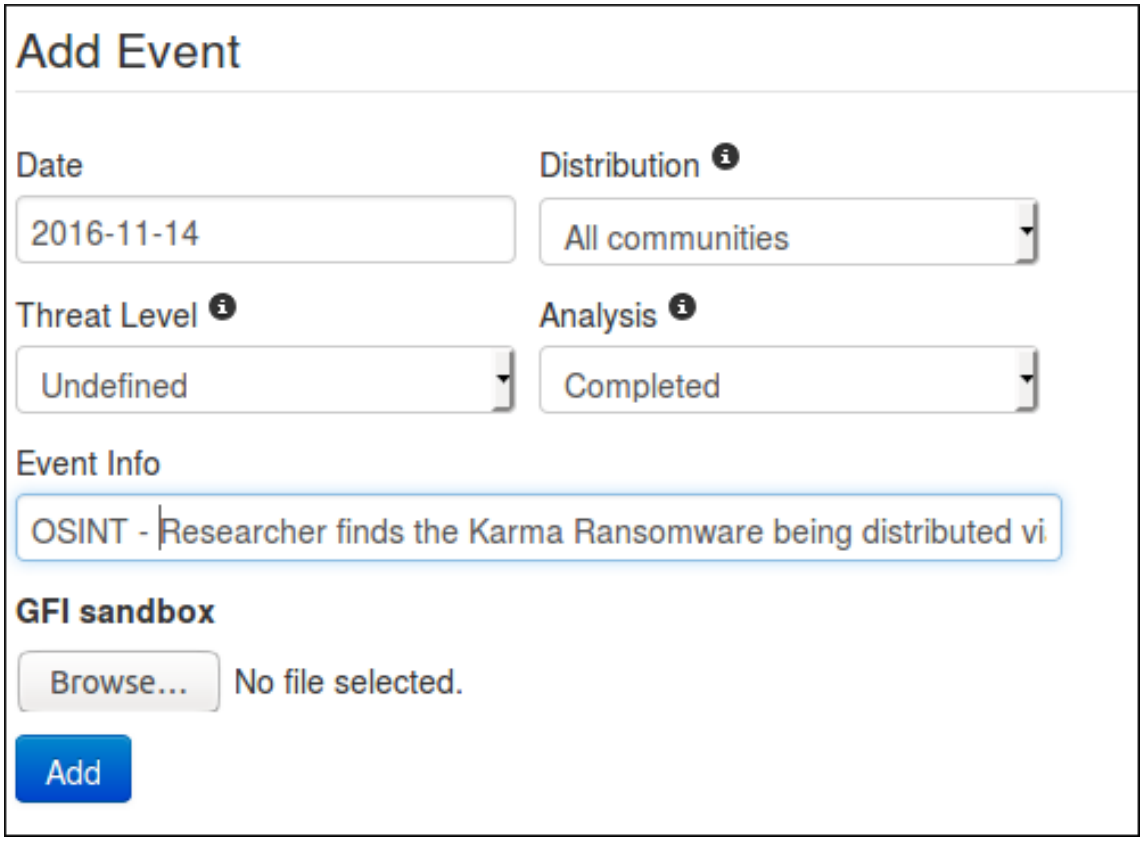
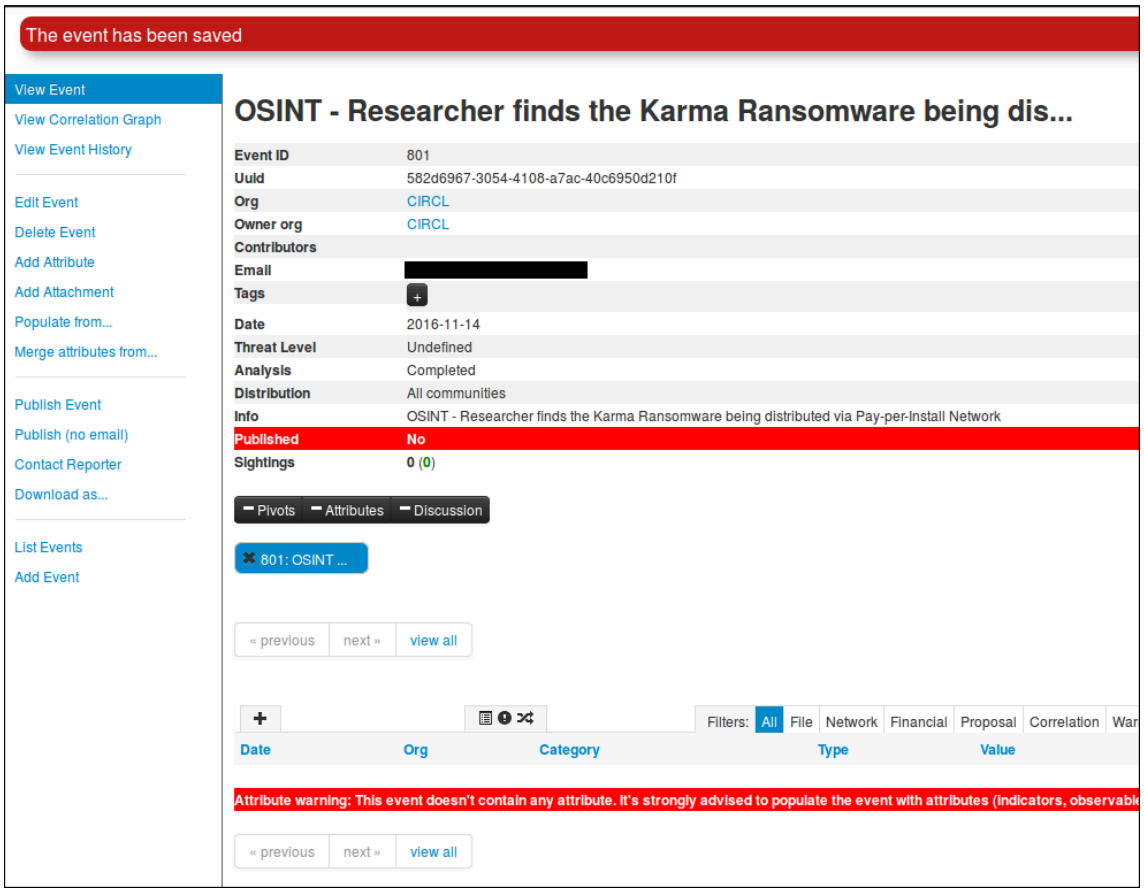


Figure 16.4: Add Event form filled

Click the blue **Add** button to create the event. You now have a brand-new, empty event.



(Displayed information can change depending on your role on the MISP instance)

16.0.2 Adding Attributes

Now it is time to populate the event. Before adding IoCs, we will add general information about the report itself: the report link and a short introduction. Click **Add Attribute** in the side menu to open the following view:

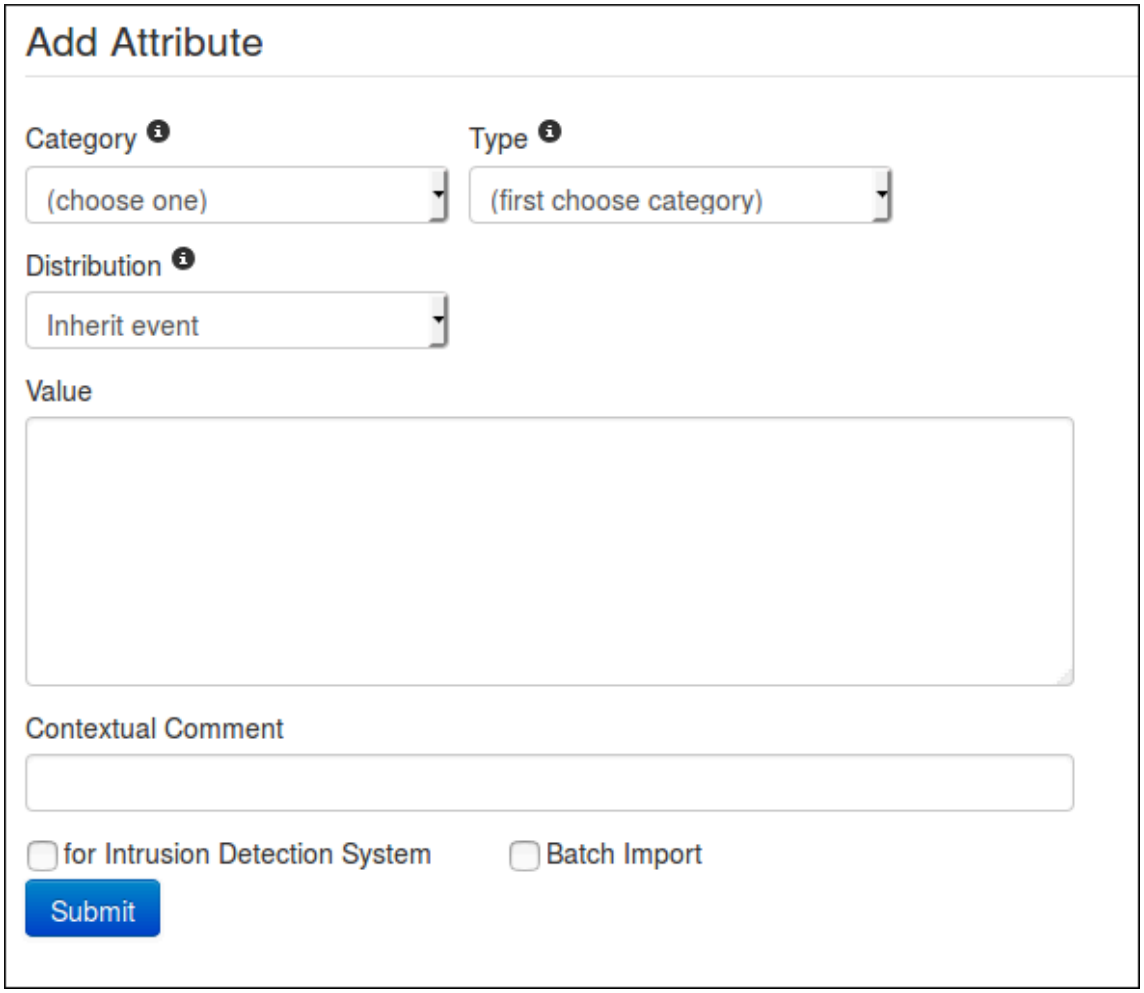


Figure 16.5: add attribute

- First, add the link to the report. Because it was written by another researcher, we will treat it as **External analysis** and choose that category.
- For the type, choose **link** because we are adding a URL.
- The distribution field can be a little tricky. We can either choose one of the option that was already available at event level or "Inherit event". If we choose the latter, the attribute will be shared the same way as the event it is included in (here to "All communities"). On the other hand, if we choose manually a distribution for the attribute, the most restritive between event distribution and attribute distribution will be applied. That is to say: if both event and attribute distributions are the same, there will be no change (similar to "Inherit event"). However, if for instance the event distribution is "all communities" while the attribute is limited to "This community only", the event will indeed be distributed to all communities but without this particular attribute which will be limited to this community only. The same works the other way around, if the attribute can be distributed to "all communities" while the related event is limited to this community, the attribute being dependant of the event, it will be shared to this community only, basing its distribution on the event (most restrictive) one.
- The value is the data we want to add, which in this case is the report URL.
- The contextual comment is not used for correlation. It is mainly there to provide supporting information about the attribute, such as a port number for an IP address or some other qualifier. Here we can note that the link is the source report.
- "for Intrusion Detection System" is used to set the IDS flag or not. If set, the attribute will be used as an IDS signature when exporting the NIDS data. In this case, we have no reason to check it.
- Batch Import is useful when you need to add several IoCs of the same category and type by separating each value with a line break. We do not need it here.

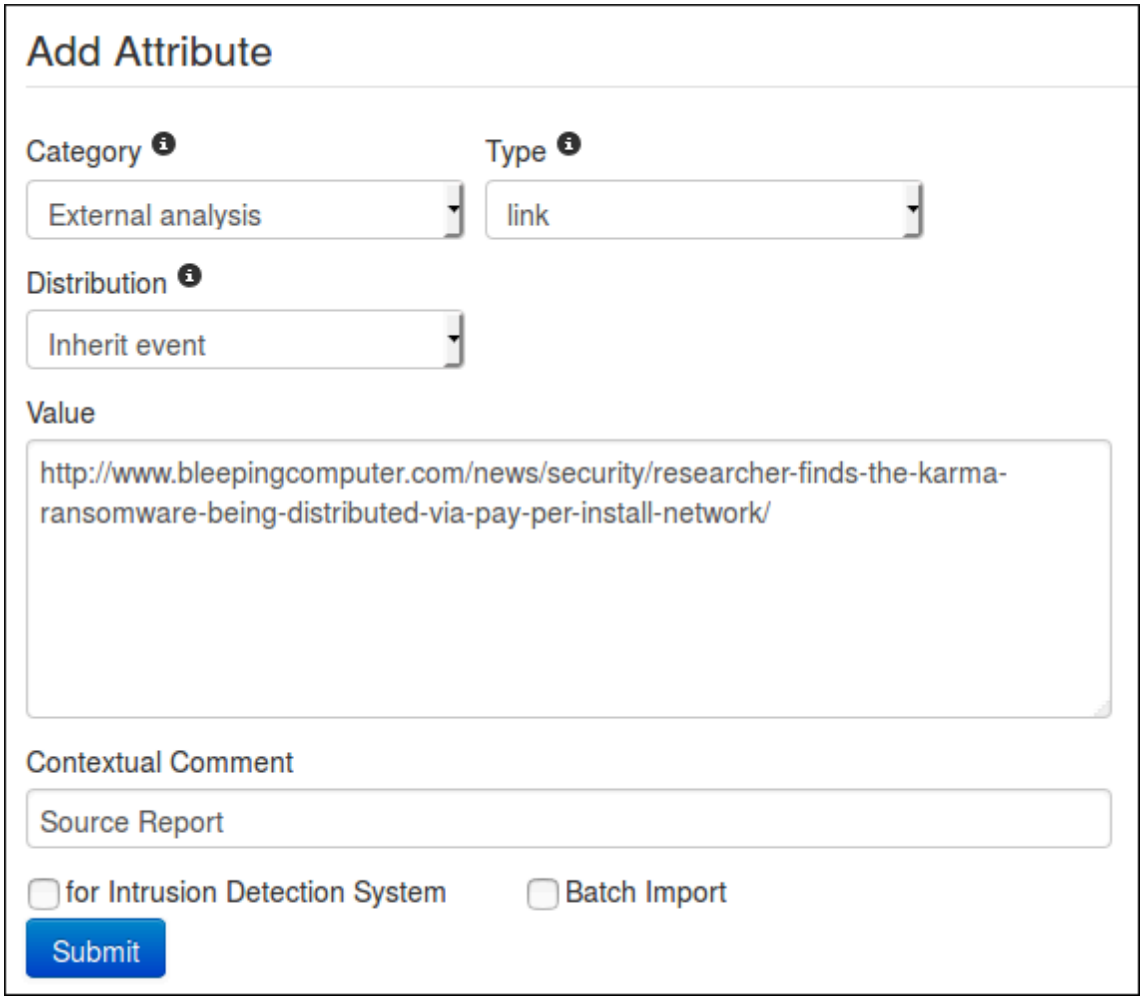


Figure 16.6: add attribute filled

Once all fields are filled in, click **Submit**.

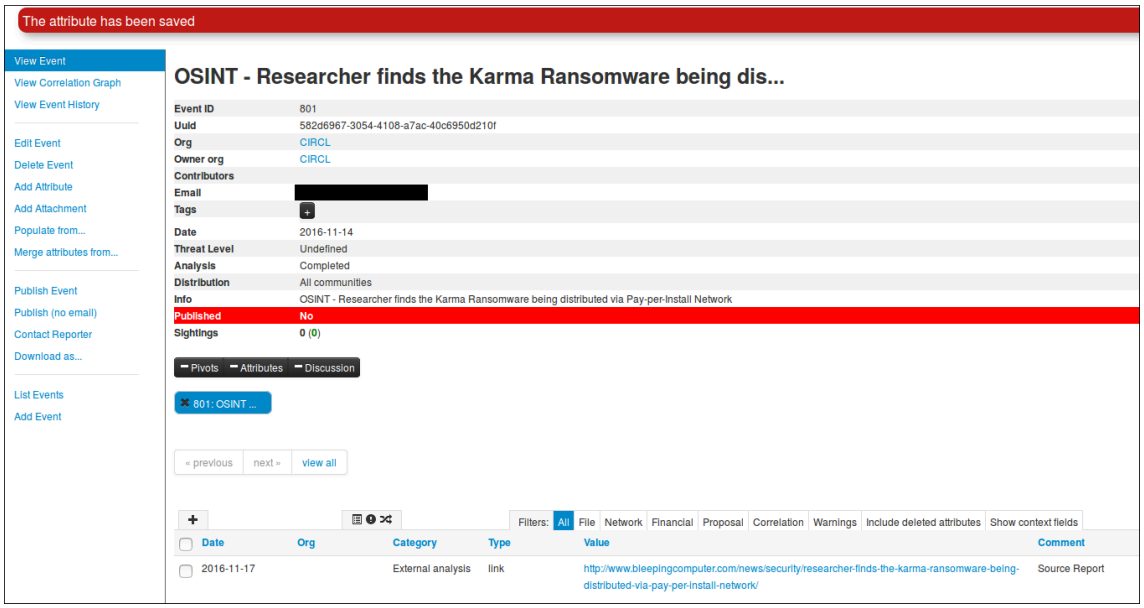


Figure 16.7: attribute added YAY

We can now follow a similar process to add an introduction to the report, meaning the first paragraph of the article. This time, change the type to **text** and open the add attribute form by clicking the small + icon next to the attribute table.



Figure 16.8: Noooo you found meeeeee

The same form appears in a pop-up.

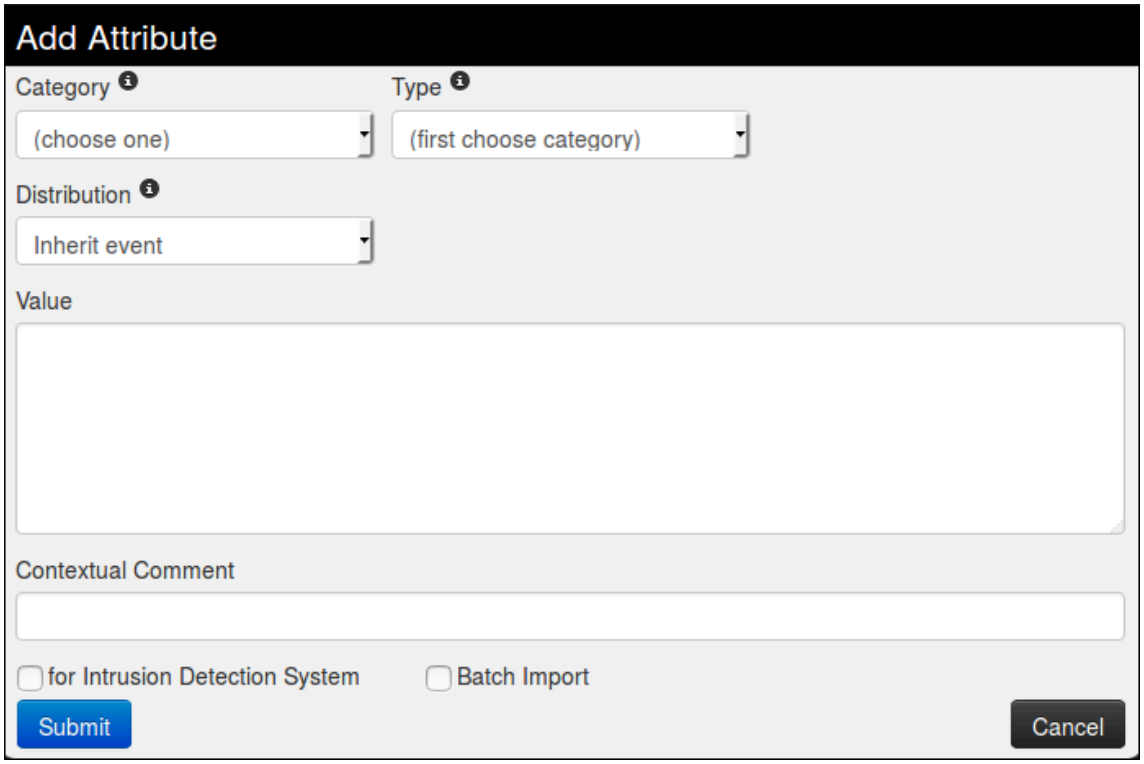


Figure 16.9: add attribute popup

Fill it in with the required data.

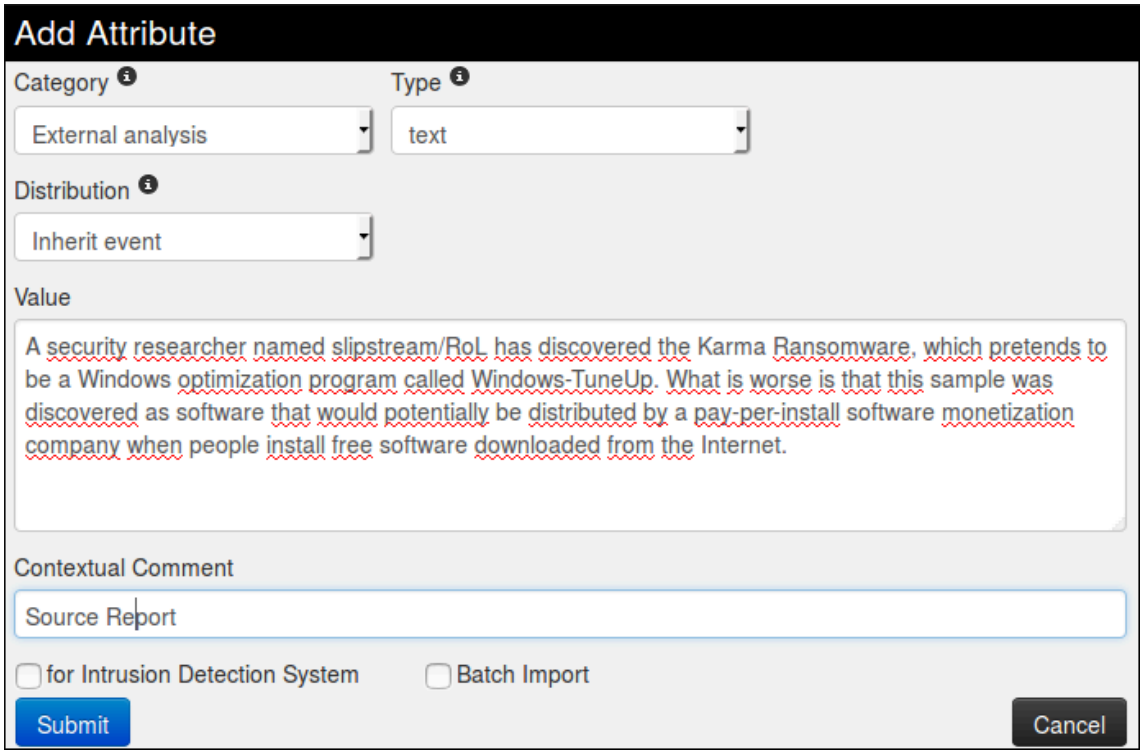


Figure 16.10: add attribute popup

Then submit it by clicking the blue button. *Et voilà!*

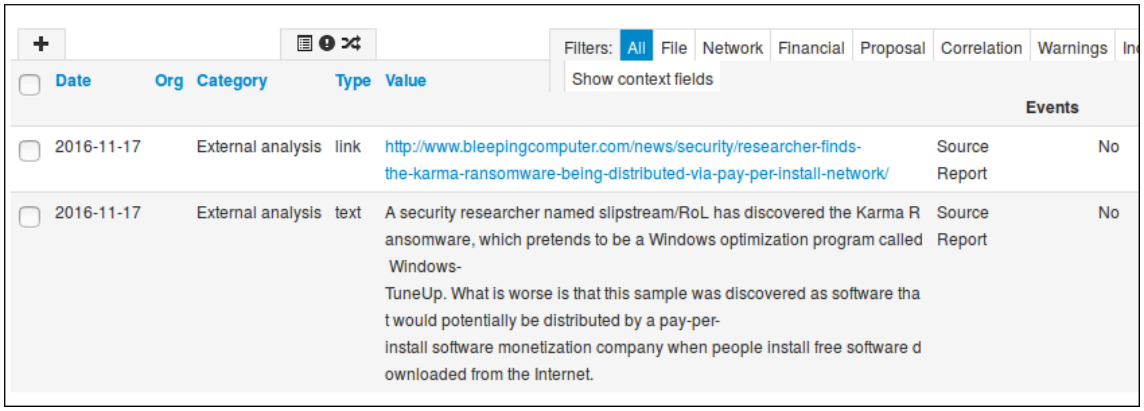
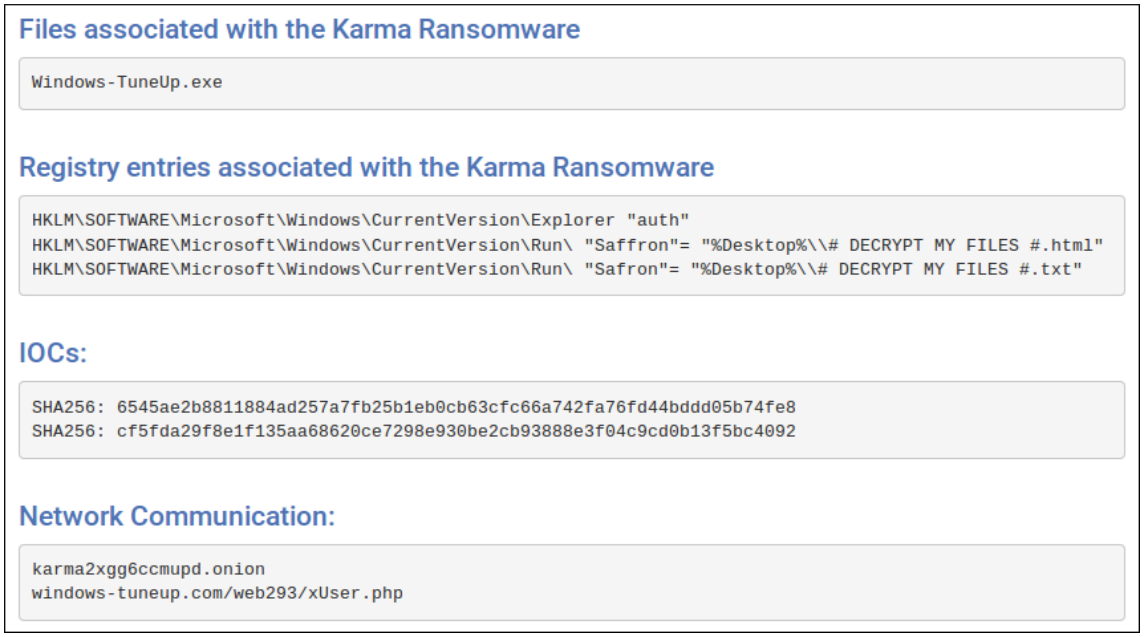


Figure 16.11: MAGIC

Okay, now it is time to add some Indicators of Compromise. In this report, they are mainly listed at the end.



Files associated with the Karma Ransomware

```
Windows-TuneUp.exe
```

Registry entries associated with the Karma Ransomware

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer "auth"
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ "Saffron"= "%Desktop%\\# DECRYPT MY FILES #.html"
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ "Safron"= "%Desktop%\\# DECRYPT MY FILES #.txt"
```

IOCs:

```
SHA256: 6545ae2b8811884ad257a7fb25b1eb0cb63cfc66a742fa76fd44bddd05b74fe8
SHA256: cf5fda29f8e1f135aa68620ce7298e930be2cb93888e3f04c9cd0b13f5bc4092
```

Network Communication:

```
karma2xgg6ccmupd.onion
windows-tuneup.com/web293/xUser.php
```

Figure 16.12: OMG IOC

Let's try to define which category/type those IoC belong to.

First, Windows-TuneUp.exe is without a doubt a *filename*, and the associated category may be *Payload delivery*.

Second the registry entries (type *regkey*) seems to be from *Artifacts dropped* category

Then the hashes that are already said to be *SHA 256*, and a quick test on VirusTotal also reveals that they correspond to the filename seen earlier. so we can add both as an association *filename|SHA256*. Once again, the category will be *Payload delivery*.

And finally the network communication. No doubt here for the category: *Network activity*, and the type might be *url* but for the example, we will let MISP decide for us.

So we begin with the filename. No real change from before for this one, except that we will set the IDS flag to true.

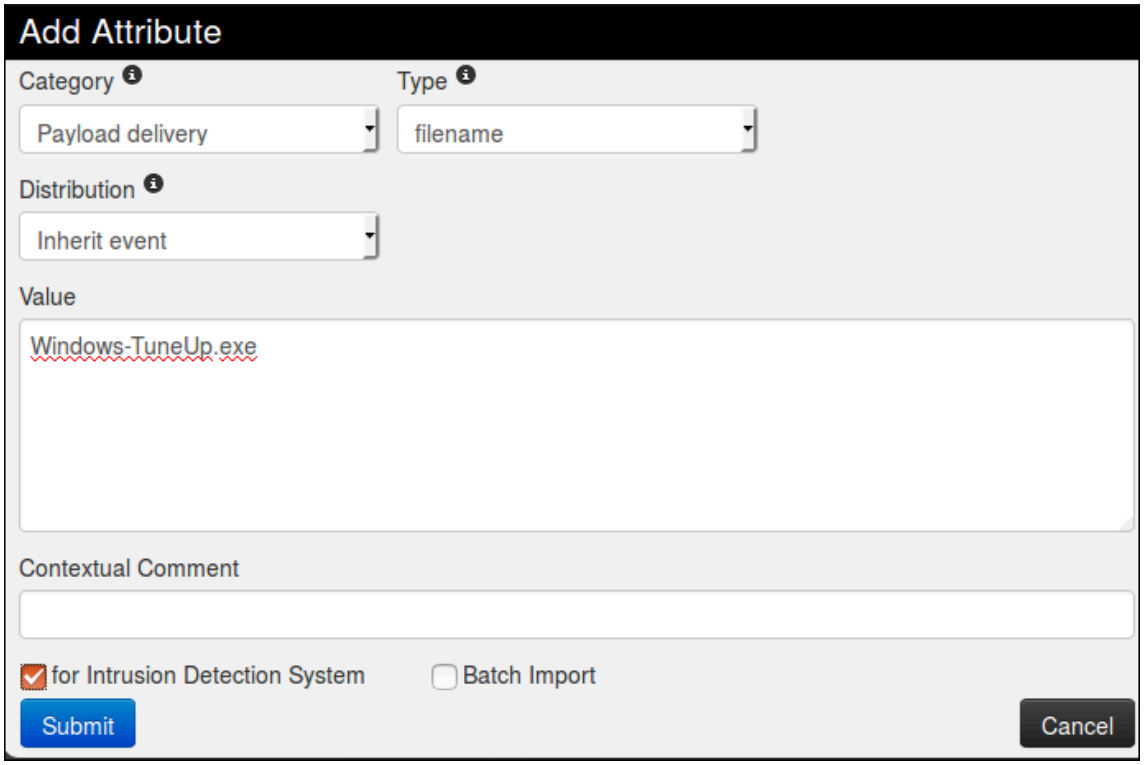


Figure 16.13: filename

16.0.3 Freetext Import Tool

Then we can add the hashes in a similar way. We will had them both alone and combined with the filename. In order to do it quickly, we are going to use the freetext import tool, hidden there



Figure 16.14: freetext import step 1

It will open a popup with a text area field where we will paste our IoC, one per line. As said previously, we add both the hashes alone and with the filename.

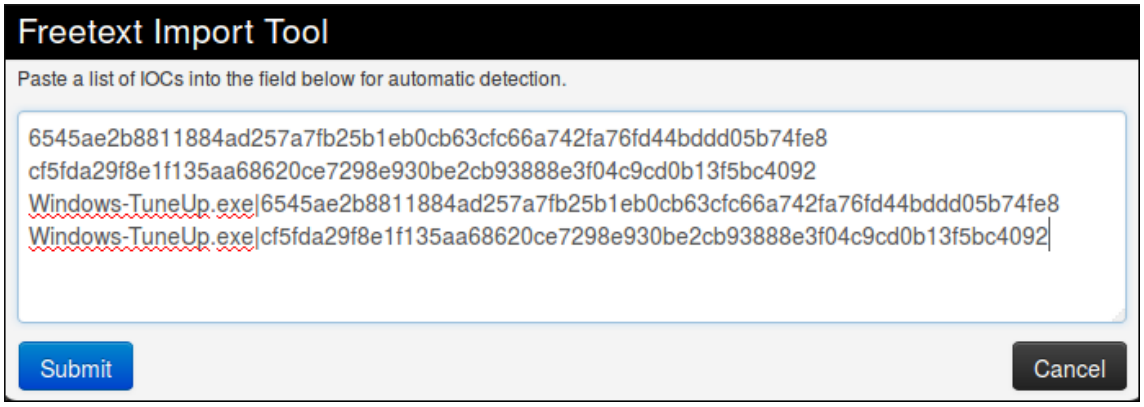


Figure 16.15: freetext import step 2

Then when we press the submit button, we are redirected on this page to control the sent data.

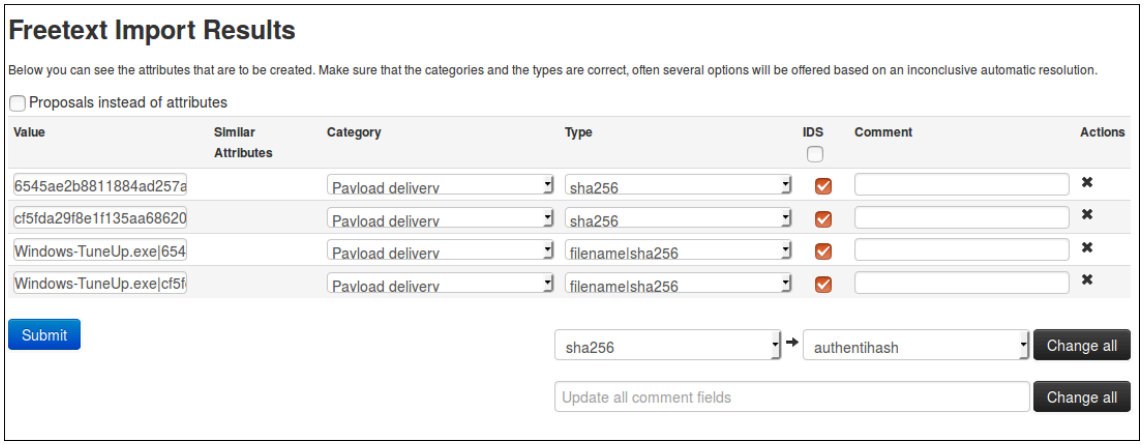
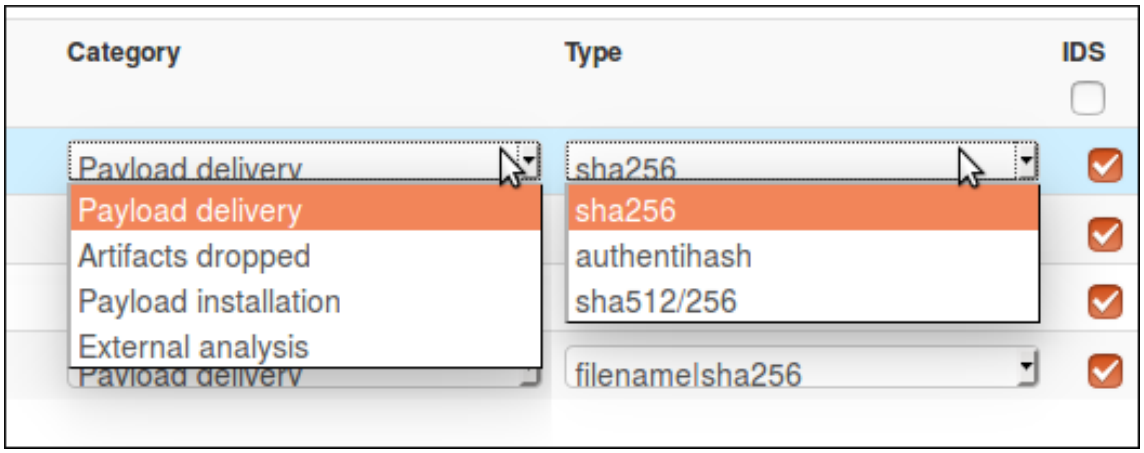


Figure 16.16: freetext import step 3

Here, MISP detected by itself what should be the category and type associated to our IoC and surprise! It matches our suppositions. Plus, it also put the IDS flag, so it is perfect. But before submitting, please double check to be sure all the values are correct and no information was lost (That can happen when the data are not formatted as expected by MISP).

If the results of MISP were not what we expected, we can still modify it, however MISP will only suggest suitable category/type regarding the format of your data. We can change for each attribute individually or all at the same time using the option on the bottom right of the form. The same principle also applies for the comments, individually or for all.



(Yes I have two cursors, MISP is magic!)

We only have the network indicators left, and as said before, we will let MISP determined for us which type is the best for the data we have.

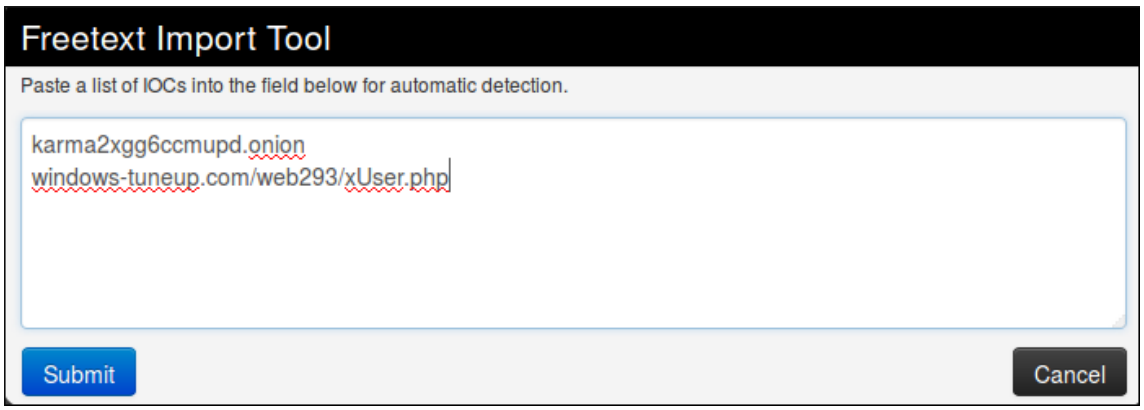


Figure 16.17: freetext import network

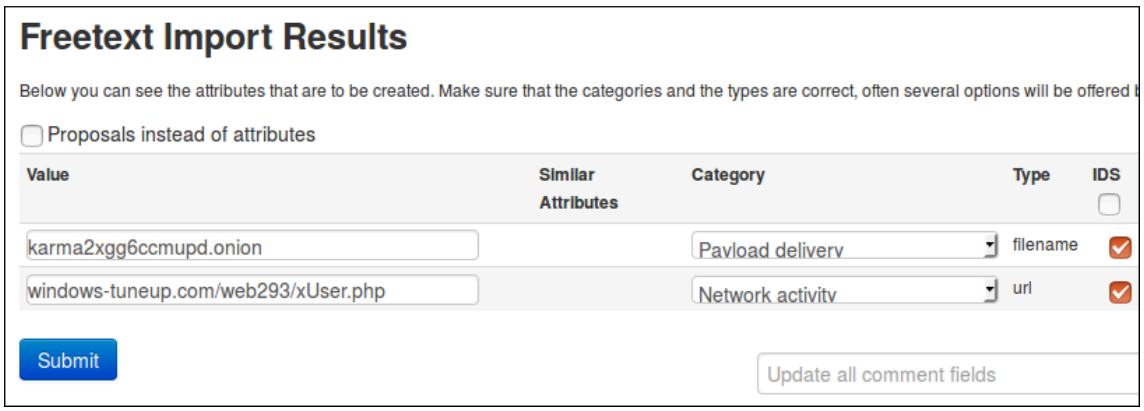


Figure 16.18: type recognition fail

Oh well, that was unexpected. In fact, it is not that surprising regarding the format of the tor address that look more like a filename than like a url but it is still a problem, since we can't change the type nor the category to a more consistent one. This is indeed one of the limitation of freetext import. To solve this issue, we will use a simple trick: we will add a slash at the end of the tor address so it won't be confused for a filename.

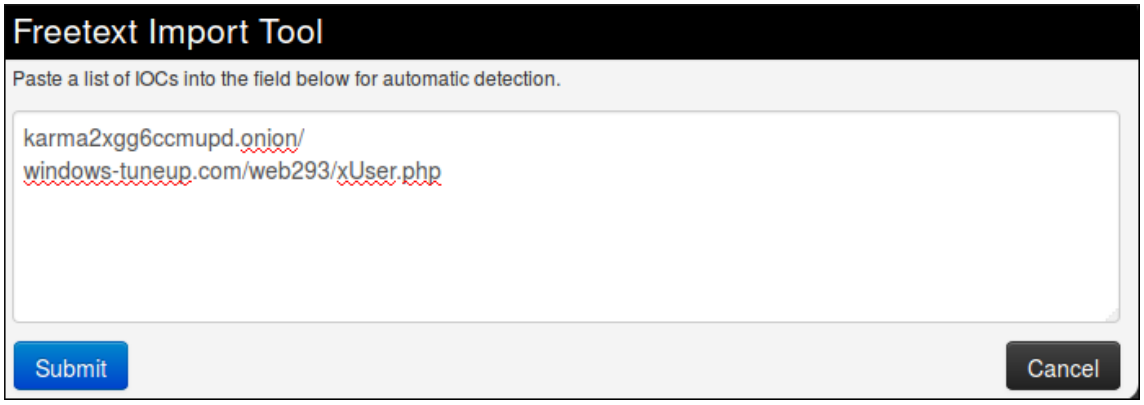


Figure 16.19: freetext import network

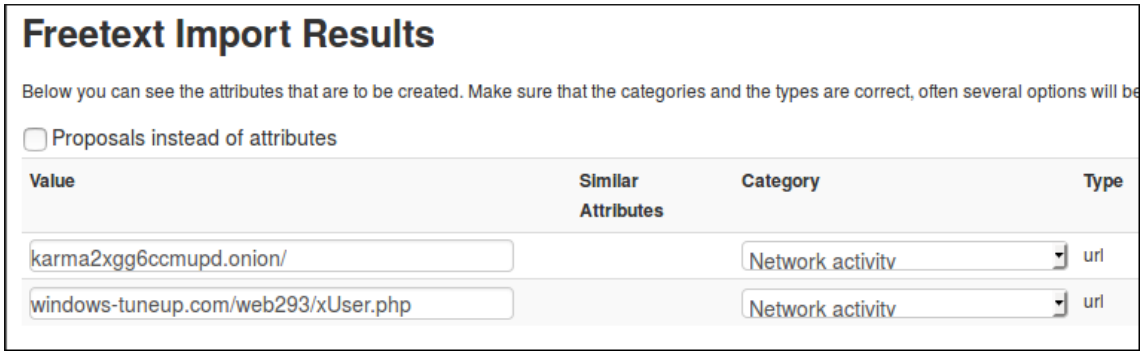


Figure 16.20: type recognition fail

Thanks to the added character, the first string is recognised as an url which is more consistent with the reality. The second also seems okay, so we can now submit both.

16.0.4 Batch Import

The Freetext Import works properly only with a string of data without any spaces in one line. But if you have lines of text with spaces between values, like e.g.

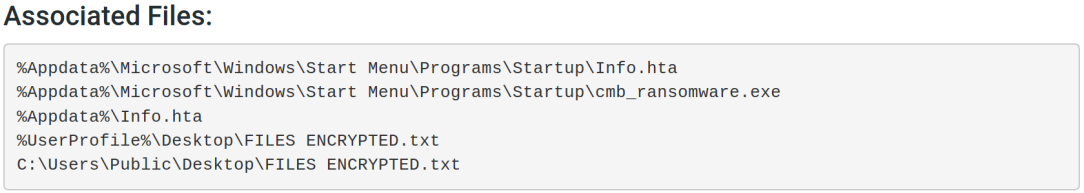


Figure 16.21: freetext_with_spaces1

you can still import them at once using the "Add Attribute" option. Click on *Add Attribute*, copy the data and paste it into the *Value* box. Choose the right category and type. Now check both checkboxes *for Intrusion Detection System* and *Batch Import*. The option *Batch Import* will import your data line for line just like the *Freetext Import* option without losing any information. Like this:

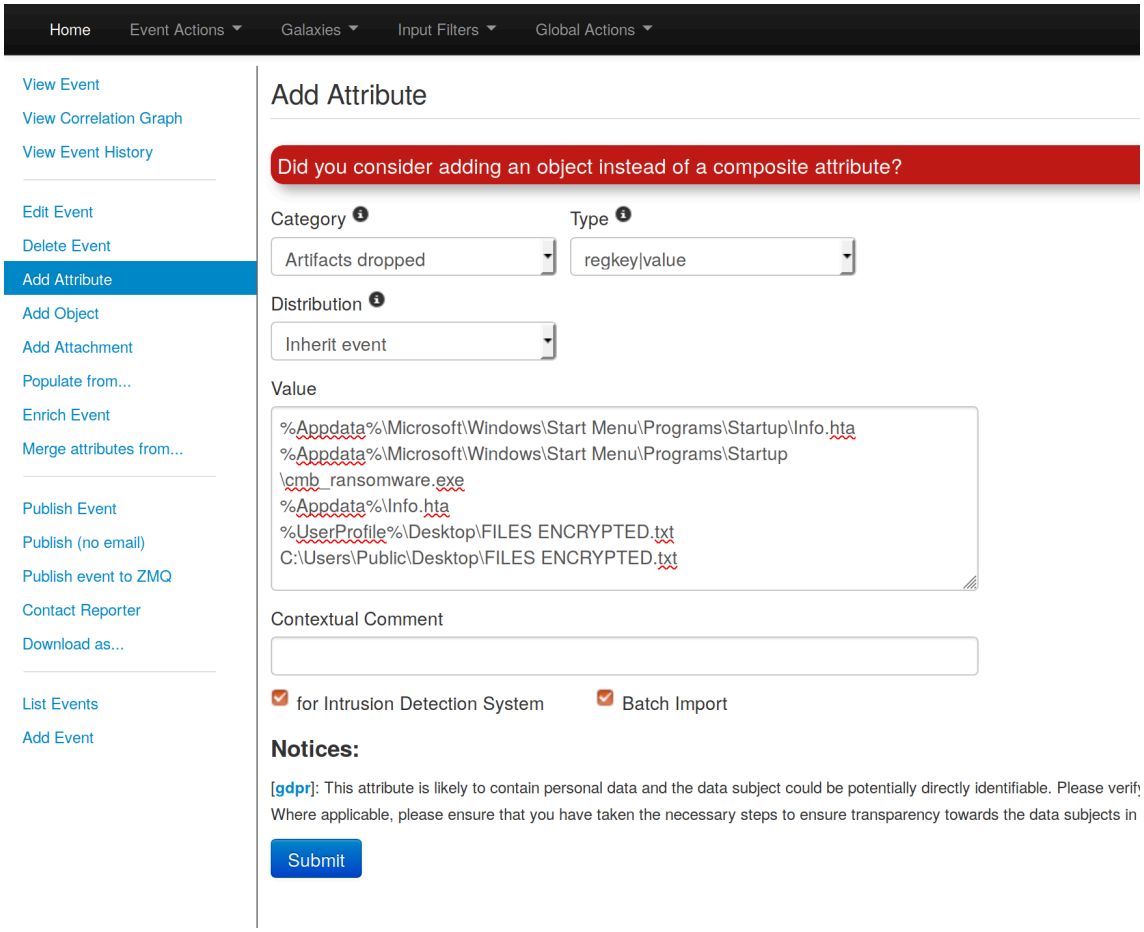


Figure 16.22: addattribute_freetext

And that is all we can get for the main informations and IoC in this report. If we search more carefully, there might still be some information left in it, like the filename of the ransomnote for instance, but we will stop here for this example.

16.0.5 Modify the event

If you want to modify your event from the home page, you can either double click on the event or click the edit symbol located in the column **Actions** on the right side. You will be redirected to the editing mode of the selected event.

17. Event Reports

Event Reports are rich-text, markdown documents attached to an event. They let an analyst write the *narrative* around an event — the story, the analysis, the context — while keeping that narrative tightly linked to the structured data (attributes, objects, tags, galaxies) it describes. A report can reference a specific attribute or object inline, so the prose and the indicators stay in sync, and MISP can even extract new indicators *out of* a report's text back into the event.

> **Note**
>
> This is distinct from the Create an event based on a report tutorial, which walks through building an event by hand from an external write-up. Here we describe the **Event Reports feature** — markdown reports stored on an event.

A report belongs to exactly one event, has a name and a markdown body, and carries its own distribution. Reports are written and read in a built-in markdown editor and appear in a collapsible **Event reports** section on the event view.

17.1 Writing a report

The editor offers several view modes — **Edit**, **Split screen** (source and preview side by side), **Markdown** (rendered) and **Raw** — plus the usual formatting toolbar (bold, italic, headings, lists, quotes, code, tables). Markdown is rendered with a GitHub-flavoured dialect, and diagrams (Mermaid) and syntax highlighting are supported.

17.1.1 Referencing MISP elements inline

Beyond ordinary markdown, reports support a special syntax to embed live references to the event's data. These render as styled inline elements and stay linked to the underlying object:

Syntax	Renders as
`@[attribute](UUID)`	The referenced attribute (type and value).
`@[object](UUID)`	The referenced object.
`@[tag](TAG_NAME)`	A tag — note tags are referenced by **name**, not UUID (e.g. `@[tag](tlp:green)`).
`@[galaxymatrix](UUID)`	The event's galaxy/ATT&CK matrix widget.
`@![attribute](UUID)`	The picture stored in an attachment-type attribute (the `!` displays the image).
`{variable_name}`	A template variable, replaced by its configured value (see below).

The editor has autocompletion and toolbar buttons that insert these for you, so you can pick the element from a list rather than typing UUIDs by hand. References resolve across an extended-event chain, so a report can point at elements of the parent or child events too.

17.2 Creating a report

From the **Event reports** section of an event you can create a report three ways:

- **Add Event Report** — start from a blank markdown document.
- **Import from URL** — fetch a web page or document and convert it to markdown automatically.

- **Generate report from Event** — build a report from the event's own data.

17.2.1 Import from URL

MISP can download a URL and convert its contents into a markdown report. HTML pages are converted directly; PDF, DOCX, PPTX, XLSX, ODT and ODS files are converted through the corresponding enrichment misp-modules. The relevant module must be enabled and permitted for your user.

> ⚠ Warning
>
> Import-from-URL is **disabled by default** and gated by the `Security.eventreport_enable_arbitrary_urls` setting, because **the MISP server itself performs the fetch** — meaning it could be pointed at internal addresses the client cannot reach. Enable it deliberately, and only if you trust the users who can create reports.

17.2.2 Generate report from Event

This produces a report from the event's existing data, driven by a REST-search-style filter. You can choose whether to include event metadata, correlations and the ATT&CK matrix, and whether to embed element **references** (`@[...]`) or raw values. It is a fast way to turn a structured event into a readable, shareable narrative.

17.3 Extracting indicators from a report

Because reports are text, MISP can parse them for indicators and pull them into the event as attributes. There are two flows:

- **Interactive suggestions** — as you edit, MISP detects candidate indicators and existing matching attributes, and offers to mark them up. Text that matches an existing attribute is turned into an `@[attribute](uuid)` reference automatically; you accept or reject each suggestion, and accepted ones become new attributes and are linked in the report.
- **Extract all IoCs** — a one-shot action that detects all indicators in the report, creates attributes for them and rewrites the report to reference them. It optionally **tags the event** with contextual elements (tags and galaxy clusters, such as detected threat actors or ATT&CK techniques) found in the text.

This closes the loop between narrative and structured data: you can paste a write-up, extract its indicators, and end up with both a readable report and a populated event.

17.4 Template variables

Reusable snippets are provided by **Event Report Template Variables** — named `{variable}` placeholders that expand to a configured value (for example a standard TLP banner or a company logo). Site administrators define the variables under the Event Reports menus; any report can then use `{variable_name}` and MISP substitutes the value when rendering and exporting. (There is no separate "report template" object in 2.5 — templating is done entirely through these variables.)

17.5 Exporting a report

From the editor's download menu you can export a report as:

- **PDF** — rendered through the `convert_markdown_to_pdf` enrichment module (which must be enabled), with all element references and template variables fully expanded; or via your browser's print-to-PDF.
- **Markdown** — the raw source.
- **GFM (simplified)** — GitHub-flavoured markdown with the MISP-specific syntax replaced by plain equivalents, for sharing outside MISP.

17.6 Distribution, visibility and sync

A report has its own distribution setting, using the event levels **0–4** plus **Inherit event** (the default), so a report can be shared exactly as widely as the event or more narrowly. Visibility follows the same rules as other event data. Reports are part of event synchronisation — they are pushed and pulled along with their event (subject to distribution and the usual sync filtering), matched by UUID so edits reconcile correctly across instances.

17.7 AI-assisted extraction (experimental)

MISP includes an **experimental** integration that can send a report's markdown to an external LLM service to extract additional context — for example an executive summary, a suspected threat actor, attributed country or motivation — and optionally tag the event with the corresponding galaxy clusters. It is configured through the `Plugin.CTIInfoExtractor_enable`, `Plugin.CTIInfoExtractor_url` and `Plugin.CTIInfoExtractor_authentication` settings and calls a generic, provider-agnostic HTTP endpoint (you supply the URL and API key of the service you want to use). It is disabled by default.

17.8 Permissions and API

Creating, editing, deleting, importing and extracting require the `perm_add` permission; viewing and rendering require only a logged-in user (subject to the report's distribution); tagging a report requires `perm_tagger`. Editing or deleting a report is limited to the event's creator organisation (and site admins).

Event Reports are fully available over the REST API under `/eventReports/` (add, view, index, edit, delete, restore, `extractAllFromReport`, `importReportFromUrl`, `reportFromEvent`, `downloadAsPDF`, and more), and template variables under `/eventReportTemplateVariables/`. See the Automation & API chapter for authentication and the request/response envelope.

18. Analyst Data

Analyst Data is a collaborative annotation layer that lets analysts enrich anything in MISP — an event, an attribute, an object, an event report, a galaxy cluster and more — *without modifying the annotated element itself*. Instead of editing another organisation's event (which you cannot do) or spawning a proposal, you attach your own record alongside it. Each record has its own identity, its own distribution, and its own author, so it can be shared and synchronised independently of the thing it describes.

There are three types of analyst data:

- **Note** — free-text commentary, with a language tag.
- **Opinion** — a numeric agreement score from 0 to 100 plus an optional justification.
- **Relationship** — a directed, labelled link from one element to another.

Because each record is identified by its own UUID and carries its creator organisation, analyst data is inherently **federated**: notes, opinions and relationships propagate between instances through synchronisation just like events do, letting a whole community build a shared, attributable body of analysis around the same intelligence.

> **Note**
>
> Creating or editing analyst data requires a role with the **Analyst Data Creator** permission (`perm_analyst_data`). Viewing analyst data only requires being logged in (subject to the record's distribution).

18.1 The three types

18.1.1 Notes

A **Note** is a piece of free-text commentary attached to a target element. Beyond the note body itself, a note carries a **language** (an RFC 5646 tag such as `en`, `fr`, `de`) so that multilingual communities can annotate in their own language and filter accordingly.

Use notes to record analysis, caveats, remediation guidance, or any context that does not fit neatly into an attribute or tag.

18.1.2 Opinions

An **Opinion** expresses how much you agree with the element it is attached to, as a score from **0 to 100**, with an optional free-text **comment** explaining your reasoning. The scale is read as five bands:

Score	Meaning
0–20	Strongly disagree
21–40	Disagree
41–60	Neutral (50 = neutral midpoint)
61–80	Agree
81–100	Strongly agree

In the interface the score is shown as a coloured gradient bar with the numeric value. Opinions are the natural way to signal (dis)agreement with a shared indicator or assessment without altering it — for example, to flag a suspected false positive or to endorse a high-value indicator.

18.1.3 Relationships

A **Relationship** is a directed edge from a *source* element to a *related* element, carrying a free-text **relationship type** label (for example `similar-to`, `derived-from`, `part-of`). The relationship type is a free-text field with autocomplete — the suggestions combine relationship types already used in analyst data with MISP's object-reference vocabulary — so you are not restricted to a fixed list.

A relationship can link any two supported elements, even across events and across instances, and both ends are referenced by UUID. A record cannot relate an element to itself.

> ! Important
>
> Analyst-data **Relationships** are not the same as two other MISP concepts that also link things together:
>
> - **MISP object references** link an attribute or object to another object *within the same event*; they are part of the event's own data model, not analyst data.
> - **Galaxy cluster relationships** link galaxy clusters to one another as part of the galaxy data model.
>
> Only analyst-data Relationships are individually authored by an organisation, individually distributable, blocklistable, and synchronised through the analyst-data pipeline. See MISP Objects and Galaxies for the other two.

18.2 What can be annotated

Analyst data can be attached to any of the following target types: **Attribute, Event, Event Report, Galaxy Cluster, Galaxy, Object, Organisation, Sharing Group**, and — because analyst data can annotate analyst data — **Note, Opinion and Relationship** themselves. Attaching a note or opinion to another note produces a **thread**, so a discussion can build up around a single element.

In practice, notes, opinions and relationships are surfaced automatically when you view an **event, attribute, object, event report or galaxy cluster**; the other target types are valid anchors but are not all rendered inline in their own views.

18.3 Working with analyst data in the UI

Analyst data appears in the event view attached to the element it annotates. From there you can add a note, opinion or relationship in place; the form asks for the target, the distribution, and the type-specific fields (note text and language, opinion score and comment, or relationship type and related element).

Opening a record shows it together with its **thread** — nested notes and opinions — and its relationships, split into:

- **Outbound** relationships (edges this element points *out* to), and
- **Inbound** relationships (edges from other elements that point *at* this element).

Inbound relationships are computed automatically, so you can see who has linked to an element even if the link was created elsewhere.

18.4 Distribution and sharing

Every analyst-data record has its own **distribution** setting, independent of the element it annotates, using the same levels as events:

Level	Meaning
0	Your organisation only
1	This community only
2	Connected communities
3	All communities
4	Sharing group (an explicit list of organisations)

The default level for new analyst data is controlled by the `Plugin.default_analyst_data_distribution` server setting (default: *This community only*). When the level is *Sharing group*, MISP validates that you are allowed to use the chosen sharing group. Visibility follows the usual rules: you can see a record if your organisation created it, if it is distributed to your community or wider, or if it is in a sharing group you belong to.

18.5 Collaboration and synchronisation

Analyst data is built for federated collaboration:

- **Identity** — each record has a globally unique UUID that is stable across instances, so the same note is recognised everywhere it travels.
- **Authorship** — every record records its *creator organisation*; only the creating organisation (or a site admin) can edit or delete it. Other organisations add their own parallel notes and opinions rather than changing yours.
- **Synchronisation** — analyst data is pushed and pulled between instances alongside events, provided both sides support the feature and the per-server **Push analyst data** / **Pull analyst data** toggles are enabled (see Sharing). Distribution degrades by one step on each outbound hop, exactly like event data, and records received from a remote instance are marked as *locked* so they are not accidentally overwritten locally.

18.6 Analyst Data Blocklist

Because analyst data is synchronised, simply deleting an unwanted record locally would not stop it being pulled back in from a remote instance. To prevent that, MISP keeps an **Analyst Data Blocklist** keyed on the record UUID. When a record is (hard-)deleted, a blocklist entry is created automatically — recording the UUID, a short human-readable summary, and the creator organisation — so the record is ignored if it is offered again during synchronisation.

The blocklist is managed by site administrators under the administration menus and behaves like the other blocklists in MISP.

18.7 API and permissions

Analyst data is fully accessible over the REST API under the `/analystData/` controller. The main actions are:

Action	Purpose	Permission
`add/{type}/{object_uuid}/{object_type}`	Create a note/opinion/relationship on a target	`perm_add` + `perm_analyst_data`
`edit/{type}/{id or uuid}`	Modify a record you own	`perm_add` + `perm_analyst_data`
`delete/{type}/{id or uuid}`	Delete a record you own	`perm_add` + `perm_analyst_data`
`view/{type}/{id or uuid}`	View a single record	any logged-in user
`index/{type}`	List records of a type	any logged-in user
`getChildren/{type}/{uuid}`	Fetch a record's thread	any logged-in user

where `{type}` is `Note`, `Opinion` or `Relationship` (or `all`, letting MISP deduce the type from the UUID). The synchronisation endpoints used between instances require the `perm_sync` permission. As with the rest of the API, requests and responses use the standard MISP JSON envelope — see the Automation & API chapter.

19. Sightings

Basically, sighting is a system allowing people to react on attributes on an event. It was originally designed to provide an easy method for user to tell when they see a given attribute, giving it more credibility.

Now sightings have been improved to also provide a method to signal false positives, but also to give an expiration date for some attributes.

19.0.1 Explanation

As said before, Sighting is a way for a user to say that they have seen or notice an attribute and confirm its validity. An attribute can been spotted several times by the same user, that is why a single user can use sighting several times on a single attribute.

Sometimes, some attributes can be considered as false positives, even if the false positive list do not detect them (for instance, if the IDS flag is set to false) so they can also be notified. As well as concerning sighting, the same user can signal a single attribute as a false positive several times.

It also happens that some attributes are only valid a certain time (for instance, in case of a phishing campaign that is assumed to be up for only one week). In this case, people can also assign an expiration date to an attribute, but this time, there can be only one valid expiration date per *organisation*.

19.0.2 Using sightings on an event (GUI)

Sighting is applied to every attribute, under the column “Sightings”, easily identifiable with its colored number. This column shows three icons and three values.

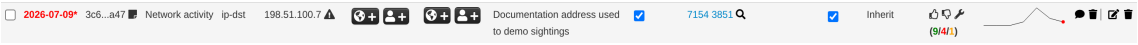


Figure 19.1: attribute

These three values show respectively: - The number of true positives detected with the attribute, in green. Malicious activity as described in the event. - The number of times the attribute has been marked as false positive, in red. Non-malicious activity or incorrect detection. - The number of different expiration dates that have been affected on this attribute, in orange

Concerning the three icons: - The first one (Thumb up) allows to add a sighting (true positive) on an attribute. - The second one (Thumb down) allows to mark the attribute as a false positive. - The third one (Tool) opens a popup for advanced sightings, showing sightings details and allowing different actions.

19.0.2.1 Advanced sightings

- The first tab, “Graph”, represents a line graph showing the evolution of sightings and false positives over time.

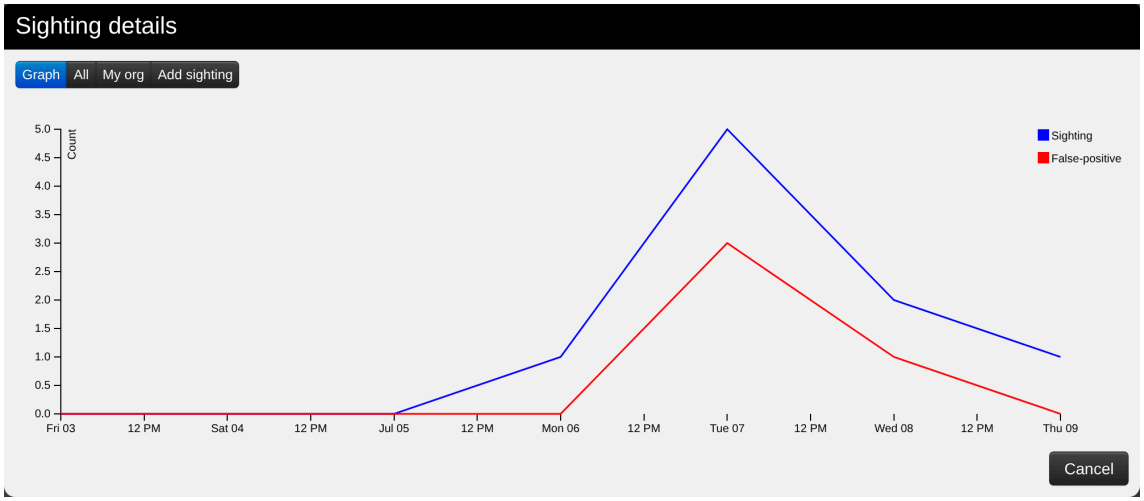


Figure 19.2: sighting graph

- The second tab gives a quick view of all the sightings applied to the attribute.

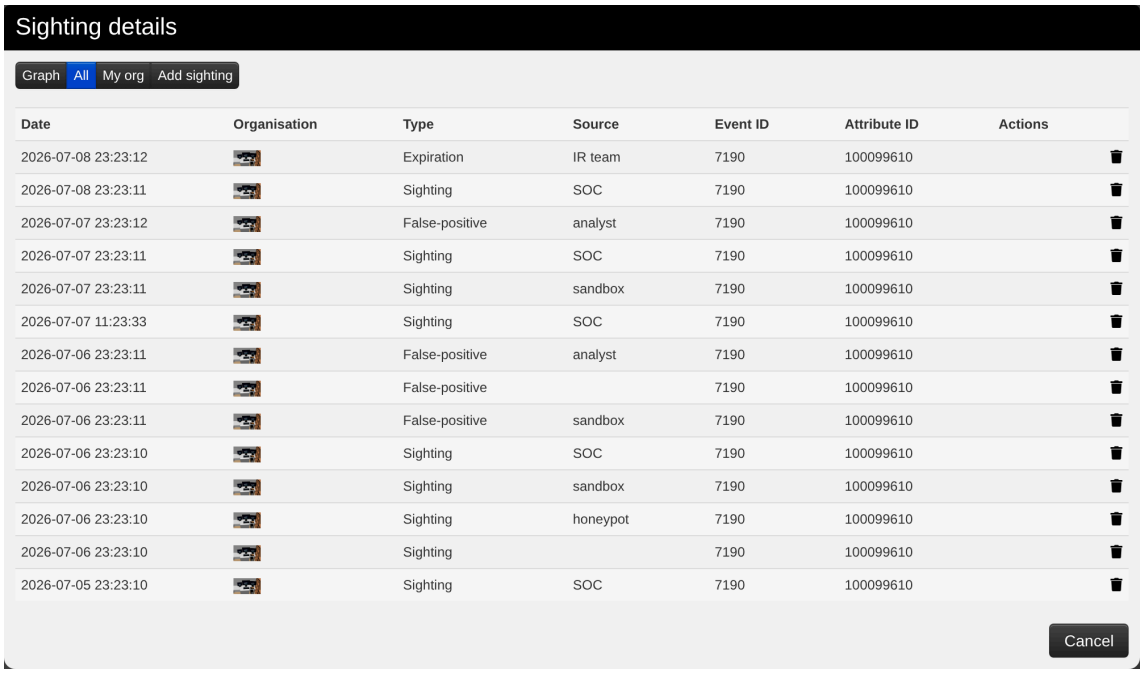


Figure 19.3: sighting all

- The third tab gives a quick view of the sightings applied to the attribute by your own organisation only.

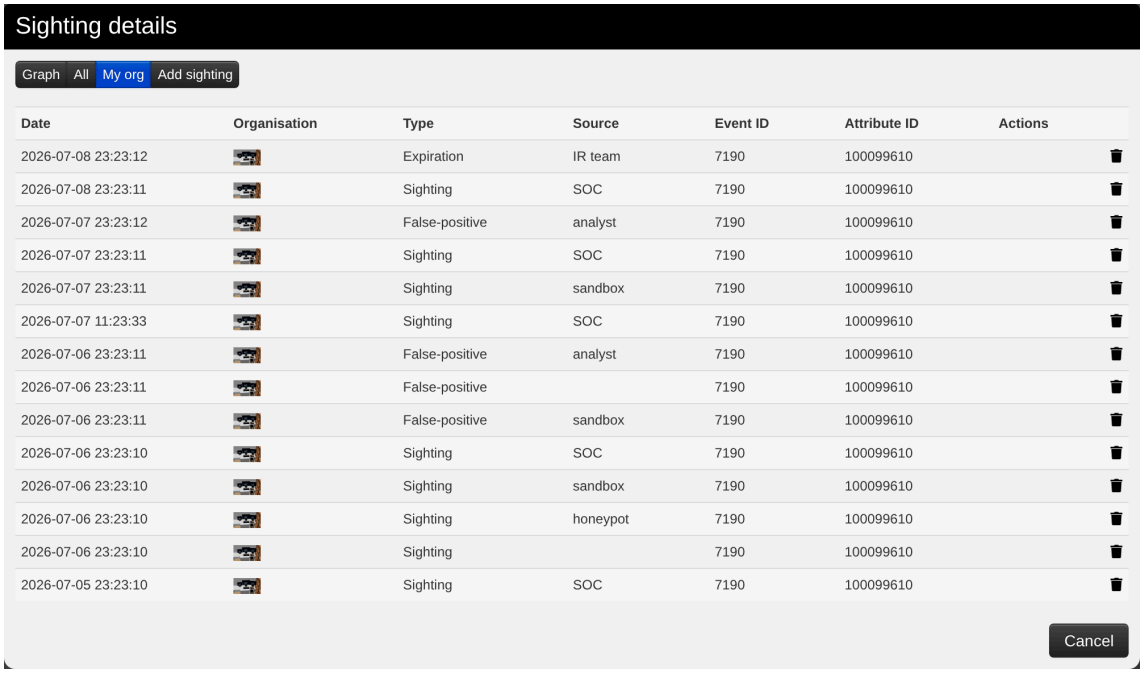


Figure 19.4: sighting my org

- The last tab can be used to add either a sighting, mark the attribute as a false positive, or define an expiration date. You can precise both the date and time of day, as well as note a particular source where the sighting comes from.

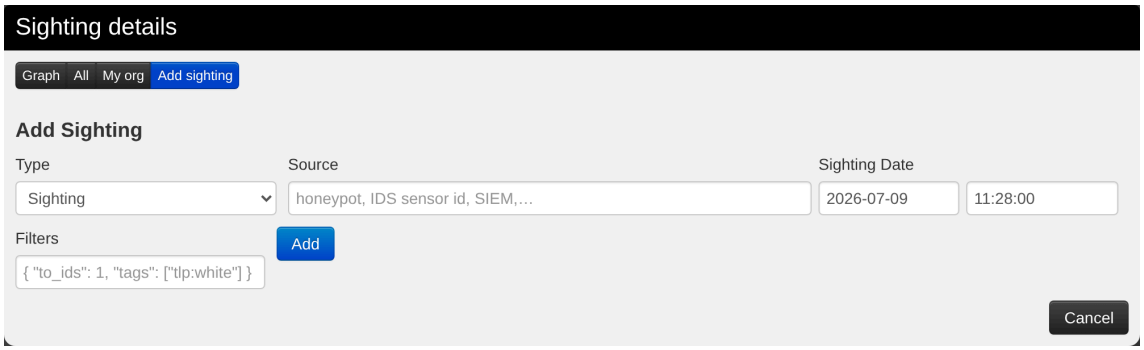


Figure 19.5: sighting add

19.0.2.2 At Event level

The total number of sightings across the whole event is shown in the event metadata, next to the **Sightings** label: the overall count, your own organisation's count in green, and — depending on the sightings policy — a note about whose sightings you are allowed to see.

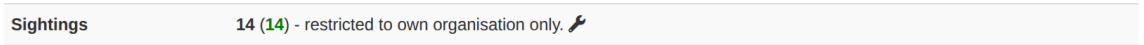


Figure 19.6: sighting event

Clicking the wrench opens the same **Sighting details** popup for the whole event, with the Graph, All and My org tabs (there is no *Add sighting* tab here — sightings are always added against a specific attribute).

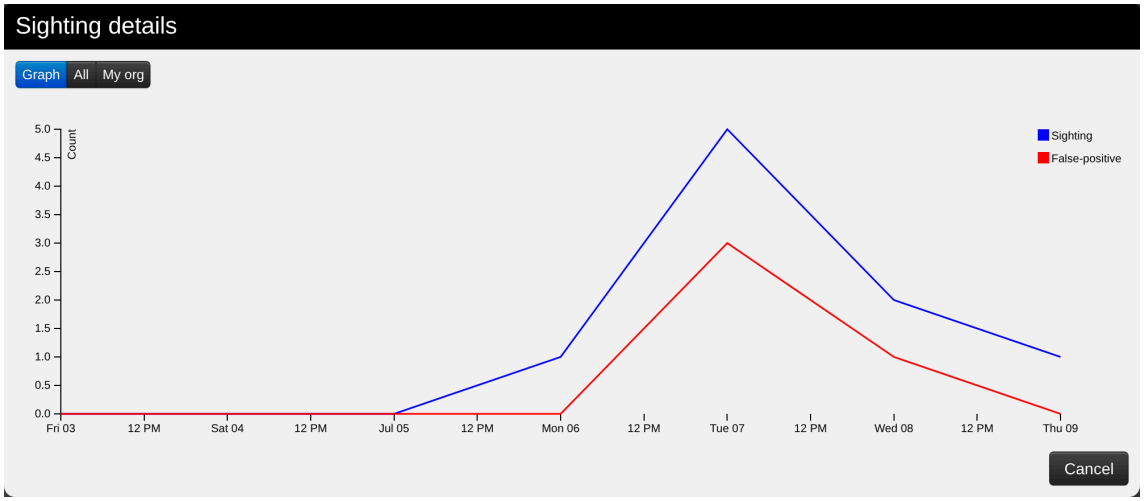


Figure 19.7: sighting event popup

19.0.3 Sighting policies and privacy

Because sightings reveal who is looking at which indicators, MISP gives administrators several controls over how much of that is shared (all under Administration → Server Settings):

- **Sightings_policy** — who can see the reported sightings on an event: the event-owner organisation only (the default; everyone always sees their own contribution), the *sighting reporters* (the owner plus any organisation that contributed a sighting), or *everyone* who can see the event.
- **Sightings_anonymise** — strip the reporting organisation from sightings so consumers see the counts but not who reported them.
- **Sightings_anonymise_as** — when pushing sightings to other servers, report them all as a single organisation, hiding the individual reporters on this instance behind one identity.
- **Sightings_range** — how many days back a sighting is considered relevant (default 365).

19.0.4 SightingDB

For high-volume sighting data, MISP can offload sightings to one or more external **SightingDB** servers instead of storing every sighting in the main database. Enable it with `Sightings_sighting_db_enable` and configure the SightingDB servers from the administration menu; matching attributes then gain an additional SightingDB column in the event view.

19.0.5 Real-time sharing and blocklists

- With `Sightings_enable_realtime_publish` set, new sightings are pushed to your sync partners as they are created, rather than only when the event is (re)published. You can also push sighting updates on demand with **Publish Sightings** in the event menu.
- To ignore sightings from a specific organisation (for example a noisy source), add it to the **sighting blocklist** — see Blocklists in the administration chapter.

19.0.6 Using sightings on an event (API)

Please have a look at the automation API

20. Decaying Models

Indicators of compromise do not stay relevant forever. An IP address used in a campaign today may be reassigned next week; a phishing domain is taken down; a malware hash stops being seen. **Decaying models** let MISP express this decay of relevance automatically, turning a static attribute into a **score** that falls over time, and a **decayed** flag once that score drops below a chosen cut-off. Instead of manually toggling the IDS flag on ageing indicators, you define how each type of indicator should age and let MISP compute an up-to-date score whenever you query.

Decaying models are based on the MISP indicator-scoring research (arxiv.org/abs/1902.03914). They are managed under *Global Actions → Decaying Models* (the "Decaying Tool") and require a role with the **Decaying Model Editor** permission (`perm_decaying`) to create or edit.

20.1 How scoring works

For a given attribute and model, MISP computes:

- a **base score** — the indicator's intrinsic importance, derived from its tags, and
- a **current score** — the base score multiplied by a time-decay factor based on how long it has been since the indicator was last seen (its most recent sighting, or its `last_seen` / last-modified time).

An attribute is considered **decayed** by a model once its current score falls below that model's **threshold**. Crucially, a single attribute can carry **several scores at once** — one for every enabled model that applies to its type — so the same indicator can be "still hot" for one detection use-case and "decayed" for another.

Scores are computed **on demand** at query time; there is no background job precomputing and storing them. This keeps scores always current but means score-heavy queries cost CPU (see the performance note at the end).

20.1.1 The base score

The base score is derived from taxonomy tags on the attribute (and its event) that carry a **numerical value** — for example a confidence or false-positive-risk taxonomy. A model's `base_score_config` maps such taxonomies to weights; MISP combines the numerical values of the matching tags into the base score. If the attribute carries none of the configured taxonomies, the model's **`default_base_score`** is used instead. Attribute-level tags take precedence over event-level tags of the same taxonomy.

20.2 Model parameters

A decaying model is defined by a formula plus a small set of parameters:

Parameter	Meaning
Lifetime	Number of days after which the score reaches 0.
Decay speed	Controls the shape of the decay curve — a higher value decays more slowly.
Threshold	The cut-off score; below it, the indicator is marked *decayed*.
Default base score	Base score used when no configured taxonomy tag is present (0–100).
Base score config	Mapping of numerical-value taxonomies to weights, used to compute the base score from tags.
Formula	The decay formula class to use (see below).

Parameter	Meaning
Settings	Optional per-formula settings (only some formulas use this).

20.3 Decay formulas

The decay curve is provided by a pluggable **formula**. MISP ships three:

- **Polynomial** *(default)* — the classic scoring formula: the score decays from the base score to zero over the model's lifetime, with the curve shaped by the decay speed.
- **Polynomial (Extended)** — the same, but additionally honours the `retention` taxonomy: an indicator tagged with a retention period is forced to a score of 0 once that period has elapsed, giving a hard end-of-life on top of the smooth decay.
- **Sightings** — scores the indicator from its true-positive versus false-positive **sightings** rather than purely from elapsed time. It returns the base score until a minimum number of sightings has accumulated, then weights the score by the true-positive ratio. This formula requires sightings to be recorded.

> **Note**
>
> Formulas are ordinary PHP classes under `app/Model/DecayingModelsFormulas/`. To add a custom decay behaviour, drop in a new class that extends the decaying-model base class and implements the score/decayed logic; it appears in the formula dropdown automatically. This is an advanced, code-level extension.

20.4 Mapping models to attribute types

A model only applies to the attribute types you associate with it. The **Decaying Tool** shows a matrix of attribute types against models, letting you link each model to the types it should score (for example, a phishing model applies to `domain`, `hostname`, `url`, `ip-dst`, and so on). Default models carry their applicable types built in; models you create store their type associations separately, which you edit from the tool.

20.5 Simulating and previewing a model

Before enabling a model you can **simulate** it. The model editor includes a simulation view that plots the score over time as an interactive chart, so you can see the effect of your lifetime, decay speed and threshold choices. You can drive the simulation two ways:

- **By REST search** — score a set of real attributes matching a filter (tags, `to_ids`, and so on), and
- **By specific attribute** — score a single attribute by ID or UUID.

The simulation also shows the base-score computation steps and lets you override model parameters on the fly, so you can tune a model against your own data before committing it.

20.6 Using decay scores in queries and the event view

Decay scores are exposed through the REST API's attribute and event search (restSearch) with these filters:

Filter	Effect
`includeDecayScore`	Attach the decay score(s) to each returned attribute.
`includeFullModel`	Include the full model definition alongside each score, not just its id/name.
`decayingModel`	Restrict scoring to a specific model.

Filter	Effect
`excludeDecayed`	Drop attributes that are decayed under *all* their applicable models (implies `includeDecayScore`).
`modelOverrides`	Override `threshold`, `lifetime` or `decay_speed` for this query only.
`score`	Shortcut to override the threshold for this query.

Each returned score contains the numeric `score`, the `base_score`, the `decayed` boolean, and the model it came from. This makes it straightforward to, for example, export only the indicators that are still relevant for detection (`excludeDecayed`), rather than everything ever recorded.

In the web interface, the event view has an **Include Decay Score** toggle in the attribute filter/query-builder. Enabling it adds a decay-score column to the attribute list so you can see, per indicator, how relevant it currently is.

20.7 Ownership, sharing and default models

Models fall into a few categories, reflected by the filters in the model index:

- **My models** — models your organisation created.
- **Shared models** — models flagged to be usable by every organisation on the instance.
- **Default models** — models shipped with MISP. These can be enabled/disabled and shared, but their formula and parameters are locked (they are maintained upstream).

A model is editable by a site admin, or by a user with the Decaying Model Editor permission when the model belongs to their organisation and is not a default model.

20.7.1 Import, export and updating defaults

You can **export** a model as JSON and **import** one from JSON (the import is always brought in as a non-default model owned by your organisation). Default models can be **updated** from the upstream `misp-decaying-models` collection: MISP upserts models by UUID and upgrades those whose shipped version is newer than the local copy.

> **Tip**
>
> Decay scoring is computed at read time and can be CPU-intensive on large result sets. Enable `includeDecayScore` (and related options) deliberately on big exports, and lean on `excludeDecayed` / model thresholds to keep result sets focused. A workflow action module, **Attach decay score**, is also available to attach scores as part of a workflow.

21. Delegation

In information sharing, the privacy of the reporting organization can be important in cases such as:

- when an incident should not be linked to a potential victim.
- when you want to avoid linking a specific organization to the shared information.

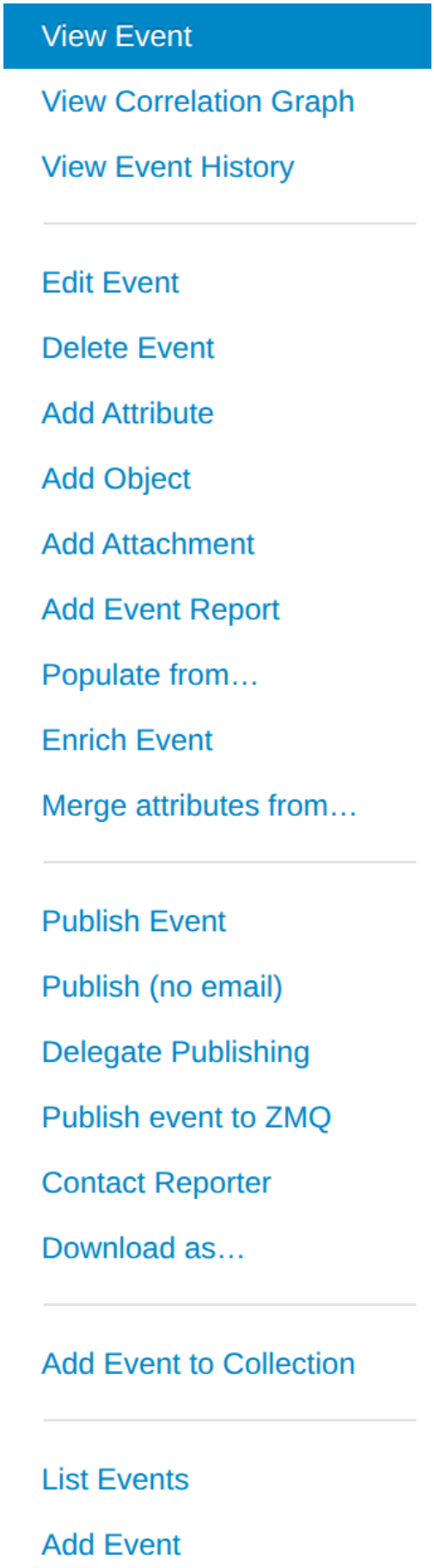
MISP includes a delegation feature that lets another organization publish an event on your behalf and removes the direct link between the shared information and your organization. If you want to publish an event without your organization being tied to it, you can delegate publication to another organization. That organization becomes the owner of the event. You can delegate only to local target organizations, not to remote ones.

> ⚠ Warning
>
> You need a role with `Delegation access` to delegate an event.

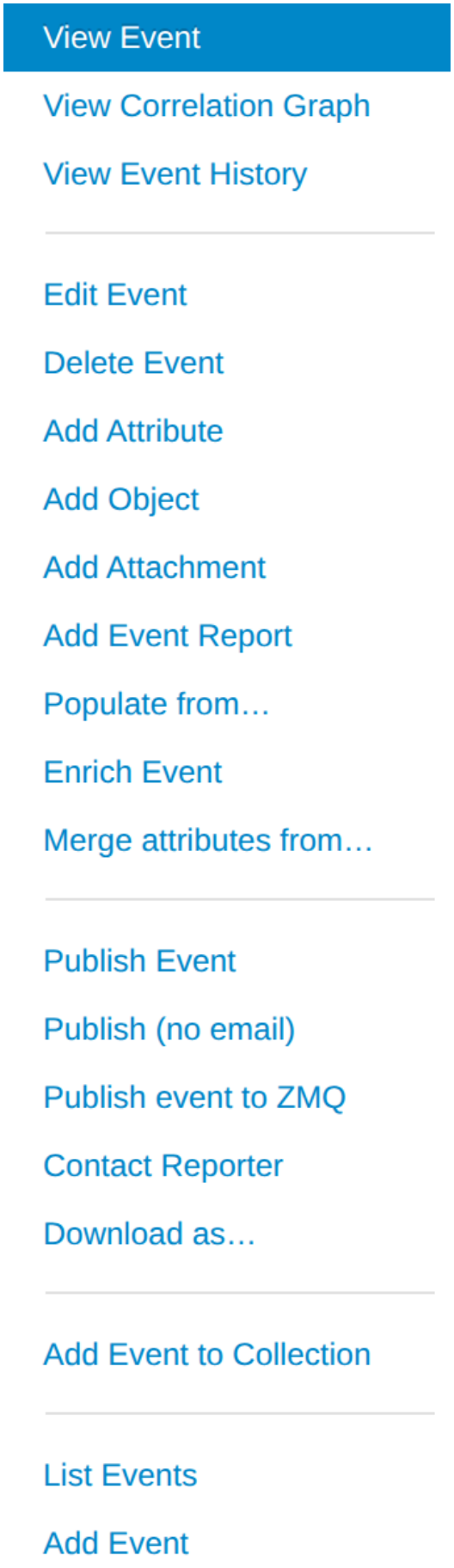
> ⚠ Warning
>
> You must also enable the `MISP.delegation` parameter on your instance.

21.0.1 Send a delegation request

To do this, you first need to set the event distribution to **Your organization only**.



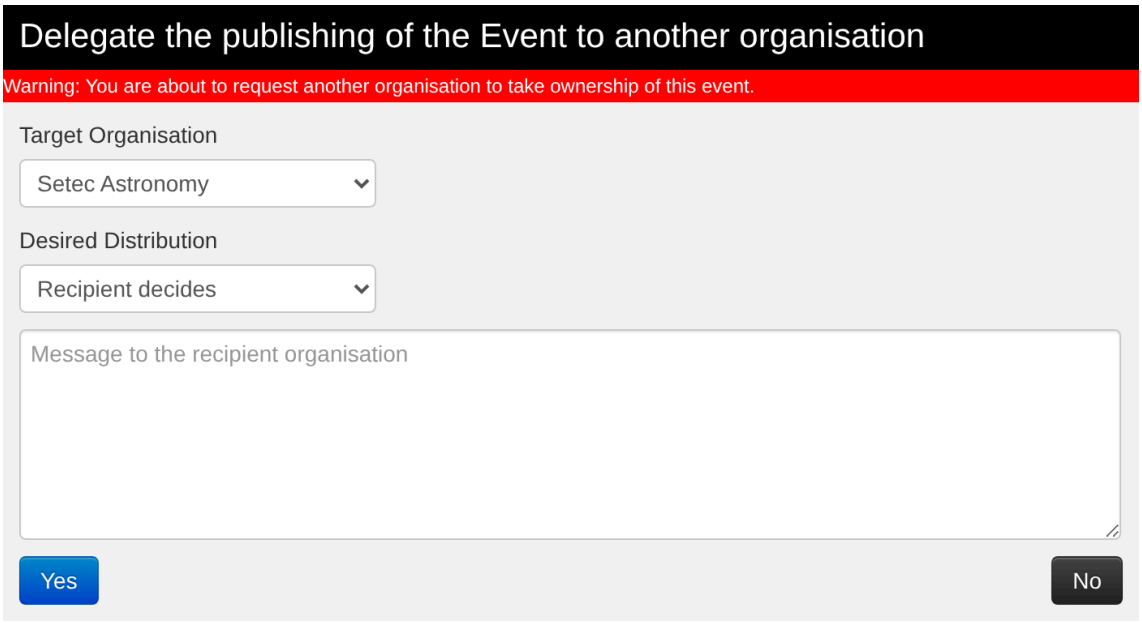
Otherwise, the delegation option will not be available.


View Event
View Correlation Graph
View Event History
Edit Event
Delete Event
Add Attribute
Add Object
Add Attachment
Add Event Report
Populate from…
Enrich Event
Merge attributes from…
Publish Event
Publish (no email)
Publish event to ZMQ
Contact Reporter
Download as…
Add Event to Collection
List Events
Add Event

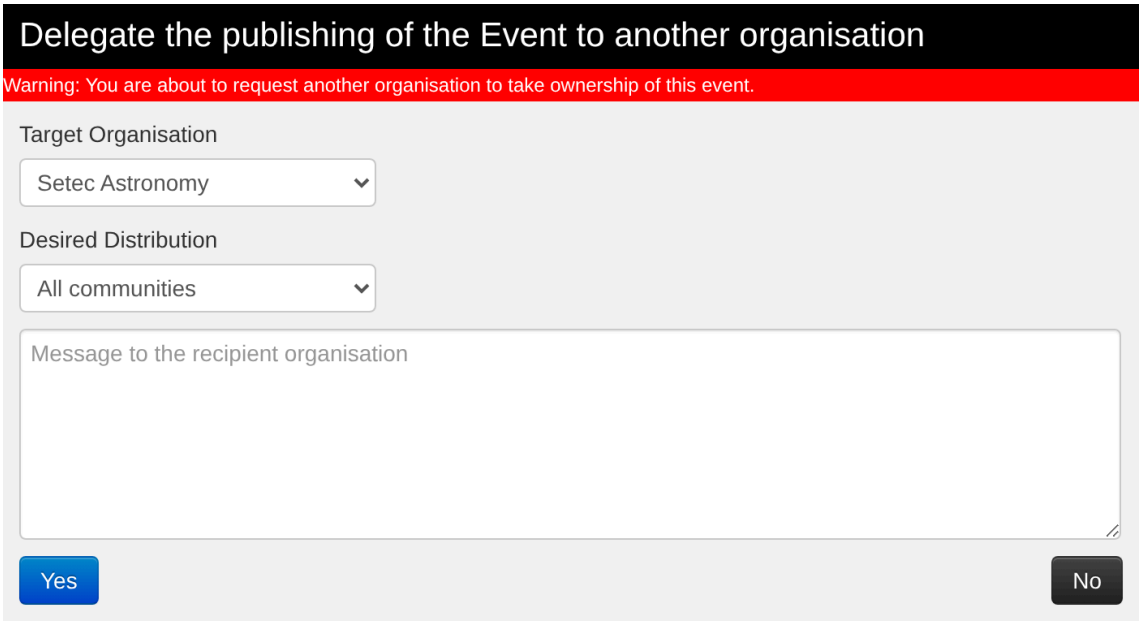
> **Note**
>
> The "distribution must be *Your organization only*" restriction is lifted if the site administrator has enabled the `MISP.unpublishedprivate` setting on the instance. When that setting is on, an event with any distribution level can be delegated.

An event can only have **one pending delegation request at a time**. If you need to change the target organization or the requested distribution, discard the existing request first, then create a new one.

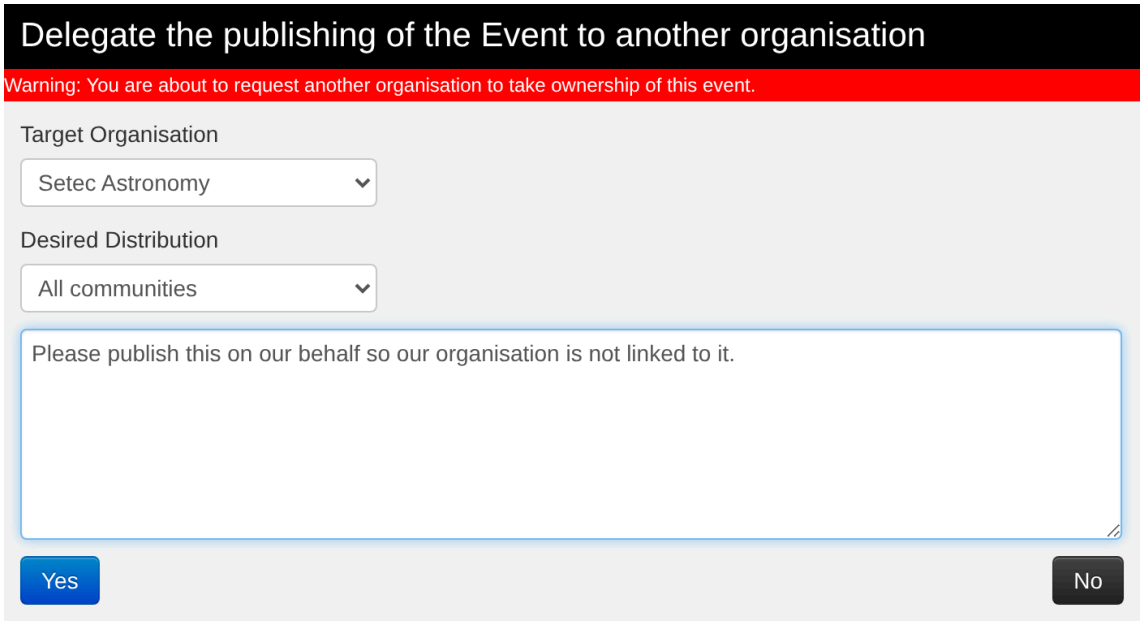
When you click **Delegate Publishing**, a pop-up appears.



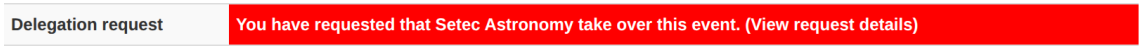
Here you can choose the following options: - The organization to which you want to delegate the event. In this example, we ask Setec Astronomy to publish the event for us. - The distribution option you would like to request for the event. You can let the recipient organization choose if you do not mind. In this example, we request **All communities**, but this is only a suggestion and the recipient can still change the distribution later.



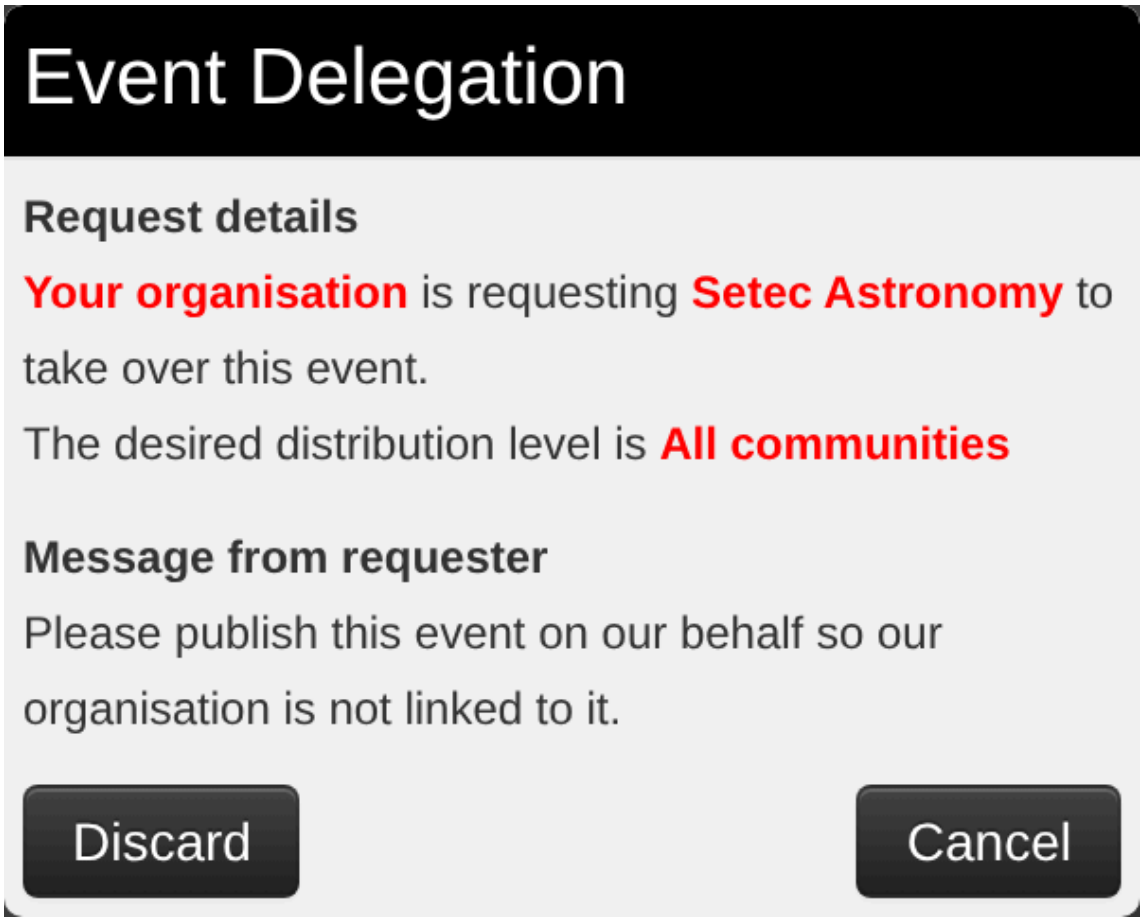
- An optional free-form message to the recipient organization.



Once the request is sent, a message appears on the event to remind you that the request is pending.



You can also view more details by clicking **View request details**.

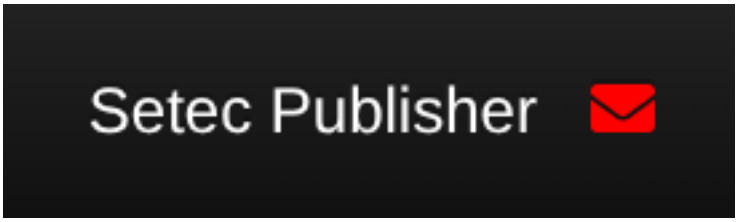


You can discard the request yourself by using either this pop-up or the link in the left-side menu.

Both the requesting and the recipient organization can review pending delegations at any time from the top navigation bar under **Global Actions → View delegation requests** (labelled *Delegation Requests*), which lists every request awaiting action.

21.0.2 Answer a delegation request

As the recipient organization, you receive the delegation request. You are notified by a red envelope in the upper-right corner of the screen.



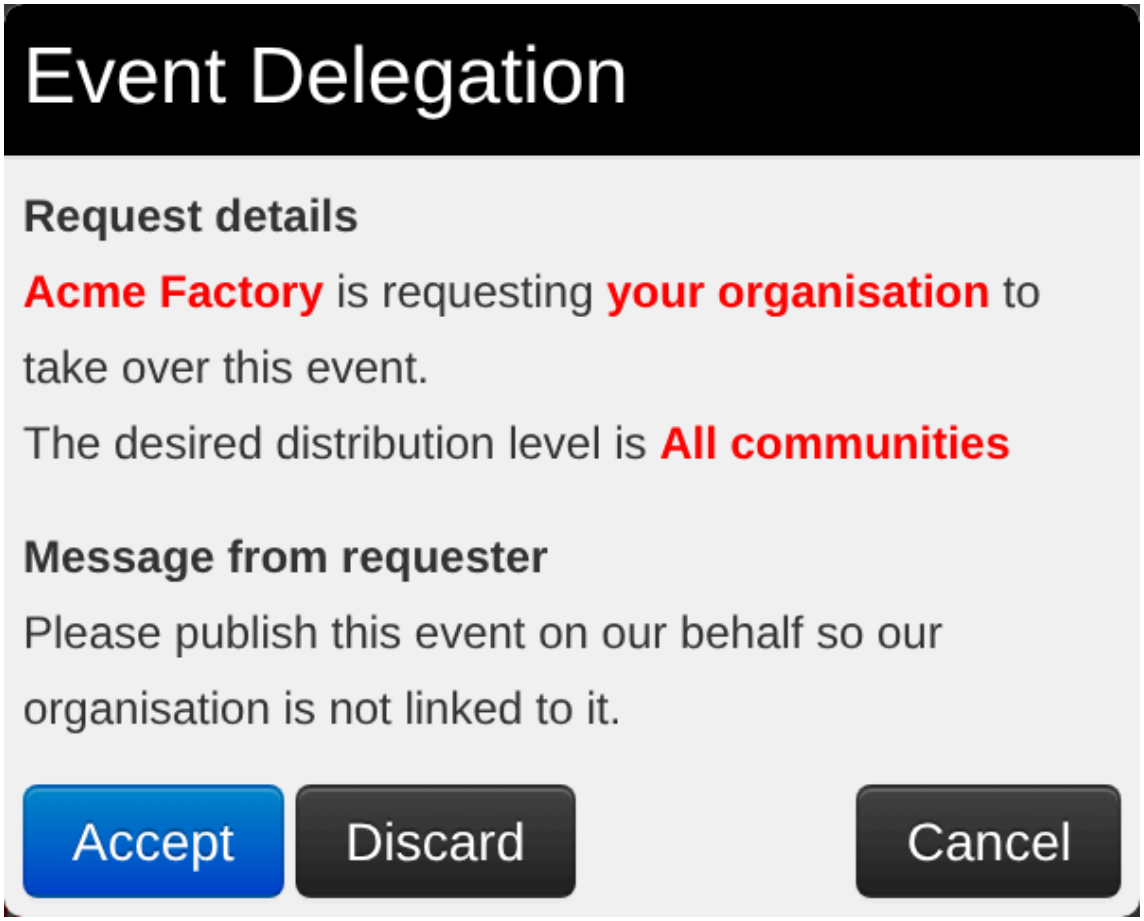
Open **Global Actions → View delegation requests** to see every event that other organizations want to delegate to your organization. In this example, there is a single request from Acme Factory.

Id	Requester	Recipient	Event id	Event info	Message
3	Acme Factory	Setec Astronomy	7191	Delegation demo: publish on our behalf	Please publish this event on our behalf so our organisation is not linked to it.

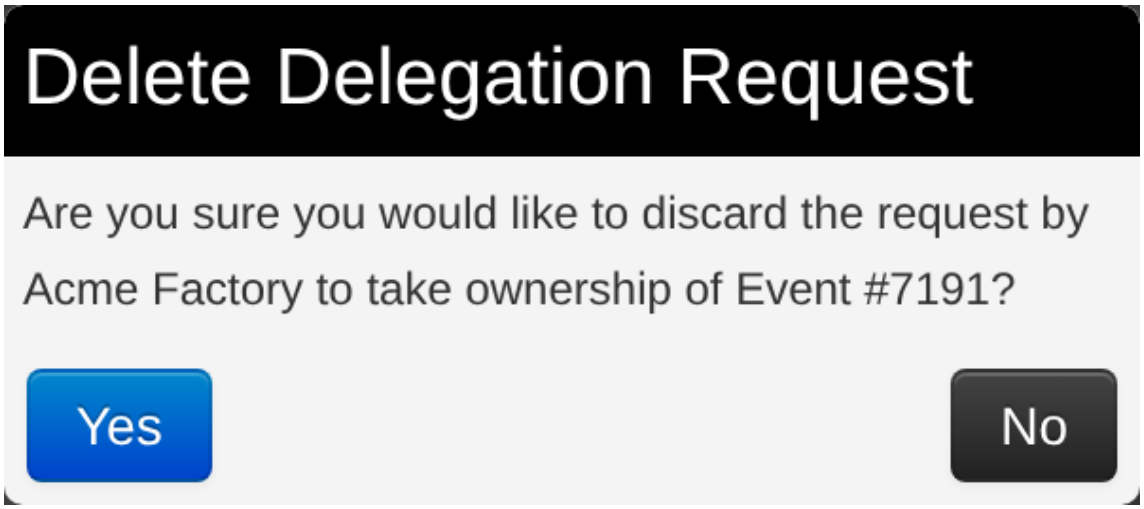
Opening the event shows its metadata, including a banner about the pending delegation.

Event ID	7191
UUID	7d2bc8ce-99c6-4cfd-a7f6-80408303088f
Source Organisation	Acme Factory
Member Organisation	Acme Factory
Protected Event (experimental)	Event is in unprotected mode.
Tags	TLP -none-
Date	2026-07-09
Threat Level	Medium
Analysis	Ongoing
Distribution	Your organisation only
Published	No
#Attributes	2 (0 Objects)
First recorded change	2026-07-09 11:36:43
Last change	2026-07-09 11:36:43
Modification map	
Sightings	0 (0) - restricted to own organisation only.
Delegation request	Acme Factory has requested that you take over this event. (View request details)
Correlation	Enabled

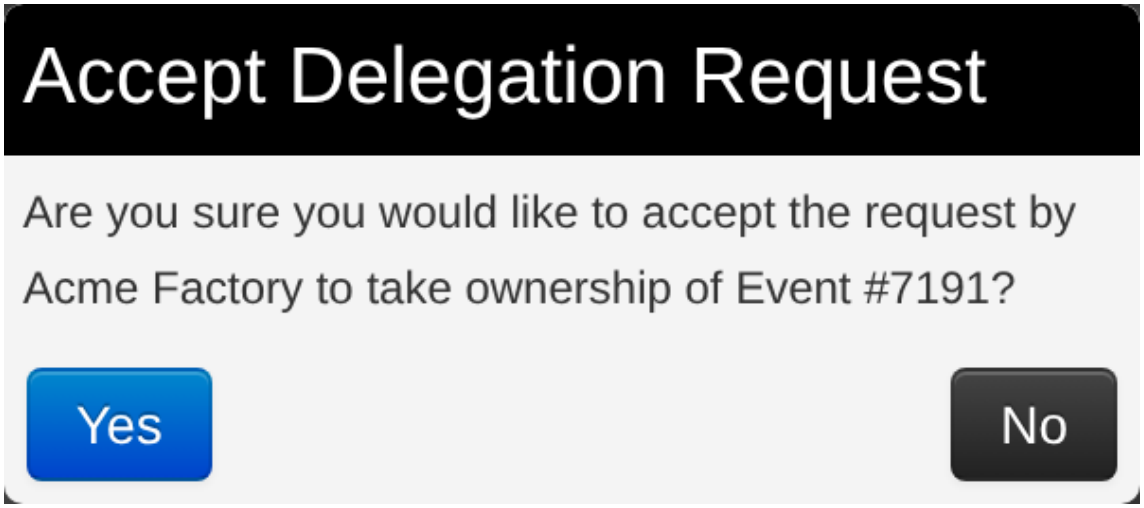
Click **View request details** to see the full request — the requesting organization, the distribution level it suggested, and its message.



If your role includes publishing rights, you can manage the delegation request by using one of the two links in the left-side menu. You can either discard it:



Or accept it:



Note that the distribution requested by the original sender is not applied automatically. If the setting is not changed, the event remains distributed only to your own organization.

22. Best Practices

The following page hosts some best practices for the usage of the MISP platform. For generic threat intelligence practices, please refer to MISP project - best practices in threat intelligence.

22.1 Tagging

22.1.1 Tagging at event level versus tagging at attribute level

You can add tags to an entire event. For a more fine-grained specification tags can also be placed at attribute level. This allows the user to put a more detailed and selective view on each attribute. If a tag applies to all or most attributes in the event, the best practice is to only include a specific tag for the attributes when they are an exception from the tag set at event level. In the below screenshot you see an example of the best practice, the tlp:white tag is only set at the event level.

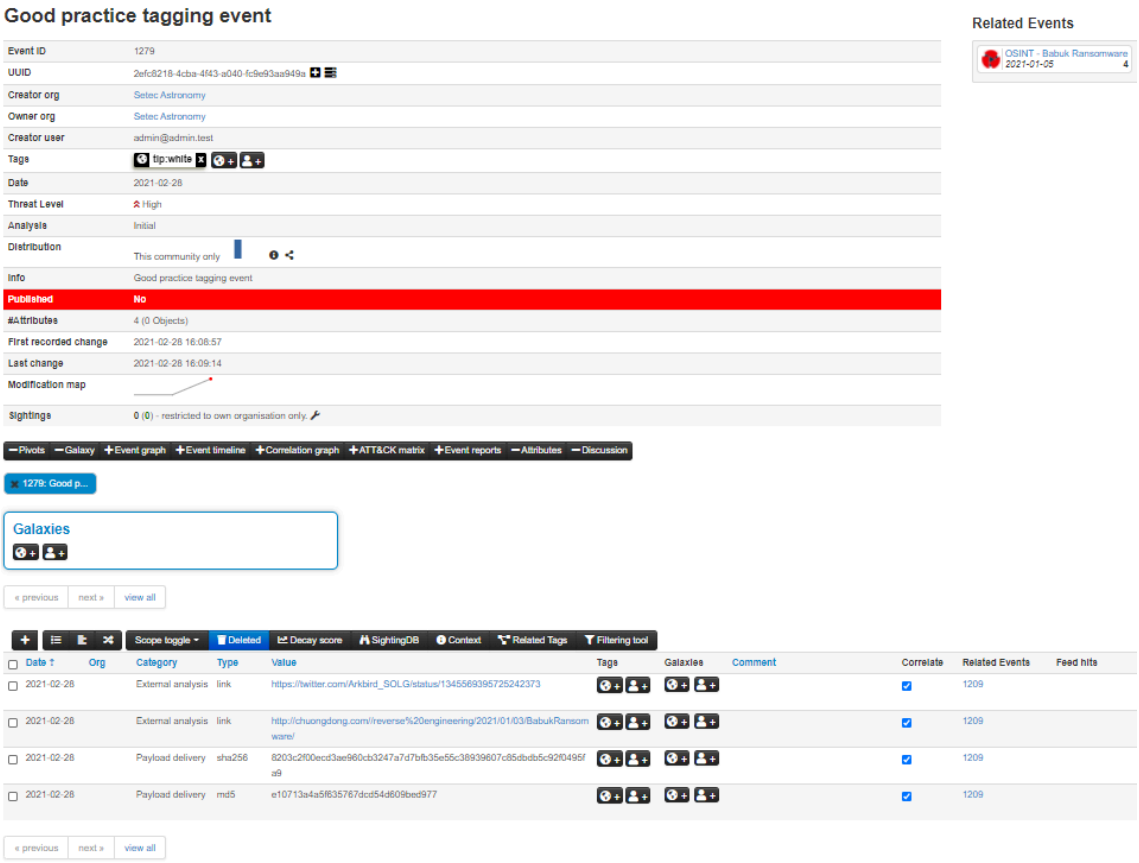
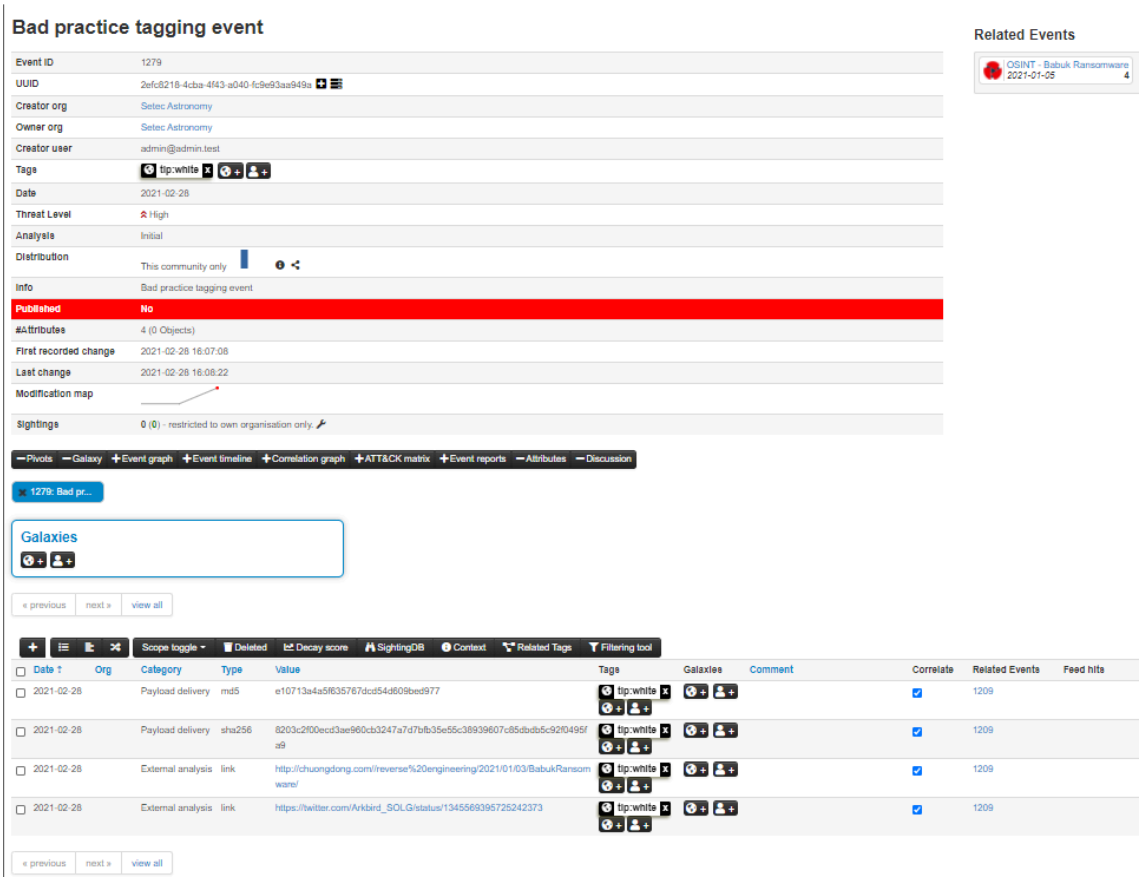


Figure 22.1: Screenshot of event tagged with tlp:white, no tlp:white tag set at attribute level

In the next screenshot you see a bad practice example. The tlp:white tag is added to both the event and all the attributes.



22.1.2 Minimal subset of tags to use for each event

22.1.2.1 Traffic Light Protocol

TLP-Tags: TLP utilizes a simple four color schema for indicating how intelligence can be shared.

22.1.2.2 Confidence

There are huge differences in the quality of data and in how well it was vetted before sharing. Communicate your confidence explicitly so recipients can weight the intelligence accordingly. Common vocabularies are the estimative-language taxonomy (confidence in an analytic judgement) and the admiralty-scale taxonomy (source reliability and information credibility).

22.1.2.3 Origin

information-origin: Describes where the information came from — for example whether it was produced in an automated fashion or through manual investigation. This gives an impression of how much the intelligence should be trusted, as manual investigation generally supersedes automatically generated data.

22.1.2.4 Permissible Actions Protocol

PAP-Tags: An even more advanced approach of data classification is using the Permissible Actions Protocol. It indicates how the received data can be used to search for compromises within the individual company or constituency.

22.2 Setting distribution

Similar to tagging, inheritance should be used wherever possible. This is especially important when using sharing groups due to limit the impact of sharing group usage on performance. Note that for distribution, the event or object distribution of the attribute should be the same or less restrictive than the attribute

distribution. Below are two good practice examples, the first using community distribution and the second using sharing groups. Note that a combination of the two can be used as well.

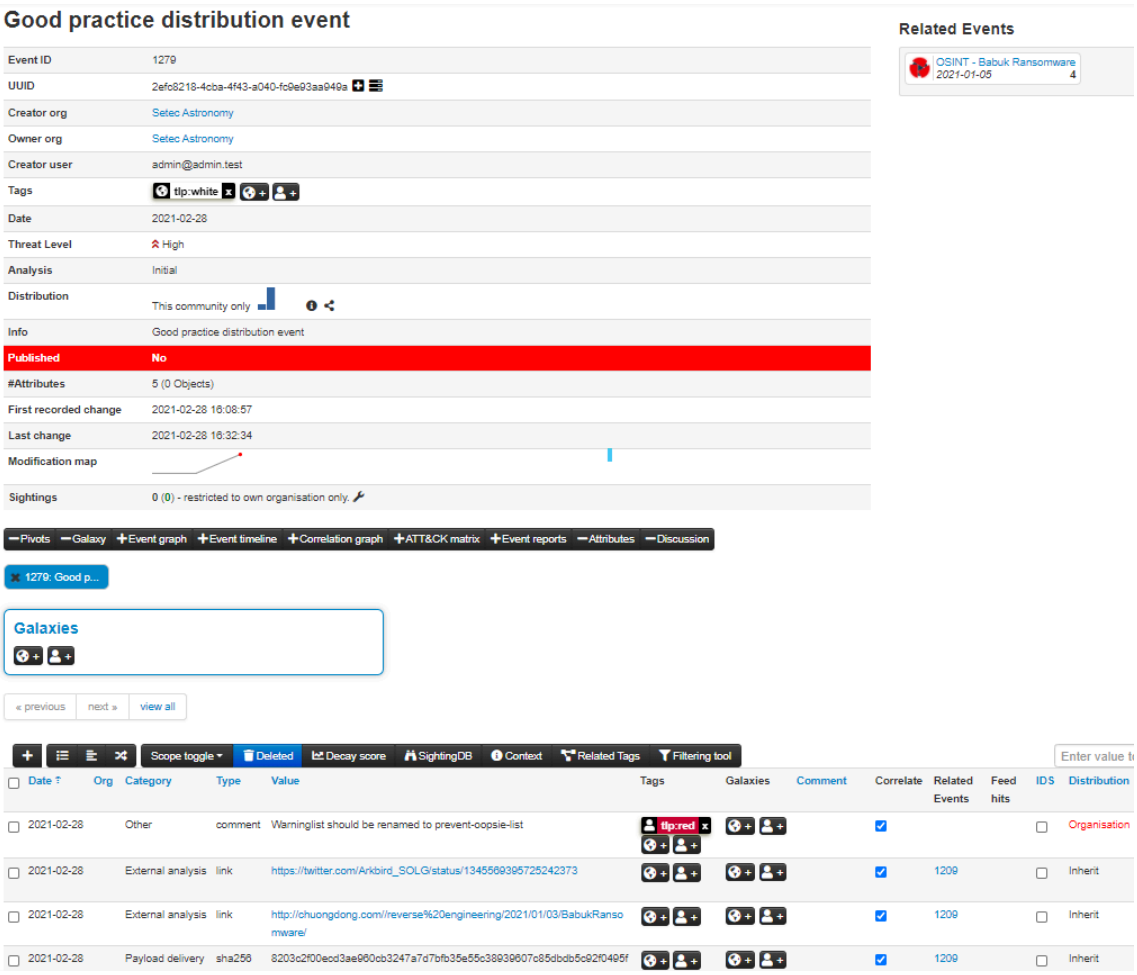


Figure 22.2: Screenshot of good practice event with this community distribution at event level and only one attribute set to your organisation only distribution

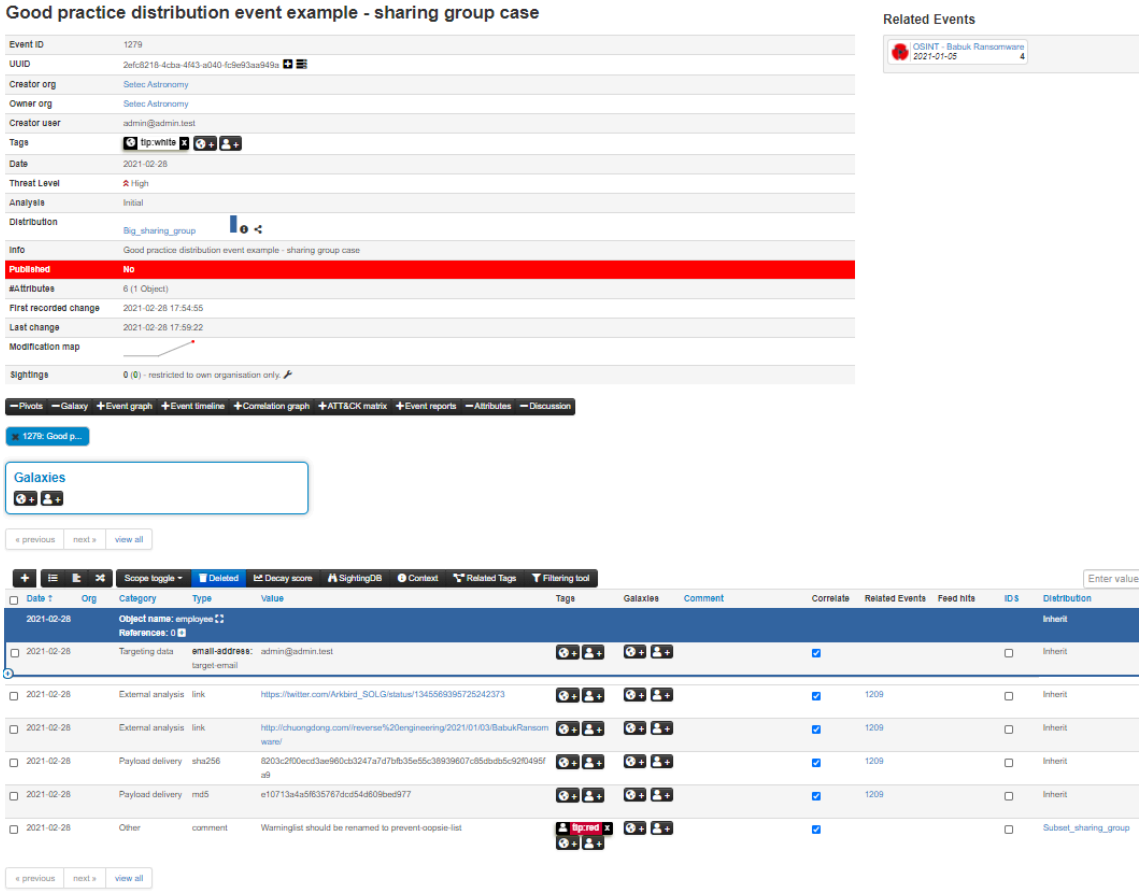


Figure 22.3: Screenshot of good practice event example with sharing group usage - inheritance is used where possible and only one attribute is set to a more restrictive sharing group

In the next to screenshots, you can see the equivalent bad practice examples.

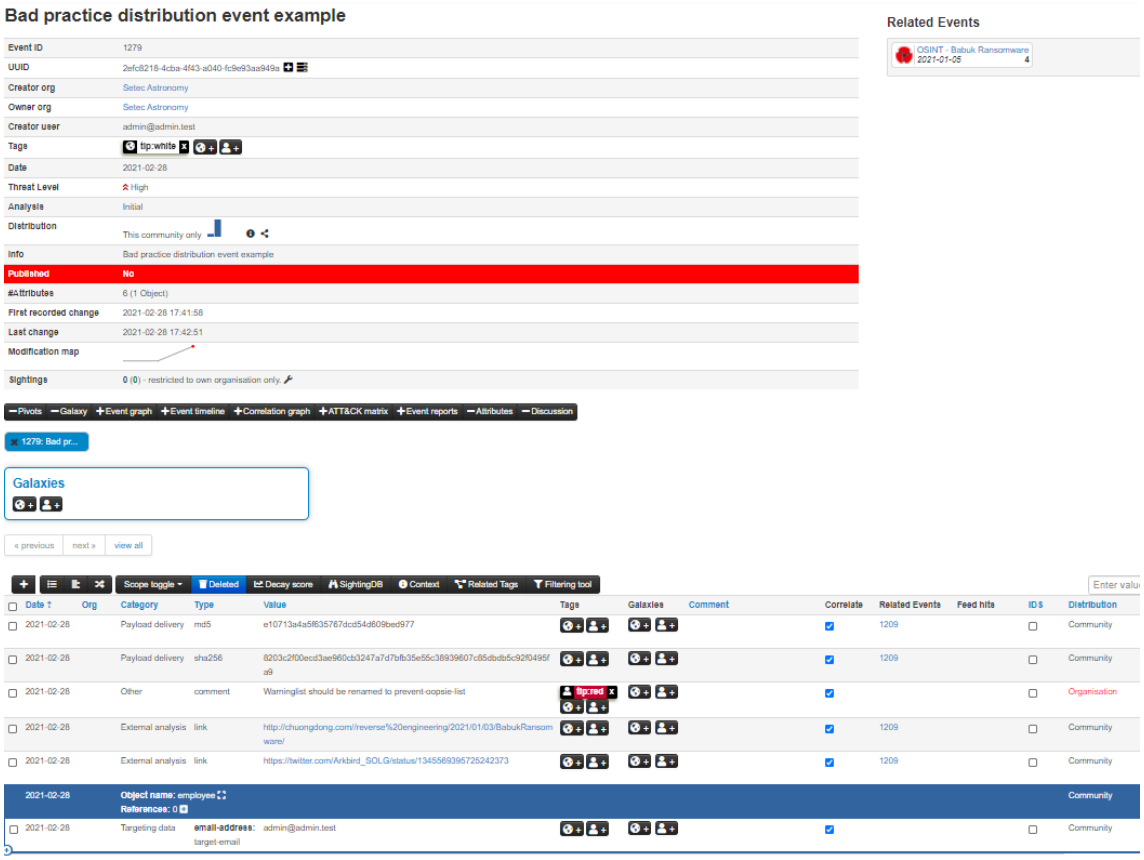


Figure 22.4: Screenshot of bad practice event example - distribution inheritance is not used

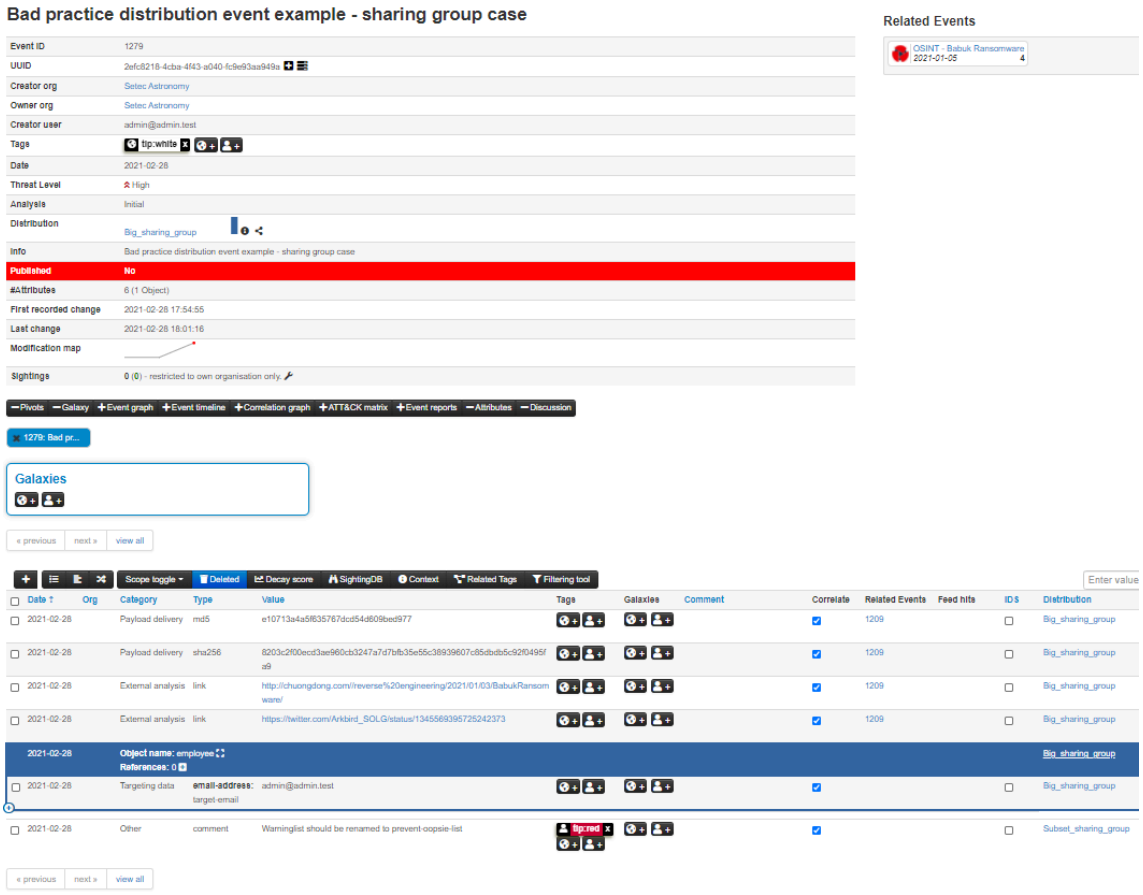


Figure 22.5: Screenshot of bad practice event example using sharing groups - Instead of using inheritance the same sharing group as the one set at event level is also specifically set for most attributes

22.3 Use objects instead of combined attributes

Some attributes like filename|sha512 and ip-dst|port now have equivalent objects such as file and ip-port. Using the object equivalent of attributes is recommended. It gives flexibility to add more context later and groups all the relevant attributes related to one concept, for example a file, together.

22.4 Use proposals when possible to request author to make changes to an event

If possible, use proposals when you want to inform the original analyst/author (e.g. a security vendor or a CSIRT) about a specific mistake or error which needs to be corrected. The advantages of using the proposal system include the lack of a need to create a new event as well as the process itself being very simple and fast.

22.5 Make use of the 2.5 collaboration and lifecycle features

MISP 2.5 adds several capabilities that materially improve the quality and longevity of shared intelligence. Build them into your day-to-day workflow.

- **Report sightings.** When you observe an indicator (or confirm it is a false positive), record a sighting. Sightings keep indicators current, feed reputation back to the author, and are the raw signal that decaying models consume. This is far more valuable than silently consuming a feed.
- **Prefer decaying models over manual `to_ids` toggling.** Rather than flipping the IDS flag by hand as indicators age, let decaying models score them over time and filter your exports on the decay score. This keeps detection content fresh automatically and consistently.

- **Add context with Analyst Data instead of altering others' events.** Use Notes, Opinions and Relationships to attach your assessment, a second opinion, or a link to related material without needing to own or copy the event. Reserve proposals for correcting factual errors; use an Opinion to express (dis)agreement, and a Note to add analysis.
- **Record temporal context with `first_seen` / `last_seen`.** Where an indicator's activity window is known, set the first_seen/last_seen fields so downstream consumers can reason about timelines rather than just presence.
- **Keep correlations meaningful.** Disable correlation on intrinsically noisy attributes, and maintain correlation exclusions for known-benign values (your own infrastructure, popular CDNs, sinkholes). Over-correlating values dilute the value of the correlation graph for everyone.

III Data Model & Context

23 Categories and Types 149

24 Taxonomies 183
24.1 Contributing to Taxonomy 185
24.2 Reserved Taxonomy 186
24.3 Adding Taxonomy in MISP 186
24.4 Adding a private taxonomy 187
24.5 How to use Taxonomy in MISP 187
24.6 More options to configure taxonomies. 188
24.7 Future functionalities related to MISP taxonomies 190

25 Galaxies 191
25.1 Managing galaxies in the UI 191
25.2 Using galaxies in events and attributes 195
25.3 Creating your own galaxy clusters (Galaxy 2.0) . 196
25.4 Forking a cluster 197
25.5 Cluster relationships 197
25.6 Blocklisting galaxy clusters 197
25.7 Authoring galaxies as JSON 197
25.8 Available galaxies 198

26 MISP Objects 199
26.1 Using objects 199
26.2 Object references and relationships 203
26.3 Editing objects 203
26.4 Object templates 203

27 Warning Lists 205
27.1 Warning Lists and False Positives 205
27.2 Usage 205

28 Notice Lists 209
28.1 How notices are triggered 209
28.2 Managing notice lists 209

23. Categories and Types

23.0.1 Attribute Categories vs. Types

> **Note**
>
> This reference is derived from MISP's own `Attribute::categoryDefinitions` and `typeDefinitions` (the authoritative `category_type_mappings`). To keep it from drifting, regenerate it from a running instance via `GET /attributes/describeTypes.json` (or the bundled `describeTypes.json`) rather than editing cells by hand. It reflects MISP 2.5.42 (16 categories, 194 types).

Category	Antivirus detection	Artifacts dropped	Attribution	External analysis	Financial fraud	Internal reference
AS				X		
aba-rtn					X	
anonymised	X	X	X	X	X	X
attachment	X	X		X		
authentihash		X				
azure-application-id						
bank-account-nr					X	
bic					X	
bin					X	
boolean						
bro				X		
btc					X	
campaign-id			X			
campaign-name			X			

Category	Antivirus detection	Artifacts dropped	Attribution	External analysis	Financial fraud	Internal reference
cc-number					X	
cd-hash		X				
chrome-extension-id						
comment	X	X	X	X	X	X
community-id				X		
cookie		X				
cortex				X		
counter						
country-of-residence						
cpe				X		
dash					X	
date-of-birth						
datetime						
dkim						
dkim-signature						
dns-soa-email			X			
domhash				X		
domain				X		
domain|ip				X		

Category	Antivirus detection	Artifacts dropped	Attribution	External analysis	Financial fraud	Internal reference
edge-extension-id						
email			X			
email-attachment						
email-body						
email-dst						
email-dst-display-name						
email-header						
email-message-id						
email-mime-boundary						
email-reply-to						
email-src						
email-src-display-name						
email-subject						
email-thread-index						
email-x-mailer						
eppn						

Category	Antivirus detection	Artifacts dropped	Attribution	External analysis	Financial fraud	Internal reference
favicon-mmh3						
filename		X		X		
filename-pattern		X		X		
filename|authentihash		X				
filename|impfuzzy		X				
filename|imphash		X				
filename|md5		X		X		
filename|pehash		X				
filename|sha1		X		X		
filename|sha224		X				
filename|sha256		X		X		
filename|sha3-224		X		X		
filename|sha3-256		X		X		
filename|sha3-384		X		X		
filename|sha3-512		X		X		
filename|sha384		X				

Category	Antivirus detection	Artifacts dropped	Attribution	External analysis	Financial fraud	Internal reference
filename|sha512		X				
filename|sha512/224		X				
filename|sha512/256		X				
filename|ssdeep		X				
filename|tlsh		X				
filename|vhash		X				
first-name						
float						
frequent-flyer-number						
full-name						
gender						
gene		X				
git-commit-id						X
github-organisation						
github-repository				X		
github-username						
hassh-md5				X		
hasshserver-md5				X		

Category	Antivirus detection	Artifacts dropped	Attribution	External analysis	Financial fraud	Internal reference
hex	X	X			X	X
hostname				X		
hostname|port						
http-method						
iban					X	
identity-card-number						
imp-fuzzy		X				
imphash		X				
integer						
ip-dst				X		
ip-dst|port				X		
ip-src				X		
ip-src|port				X		
issue-date-of-the-visa						
ja3-fingerprint-md5				X		
jabber-id						
jarm-fingerprint				X		
kusto-query		X				
last-name						

Category	Antivirus detection	Artifacts dropped	Attribution	External analysis	Financial fraud	Internal reference
link	X			X		X
mac-address				X		
mac-eui-64				X		
malware-sample		X		X		
malware-type						
md5		X		X		
middle-name						
mime-type		X				
mobile-application-id						
mutex		X				
named pipe		X				
nationality						
onion-address				X		
other	X	X	X	X	X	X
passenger-name-record-locator-number						
passport-						

Category	Antivirus detection	Artifacts dropped	Attribution	External analysis	Financial fraud	Internal reference
country						
passport-expiration						
passport-number						
pattern-in-file		X		X		
pattern-in-memory		X		X		
pattern-in-traffic				X		
payment-details						
pdb		X				
pehash						
pgp-private-key		X				
pgp-public-key		X				
phone-number					X	
place-of-birth						
place-port-of-clearance						

Category	Antivirus detection	Artifacts dropped	Attribution	External analysis	Financial fraud	Internal reference
place-port-of-onward-foreign-destination						
place-port-of-original-embarkation						
port						
primary-residence						
process-state		X				
prtn					X	
redress-number						
regkey		X		X		
regkey|value		X		X		
sha1		X		X		
sha224		X				
sha256		X		X		
sha3-224		X		X		
sha3-256		X		X		
sha3-384		X		X		
sha3-512		X		X		
sha384		X				
sha512		X				
sha512/224		X				
sha512/256		X				
sigma		X				

Category	Antivirus detection	Artifacts dropped	Attribution	External analysis	Financial fraud	Internal reference
size-in-bytes						
snort				X		
special-service-request						
ssdeep		X				
ssh-fingerprint						
stix2-pattern		X				
suricata				X		
target-email						
target-external						
target-location						
target-machine						
target-org						
target-user						
telfhash		X				
text	X	X	X	X	X	X
threat-actor			X			

Category	Antivirus detection	Artifacts dropped	Attribution	External analysis	Financial fraud	Internal reference
tlsh						
travel-details						
twitter-id						
uri						
url				X		
user-agent				X		
uuid						
vhash		X				
visa-number						
vulnerability				X		
weakness				X		
whois-creation-date			X			
whois-registrant-email			X			
whois-registrant-name			X			
whois-registrant-org			X			
whois-registrant-phone			X			
whois-registrar			X			
windows-sched-		X				

Category	Antivirus detection	Artifacts dropped	Attribution	External analysis	Financial fraud	Internal reference
uled-task						
windows-service-display-name		X				
windows-service-name		X				
x509-fingerprint-md5		X	X	X		
x509-fingerprint-sha1		X	X	X		
x509-fingerprint-sha256		X	X	X		
xmr					X	
yara		X				
zeek				X		

Category	Network activity	Other	Payload delivery	Payload installation	Payload type	Persistence mechanism
AS	X		X			
abartn						
anonymised	X	X	X	X	X	X
attachment	X		X	X		
authentihash			X	X		
azure-applica-			X	X		

Category	Network activity	Other	Payload delivery	Payload installation	Payload type	Persistence mechanism
tion-id						
bank-account-nr						
bic						
bin						
boolean		X				
bro	X					
btc						
campaign-id						
campaign-name						
cc-number						
cd-hash			X	X		
chrome-extension-id			X	X		
comment	X	X	X	X	X	X
community-id	X					
cookie	X					
cortex						
counter		X				
country-of-residence						
cpe		X	X	X		
dash						

Category	Network activity	Other	Payload delivery	Payload installation	Payload type	Persistence mechanism
date-of-birth						
date-time		X				
dkim	X					
dkim-signature	X					
dns-soa-email						
dom-hash	X					
domain	X		X			
domain|ip	X					
edge-extension-id			X	X		
email	X		X			
email-attachment			X			
email-body			X			
email-dst	X		X			
email-dst-display-name			X			
email-header			X			
email-message-id			X			
email-mime-boundary			X			

Category	Network activity	Other	Payload delivery	Payload installation	Payload type	Persistence mechanism
email-reply-to			X			
email-src	X		X			
email-src-display-name			X			
email-subject	X		X			
email-thread-index			X			
email-x-mailer			X			
eppn	X					
favicon-mmh3	X					
filename			X	X		X
filename-pattern	X		X	X		
filename|authentihash			X	X		
filename|imp-fuzzy			X	X		
filename|imphash			X	X		
filename|md5			X	X		
filename|			X	X		

Category	Network activity	Other	Payload delivery	Payload installation	Payload type	Persistence mechanism
pehash						
filename|sha1			X	X		
filename|sha224			X	X		
filename|sha256			X	X		
filename|sha3-224			X	X		
filename|sha3-256			X	X		
filename|sha3-384			X	X		
filename|sha3-512			X	X		
filename|sha384			X	X		
filename|sha512			X	X		
filename|sha512/224			X	X		
filename|sha512/256			X	X		
filename|ssdeep			X	X		
filename|tlsh			X	X		
filename|vhash			X	X		
firstname						
float		X				
frequentflyernumber						
fullname						

Category	Network activity	Other	Payload delivery	Payload installation	Payload type	Persistence mechanism
gender						
gene						
git-commit-id						
github-organisation						
github-repository						
github-username						
hassh-md5	X		X			
hasshserver-md5	X		X			
hex	X	X	X	X		X
hostname	X		X			
hostname|port	X		X			
http-method	X					
iban						
identity-card-number						
imp-fuzzy			X	X		
imphash			X	X		
integer		X				
ip-dst	X		X			
ip-dst|port	X		X			
ip-src	X		X			

Category	Network activity	Other	Payload delivery	Payload installation	Payload type	Persistence mechanism
ip-src|port	X		X			
issue-date-of-the-visa						
ja3-fingerprint-md5	X		X			
jabber-id						
jarm-fingerprint	X		X			
kusto-query						
last-name						
link			X			
mac-address	X		X			
mac-eui-64	X		X			
malware-sample			X	X		
malware-type			X	X		
md5			X	X		
middle-name						
mime-type			X	X		
mobile-application-			X	X		

Category	Network activity	Other	Payload delivery	Payload installation	Payload type	Persistence mechanism
tion-id						
mutex						
named pipe						
nationality						
onion-address	X		X			
other	X	X	X	X	X	X
passenger-name-record-locator-number						
passport-country						
passport-expiration						
passport-number						
pattern-in-file	X		X	X		
pattern-in-memory				X		
pattern-in-traffic	X		X	X		
payment-details						

Category	Network activity	Other	Payload delivery	Payload installation	Payload type	Persistence mechanism
pdb						
pehash			X	X		
pgp-private-key		X				
pgp-public-key		X				
phone-number		X				
place-of-birth						
place-port-of-clearance						
place-port-of-onward-foreign-destination						
place-port-of-original-embarkation						
port	X	X				
primary-residence						
process-state						
prtn						
re-dress-						

Category	Network activity	Other	Payload delivery	Payload installation	Payload type	Persistence mechanism
number						
regkey						X
regkey|value						X
sha1			X	X		
sha224			X	X		
sha256			X	X		
sha3-224			X	X		
sha3-256			X	X		
sha3-384			X	X		
sha3-512			X	X		
sha384			X	X		
sha512			X	X		
sha512/224			X	X		
sha512/256			X	X		
sigma			X	X		
size-in-bytes		X				
snort	X					
special-service-request						
ssdeep			X	X		
ssh-fingerprint	X					
stix2-pattern	X		X	X		
suricata	X					
target-email						

Category	Network activity	Other	Payload delivery	Payload installation	Payload type	Persistence mechanism
target-external						
target-location						
target-machine						
target-org						
target-user						
telfhash			X	X		
text	X	X	X	X	X	X
threat-actor						
tlsh			X	X		
travel-details						
twitter-id						
uri	X					
url	X		X			
user-agent	X		X			
uuid		X				
vhash			X	X		
visa-number						
vulnerability			X	X		
weakness			X	X		
whois-cre-						

Category	Network activity	Other	Payload delivery	Payload installation	Payload type	Persistence mechanism
ation-date						
whois-registrant-email			X			
whois-registrant-name						
whois-registrant-org						
whois-registrant-phone						
whois-registrar						
windows-scheduledtask						
windows-servicedisplayname						
windows-servicename						
x509-fingerprint-md5	X		X	X		
x509-fingerprint-sha1	X		X	X		

Category	Network activity	Other	Payload delivery	Payload installation	Payload type	Persistence mechanism
x509-fingerprint-sha256	X		X	X		
xmr						
yara			X	X		
zeek	X					

Category	Person	Social network	Support Tool	Targeting data
AS				
aba-rtn				
anonymised	X	X	X	X
attachment			X	
authentihash				
azure-application-id				
bank-account-nr				
bic				
bin				
boolean				
bro				
btc				
campaign-id				
campaign-name				
cc-number				
cdhash				
chrome-extension-id				
comment	X	X	X	X
community-id				
cookie				
cortex				
counter				
country-of-residence	X			
cpe				
dash				
date-of-birth	X			
datetime				
dkim				

Category	Person	Social network	Support Tool	Targeting data
dkim-signature				
dns-soa-email				
dom-hash				
domain				
domain|ip				
edge-extension-id				
email	X	X		
email-attachment				
email-body				
email-dst		X		
email-dst-display-name				
email-header				
email-message-id				
email-mime-boundary				
email-reply-to				
email-src		X		
email-src-display-name				
email-subject				
email-thread-index				
email-x-mailer				
eppn		X		
favicon-mmh3				
filename				
filename-pattern				
filename|authentihash				
filename|impfuzzy				
filename|imphash				
filename|md5				
filename|pehash				
filename|sha1				
filename|sha224				
filename|sha256				
filename|sha3-224				
filename|sha3-256				
filename|sha3-384				

Category	Person	Social network	Support Tool	Targeting data
filename|sha3-512				
filename|sha384				
filename|sha512				
filename|sha512/224				
filename|sha512/256				
filename|ssdeep				
filename|tlsh				
filename|vhash				
first-name	X			
float				
frequent-flyer-number	X			
full-name	X			
gender	X			
gene				
git-commit-id				
github-organisation		X		
github-repository		X		
github-username		X		
hassh-md5				
hasshserver-md5				
hex			X	
hostname				
hostname|port				
http-method				
iban				
identity-card-number	X			
impfuzzy				
imphash				
integer				
ip-dst				
ip-dst|port				
ip-src				
ip-src|port				
issue-date-of-the-visa	X			
ja3-fingerprint-md5				

Category	Person	Social network	Support Tool	Targeting data
jabber-id		X		
jarm-fingerprint				
kusto-query				
last-name	X			
link			X	
mac-address				
mac-eui-64				
malware-sample				
malware-type				
md5				
middle-name	X			
mime-type				
mobile-application-id				
mutex				
named pipe				
nationality	X			
onion-address				
other	X	X	X	
passenger-name-record-locator-number	X			
passport-country	X			
passport-expiration	X			
passport-number	X			
pattern-in-file				
pattern-in-memory				
pattern-in-traffic				
payment-details	X			
pdb				
pehash				
pgp-private-key	X	X		
pgp-public-key	X	X		
phone-number	X			
place-of-birth	X			
place-port-of-clearance	X			
place-port-of-onward-foreign-destination	X			
place-port-of-original-embarkation	X			

Category	Person	Social network	Support Tool	Targeting data
port				
primary-residence	X			
process-state				
prtn				
redress-number	X			
regkey				
regkey|value				
sha1				
sha224				
sha256				
sha3-224				
sha3-256				
sha3-384				
sha3-512				
sha384				
sha512				
sha512/224				
sha512/256				
sigma				
size-in-bytes				
snort				
special-service-request	X			
ssdeep				
ssh-fingerprint				
stix2-pattern				
suricata				
target-email				X
target-external				X
target-location				X
target-machine				X
target-org				X
target-user				X
telfhash				
text	X	X	X	
threat-actor				

Category	Person	Social network	Support Tool	Targeting data
tlsh				
travel-details	X			
twitter-id		X		
uri				
url				
user-agent				
uuid				
vhash				
visa-number	X			
vulnerability				
weakness				
whois-creation-date				
whois-registrant-email		X		
whois-registrant-name				
whois-registrant-org				
whois-registrant-phone				
whois-registrar				
windows-scheduled-task				
windows-service-displayname				
windows-service-name				
x509-fingerprint-md5				
x509-fingerprint-sha1				
x509-fingerprint-sha256				
xmr				
yara				
zeek				

23.0.2 Categories

- **Antivirus detection**: All the info about how the malware is detected by the antivirus products
- **Artifacts dropped**: Any artifact (files, registry keys etc.) dropped by the malware or other modifications to the system
- **Attribution**: Identification of the group, organisation, or country behind the attack
- **External analysis**: Any other result from additional analysis of the malware like tools output
- **Financial fraud**: Financial Fraud indicators
- **Internal reference**: Reference used by the publishing party (e.g. ticket number)
- **Network activity**: Information about network traffic generated by the malware
- **Other**: Attributes that are not part of any other category or are meant to be used as a component in MISP objects in the future
- **Payload delivery**: Information about how the malware is delivered
- **Payload installation**: Info on where the malware gets installed in the system

- **Payload type**: Information about the final payload(s)
- **Persistence mechanism**: Mechanisms used by the malware to start at boot
- **Person**: A human being - natural person
- **Social network**: Social networks and platforms
- **Support Tool**: Tools supporting analysis or detection of the event
- **Targeting data**: Internal Attack Targeting and Compromise Information

23.0.3 Types

- **AS**: Autonomous system
- **aba-rtn**: ABA routing transit number
- **anonymised**: Anonymised value - described with the anonymisation object via a relationship
- **attachment**: Attachment with external information
- **authentihash**: Authenticode executable signature hash
- **azure-application-id**: Azure Application ID.
- **bank-account-nr**: Bank account number without any routing number
- **bic**: Bank Identifier Code Number also known as SWIFT-BIC, SWIFT code or ISO 9362 code
- **bin**: Bank Identification Number
- **boolean**: Boolean value - to be used in objects
- **bro**: An NIDS rule in the Bro rule-format
- **btc**: Bitcoin Address
- **campaign-id**: Associated campaign ID
- **campaign-name**: Associated campaign name
- **cc-number**: Credit-Card Number
- **cdhash**: An Apple Code Directory Hash, identifying a code-signed Mach-O executable file
- **chrome-extension-id**: Chrome extension id
- **comment**: Comment or description in a human language
- **community-id**: A community ID flow hashing algorithm to map multiple traffic monitors into common flow id
- **cookie**: HTTP cookie as often stored on the user web client. This can include authentication cookie or session cookie.
- **cortex**: Cortex analysis result
- **counter**: An integer counter, generally to be used in objects
- **country-of-residence**: The country of residence of a natural person
- **cpe**: Common Platform Enumeration - structured naming scheme for information technology systems, software, and packages.
- **dash**: Dash Address
- **date-of-birth**: Date of birth of a natural person (in YYYY-MM-DD format)
- **datetime**: Datetime in the ISO 8601 format
- **dkim**: DKIM public key
- **dkim-signature**: DKIM signature
- **dns-soa-email**: RFC 1035 mandates that DNS zones should have a SOA (Statement Of Authority) record that contains an email address where a PoC for the domain could be contacted. This can sometimes be used for attribution/linkage between different domains even if protected by whois privacy
- **dom-hash**: A dom-hash algorithm is a structural fingerprint of an HTML Document Object Model where all tag names are contained in a single string separated by a pipe. The truncated SHA252 value by the first 32-character serves as fingerprint.
- **domain**: A domain name used in the malware
- **domain|ip**: A domain name and its IP address (as found in DNS lookup) separated by a |
- **edge-extension-id**: A browser extension identifier (for example, a Microsoft Edge / Chromium extension ID)
- **email**: An email address
- **email-attachment**: File name of the email attachment.
- **email-body**: Email body
- **email-dst**: The destination email address. Used to describe the recipient when describing an e-mail.
- **email-dst-display-name**: Email destination display name
- **email-header**: Email header
- **email-message-id**: The email message ID
- **email-mime-boundary**: The email mime boundary separating parts in a multipart email

- **email-reply-to**: Email reply to header
- **email-src**: The source email address. Used to describe the sender when describing an e-mail.
- **email-src-display-name**: Email source display name
- **email-subject**: The subject of the email
- **email-thread-index**: The email thread index header
- **email-x-mailer**: Email x-mailer header
- **eppn**: eduPersonPrincipalName - eppn - the NetId of the person for the purposes of inter-institutional authentication. Should be stored in the form of user@univ.edu, where univ.edu is the name of the local security domain.
- **favicon-mmh3**: favicon-mmh3 is the murmur3 hash of a favicon as used in Shodan.
- **filename**: Filename
- **filename-pattern**: A pattern in the name of a file
- **filename|authentihash**: A filename and Authenticode executable signature hash
- **filename|impfuzzy**: Import fuzzy hash - a fuzzy hash created based on the imports in the sample.
- **filename|imphash**: Import hash - a hash created based on the imports in the sample.
- **filename|md5**: A filename and an MD5 hash separated by a |
- **filename|pehash**: A filename and a peHash separated by a |
- **filename|sha1**: A filename and an SHA1 hash separated by a |
- **filename|sha224**: A filename and a SHA-224 hash separated by a |
- **filename|sha256**: A filename and an SHA256 hash separated by a |
- **filename|sha3-224**: A filename and an SHA3-224 hash separated by a |
- **filename|sha3-256**: A filename and an SHA3-256 hash separated by a |
- **filename|sha3-384**: A filename and an SHA3-384 hash separated by a |
- **filename|sha3-512**: A filename and an SHA3-512 hash separated by a |
- **filename|sha384**: A filename and a SHA-384 hash separated by a |
- **filename|sha512**: A filename and a SHA-512 hash separated by a |
- **filename|sha512/224**: A filename and a SHa-512/224 hash separated by a |
- **filename|sha512/256**: A filename and a SHA-512/256 hash separated by a |
- **filename|ssdeep**: A checksum in ssdeep format
- **filename|tlsh**: A filename and a Trend Micro Locality Sensitive Hash separated by a |
- **filename|vhash**: A filename and a VirusTotal hash separated by a |
- **first-name**: First name of a natural person
- **float**: A floating point value.
- **frequent-flyer-number**: The frequent flyer number of a passenger
- **full-name**: Full name of a natural person
- **gender**: The gender of a natural person (Male, Female, Other, Prefer not to say)
- **gene**: GENE - Go Evtx sigNature Engine
- **git-commit-id**: A Git commit ID.
- **github-organisation**: A GitHub organisation
- **github-repository**: A Github repository
- **github-username**: A GitHub user name
- **hassh-md5**: hassh is a network fingerprinting standard which can be used to identify specific Client SSH implementations. The fingerprints can be easily stored, searched and shared in the form of an MD5 fingerprint.
- **hasshserver-md5**: hasshServer is a network fingerprinting standard which can be used to identify specific Server SSH implementations. The fingerprints can be easily stored, searched and shared in the form of an MD5 fingerprint.
- **hex**: A value in hexadecimal format
- **hostname**: A full host/dnsname of an attacker
- **hostname|port**: Hostname and port number separated by a |
- **http-method**: HTTP method used by the malware (e.g. POST, GET, ...).
- **iban**: International Bank Account Number
- **identity-card-number**: Identity card number
- **impfuzzy**: A fuzzy hash of import table of Portable Executable format
- **imphash**: Import hash - a hash created based on the imports in the sample.
- **integer**: A generic integer generally to be used in objects
- **ip-dst**: A destination IP address of the attacker or C&C server
- **ip-dst|port**: IP destination and port number separated by a |
- **ip-src**: A source IP address of the attacker

- **ip-src|port**: IP source and port number separated by a |
- **issue-date-of-the-visa**: The date on which the visa was issued
- **ja3-fingerprint-md5**: JA3 is a method for creating SSL/TLS client fingerprints that should be easy to produce on any platform and can be easily shared for threat intelligence.
- **jabber-id**: Jabber ID
- **jarm-fingerprint**: JARM is a method for creating SSL/TLS server fingerprints.
- **kusto-query**: Kusto query - Kusto from Microsoft Azure is a service for storing and running interactive analytics over Big Data.
- **last-name**: Last name of a natural person
- **link**: Link to an external information
- **mac-address**: MAC address
- **mac-eui-64**: MAC EUI-64 address
- **malware-sample**: Attachment containing encrypted malware sample
- **malware-type**:
- **md5**: A checksum in MD5 format
- **middle-name**: Middle name of a natural person
- **mime-type**: A media type (also MIME type and content type) is a two-part identifier for file formats and format contents transmitted on the Internet
- **mobile-application-id**: The application id of a mobile application
- **mutex**: Mutex, use the format <Mutex>
- **named pipe**: Named pipe, use the format .<PipeName>
- **nationality**: The nationality of a natural person
- **onion-address**: Onion service (formerly known as “hidden service”) address
- **other**: Other attribute
- **passenger-name-record-locator-number**: The Passenger Name Record Locator is a key under which the reservation for a trip is stored in the system. The PNR contains, among other data, the name, flight segments and address of the passenger. It is defined by a combination of five or six letters and numbers.
- **passport-country**: The country in which the passport was issued
- **passport-expiration**: The expiration date of a passport
- **passport-number**: The passport number of a natural person
- **pattern-in-file**: Pattern in file that identifies the malware
- **pattern-in-memory**: Pattern in memory dump that identifies the malware
- **pattern-in-traffic**: Pattern in network traffic that identifies the malware
- **payment-details**: Payment details
- **pdb**: Microsoft Program database (PDB) path information
- **pehash**: peHash - a hash calculated based of certain pieces of a PE executable file
- **pgp-private-key**: A PGP private key
- **pgp-public-key**: A PGP public key
- **phone-number**: Telephone Number
- **place-of-birth**: Place of birth of a natural person
- **place-port-of-clearance**: The port of clearance
- **place-port-of-onward-foreign-destination**: A Port where the passenger is transiting to
- **place-port-of-original-embarkation**: The original port of embarkation
- **port**: Port number
- **primary-residence**: The primary residence of a natural person
- **process-state**: State of a process
- **prtn**: Premium-Rate Telephone Number
- **redress-number**: The Redress Control Number is the record identifier for people who apply for redress through the DHS Travel Redress Inquiry Program (DHS TRIP). DHS TRIP is for travelers who have been repeatedly identified for additional screening and who want to file an inquiry to have erroneous information corrected in DHS systems
- **regkey**: Registry key or value
- **regkey|value**: Registry value + data separated by |
- **sha1**: A checksum in SHA1 format
- **sha224**: A checksum in SHA-224 format
- **sha256**: A checksum in SHA256 format
- **sha3-224**: A checksum in SHA3-224 format
- **sha3-256**: A checksum in SHA3-256 format
- **sha3-384**: A checksum in SHA3-384 format

- **sha3-512**: A checksum in SHA3-512 format
- **sha384**: A checksum in SHA-384 format
- **sha512**: A checksum in SHA-512 format
- **sha512/224**: A checksum in the SHA-512/224 format
- **sha512/256**: A checksum in the SHA-512/256 format
- **sigma**: Sigma - Generic Signature Format for SIEM Systems
- **size-in-bytes**: Size expressed in bytes
- **snort**: An IDS rule in Snort rule-format
- **special-service-request**: A Special Service Request is a function to an airline to provide a particular facility for A Passenger or passengers.
- **ssdeep**: A checksum in ssdeep format
- **ssh-fingerprint**: A fingerprint of SSH key material
- **stix2-pattern**: STIX 2 pattern
- **suricata**: An IDS rule in Suricata rule-format
- **target-email**: Attack Targets Email(s)
- **target-external**: External Target Organizations Affected by this Attack
- **target-location**: Attack Targets Physical Location(s)
- **target-machine**: Attack Targets Machine Name(s)
- **target-org**: Attack Targets Department or Organization(s)
- **target-user**: Attack Targets Username(s)
- **telfhash**: telfhash is symbol hash for ELF files, just like imphash is imports hash for PE files.
- **text**: Name, ID or a reference
- **threat-actor**: A string identifying the threat actor
- **tlsh**: A checksum in the Trend Micro Locality Sensitive Hash format
- **travel-details**: Travel details
- **twitter-id**: Twitter ID
- **uri**: Uniform Resource Identifier
- **url**: Uniform Resource Locator
- **user-agent**: The user-agent used by the malware in the HTTP request.
- **uuid**: A universally unique identifier referencing another object, event, or resource
- **vhash**: A VirusTotal checksum
- **visa-number**: Visa number
- **vulnerability**: A reference to the vulnerability used in the exploit
- **weakness**: A reference to the weakness (CWE) used in the exploit
- **whois-creation-date**: The date of domain's creation, obtained from the WHOIS information.
- **whois-registrant-email**: The e-mail of a domain's registrant, obtained from the WHOIS information.
- **whois-registrant-name**: The name of a domain's registrant, obtained from the WHOIS information.
- **whois-registrant-org**: The org of a domain's registrant, obtained from the WHOIS information.
- **whois-registrant-phone**: The phone number of a domain's registrant, obtained from the WHOIS information.
- **whois-registrar**: The registrar of the domain, obtained from the WHOIS information.
- **windows-scheduled-task**: A scheduled task in windows
- **windows-service-displayname**: A windows service's displayname, not to be confused with the windows-service-name. This is the name that applications will generally display as the service's name in applications.
- **windows-service-name**: A windows service name. This is the name used internally by windows. Not to be confused with the windows-service-displayname.
- **x509-fingerprint-md5**: X509 fingerprint in MD5 format
- **x509-fingerprint-sha1**: X509 fingerprint in SHA-1 format
- **x509-fingerprint-sha256**: X509 fingerprint in SHA-256 format
- **xmr**: Monero Address
- **yara**: YARA signature
- **zeek**: An NIDS rule in the Zeek rule-format

24. Taxonomies

MISP supports a flexible mechanism for taxonomies of classification — curated sets of machine tags you can enable, cherry-pick, and share across instances.

You can access the taxonomy by going into 'Event Actions' and select 'List Taxonomies'. For fresh install, make sure to click 'Update Taxonomies' to view available taxonomies.

A complete list of the available taxonomies PDF are available on the MISP project website.

ID ↓	Namespace	Description	Version	Enabled	Required	Highlighted	Active Tags	Actions
1	CERT-XLM	CERT-XLM Security Incident Classification.	2	✖	☐	☐	**5** / 37	▶ 🗑 👁
2	DFRLab-dichotomies-of-disinformation	DFRLab Dichotomies of Disinformation.	1	✖	☐	☐	**0** / 608	▶ 🗑 👁
3	DML	The Detection Maturity Level (DML) model is a capability maturity model for referencing ones maturity in detecting cyber attacks. It's designed for organizations who perform intel-driven detection and response and who put an emphasis on having a mature detection program.	1	✖	☐	☐	**0** / 9	▶ 🗑 👁
5	access-method	The access method used to remotely access a system.	1	✖	☐	☐	**1** / 8	▶ 🗑 👁
6	accessnow	Access Now classification to classify an issue (such as security, human rights, youth rights).	3	✖	☐	☐	**0** / 22	▶ 🗑 👁
7	action-taken	Action taken in the case of a security incident (CSIRT perspective).	2	✖	☐	☐	**0** / 6	▶ 🗑 👁
8	admiralty-scale	The Admiralty Scale or Ranking (also called the NATO System) is used to rank the reliability of a source and the credibility of an information. Reference based on FM 2-22.3 (FM 34-52) HUMAN INTELLIGENCE COLLECTOR OPERATIONS and NATO documents.	5	✖	☐	☐	**10** / 13	▶ 🗑 👁
10	ais-marking	The AIS Marking Schema implementation is maintained by the National Cybersecurity and Communication Integration Center (NCCIC) of the U.S. Department of Homeland Security (DHS)	2	✖	☐	☐	**0** / 10	▶ 🗑 👁
11	approved-category-of-action	A pre-approved category of action for indicators being shared with partners (MIMIC).	1	✖	☐	☐	**0** / 6	▶ 🗑 👁
12	binary-class	Custom taxonomy for types of binary file.	2	✖	☐	☐	**0** / 3	▶ 🗑 👁
13	cccs	Internal taxonomy for CCCS.	2	✖	☐	☐	**0** / 129	▶ 🗑 👁
15	coa	Course of action taken within organization to discover, detect, deny, disrupt, degrade, deceive and/or destroy an attack.	2	✖	☐	☐	**0** / 61	▶ 🗑 👁

Figure 24.1: MISP Taxonomy index

Taxonomies can be used in MISP (as local or distributed tags) or in other tools willing to share common taxonomies among security information sharing tools.

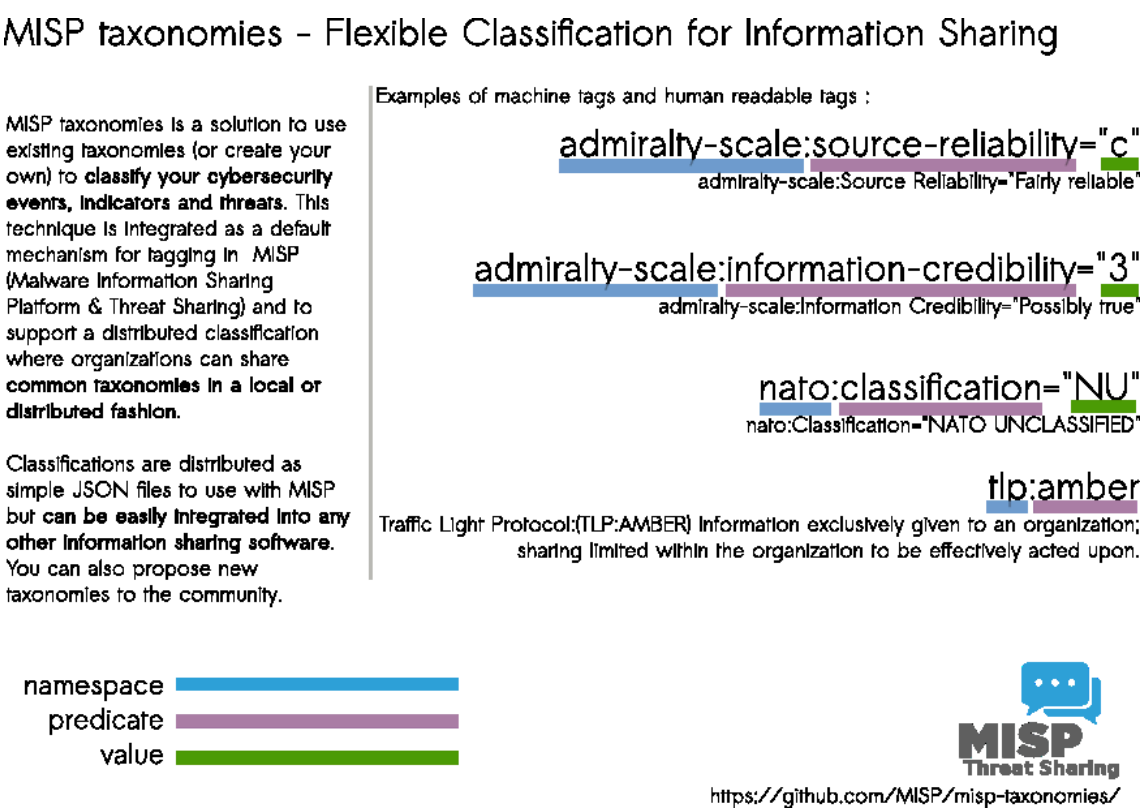


Figure 24.2: Overview of the MISP taxonomies

MISP ships **166 taxonomies** in 2.5.42 and the catalogue grows regularly, so this guide no longer reproduces the full list. Browse the always-current, authoritative catalogue on the MISP taxonomies page (PDF), or in your own instance under **Event Actions → List Taxonomies** (on a fresh install, click **Update Taxonomies** first). Widely used examples include TLP, PAP, Admiralty Scale, Estimative Language and the MISP taxonomy.

A taxonomy contains a series of tags that can be used as normal tags in your MISP instance.

Tagging is a simple way to attach a classification to an event. In the early version of MISP, tagging was local to an instance. Classification must be globally used to be efficient. After evaluating different solutions of classification, we build a new scheme using the concept of machine tags.

Taxonomy is a classification of informations. Taxonomies are implemented in a simple JSON format. Anyone can create their own taxonomy or reuse an existing one.

Taxonomies are in an independent git repository.

These can be **freely reused** and **integrated** in other threat intel tools.

The advantage is that you can set a specific tag as being exportable. This means that you can **export** your classification with other MISP instance and **share** the same taxonomies. Tagging is a simple way to attach a classification to an event.

Classification must be globally used to be efficient.

If you want to enable a specific taxonomy, you can click on the cross to enable it.

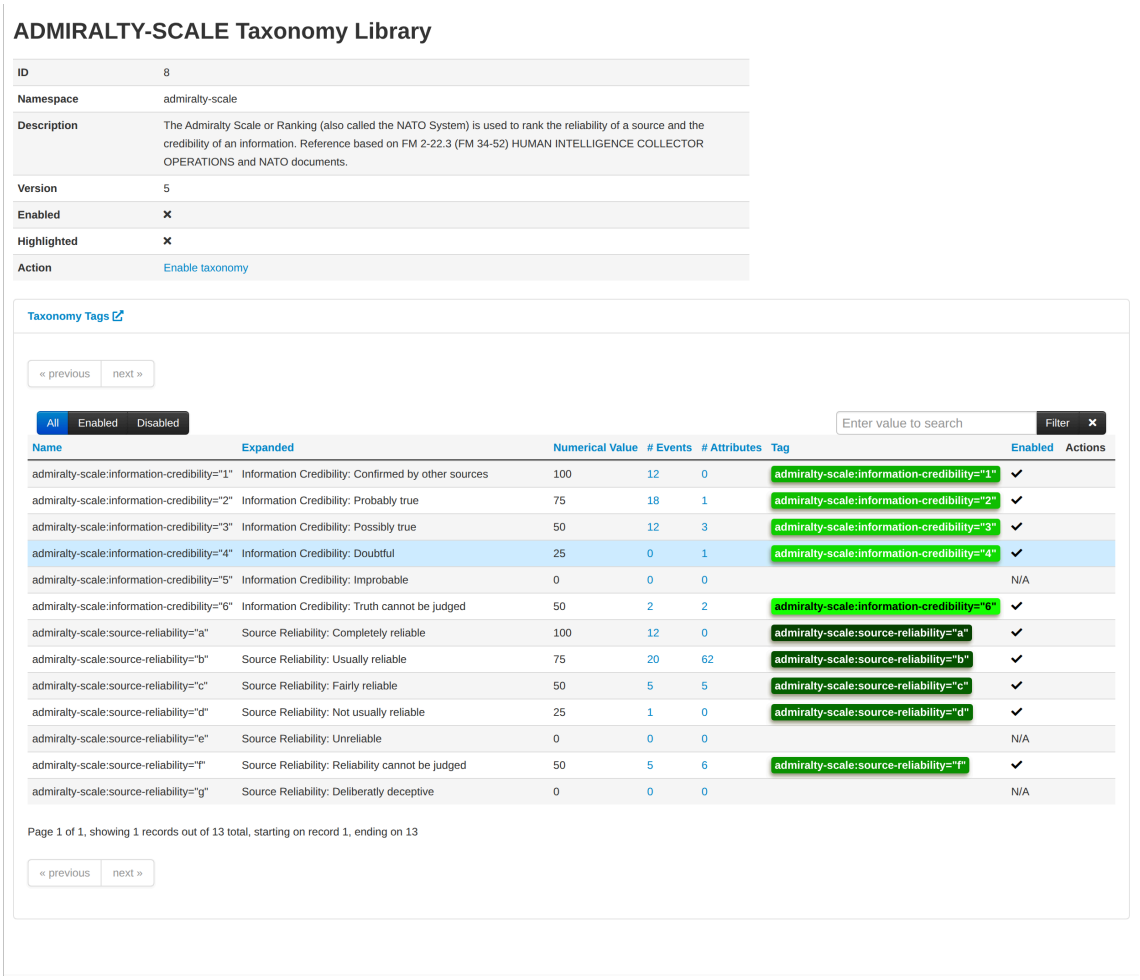


Figure 24.3: enableTaxonomy

Then you can even cherry-pick the tags you want to use on the system. If you want to use the whole taxonomy, select all and then click on the cross in the top left.

24.1 Contributing to Taxonomy

It is quite easy. Create a JSON file describing your taxonomy as triple tags.

```
{
  "namespace": "admiralty-scale",
  "description": "The Admiralty Scale (also called the NATO
      System) is used to rank the reliability of a source and
      the credibility of an information.",
  "version": 1,
  "predicates": [
    {
      "value": "source-reliability",
      "expanded": "Source Reliability"
    },
    {
      "value": "information-credibility",
      "expanded": "Information Credibility"
    }
  ],
....
```

```
{
  "values": [
    {
      "predicate": "source-reliability",
      "entry": [
        {
          "value": "a",
          "expanded": "Completely reliable"
        },
....
```

(e.g. check an existing one like Admiralty Scale), create a directory matching your name space, put your machinetag file in the directory and pull your request. Publishing your taxonomy is as easy as a simple git pull request on misp-taxonomies (https://github.com/MISP/misp-taxonomies). That's it. Everyone can benefit from your taxonomy and can be automatically enabled in information sharing tools like MISP.

24.2 Reserved Taxonomy

The following taxonomy namespaces are reserved and used internally to MISP.

- galaxy mapping taxonomy with cluster:element:"value".

24.3 Adding Taxonomy in MISP

How are taxonomies integrated in MISP?

MISP administrators have only to import (or even cherry pick) the namespace or predicates they want to use as tags.

Tags can be exported to other instances.

Tags are also accessible via the MISP REST API.

For more information, "Information Sharing and Taxonomies Practical Classification of Threat Indicators using MISP" presentation given to the last MISP training in Luxembourg.

24.4 Adding a private taxonomy

```
$ cd /var/www/MISP/app/files/taxonomies/
$ mkdir privatetaxonomy
$ cd privatetaxonomy
$ vi machinetag.json
```

Create a JSON file describing your taxonomy as triple tags.

```
For example :
mkdir sample
cd sample
vim machinetag.json
```

Sample JSON with triple tags. You can use the JSON validator to be sure that there is no syntax error.

```
{
  "namespace": "sample",
  "description": "Some descriptive words",
  "version": 1,
  "predicates": [
    {
      "value": "my-predicate",
      "expanded": "my-predicate"
    }
  ],
  "values": [
    {
      "predicate": "my-predicate",
      "entry": [
        {
          "value": "a-value",
          "expanded": "A value"
        }
      ]
    }
  ]
}
```

Go to MISP Web GUI taxonomies/index and update the taxonomies once you are happy with your file. The newly created taxonomy should be visible. Now you need to activate the tags within your taxonomy.

24.5 How to use Taxonomy in MISP

24.5.1 Filtering the distribution of events among MISP instances

Taxonomy tags can drive what MISP shares with connected instances. When you configure a synchronisation server (**Sync Actions → List Servers**), the push and pull rules let you allow or block events by tag — as well as by organisation, attribute type, or object. This makes tags such as `tlp:*` a practical distribution control. See Synchronisation and Sharing for the full push/pull rule set.

24.5.2 MISP Taxonomies - tools

- machinetag.py is a parsing tool to dump taxonomies expressed in Machine Tags (Triple Tags) and list all valid tags from a specific taxonomy.

```
% cd tools
% python machinetag.py
        admiralty-scale:source-reliability="a"
        admiralty-scale:source-reliability="b"
        admiralty-scale:source-reliability="c"
```

```
        admiralty-scale:source-reliability="d"
        admiralty-scale:source-reliability="e"
        admiralty-scale:source-reliability="f"
        admiralty-scale:information-credibility="1"
        admiralty-scale:information-credibility="2"
        admiralty-scale:information-credibility="3"
        admiralty-scale:information-credibility="4"
        admiralty-scale:information-credibility="5"
        admiralty-scale:information-credibility="6"
        ...
```

- PyTaxonomies - Python module to use the MISP Taxonomies

24.5.3 Other use cases using MISP taxonomies

Tags can be used to:

- Set events for further processing by external tools (for example to trigger automated enrichment or export via a workflow or an external integration).
- Ensure a classification manager classes the events before release (e.g. release of information from air-gapped/classified networks).
- Enrich IDS export with tags to fit your NIDS deployment.

24.6 More options to configure taxonomies.

For MISP users and organisations, it's important to show the important contextualised information and especially the taxonomies which are important to your use-case. Once a taxonomy is enabled and available for use in MISP, there are two more options a admin can be set to encourage the use of particular taxonomies. Both are found in Event Actions > List Taxonomies menu.

ID ↓	Namespace	Description	Version	Enabled	Required	Highlighted	Active Tags	Actions
1	CERT-XLM	CERT-XLM Security Incident Classification.	2	✗	☐	☐	**5** / 37	▶ 🗑 👁
2	DFRLab-dichotomies-of-disinformation	DFRLab Dichotomies of Disinformation.	1	✗	☐	☐	**0** / 608	▶ 🗑 👁
3	DML	The Detection Maturity Level (DML) model is a capability maturity model for referencing ones maturity in detecting cyber attacks. It's designed for organizations who perform intel-driven detection and response and who put an emphasis on having a mature detection program.	1	✗	☐	☐	**0** / 9	▶ 🗑 👁
5	access-method	The access method used to remotely access a system.	1	✗	☐	☐	**1** / 8	▶ 🗑 👁
6	accessnow	Access Now classification to classify an issue (such as security, human rights, youth rights).	3	✗	☐	☐	**0** / 22	▶ 🗑 👁
7	action-taken	Action taken in the case of a security incident (CSIRT perspective).	2	✗	☐	☐	**0** / 6	▶ 🗑 👁
8	admiralty-scale	The Admiralty Scale or Ranking (also called the NATO System) is used to rank the reliability of a source and the credibility of an information. Reference based on FM 2-22.3 (FM 34-52) HUMAN INTELLIGENCE COLLECTOR OPERATIONS and NATO documents.	5	✗	☐	☐	**10** / 13	▶ 🗑 👁
10	ais-marking	The AIS Marking Schema implementation is maintained by the National Cybersecurity and Communication Integration Center (NCCIC) of the U.S. Department of Homeland Security (DHS)	2	✗	☐	☐	**0** / 10	▶ 🗑 👁
11	approved-category-of-action	A pre-approved category of action for indicators being shared with partners (MIMIC).	1	✗	☐	☐	**0** / 6	▶ 🗑 👁
12	binary-class	Custom taxonomy for types of binary file.	2	✗	☐	☐	**0** / 3	▶ 🗑 👁

Figure 24.4: taxonomies-list

24.6.1 Setting a taxonomy as "Highlighted"

If a taxonomy is highlighted, its namespace will appear in a visible box, even if it is not set in the event.



Figure 24.5: highlight-event

Tags are also hilighted in the event list.



Figure 24.6: highlight-list-event

It is also easier to add an highlighted tag to an event.

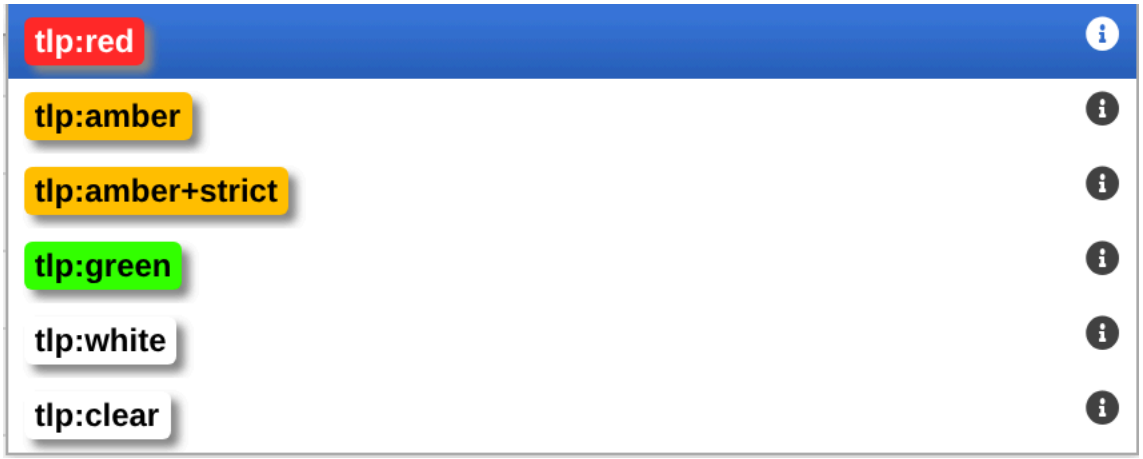


Figure 24.7: tag-box

24.6.2 Setting a taxonomy as "Required"

If taxonomies are set as required, a message will be visible on the tag list of the event, enumerating the missing required taxonomies still missing.

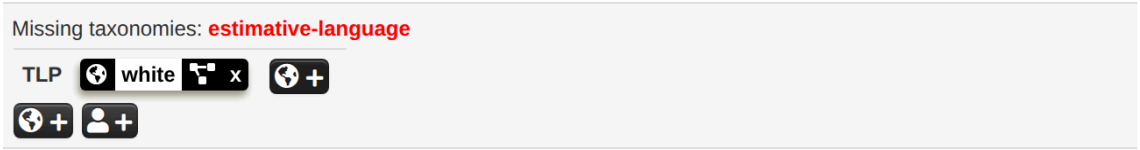


Figure 24.8: missing-taxo

An event will not be published if it is not tagged with at least one of tag of each required taxonomy.

Figure 24.9: publishing-without-required

24.6.3 Exclusive taxonomies

A taxonomy — or an individual predicate within one — can be marked **exclusive**. Exclusivity means only one value may apply at a time: for an exclusive *taxonomy*, an event may carry only a single tag from that whole namespace; for an exclusive *predicate*, only one value of that predicate may be set. `tlp` is the classic example — an event should have exactly one TLP level. When an event ends up carrying more

than one value from an exclusive taxonomy or predicate — for example both `tlp:white` and `tlp:red` — MISP does not silently reconcile them: it applies the tags but flags the conflict with a prominent warning in the event's tag area, so you can resolve it. (Such a state most often arises from synchronised or API-tagged events.) Exclusivity is defined in the taxonomy's JSON, not toggled from the UI.

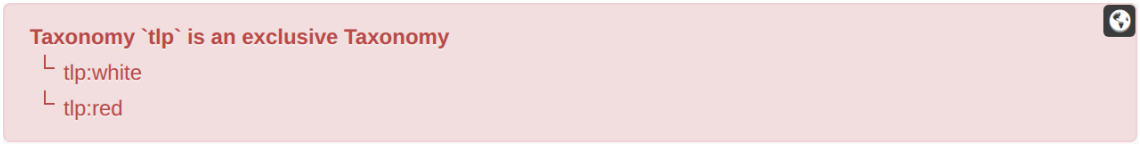


Figure 24.10: An exclusivity conflict: because `tlp` is an exclusive taxonomy, an event tagged both `tlp:white` and `tlp:red` is flagged with a warning in its tag area.

24.6.4 Bulk actions and normalising custom tags

The *List Taxonomies* index supports **bulk actions** across selected taxonomies — enable, disable, set required/optional, and highlight/remove highlight — so you can configure many taxonomies in one step instead of one at a time.

If your instance has accumulated free-text (custom) tags that happen to match a taxonomy's machine-tag format, the **Normalise custom tags to taxonomy format** action converts them into proper taxonomy tags, so they benefit from the taxonomy's metadata (expanded labels, exclusivity, numerical values) and display consistently.

24.7 Future functionalities related to MISP taxonomies

- Sighting support (thanks to NCSC-NL) is integrated in MISP allowing to auto expire IOC based on user detection.
- Adjusting taxonomies (adding/removing tags) based on their score or visibility via sighting.
- Simple taxonomy editors to help non-technical users to create their taxonomies.
- Filtering mechanisms in MISP to rename or replace taxonomies/tags at pull and push synchronisation.
- More public taxonomies to be included

25. Galaxies

Galaxies are MISP's way of attaching rich, structured knowledge objects — **clusters** — to events and attributes. A cluster is a named entry (a threat actor, a malware family, an ATT&CK technique, a sector, a country, …) carrying a description, synonyms, and a set of key/value metadata. Where a plain tag gives you one label and one colour, a galaxy cluster gives you a whole reusable object with context and relationships.

MISP ships with a large library of community-maintained galaxies (from misp-galaxy, and drawing on standards such as ATT&CK, STIX and others). You can use these default clusters as they are, and — since "Galaxy 2.0" — create, edit, fork and relate **your own** clusters directly in the web interface. Each cluster carries a MISP distribution level, so custom knowledge can be kept local or shared as widely as you like.

Under the hood a galaxy cluster is stored as a tag of the form `misp-galaxy:<type>="<uuid>"`, which is what actually gets attached to an event or attribute — but the UI presents it as a first-class object rather than a flat tag. For the difference between galaxies and taxonomy tags, see the Taxonomies chapter.

25.1 Managing galaxies in the UI

Open **Galaxies** from the top menu to see every galaxy on the instance.

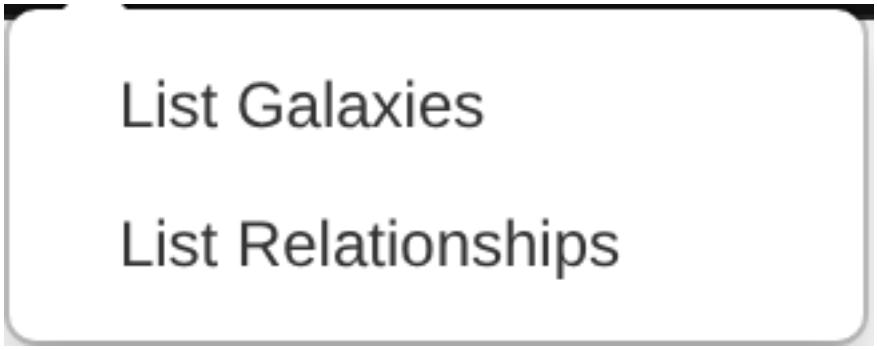


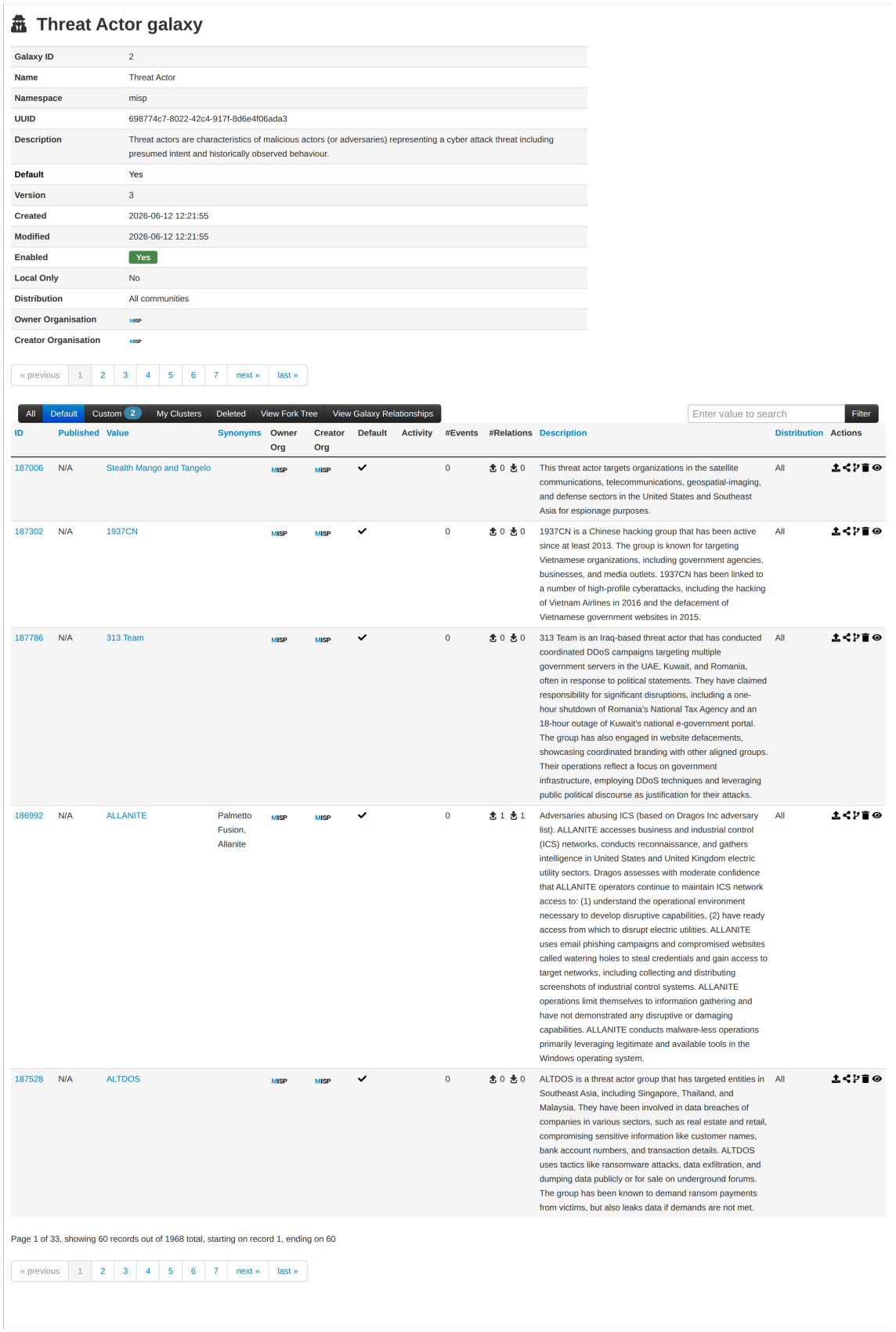
Figure 25.1: The Galaxies menu.

The index lists each galaxy with its icon, name, version, namespace and description, and shows whether it is **enabled** and whether it is **local only**. Use the *All / Enabled / Disabled* tabs and the search box to filter.

ID ↓	Icon	Name	Version	Namespace	Description	Enabled	Local Only	Actions
1		Mobile Attack - Tool	5	deprecated	Name of ATT&CK software	✓	×	
2		Threat Actor	3	misp	Threat actors are characteristics of malicious actors (or adversaries) representing a cyber attack threat including presumed intent and historically observed behaviour.	✓	×	
3		Bhadra Framework	3	misp	Bhadra Threat Modelling Framework	✓	×	
4		Country	1	misp	Country meta information based on the database provided by geonames.org.	✓	×	
5		Cert EU GovSector	2	misp	Cert EU GovSector	✓	×	
6		Intrusion Set	8	mitre-attack	Name of ATT&CK Group	✓	×	
7		Regions UN M49	2	misp	Regions based on UN M49.	✓	×	
8		mitre-tool	7	mitre-attack	Name of ATT&CK software	✓	×	
9		RAT	3	misp	remote administration tool or remote access tool (RAT), also called sometimes remote access trojan, is a piece of software or programming that allows a remote "operator" to control a system as if they have physical access to that system.	✓	×	
10		Malpedia	2	misp	Malware galaxy based on Malpedia archive.	✓	×	
11		Enterprise Attack - Tool	5	deprecated	Name of ATT&CK software	✓	×	
12		Assets	1	mitre-attack-ics	ATT&CK for ICS Assets	✓	×	

Figure 25.2: The galaxy index.

Each galaxy can be opened with the **View** action. The galaxy page shows its metadata (namespace, UUID, distribution, owner and creator organisation, whether it is a default galaxy, and its kill-chain layout for matrix galaxies) and, below it, the list of its clusters.



Threat Actor galaxy

Galaxy ID	2
Name	Threat Actor
Namespace	misp
UUID	698774c7-8022-42c4-917f-8d6e4f06ada3
Description	Threat actors are characteristics of malicious actors (or adversaries) representing a cyber attack threat including presumed intent and historically observed behaviour.
Default	Yes
Version	3
Created	2026-06-12 12:21:55
Modified	2026-06-12 12:21:55
Enabled	Yes
Local Only	No
Distribution	All communities
Owner Organisation	MISP
Creator Organisation	MISP

« previous 1 2 3 4 5 6 7 next » last »

All Default Custom 2 My Clusters Deleted View Fork Tree View Galaxy Relationships

Enter value to search Filter

ID	Published	Value	Synonyms	Owner Org	Creator Org	Default	Activity	#Events	#Relations	Description	Distribution	Actions
187006	N/A	Stealth Mango and Tangelo		MISP	MISP	✔		0	0 0	This threat actor targets organizations in the satellite communications, telecommunications, geospatial-imaging, and defense sectors in the United States and Southeast Asia for espionage purposes.	All	
187302	N/A	1937CN		MISP	MISP	✔		0	0 0	1937CN is a Chinese hacking group that has been active since at least 2013. The group is known for targeting Vietnamese organizations, including government agencies, businesses, and media outlets. 1937CN has been linked to a number of high-profile cyberattacks, including the hacking of Vietnam Airlines in 2016 and the defacement of Vietnamese government websites in 2015.	All	
187786	N/A	313 Team		MISP	MISP	✔		0	0 0	313 Team is an Iraq-based threat actor that has conducted coordinated DDoS campaigns targeting multiple government servers in the UAE, Kuwait, and Romania, often in response to political statements. They have claimed responsibility for significant disruptions, including a one-hour shutdown of Romania's National Tax Agency and an 18-hour outage of Kuwait's national e-government portal. The group has also engaged in website defacements, showcasing coordinated branding with other aligned groups. Their operations reflect a focus on government infrastructure, employing DDoS techniques and leveraging public political discourse as justification for their attacks.	All	
186992	N/A	ALLANITE	Palmetto Fusion, Allanite	MISP	MISP	✔		0	1 1	Adversaries abusing ICS (based on Dragos Inc adversary list). ALLANITE accesses business and industrial control (ICS) networks, conducts reconnaissance, and gathers intelligence in United States and United Kingdom electric utility sectors. Dragos assesses with moderate confidence that ALLANITE operators continue to maintain ICS network access to: (1) understand the operational environment necessary to develop disruptive capabilities, (2) have ready access from which to disrupt electric utilities. ALLANITE uses email phishing campaigns and compromised websites called watering holes to steal credentials and gain access to target networks, including collecting and distributing screenshots of industrial control systems. ALLANITE operations limit themselves to information gathering and have not demonstrated any disruptive or damaging capabilities. ALLANITE conducts malware-less operations primarily leveraging legitimate and available tools in the Windows operating system.	All	
187528	N/A	ALTDOS		MISP	MISP	✔		0	0 0	ALTDOS is a threat actor group that has targeted entities in Southeast Asia, including Singapore, Thailand, and Malaysia. They have been involved in data breaches of companies in various sectors, such as real estate and retail, compromising sensitive information like customer names, bank account numbers, and transaction details. ALTDOS uses tactics like ransomware attacks, data exfiltration, and dumping data publicly or for sale on underground forums. The group has been known to demand ransom payments from victims, but also leaks data if demands are not met.	All	

Page 1 of 33, showing 60 records out of 1968 total, starting on record 1, ending on 60

« previous 1 2 3 4 5 6 7 next » last »

Figure 25.3: Viewing a galaxy and its clusters.

The cluster list has its own context tabs — **All**, **Default** (shipped clusters), **Custom** (clusters created on this instance), **My Clusters**, **Deleted**, **View Fork Tree** and **View Galaxy Relationships** — and, for each cluster, an activity sparkline (from sightings) alongside the number of events and relationships it appears in.

25.1.1 Enabling, disabling and updating galaxies

Site admins can **enable** or **disable** a whole galaxy from the index (a disabled galaxy is hidden from the attach picker). The bundled galaxy library is refreshed from the misp-galaxy files on disk with the side-menu actions:

- **Update Galaxies** — reimport all galaxies from the library, adding new default clusters and relationships.
- **Force Update Galaxies** — drop and reimport, useful when a definition changed in place.
- **Wipe Default Galaxy Clusters** — remove all shipped clusters (custom clusters are kept).

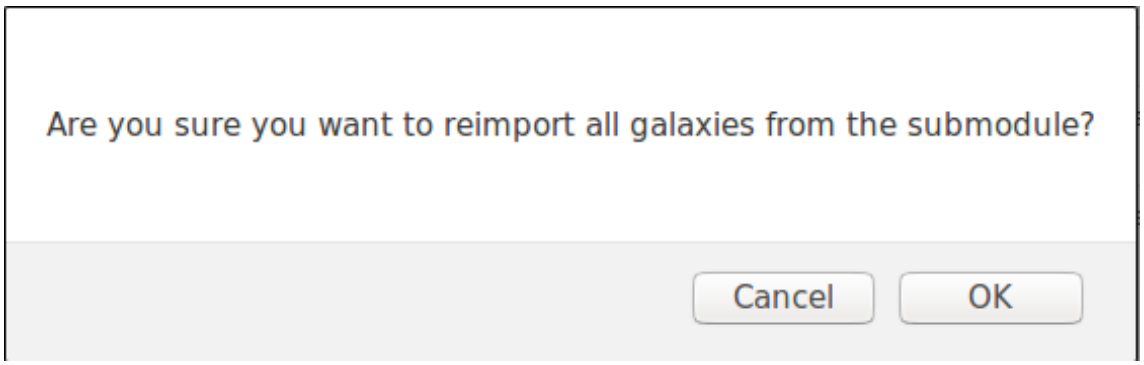


Figure 25.4: Confirming a galaxy update.

The same reimport is available on the command line with `cake Admin updateGalaxies [force]`, which is convenient for a cron job (see the administration chapter). The library files themselves are updated by pulling the misp-galaxy submodule (part of the standard `git submodule` update during a MISP upgrade).

25.1.2 Matrix galaxies

Galaxies that define a `kill_chain_order` — most notably the MITRE ATT&CK galaxies — are rendered as an interactive **matrix**, with the techniques present in an event highlighted as a heat-map across the tactics.

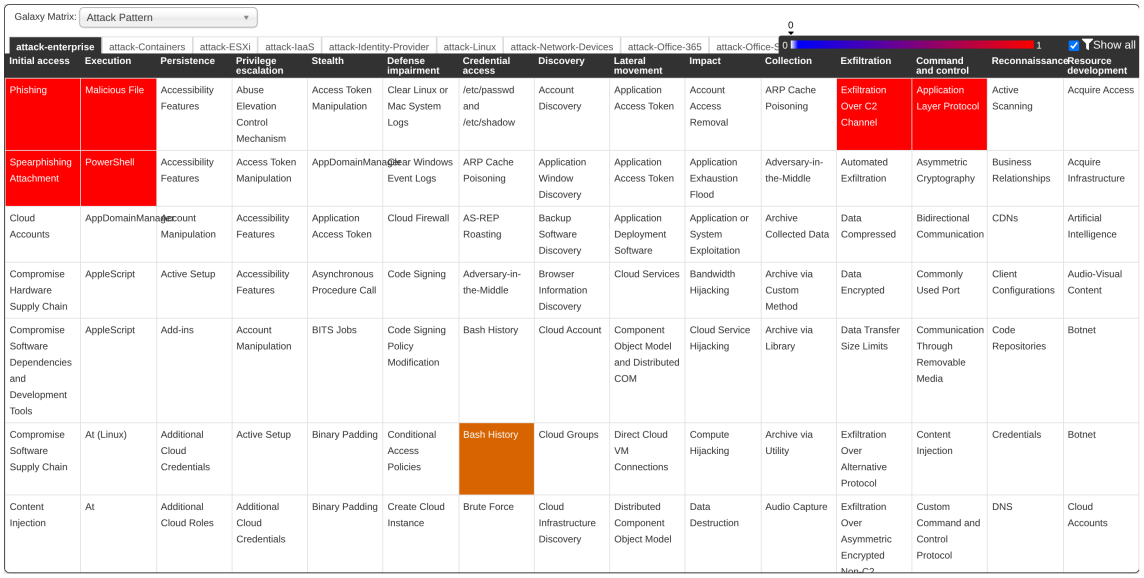


Figure 25.5: A matrix (ATT&CK-style) galaxy.

25.2 Using galaxies in events and attributes

Galaxy clusters are attached from the **Galaxies** panel on an event (or on an individual attribute). The panel offers two add buttons: **Add new cluster** (global — the cluster is shared and synchronised with the event) and **Add new local cluster** (the attachment stays on your instance only). Local attachments are useful for adding your own context to data you intend to share without leaking that context.

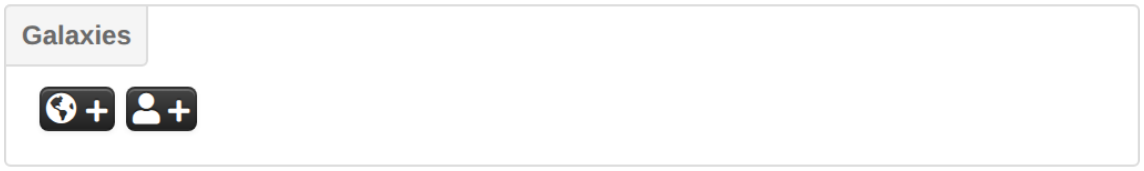


Figure 25.6: An event before any galaxy cluster is attached — note the Galaxies panel and its “Add new cluster” button.

Clicking either button opens a picker where you first choose a galaxy (or search across all of them), then select one or more clusters to attach.

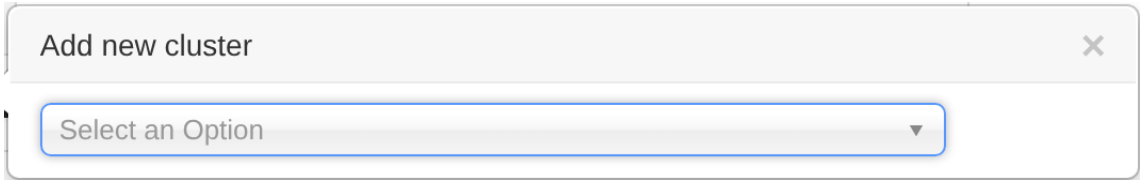


Figure 25.7: The galaxy picker.

Clusters carry synonyms, so if you cannot find an actor or family under the name you know, search for it — an alias will lead you to the canonical cluster. In the example below, searching for “Sneaky Panda” surfaces the cluster under its primary name.

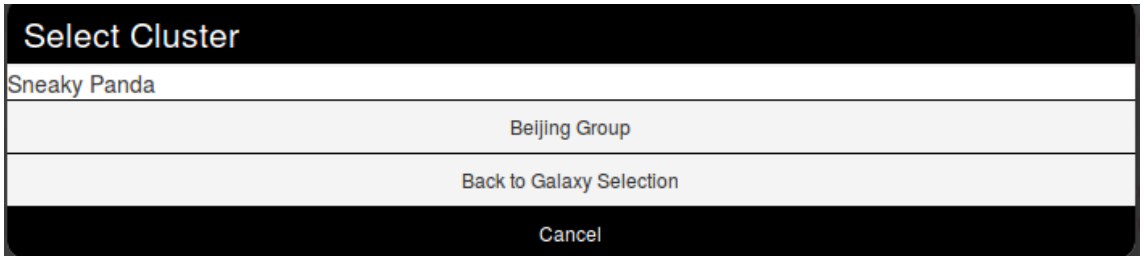


Figure 25.8: Searching the cluster list.

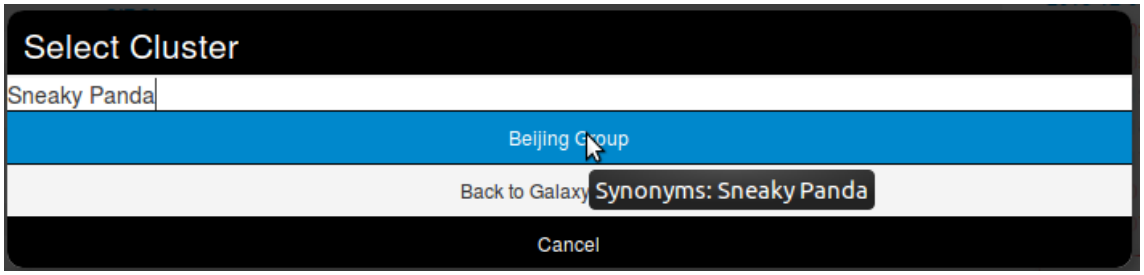


Figure 25.9: The synonym reveals the matching cluster.

Once attached, each cluster in the Galaxies panel offers inline actions: a magnifier to **view the cluster**, an icon to **list the events** that use it, a control to **modify the tag relationship**, and a trash icon to **detach** it. A globe or user icon indicates whether the attachment is global or local.

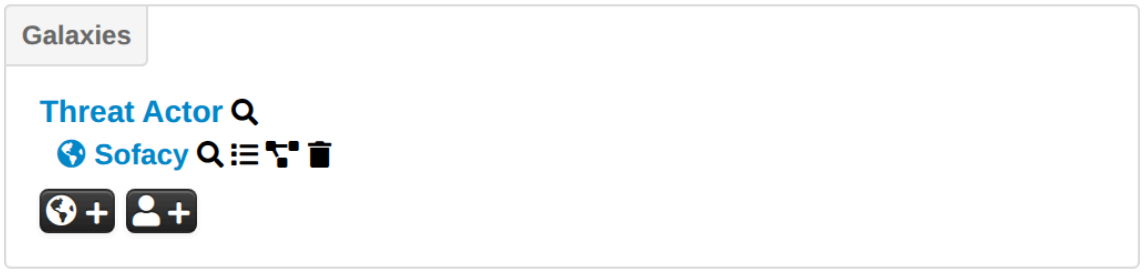


Figure 25.10: A cluster attached to the event.

25.3 Creating your own galaxy clusters (Galaxy 2.0)

Users with the **Galaxy Editor** role permission (or site admins) can create custom clusters directly in the UI. Default (shipped) clusters are read-only; your custom clusters are owned by your organisation and are fully editable.

From a galaxy's view, use **Add Cluster** to open the cluster form:

- **Name** — the cluster's value (for example the actor or family name).
- **Distribution** / **Sharing Group** — who the cluster is shared with, using the standard MISP distribution levels (Your organisation only → All communities, or a sharing group).
- **Description** — free text describing the cluster.
- **Authors** — a JSON array or comma-separated list of authors.
- **Source** — where the information comes from.
- **Galaxy Cluster Elements** — the key/value metadata (synonyms, references, external IDs, and any other fields). You can paste a JSON array of `{key, value}` objects, or use the **Toggle UI** button to fill in the key/value pairs through a form.

You can also create an entirely new **custom galaxy** to hold your clusters with **Add Custom Galaxy** (from the galaxies index side menu) — for example a galaxy specific to your organisation's internal taxonomy of adversaries.

25.3.1 Publishing custom clusters

A custom cluster must be **published** before it can be attached to data or synchronised — an unpublished cluster is hidden from the attach picker, and the cluster view warns you that users cannot yet use it. Publishing requires both the *Galaxy Editor* and *Publish* permissions. Note that editing a cluster (or changing one of its relationships) automatically un-publishes it, so you republish when your changes are ready. Published clusters are pushed to the sync servers you are connected to, according to their distribution level.

25.4 Forking a cluster

Sometimes you want to build on a shipped cluster without losing the link to the original — for example to add your own metadata or relationships to a MITRE technique. **Forking** creates an editable copy of any cluster in the same galaxy: use the **Fork** action on a cluster row (or *Fork Cluster* in the cluster menu).

The fork is a new, independent custom cluster owned by your organisation, with its own UUID, but it records the parent it was forked from. The cluster view shows **Forked From** / **Forked By** links, and when the parent later advances to a newer version you get a **New version available** banner with the option to pull the parent's changes into your fork.

25.5 Cluster relationships

Clusters can be linked to one another to build a graph of related knowledge — for example marking one actor as *is-similar* to another, or a malware family as *used-by* an actor. A relationship connects a **source** cluster to a **target** cluster with a **relationship type** (a free-form string; the picker suggests existing types such as `is-similar`), can carry its own **tags** (for example estimative-language or false-positive tags to qualify the link), and has its own distribution level.

Add relationships from a cluster's **View Galaxy Relationships** panel using the inline **Add relationship** form (the source is pre-filled; pick a type, choose the target cluster, and optionally add tags), or from the dedicated relationship add form. The panel offers both a **tree** and a **tabular** view, with an option to include inbound relations. Because a relationship is part of the cluster's shared data, adding, editing or deleting one un-publishes the source cluster until you republish.

25.6 Blocklisting galaxy clusters

If you want to keep a specific cluster from ever being captured or synchronised onto your instance, add its UUID to the **galaxy cluster blocklist** (List Cluster Blocklists in the galaxies menu; hard-deleting a cluster also adds it to the blocklist). See Blocklists in the administration chapter.

25.7 Authoring galaxies as JSON

The UI editor is the recommended way to build clusters on a running instance. If instead you want to author galaxies as JSON files — for example to contribute them back to the community misp-galaxy repository — a galaxy is made up of two paired files that share the same `type` and `uuid`:

- a **galaxy definition** (`galaxies/<type>.json`) — metadata: name, description, icon, namespace, and (for matrix galaxies) the `kill_chain_order` that lays out the matrix columns; and
- a **cluster collection** (`clusters/<type>.json`) — the actual clusters, each with a value, description, and `meta` key/values (and, for matrix galaxies, a `kill_chain` entry per value that places it in the matrix).

Rather than hand-editing and validating these files, use the **MISP Galaxy Editor**, part of the MISP Engineering Bay toolset. It is a small self-hosted web application that provides a guided editor for both simple and matrix galaxies: a unified galaxy-and-cluster editor, a drag-and-drop matrix builder, autocomplete for the common meta keys, a relationship editor, real-time validation with a live JSON preview, and one-click export of a zip that matches the misp-galaxy repository layout (`galaxies/` + `clusters/`). Once you

have your files, drop them into `app/files/misp-galaxy/` and run **Update Galaxies** (or `cake Admin updateGalaxies force`) to load them.

25.8 Available galaxies

The set of bundled galaxies changes with every misp-galaxy release, so this chapter does not reproduce the list. To see exactly what is available on your instance, browse **Galaxies** in the UI; for the authoritative, always-current catalogue see the misp-galaxy repository and the MISP galaxy page.

26. MISP Objects

A **MISP object** is a structured group of related attributes, built from a reusable **template** — for example a `file` object bundling a filename with its hashes and size, or a `domain-ip` object. Objects sit alongside plain attributes and let you express richer, real-world constructs while keeping each value a first-class, correlatable attribute. The object templates are community-maintained in the misp-objects repository and are designed around real cyber-security use-cases and sharing practices.

26.1 Using objects

To add an object to an event, open the event and choose **Add Object** from the left-hand menu:

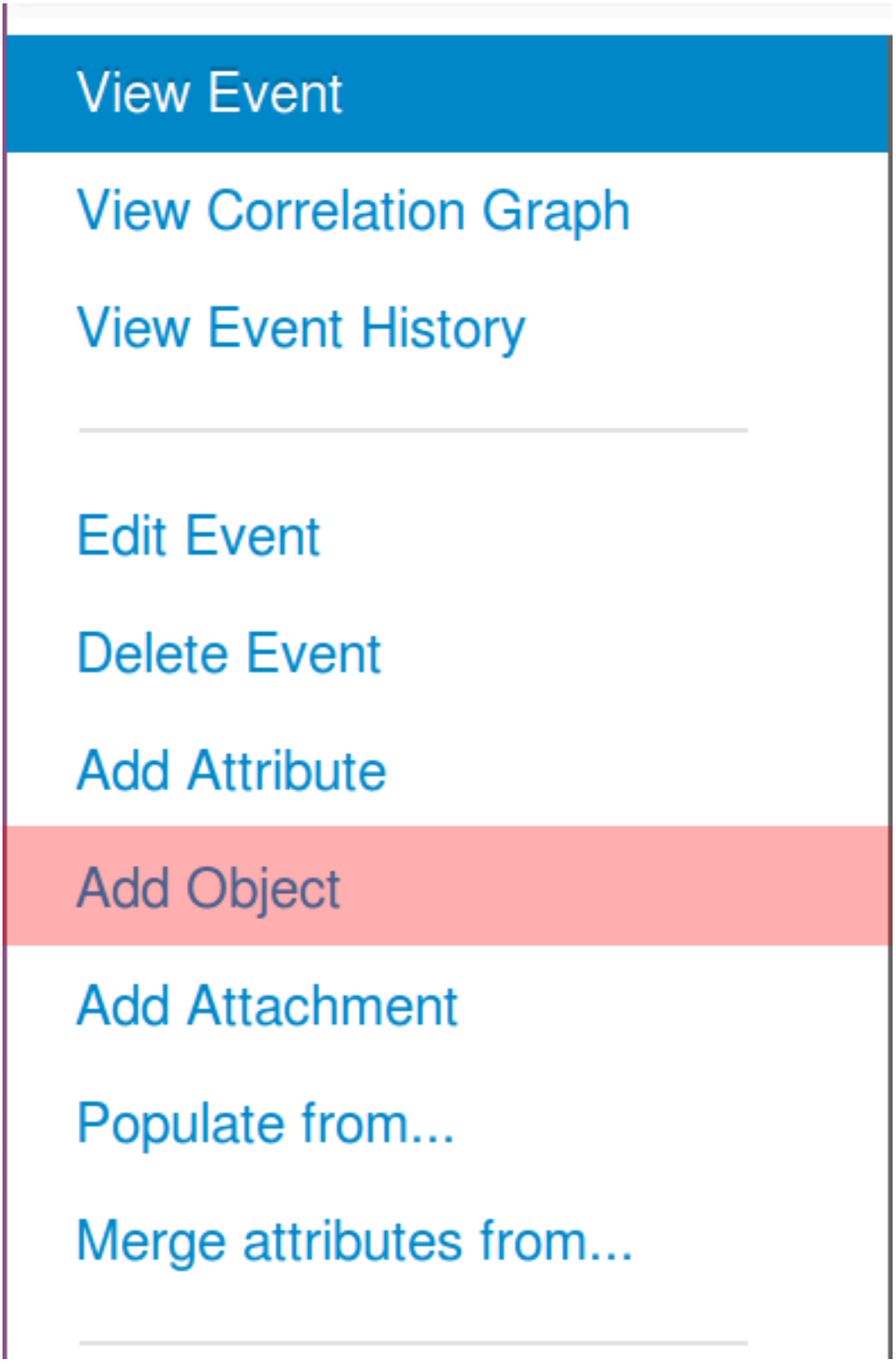


Figure 26.1: The event side menu, with Add Object highlighted.

This opens a picker that lists every available object template in a single searchable dropdown. Start typing to filter the list by name — each entry shows the template's **meta-category** beneath its name, and its info icon reveals a short description of what the object is for:

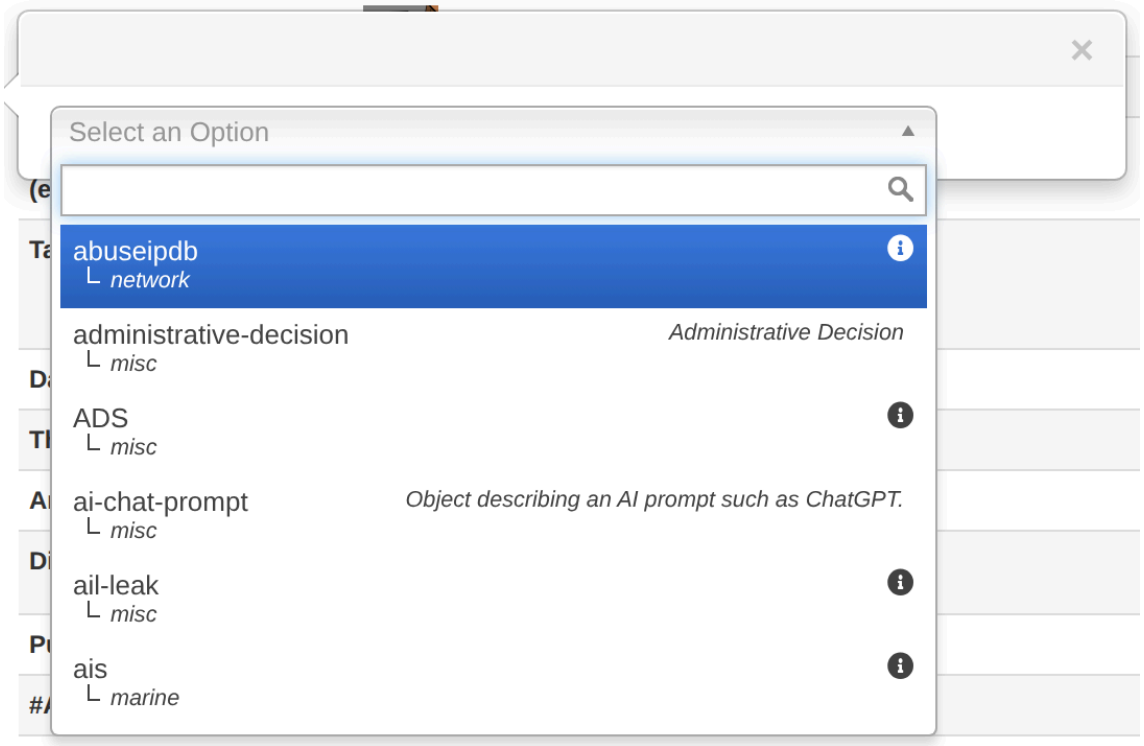


Figure 26.2: Choosing a template from the searchable object picker.

For this example we will add an `ip-port` object. Selecting a template opens its form, which lists every attribute the template defines:

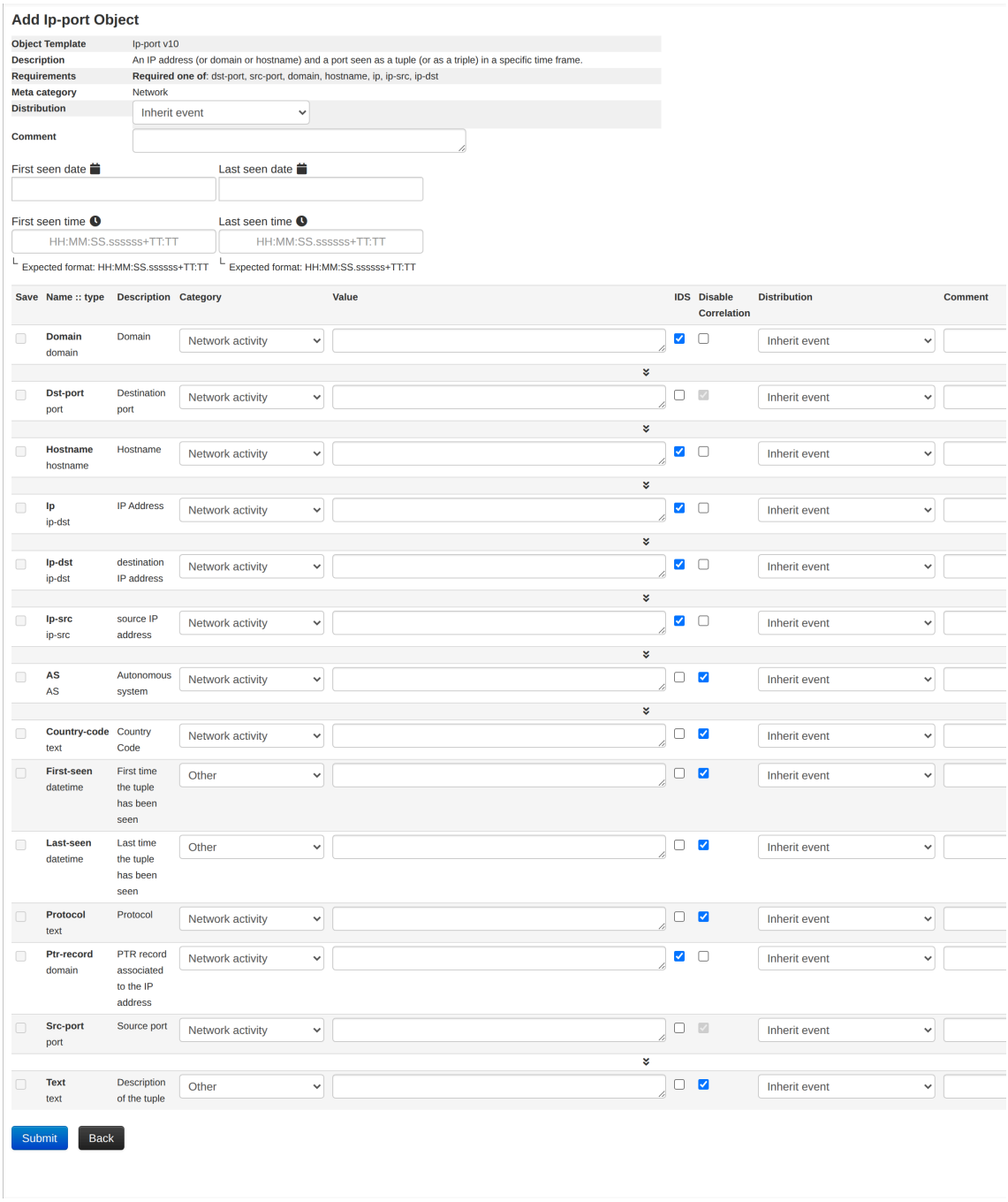


Figure 26.3: The ip-port object form.

Each template sets its own **requirements**, shown at the top of the form. The `ip-port` template requires *one of* `dst-port`, `src-port`, `domain`, `hostname`, `ip`, `ip-src` or `ip-dst` to be set — you do not have to fill in every field, but an object with no attributes at all, or one that does not satisfy its template's requirements, is invalid and will not be added to the event.

After pressing **Submit** you are shown a pre-save review, so you can confirm the object reflects what you intended before it is created:



Figure 26.4: The object pre-save review.

Objects can also be created for you by the import tools (freetext and the various "Populate from…" formats), which build the appropriate objects from the data you paste in.

26.2 Object references and relationships

The real power of objects comes from linking them together. An object can hold **references** to other objects — or to individual attributes — each labelled with a **relationship type** that says *how* they are related (for example a `network-connection` object *`connects-to`* an `ip-port` object, or a `file` object was *`downloaded-from`* a `url`). Together these references form the **event graph** you can explore from the event view.

To add one, use the **Add reference** action on an object and pick the target object or attribute plus a relationship type. MISP offers a standard set of relationship types (maintained in the misp-objects repository), or you can enter a custom one. References are directional (source → target) and are shown both on the objects themselves and in the event graph.

26.3 Editing objects

An object's attributes are edited just like any other attribute — inline in the event view, or through the object's edit form. As with attributes, changing an object un-publishes the event, so you republish once you are done.

26.4 Object templates

Every object is created from an **object template** that defines which attributes it can contain, which are required, and how they are presented. The template library ships with MISP and is kept up to date from the misp-objects repository — refresh it with **Update Objects** in the administration menu, or on the command line with `cake Admin updateObjectTemplates` (a good candidate for a cron job).

26.4.1 Creating object templates

An object template is a JSON file that follows the object schema.

An object is basically a combination of two or more attributes that can be used together to represent real cyber security use-cases. These attributes are listed in a JSON object.

Each attribute is an JSON object defined by a name, a description, a misp-attribute and an ui-priority value. - Name and description are self-explanatory. - misp-attribute is an existing type of attribute in misp that matches the attribute. - Concerning ui-priority, the higher the number is, the most it is expected to be seen.

There are also others options that can be added to define an attribute more precisely. - sane_default is a list of default valid value for this attribute. The user can pick a value from this list or choose "Enter value manually" - disable_correlation will disable correlation for this value. Useful for dates for instance - recommended value for this field - multiple, if set to true, allow the user to add multiple instances of this attribute.

Not all attributes are mandatory, but some can be required. If so, they need to be listed in a list called "required". The object will only be valid if the listed attributes are set. The same way, there are sometimes when only one attribute in a set is needed. This set can be put in a list called "requiredOneOf". If at least one of the attributes in this list is set, the object will be valid.

Rather than hand-editing and validating these JSON files, you can use the **Object Template Creator**, part of the MISP Engineering Bay toolset. It is a small self-hosted web application that provides a guided editor for object templates — a searchable attribute-type picker driven by MISP's `describeTypes`, real-time validation, and import/export of `definition.json` files — and can load any of the existing templates as a starting point. Once your template is ready, add it to `app/files/misp-objects/` and run **Update Objects** (or `cake Admin updateObjectTemplates`) to load it.

27. Warning Lists

MISP warning lists contain well-known indicators that may correspond to false positives, errors, or mistakes. There is also a Python module for working with warning lists in a Pythonic way: PyMISPWarningLists. MISP Warning Lists GitHub Repository

27.1 Warning Lists and False Positives

False positives are a common issue in threat intelligence sharing. They are often contextual: - What counts as a false positive may differ from one sharing community to another. - Organizations may also have their own view of what should be treated as a false positive.

27.2 Usage

By default, MISP triggers warning list hits only when the attribute IDS flag is set. You can change this behavior by setting the `MISP.warning_for_all` configuration parameter to `true`.

When an attribute matches a warninglist entry, an info/warning box is displayed at the event and attribute level, as can be seen in the screenshot below.

Date ↑	Context	Category	Type	Value	Tags	Galaxies	Comment	Correlate	Related Events	Feed hits	IDS	Distribution	Sightings	Activity	Actions
2026-07-08*	5a3...a42	Network activity	ip-dst	192.168.0.1			RFC 1918 private address (used to demo warninglist hits)	☑		72	☑	Inherit	(0/0/0)		
2026-07-08*	357...ae7	Network activity	ip-src	192.168.0.53			RFC 1918 private address	☑			☑	Inherit	(0/0/0)		

Individual warning lists can be enabled or disabled at the instance level from the warning lists index page. Examples of default warning lists include known public DNS resolvers, multicast IP addresses, hashes for empty values, RFC1918 ranges, TLDs, and known Google domains.

ID ↓	Name	Event ID translator		Category	Type	Entries	Default	Enabled	Actions
3	List of known Akamai IP ranges	20240422	Akamai IP ranges from BGP search	False positive	cidr	536	✓	×	
4	Top 1000 website from Alexa	20230511	Event contains one or more entries from the top 1000 of the most used website (Alexa).	False positive	hostname	1000	✓	×	
5	List of known Amazon AWS IP address ranges	20251223	Amazon AWS IP address ranges (https://ip-ranges.amazonaws.com/ip-ranges.json)	False positive	cidr	3314	✓	×	
6	List of known domains used by automated malware analysis services & security vendors	5	Domains used by automated malware analysis services & security vendors	False positive	substring	26	✓	×	
7	List of known bank domains	8	Event contains one or more entries of known banking website	False positive	hostname	2153	✓	×	
8	Top 1000 websites from Cisco Umbrella	20251223	Event contains one or more entries from the top 1000 of the most used websites (Cisco Umbrella).	False positive	string	1000	✓	×	
9	Top 10 000 websites from Cisco Umbrella	20251223	Event contains one or more entries from the top 10 000 of the most used websites (Cisco Umbrella).	False positive	string	10000	✓	×	
10	Top 20 000 websites from Cisco Umbrella	20251223	Event contains one or more entries from the top 20 000 of the most used websites (Cisco Umbrella).	False positive	string	20000	✓	×	
11	Top 5000 websites from Cisco Umbrella	20251223	Event contains one or more entries from the top 5000 of the most used websites (Cisco Umbrella).	False positive	string	5000	✓	×	
12	List of known Cloudflare IP ranges	20251223	List of known Cloudflare IP ranges (https://www.cloudflare.com/ips/)	False positive	cidr	22	✓	×	
13	Common contact e-mail addresses	20200226	A list of commonly used abuse and contact e-mail addresses, including the ones denoted in RFC2142.	False positive	regex	1	✓	×	
14	List of known hashes with common false-positives (based on Florian Roth input list)	2	Event contains one or more entries with common false-positives	False positive	string	69	✓	×	

Warning lists can be extended locally in JSON or contributed through pull requests to https://github.com/MISP/misp-warninglists. They can also be used for critical infrastructure, core infrastructure, or personally identifiable information.

27.2.1 Warninglists and data export

The enforceWarninglist parameter of MISP restSearch can be used to exclude attributes that have a warninglist hit from the export. For more information on the MISP API, please refer to the Automation and MISP API chapter.

27.2.2 Check individual values for warning list hits

It is also possible to do a lookup for a specific value in the warninglists. This functionality is accessible by using the top menu "Input Filters" > "List Warninglists" and then using the link in the left side menu bar (or by browsing directly to [misp_base_url]/warninglists/checkValue). Only enabled warninglists will be searched.

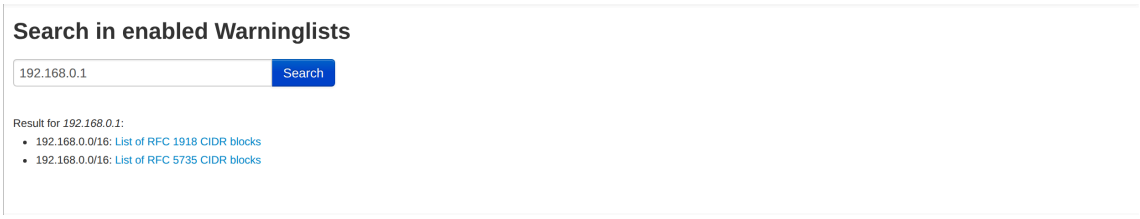


Figure 27.1: Screenshot of example search in warninglists using check value, for value 192.168.0.1. The result shows two hits, one for RFC 5735 CIDR blocka nd one for RFC 1918 CIDR blocks

27.2.3 Updating warninglists

An update of the warninglists can be triggered via the GUI using the "Update Warninglists" button in the side menu bar when viewing any of the relevant warninglists pages, for example the index page.

Alternatively, it is also possible to trigger an update using a CLI command. ~ MISP/app/Console/cake Admin updateWarningLists ~

If you are updating an existing warning list, make sure you increment the version number before triggering the update in MISP. You can also contribute to the existing warning lists by forking the MISP Warning Lists GitHub Repository, making your changes, and then opening a pull request.

27.2.4 How matching works: types and categories

Each warninglist declares a **type** that controls how its entries are matched against attribute values:

Type	Matches when the value…
`string`	equals an entry exactly
`substring`	contains an entry as a substring
`hostname`	is (or is a subdomain of) an entry hostname
`cidr`	is an IP address inside an entry CIDR range
`regex`	matches an entry regular expression

Each list also has a **category** — either `false_positive` (values likely to be false positives, such as RFC1918 ranges or public DNS resolvers) or `known` (well-known values, such as your own infrastructure) — which is shown alongside the hit so you understand *why* a value was flagged.

27.2.5 Managing warninglists from the GUI

As well as the file-based method below, MISP lets site administrators create and manage warninglists directly in the web interface from *Input Filters → List Warninglists*:

- **New Warninglist** — create a list by giving it a name and description, choosing its **type** and **category**, and pasting its entries (one per line). This is the quickest way to add, for example, a list of your organisation's own domains or IP ranges.
- **Edit** — update an existing GUI-created list's metadata or entries.
- **Delete** — remove a list you no longer need (individually or via the bulk selection).
- **Enable / disable in bulk** — the index supports mass enable/disable across selected lists.

> **Note**
>
> Lists that ship with MISP (from the `misp-warninglists` repository) are maintained upstream and refreshed by the update mechanism; use the GUI editor mainly for your own local lists so your changes are not overwritten on the next update.

27.2.6 Creating a custom warninglist on disk

You can also add a warninglist as a file on the server:

1. Create a new directory for your warninglist in /var/www/MISP/app/files/warninglists/lists.
2. Add a file called list.json to the newly created directory and add the content you want. You can use any of the existing warning lists in https://github.com/MISP/misp-warninglists as reference.
3. Trigger an update of the warning lists on the instance to load in your new warning list.

Example use cases include a list of domain names owned by your organization or a list of employee email addresses.

28. Notice Lists

Notice lists inform MISP users about the legal, privacy, policy or technical implications of using specific attributes, categories or objects. They are a way for a community to attach guidance to particular kinds of data — for example reminding users of a handling requirement when they add an attribute that matches a sensitive pattern.

28.1 How notices are triggered

Each notice list is a simple JSON description that pairs a set of triggers (by attribute type, category, object, or value pattern) with an informational message. When you enter or edit data that matches an active notice list — for example while adding an attribute — MISP shows the corresponding notice inline on the form, so you are made aware of the implication before you save. Notices are purely informational: they warn and inform, they do not block the action.

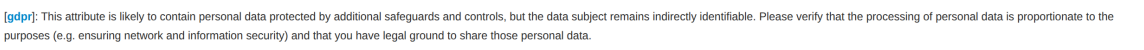

[gdpr]: This attribute is likely to contain personal data protected by additional safeguards and controls, but the data subject remains indirectly identifiable. Please verify that the processing of personal data is proportionate to the purposes (e.g. ensuring network and information security) and that you have legal ground to share those personal data.

Figure 28.1: The GDPR notice list message displayed inline while adding a personal-data attribute.

28.2 Managing notice lists

Notice lists are managed from the administration menu:

- **List Noticelists** shows every notice list on the instance, whether it is enabled, and its contents. Only enabled notice lists produce notices.
- **Enable / disable** a notice list to control whether it is active.
- **Update Noticelists** re-imports the bundled notice lists from the misp-noticelist library. The same update is available on the command line with `cake Admin updateNoticeLists` (handy for a cron job — see the administration chapter).



ID ↑	Name	Expanded Name	Ref	Geographical area	Version	Enabled	Actions
1	gdpr	General Data Protection Regulation	http://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX:32016R0679	EU	1	☑	👁

Figure 28.2: The notice lists index, showing the bundled GDPR notice list.

The set of notice lists is maintained by the community in the MISP notice list repository; you can contribute new ones there, or add your own locally.

IV Sharing & Synchronisation

29 Synchronization and Sharing 213

30 Sharing / Synchronisation 215

31 Synchronisation 217
31.1 Concept 217
31.2 Pull vs Push synchronisation 218
31.3 Adding a server 219
31.4 Test connection 221
31.5 Rules 221
31.6 Troubleshooting 222
31.7 Sharing and distribution 222
31.8 Collaboration 226
31.9 Cerebrate 230
31.10 Sharing group blueprints 230
31.11 TAXII 2.x servers 231
31.12 Recommendation 231

32 Managing Feeds 233

33 Feeds 235
33.1 Managing feeds 235
33.2 Feed correlation 245

34 Collections 247
34.1 What a collection contains 247
34.2 Creating and populating a collection 247
34.3 Distribution and sharing 248
34.4 Synchronising collections between instances . 248
34.5 API 248

35 Cryptographic Keys & Event Signing 251
35.1 How it works 251
35.2 Protecting an event 251
35.3 Verification on synchronisation 252
35.4 Permissions and the server-signing endpoint . 252
35.5 API 252

29. Synchronization and Sharing

30. Sharing / Synchronisation

- MISP's core functionality is sharing where everyone can be a consumer and/or a contributor/producer.
- Quick benefit without the obligation to contribute
- Low barrier access to get acquainted to the system

31. Synchronisation

31.1 Concept

The following figure shows the concept how different MISP instances could tie together.

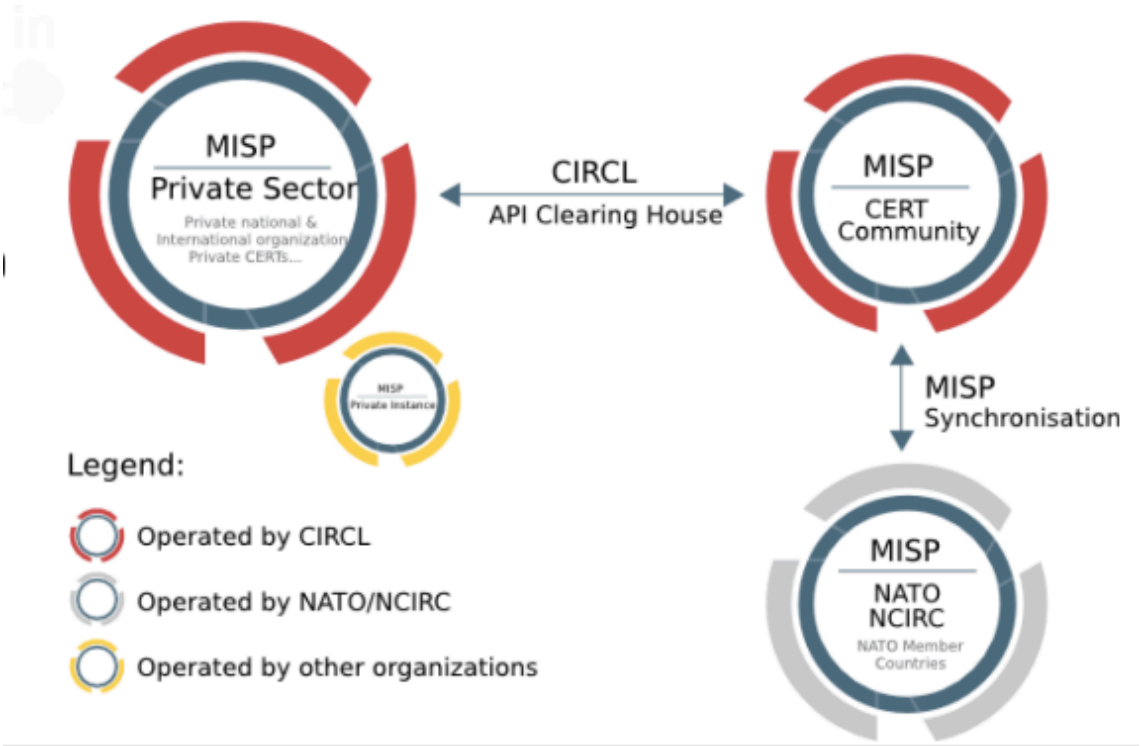


Figure 31.1: Scenario example

In MISP, two ways exist to get events from remote sources:

- **Use case 1**: From another MISP server (also called MISP instance), by synchronising two MISP servers.
- **Use case 2**: From a link, by using Feeds.

The example below illustrate the synchronisation between two MISP servers (use case 1). An organisation B (OrgB) wants to synchronise its MISP server, called ServerB, with the MISP server of an organisation A (Org A), called ServerA. The following steps can be taken to syncronise ServerB with ServerA:

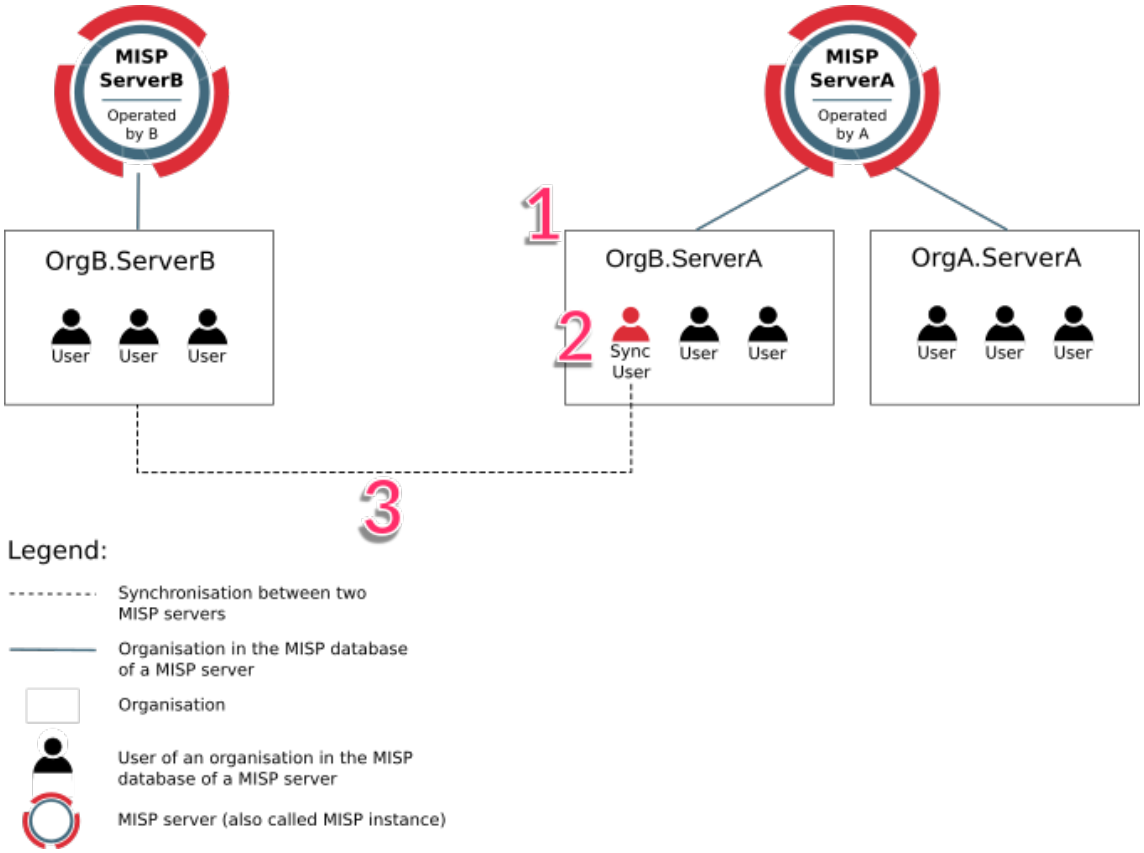


FIGURE: Illustration of the synchronisation between two MISP servers

- **Step 1**: Add OrgB as a local organisation on ServerA (OrgB.ServerA) using OrgB's existing UUID from their local organisation on ServerB.
- **Step 2**: Add a Sync User (syncuser@OrgB.ServerA) in the organisation OrgB.ServerA on the MISP ServerA.
- **Step 3**: Set up a sync server on MISP ServerB using the key (called Authkey) from the sync user (syncuser@OrgB.ServerA) created on MISP ServerA.

For additional information on the synchronisation process, refer to the MISP GitHub issues, for example, issue 2595.

31.2 Pull vs Push synchronisation

Synchronisation can be set up using the pull or push option. A combination of the two is possible as well. This section discusses the main practical differences between the two and the benefit of using both in conjunction.

To automatically execute pull and push synchronisations on a schedule, please use a scheduling service such as cron to execute the relevant command line functions mentioned in the related administration section.

For a table overview of which community distributions lead to a sync on push or pull scenarios as well as how the distribution level automatically decreases on sync, please refer to the syncing scenarios with communities distribution section.

31.2.1 Push

The push mechanism provides a seemingly real time experience. If a push connection is enabled, data created (and synchronized from other peers) will be synchronized as fast as possible. In practice, this means that an attempt to synchronise is made immediately after publication of an event.

A disadvantage of the push mechanism is that connection issues or dead workers might prevent the push from being successful.

Data with distribution set to 'Your organisation only' or 'This community only' will not be synchronised using the push mechanism (unless the internal instance flag is set for the server to be synced to).

31.2.2 Pull

In contrast to a push, pull synchronisation is only performed on command. This could be a manual trigger by an admin from the servers page or a cron job executing a pull command. See administration for the available command line interface commands.

A pull will also fetch objects with a distribution set to 'this community only' and even your 'organisation only', if the auth key belongs to the organisation owning the related event.

31.2.3 Benefits of using push and pull in conjunction

- If an unexpected issue prevented the push from happening, the data can be pulled in later
- Different rules can be set for the push and pull sync

31.3 Adding a server

Servers can be added by users via

```
https://<misp url>/servers/add
```

The Add Server form is grouped into sections — *Instance identification*, *Instance ownership and credentials*, *Enabled synchronisation methods* and *Misc settings* — and has several input fields:

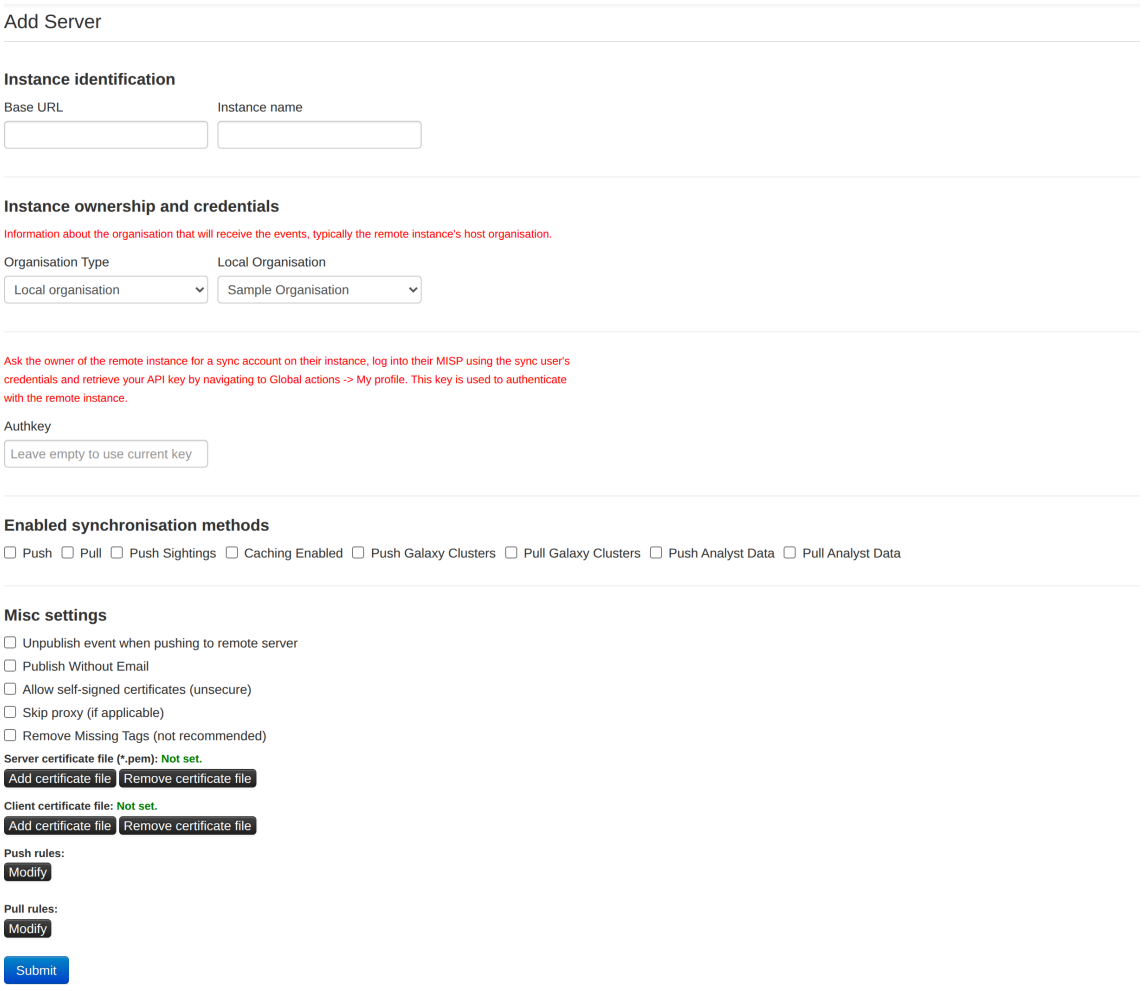


Figure 31.2: Add Server

1. **Base URL** The base-url to the external server you want to sync with. Example: https://foo.sig.mil.be
2. **Instance Name** A name that will make it clear to your users what this instance is. For example: Organisation A's instance
3. **Internal instance** This checkbox will only show up if the set 'local organisation', see 5, is the same as the owning organisation of the instance the remote server is being configured on. If this flag is set, the distribution of pushed data will not be automatically downgraded. Please refer to the syncing scenarios with communities distribution section for a table overview of the differences.
4. **Remote Sync Organisation Type** MISP has several organisation "pools", one for local and one for known external organisations. When adding a synchronisation connection, you need to define the host organisation of the remote instance. Select which pool you wish to pick the organisation from using this drop-down. You also have the option of adding a new external organisation directly from this interface.
5. **Local/Known remote Organisation** Choose the organisation from the selected pool that defines the host organisation on the remote side. Make sure that the remote instance is actually run by the organisation you select. When selecting data to push, this organisation will be used to determine membership of sharing groups. As a result, this setting is very important, since selecting the wrong organisation can lead to leaking confidential data (oversharing) or sharing less than intended.
6. **Authkey** You can find the authentication key on your profile on the external server. If advanced auth keys is enabled you might have to create an auth key manually. This can be done on your profile view as well. Please refer to the automation section for more information about creating an automation key.
7. **Push** Allow the upload of events and their attributes. Only Events that match the given push rules (see 19) will be pushed to the server. Sightings and relevant galaxy clusters will not be pushed unless 'Push Sightings' and 'Push Galaxy Clusters' are enabled as well.
8. **Pull** Allow the download of events and their attributes from the server. Only data matching the given pull rules (see 20) will be pulled. Relevant galaxy clusters will not be pulled in unless the 'Pull Galaxy Clusters' checkbox is ticked as well.
9. **Push sightings** If checked, sightings will be pushed out as well on push.
10. **Caching Enabled** Allow caching of the remote server. Once cached, caching related functionalities such as server overlap analysis can be done.
11. **Push Galaxy Clusters** If checked, relevant galaxy clusters will be pushed out as well on push.
12. **Pull Galaxy Clusters** If checked, relevant galaxy clusters will be pulled in as well on push. This also enables you to do a pull for galaxy clusters only if pull, see 8, is checked as well.
13. **Unpublish Event** Unpublish the synced event. This only works on push.
14. **Publish Without Email** Publish the event without sending out an email.
15. **Self Signed** Check this if you would like to allow a connection despite the other instance using a self-signed certificate (not recommended).
16. **Skip proxy (if applicable)** Do not connect to this server using the configured proxy (if any proxy host is set in the MISP configuration).
17. **Server certificate file** You can upload a certificate file if the instance you are trying to connect to has its own signing authority. (*.pem)
18. **Client certificate file** Set a client certificate to use when connecting to this server.
19. **Push rules** Allows you to set filtering rules for data to be pushed out (tag based and organisation based). Please see rules for more info.
20. **Pull rules** Allows you to set filtering rules for data to be pulled in (tag based and organisation based). Please see rules for more info.

31.3.1 Synchronising analyst data and collections (2.5)

MISP 2.5 adds four more synchronisation toggles to the server form, alongside the galaxy-cluster options, so that the newer collaborative data types can travel between instances:

- **Push analyst data** / **Pull analyst data** — synchronise Analyst Data (notes, opinions and relationships) with the remote instance.
- **Push collections** / **Pull collections** — synchronise Collections (bundles of events and galaxy clusters) with the remote instance.

Like the other sync options, each direction is independent, both instances must support the feature, and the usual distribution-degradation rules apply.

Adding, editing and removing sync servers is restricted to site administrators.

31.4 Test connection

The connection test can be used to test the connection to the remote server and will give a feedback about local and remote version of MISP.



***

31.5 Rules

Rules are used to limit sharing when synchronising events and attributes. In the example below, data tagged with tlp:red or owned by organisation Setec Astronomy will not be pushed out to this server. Please note any other filtering set up on the instance will still apply as well (blocklists for example).

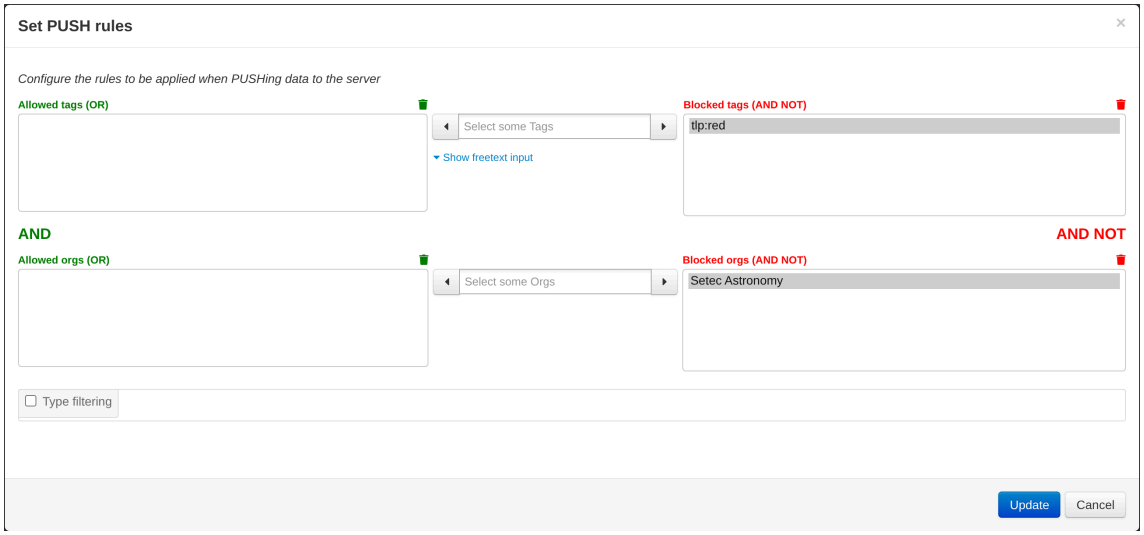


Figure 31.3: Screenshot of configured push rules, tlp:red tag is added to blocked tags and organisation Setec Astronomy to Blocked Orgs

31.6 Troubleshooting

If you have issues connecting to a remote servers try to do the following things: - manually try to do an API request, to ensure that your API key is valid - try to connect with your user account to the remote server and check your roles on the remote server - perform a package capture and analyse the traffic - if you have a SSL connection issue to a remote server with a certificate signed by a CA that is not included in OS, make sure the whole certificate path is included in the path.

31.7 Sharing and distribution

The following section describes how distribution mechanisms of events and attributes work.

31.7.1 Distribution settings

The below five distribution settings are available for events and attributes. Descriptions of those settings can be found here.

- Your organisation only
- This community only
- Connected communities
- All communities
- Sharing group

Events that are not published are only distributed/shared to the local organisations on the same MISP server/instance (within the limit of the distribution model). Only events that are **published** will be shared with remote organisations on other MISP servers via push/pull mechanisms. More details on publishing events here.

31.7.1.1 Visibility of data - events objects and attributes

Assuming that none of the ancestors of the object of attribute are too restrictive, you can view an event/object/attribute on an instance if: - You are a site admin - Your organisation is owner of the event - Neither of the above apply and one of the below statements is true: - The distribution of the event/object/attribute is 'This community only', 'Connected communities' or 'All communities' - Your organisation belongs to the sharing group of the event/object/attribute or the sharing group has the 'all orgs' flag set for the instance

Note: - As of MISP version 2.4.140: When pulling in data you can only see due to the 'all orgs' flag being set in the sharing group, your organisation will be added to the sharing group when creating the data locally.

31.7.2 Community

A community is composed of the local organisations on a MISP server and the remote organisations connected by the sync users. For more information on the concept of community, refer to an article on MISP information sharing following ISO/IEC 27010, explaining the concept of community.

Specifically, communities are not reversible. Taking the example of the above figure, illustrating the synchronisation between two MISP servers, OrgB.ServerB is part of the MISP ServerA community but OrgB.ServerA is not part of MISP ServerB community.

31.7.3 Sharing-groups

There is an article about sharing groups in here

31.7.4 Distribution mechanisms

The distribution level of an event is automatically decreased as it is synchronised with other MISP instances, when it was originally set to:

- Community only (to organisation only)
- Connected community (to community only)

It is not decreased when it was originally set to:

- Organisation only
- All communities
- Sharing group

[!] This rule does not apply if “Internal instance” has been checked when creating the server.

As an example, the figure below illustrates two events **e** and **e’** created by OrgA and respectively shared as “This community only” and “Connected communities” and how they propagate in an illustrative MISP set of intances synchronised with each others.

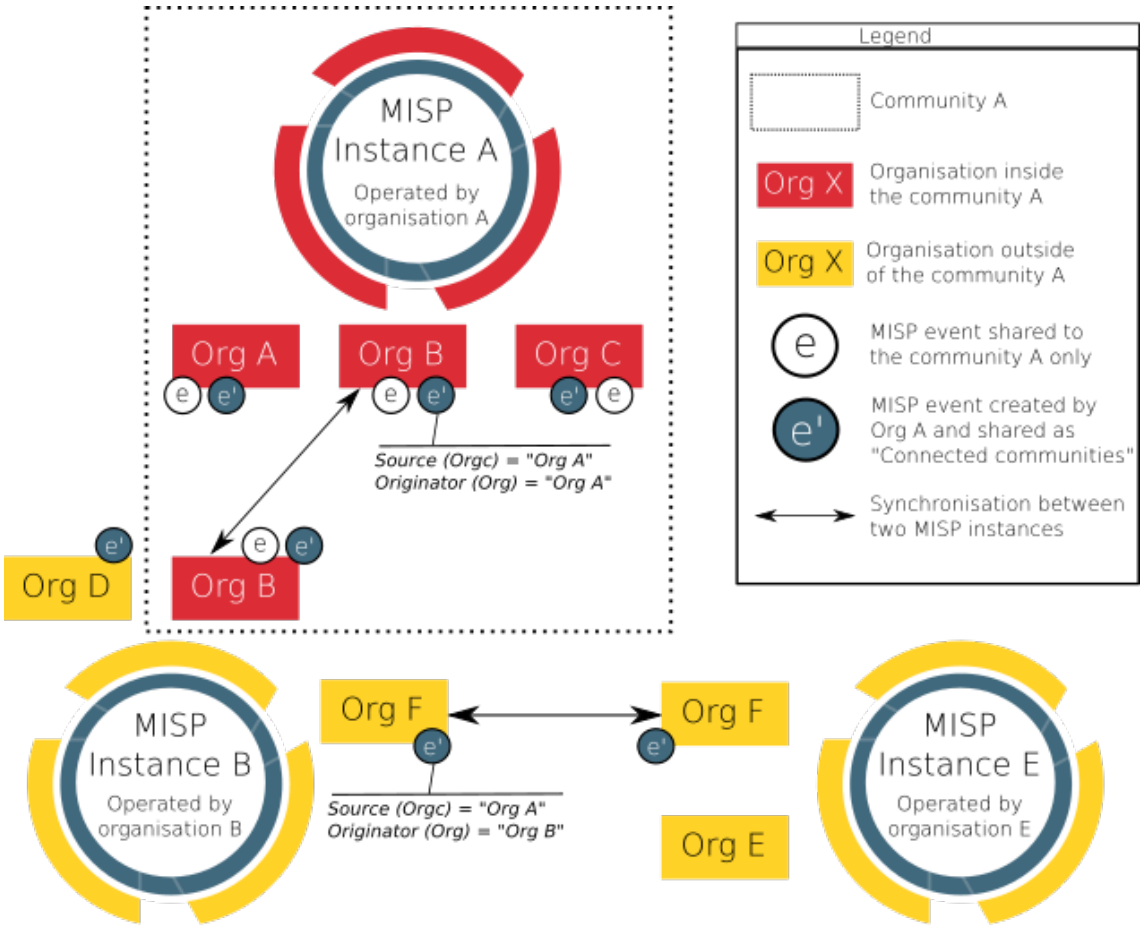


Figure 31.4: Illustration of MISP organisations and community interactions

31.7.4.1 General syncing rules

- The owner organisation of the event on instance B is set to the organisation of the sync user.
- The creator user is the authkey user when pushing
- The creator user is the user triggering the pull when pulling. This user can be different than the authkey user.
- Rule of thumb: if the user configured to pull from instance A to B can see the event on instance A, the event will be synced.

31.7.4.2 Syncing scenarios with communities distribution

Internal instance flag not set

The below scenarios are if "Internal instance" has not been checked when creating the server. This is the usual scenario.

Push from instance A to instance B - usual scenario

Which organisation the remote sync user belongs to has no impact on which events are pushed.

Instance A	Instance B
Your organisation only	Event/object/attribute not pushed

Instance A	Instance B
This community only	Event/object/attribute not pushed
Connected communities	Event/object/attribute distribution decreased to 'This community only' on B
All communities	Event/object/attribute distribution stays 'all communities'

Pulling from instance A to instance B - usual scenario

Rule of thumb: if the user configured to pull from instance A can see the event on instance A, the event will be synced.

Instance A	Instance B
Your organisation only	Event/object/attribute pulled in only if the sync user is member of the event's owner organisation on A. Event distribution stays 'Your organisation only' on instance B
This community only	Event/object/attribute distribution decreased to 'Your organisation only' on B
Connected communities	Event/object/attribute distribution decreased to 'This community only' on B
All communities	Event/object/attribute distribution stays all communities on B

Internal instance flag set

The below scenarios are if "Internal instance" has been checked when creating the server. This is the *not* the usual scenario and *potentially dangerous*. The internal instance flag can be used when both instances have the same hosting organisation.

Push from instance A to instance B - internal flag set scenario

Instance A	Instance B
Your organisation only	Event/object/attribute not pushed if triggering push of already locally (on instance A) published event. Event/object/attribute synced on publication of an event, even if the organisation publishing is not the host organisation of the instance

Instance A	Instance B
This community only	Event/object/attribute distribution stays ‘This community only’ on B
Connected communities	Event/object/attribute distribution stays ‘Connected communities’ on B
All communities	Event/object/attribute distribution stays ‘All communities on B’

Pulling from instance A to instance B - internal flag set scenario

Rule of thumb: if the user configured to pull from instance A can see the event on instance A, the event will be synced.

Instance A	Instance B
Your organisation only	Event/object/attribute pulled in only if the sync user is member of the event’s owner organisation. Event distribution stays ‘Your organisation only’ on B
This community only	Event/object/attribute distribution decreased to ‘Your organisation only’ on B
Connected communities	Event/object/attribute distribution decreased to ‘This community only’ on B
All communities	Event/object/attribute distribution stays ‘All communities’ on B

31.8 Collaboration

31.8.1 Proposals

Proposals can be used to propose new attribute values that can be reviewed by the event owner.

31.8.2 Forums / Threats

Forums can be used to discuss non event related topics.
Discussions can be accessed on the top “Global Actions - List Discussions”
Discussions will and can not be shared with other servers
and via URL:

```
https://<misp url>/threads/index
```

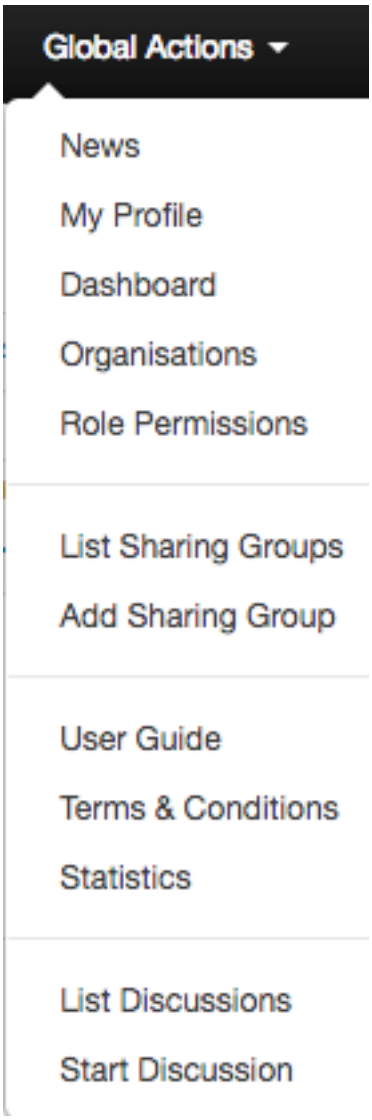


Figure 31.5: Discussions

31.8.2.1 Create a new Topic

To create a new topic

```
https://<misp url>/posts/add
```

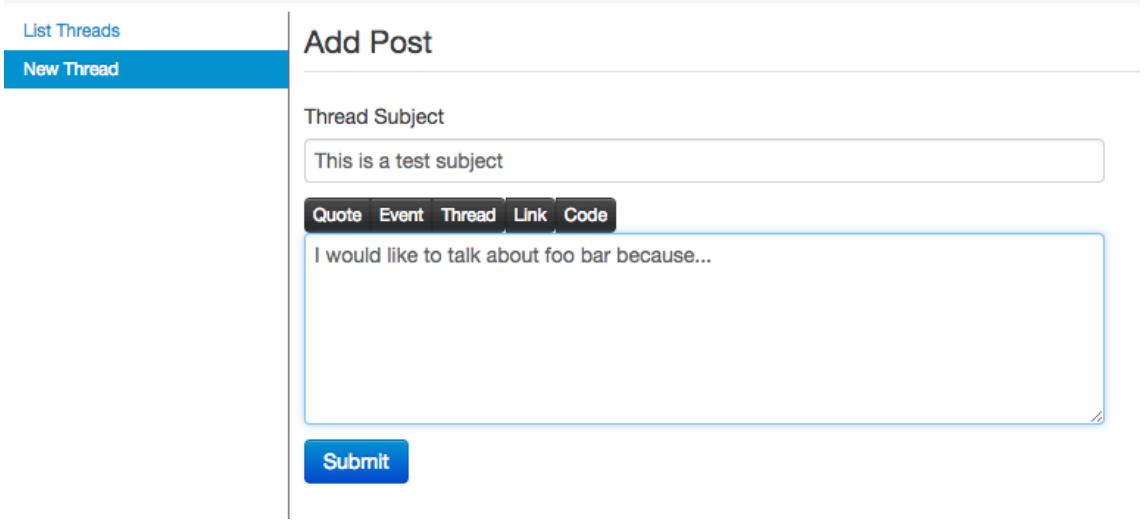


Figure 31.6: Start a topic

31.8.2.2 Comment a topic

A topic can be commented by any user

```
https://<misp url>/threads/view/<topic id>
```

31.8.3 Comments to events

In MISP ongoing events can be commented by every user to ask free text question to events. **Comments to events will not be shared with other servers**

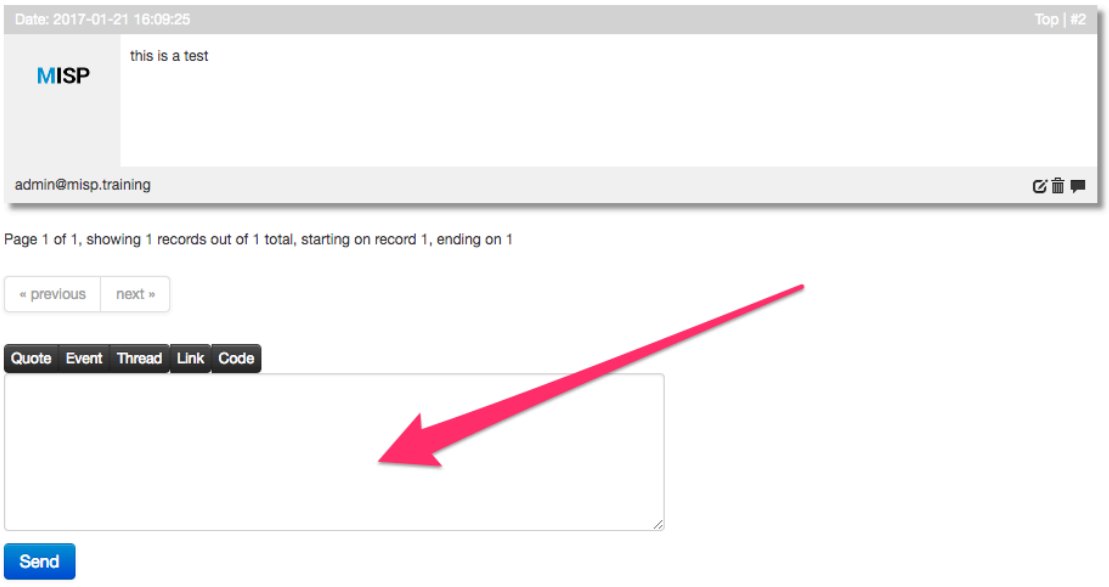


Figure 31.7: Contact reporter

31.8.4 Contact a reporter

This feature can be used to contact the person or the organisation that the person belongs to that has created the event.

All E-Mails can be enforced to be encrypted



Figure 31.8: Contact reporter

31.8.5 Receive alerts

It is possible to get alerts via encrypted mail in the following cases:

- published events by other user of the MISP instance
- events pushed to the MISP instance
- events pulled by the MISP instance

These E-Mail alerts are an opt-in feature

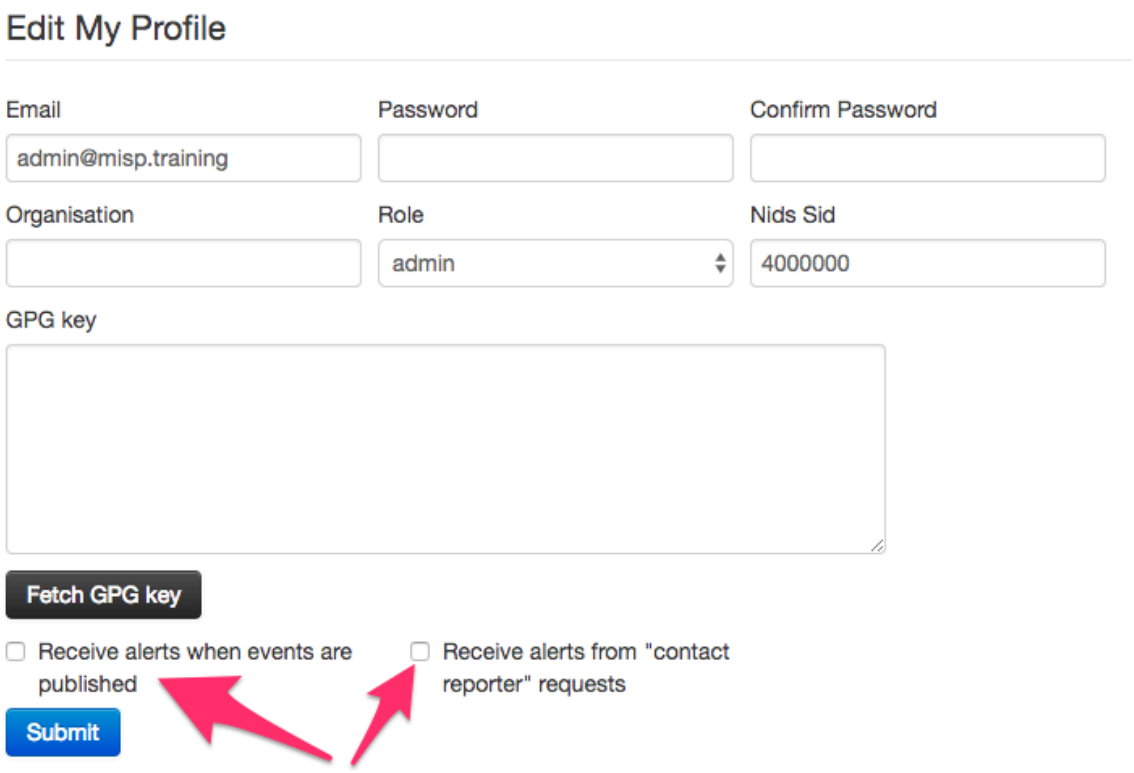


Figure 31.9: Change user settings

31.9 Cerebrate

Cerebrate is a companion open-source platform that acts as a **directory and distributed trust source** for a community: it holds authoritative records of **organisations** and **sharing groups** that many MISP instances can then draw from, instead of each instance maintaining that information by hand. Connecting your MISP to a Cerebrate lets you pull those organisations and sharing groups in, keeping identities and sharing-group memberships consistent across the community.

Cerebrate connections are managed under *Sync Actions* → *Cerebrates* and are **site-admin only**. To add one, provide:

- **Name** and **URL** of the Cerebrate node,
- an **Authkey** for its API (stored encrypted),
- the local **owner organisation** that represents this connection, an optional **description**, and a **Skip proxy** option, plus
- **Pull organisations** and **Pull sharing groups** toggles for what you want to retrieve.

From a configured Cerebrate you can **preview** the remote organisations and sharing groups — each is annotated with whether it already exists locally and what differs — before pulling. You can then **pull all** organisations or sharing groups (creating the missing ones and updating the rest), or **download** an individual organisation or sharing group. Pulled organisations are matched by UUID; a name clash is resolved by suffixing the imported name.

31.10 Sharing group blueprints

Maintaining sharing groups by hand becomes tedious when membership follows a rule ("every national CSIRT", "all organisations in the finance sector", "everyone except this org"). A **Sharing Group Blueprint** captures that rule as JSON and lets MISP **generate and keep a sharing group in sync automatically**.

Blueprints are managed under *Global Actions* → *List Sharing Group Blueprints*. Creating or editing one requires the sharing-group permission, and non-site-admins only see their own organisation's blueprints. A blueprint has a **name** and a **rules** definition (edited as JSON). The rules select organisations by

properties — organisation id/uuid/name/sector/nationality/type, or sharing-group id/uuid — combined with the boolean operators **OR**, **AND** and **NOT**.

Once defined, you can:

- **Execute** the blueprint — MISP resolves the rules to a set of organisations and either creates a new sharing group or updates the membership of the one it already manages.
- **Preview the organisations** the rules currently resolve to before executing.
- **Detach** the generated sharing group from the blueprint (making it a normal, manually managed sharing group).
- **Encode as a sync rule** (site admins) — transpose the resolved organisations directly into a sync server's push or pull rules.

31.11 TAXII 2.x servers

MISP can act as a **TAXII 2.1 client**, pushing your intelligence — converted to STIX 2.x — to a remote TAXII 2.1 collection, and browsing the collections and objects a TAXII server exposes. This is the standards-based path for sharing with platforms that speak TAXII rather than MISP's native synchronisation. The STIX conversion relies on the misp-stix library.

TAXII servers are configured under *Sync Actions* → *TAXII Servers* and are **site-admin only**. The form asks for:

- a **name** and **owner** label,
- the **discovery URL** of the TAXII server and a **Skip proxy** option,
- an **Enabled** toggle,
- the **authentication type** — *basic* (username and password) or *bearer* (an API token),
- the **API root** and **collection**, which are populated dynamically by querying the server (only collections you can write to are offered for pushing), and
- a **filters** definition (a MISP REST search filter, as JSON) that decides which events are pushed.

Once a server is configured you can **push** events to it (filtered by the rules you set — run as a background job when jobs are enabled), and **browse** the server's collections and the STIX objects within a collection. TAXII push complements the automation chapter's STIX export formats.

31.12 Recommendation

The following section will describe what is the best practice how many MISP instances that showed to be good for orgs. Of course depending on your specific requirements an architecture could be more spread or simplified.

The architecture is divided into several systems / stages beginning with:

31.12.1 MISP Staging System

This systems purpose is to be linked to all available external MISP systems that you have access to. It will download all events and do enrichment between these events.

31.12.2 MISP SECOps System

This system is the main system used by human analysts. It will it is not linked to any external MISP instance other then the Staging System.

To publish events to the community assign the right tags to match your push Rules and **publish the event**

32. Managing Feeds

33. Feeds

Feeds are remote or local resources containing indicators that can be automatically imported into MISP at regular intervals. Feeds can be structured in MISP format, CSV format or even free-text format. You can easily import any remote or local URL to store the data in your MISP instance. It's a simple way to gather many external sources of information into MISP without any programming skills.

Feed descriptions can be also easily shared among different MISP instances as you can export a feed description as JSON and import it back in another MISP instance.

33.1 Managing feeds

> ⚠ Warning
>
> A site admin role is required to perform these actions.

To manage feeds you first need to access the Feeds page using the **Feeds** item on the **Sync Actions** top menu:

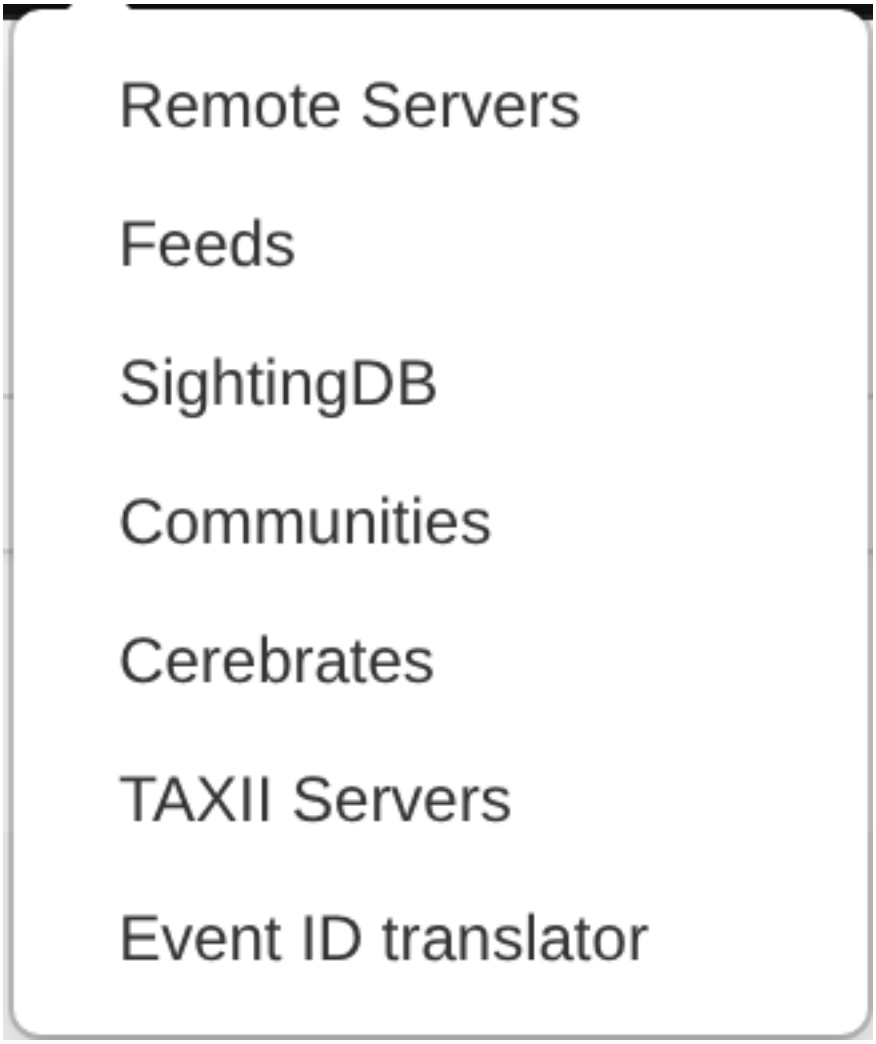


Figure 33.1: List feeds menu

Then you will see the Feeds page with a table of configured feeds.

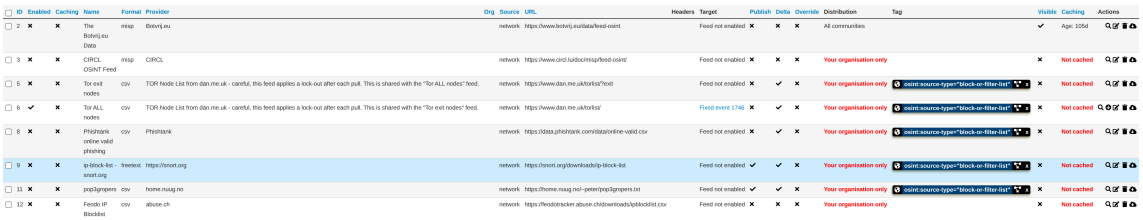


Figure 33.2: List feeds side menu

The index can be filtered by **scope** using the tabs at the top: **All**, **Default** (the feeds shipped with MISP) and **Custom** (feeds you added yourself). Selecting feeds with the checkboxes reveals **bulk actions**, so you can enable or disable many feeds at once instead of toggling each individually.

33.1.1 Default feeds

The MISP project supplies a list of open-source feeds. You can load these feed definitions by using the 'Load default feed metadata' button on the Feeds page. This feature creates new feeds by importing the entries in file app/files/feed-metadata/defaults.json to the database. Existing feeds are not changed. The

feature checks for duplicates using the feed URL. If a feed with the same URL already exists in the database, that entry is not imported. This ensures that local modifications such as name, distribution or enabled status are never overwritten.

33.1.2 Caching feeds

Caching downloads the feed content to the Redis server in your instance, and allows you to correlate attributes and see matching "Feed hits" (similar to correlated "Related Events") in the event view on each attribute row. Caching does not create any events in the database server in your instance.

Use the buttons at the top of the Feeds screen to retrieve data from feeds and store the data in the Redis cache. The buttons let you cache data from all feeds, cache data from freetext/CSV-format feeds only, or cache data from MISP-format feeds only.

33.1.3 Fetching feeds

Use the button at the top right of the Feeds screen to fetch data from all feeds and ingest the data to the MISP database.

33.1.4 Search feed caches

To search the feed caches, select the Search Feed Caches option on the side menu. This displays a table where you can search for values potentially contained in the cached feeds and servers.

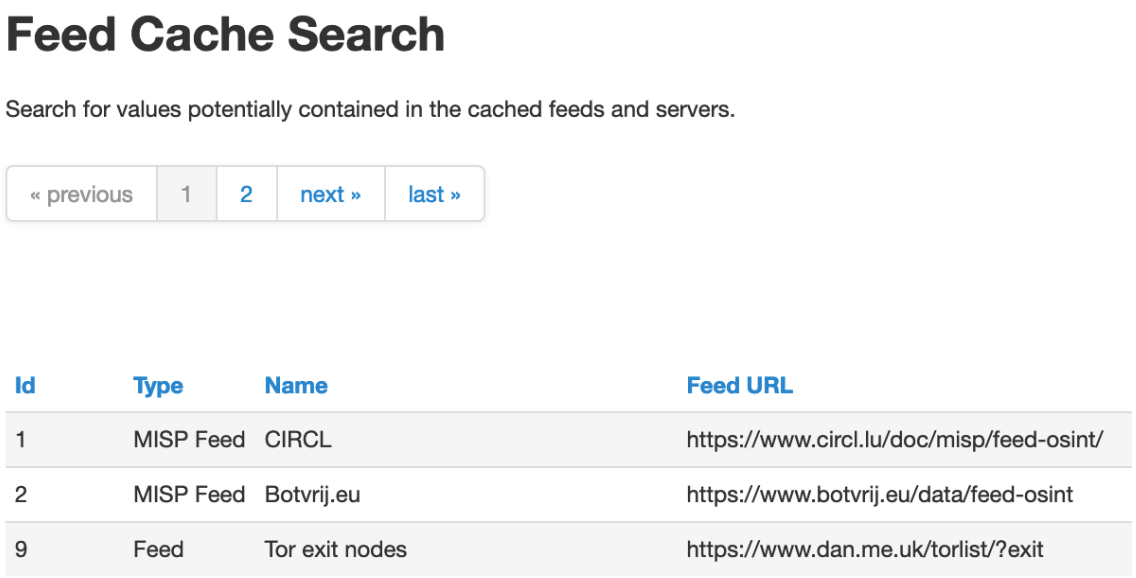


Figure 33.3: Feed cache search

33.1.5 Adding feeds

To add a new feed, select the Add Feed option on the side menu. Here you will have access to a dynamic form. Let's check each field by order. The form shows or hides fields based on the selections in the drop-down fields.

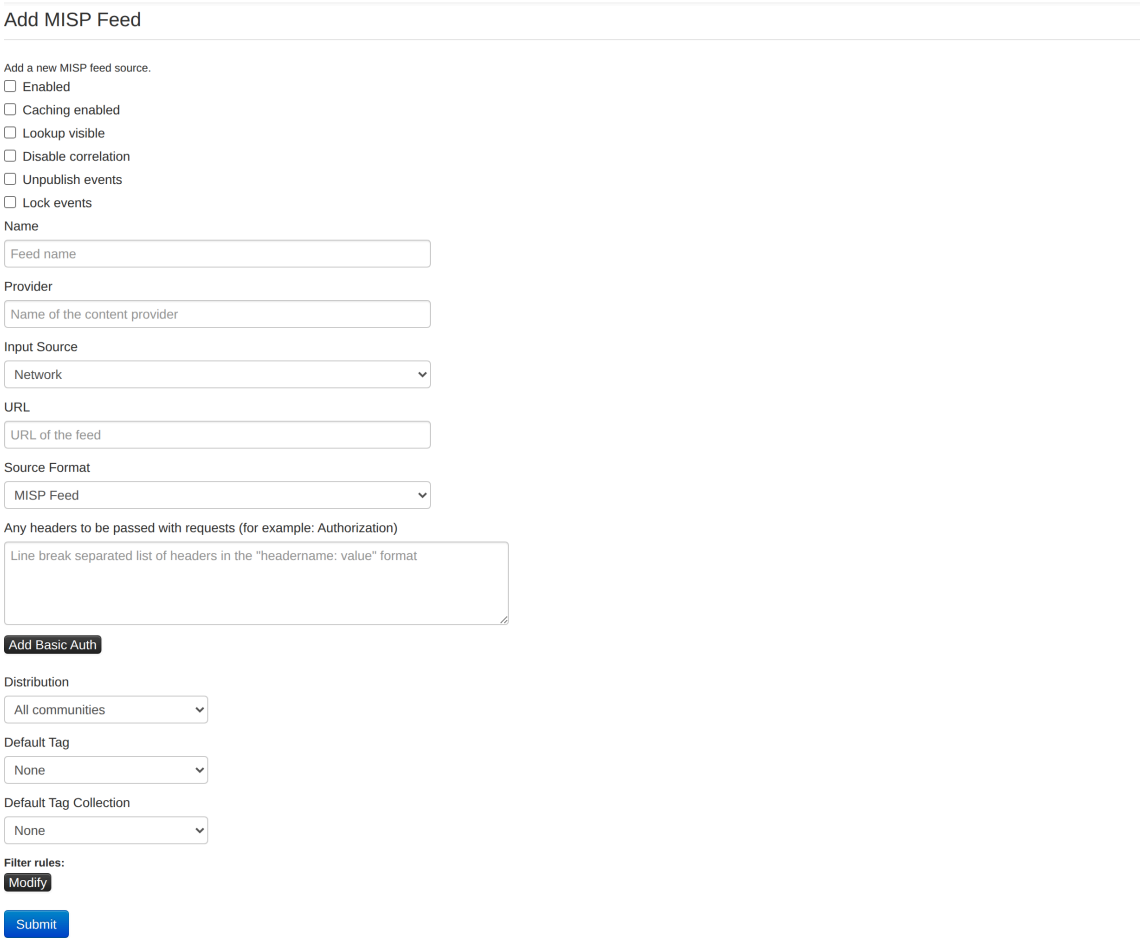


Figure 33.4: Add feed form

- Enabled: Is the feed active or not
- Caching enabled: Should the feed data be cached
- Lookup visible: If this is not checked, correlations will only show up for you; if checked, correlations are visible for other users as well
- Disable correlation: If this is checked, correlations will be disabled for all events coming from this Feed
- Unpublish event: If checked, the target event is unpublished whenever the feed updates it (useful when you want to review changes before they are re-published)
- Name: Name to identify the feed; not required to be unique
- Provider: Name of the content provider
- Input Source: Where does the input come from

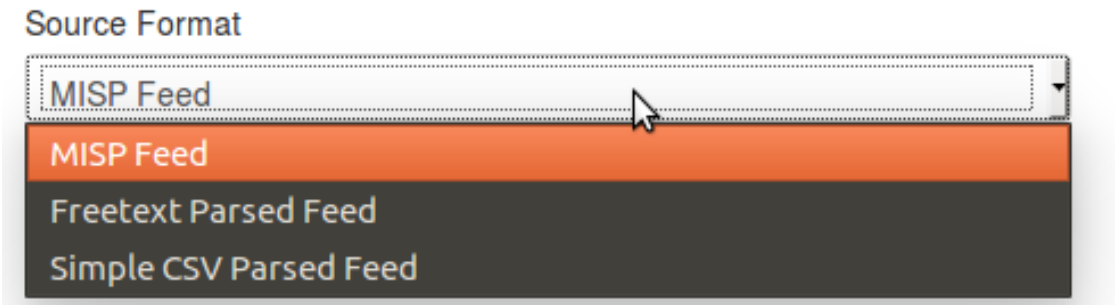


  - Network: hosted somewhere outside the platform

  - Local: Hosted on the local server. In this case, a new checkbox “Remove input after ingestion” will appear. If checked, the source is deleted after usage.

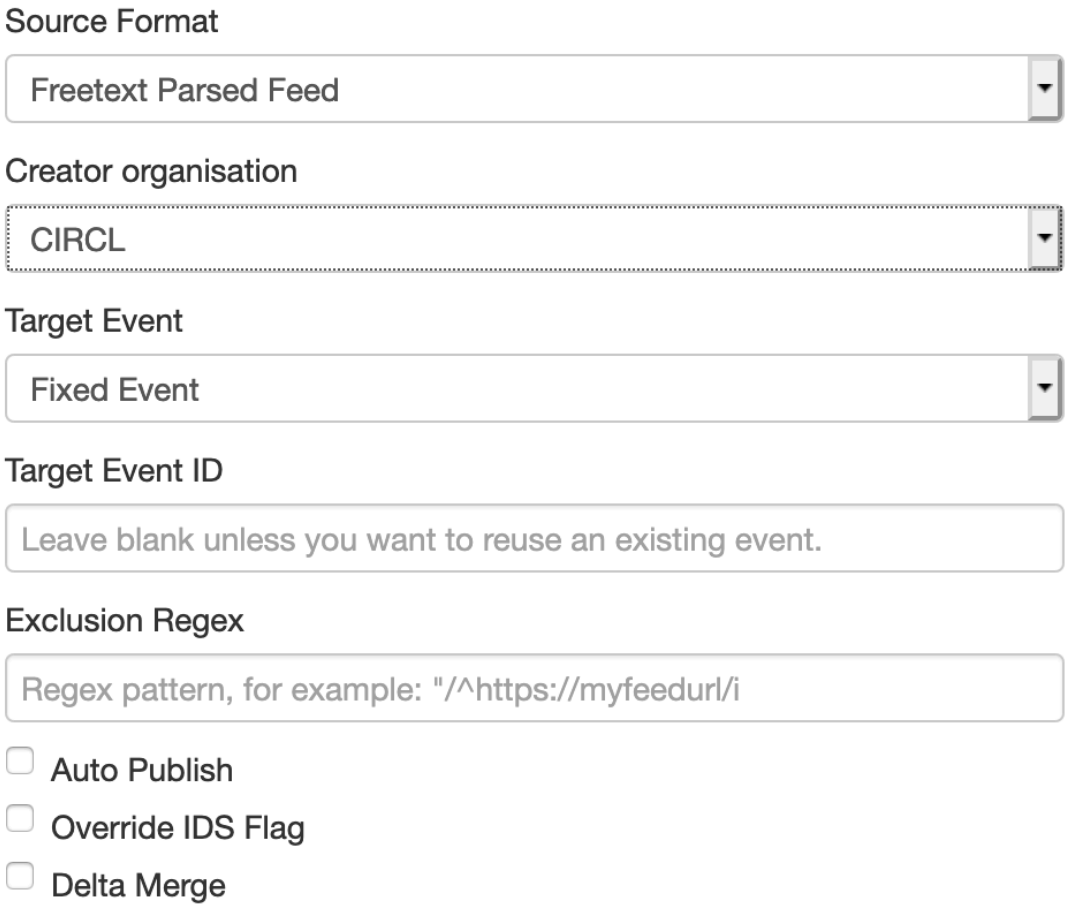


- URL: URL of the feed, where it is located (for Local hosted files, point to the manifest.json; e.g., /home/user/feed-generator/output/manifest.json)
- Headers: Any HTTP headers to pass with the feed request, as a line-break-separated list in `headername: value` format. This is how you consume a feed that requires authentication — for example an `Authorization` header carrying an API token or bearer key.
- The Source Format can be:

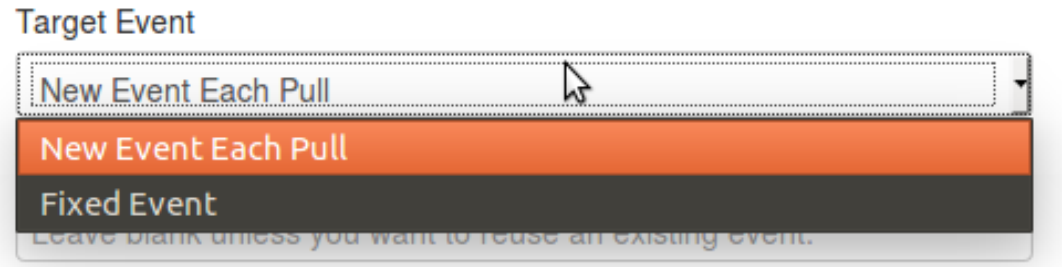


  - MISP Feed: The source points to a list of JSON formatted files like MISP events.
    Example: https://www.circl.lu/doc/misp/feed-osint

– Freetext Parsed Feed: The options for a freetext-parsed feed are shown below.

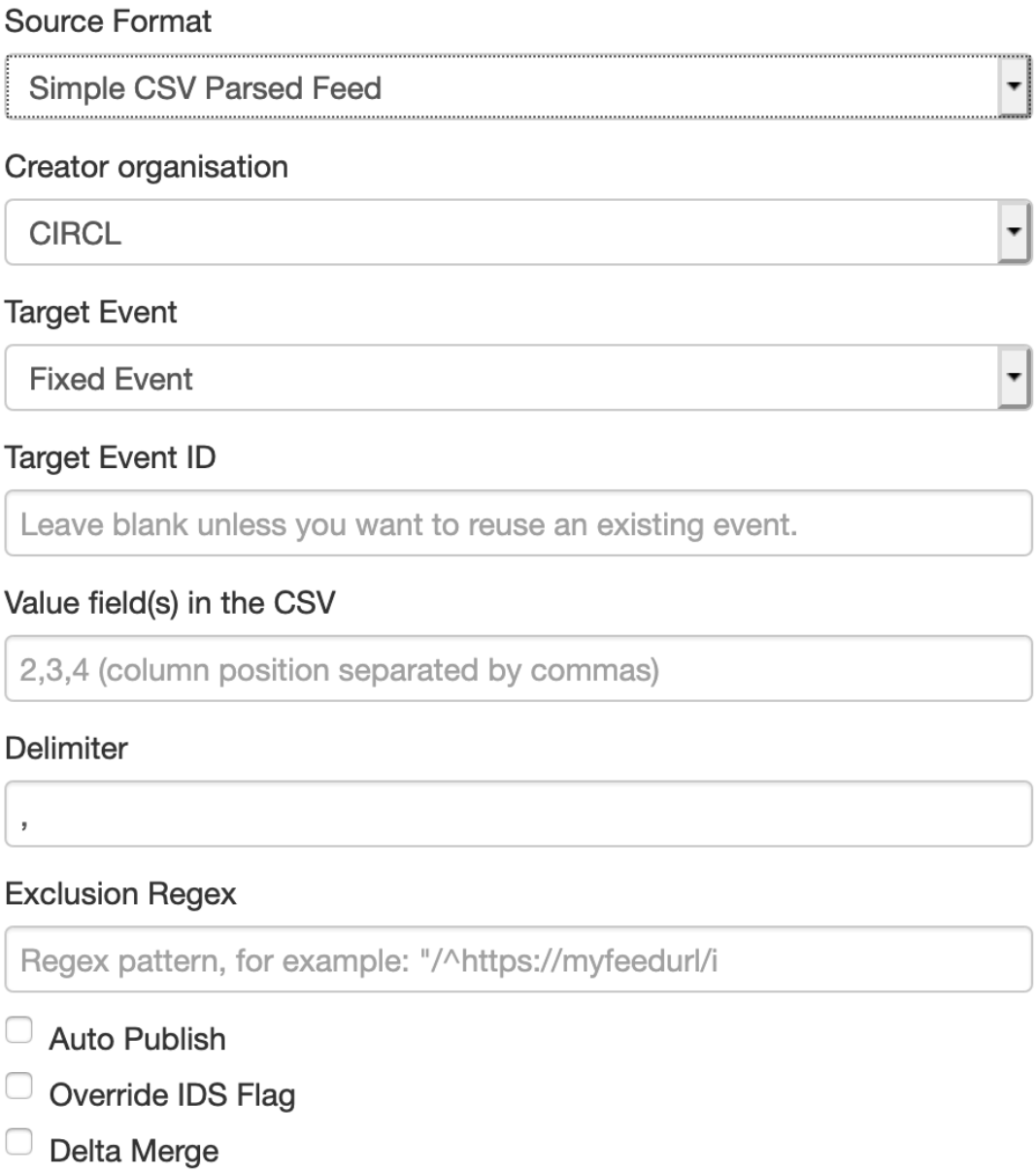


- Creator organisation: The creator organisation (orgc_id) for the event created from this feed. Appears in the Org column on the List Feeds screen.
- Target Event: The event to hold data from the feed. Can be either "New Event Each Pull" (a new event will be created each time the feed is pulled) or "Fixed Event" (a unique event will be updated with the new data, as determined by the next field).



- Target Event ID: The ID of the event where the data will be added; if not set, the field will be set the first time the feed is fetched.
- Exclusion Regex: Add a regex pattern for detecting IoCs that should be skipped. This can be useful to exclude any references to the actual report / feed for example.
- Auto Publish: If checked, the event created from the feed will be automatically published
- Override IDS Flag: If checked, the IDS flag will be set to false
- Delta Merge: If checked, only attributes from the most recent fetch are kept, the old ones are (soft-) deleted.

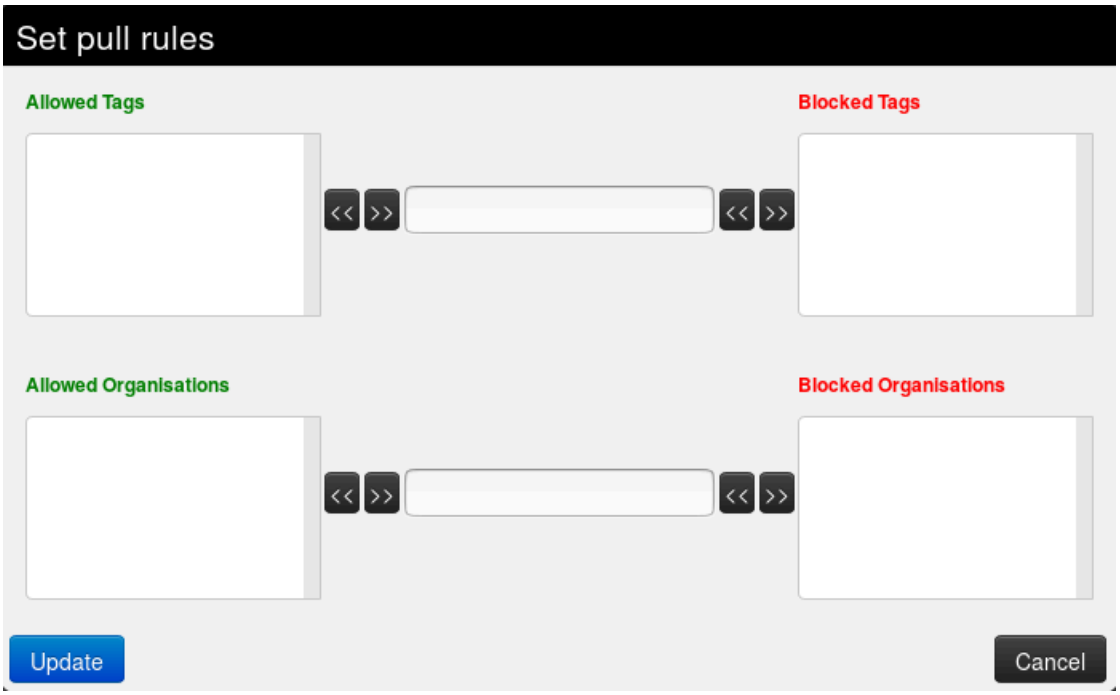
– Simple CSV Parsed Feed: The options for a chacter-separated feed are shown below.



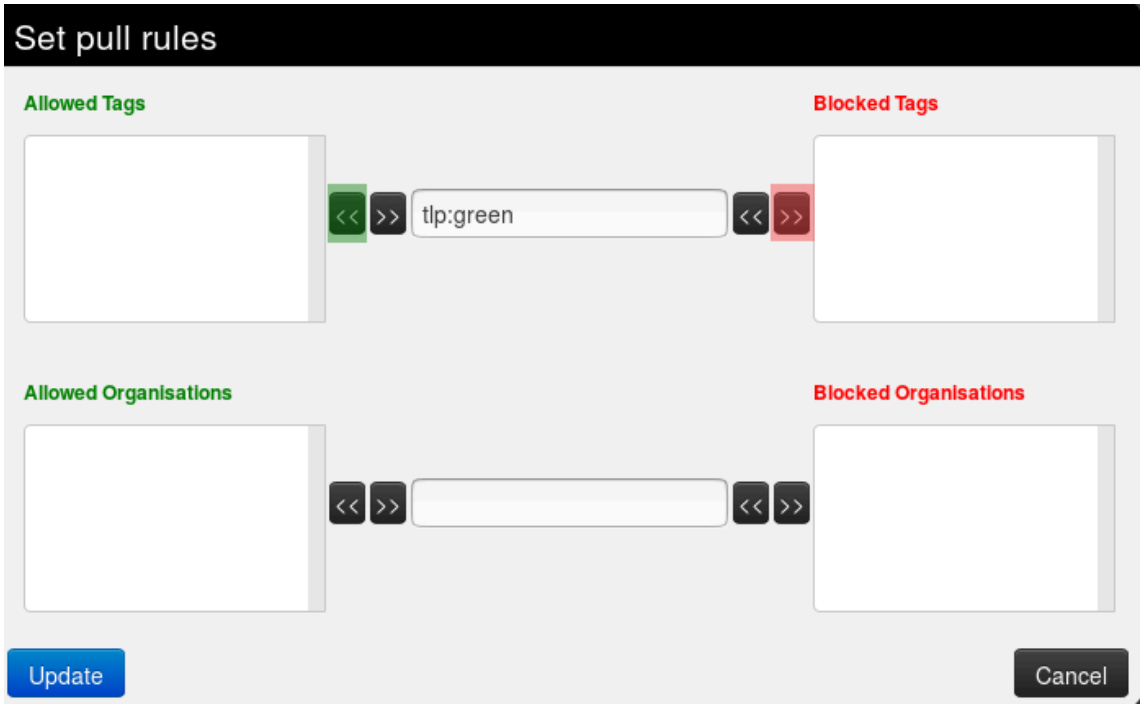
  - Creator organisation: The creator organisation (orgc_id) for the event created from this feed. Appears in the Org column on the List Feeds screen.
  - Target Event: The event to hold data from the feed. Can be either "New Event Each Pull" (a new event will be created each time the feed is pulled) or "Fixed Event" (a unique event will be updated with the new data, as determined by the next field).
  - Target Event ID: The ID of the event where the data will be added; if not set, the field will be set the first time the feed is fetched.
  - Value field(s) in the CSV: Select one or several fields that should be parsed by the CSV parser and converted into MISP attributes; specify column position separated by commas
  - Delimiter: Set the field separator; default is comma ","
  - Exclusion Regex: Add a regex pattern for detecting IoCs that should be skipped. This can be useful to exclude any references to the actual report / feed for example.
  - Auto Publish: If checked, the event created from the feed will be automatically published
  - Override IDS Flag: If checked, the IDS flag will be set to false
  - Delta Merge: If checked, only attributes from the most recent fetch are kept, the old ones are (soft-) deleted.

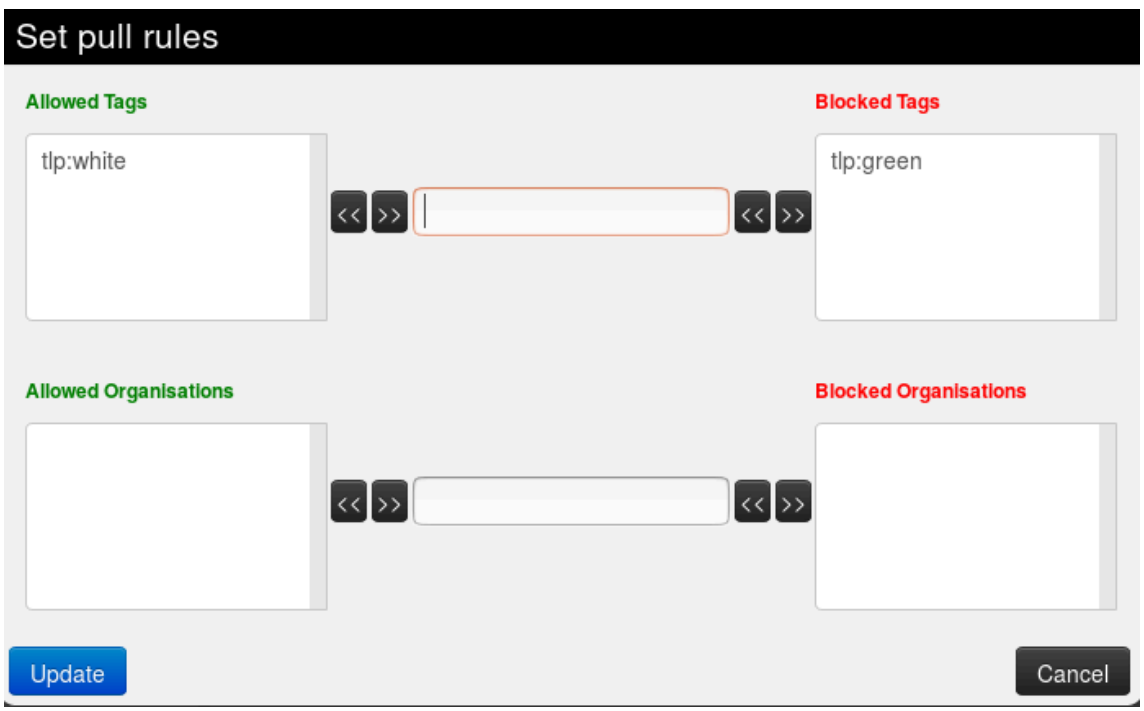
- Distribution: The distribution option that will be set on the event created from the feed. The choices are:

  - Your organisation only
  - This community only
  - Connected communities
  - All communities
  - Sharing Group. In this case, a new field Sharing Group appears where you must select a group.
  - Inherit from Feed (only for MISP feeds). In this case the original distribution and Sharing Group from the feed will be used. Note that the `feed_generator.py` needs to be configured with `with_distribution = True`. Data will be imported as "Your organisation only" if the distribution is not present in the feed.
- Default Tag: A default tag can be added to the created event(s). You can select either a single tag or a tag collection (a preset bundle of tags), so that every event created from the feed is consistently labelled.
- Filter rules: Here you can define which tags or organisations are allowed or blocked.

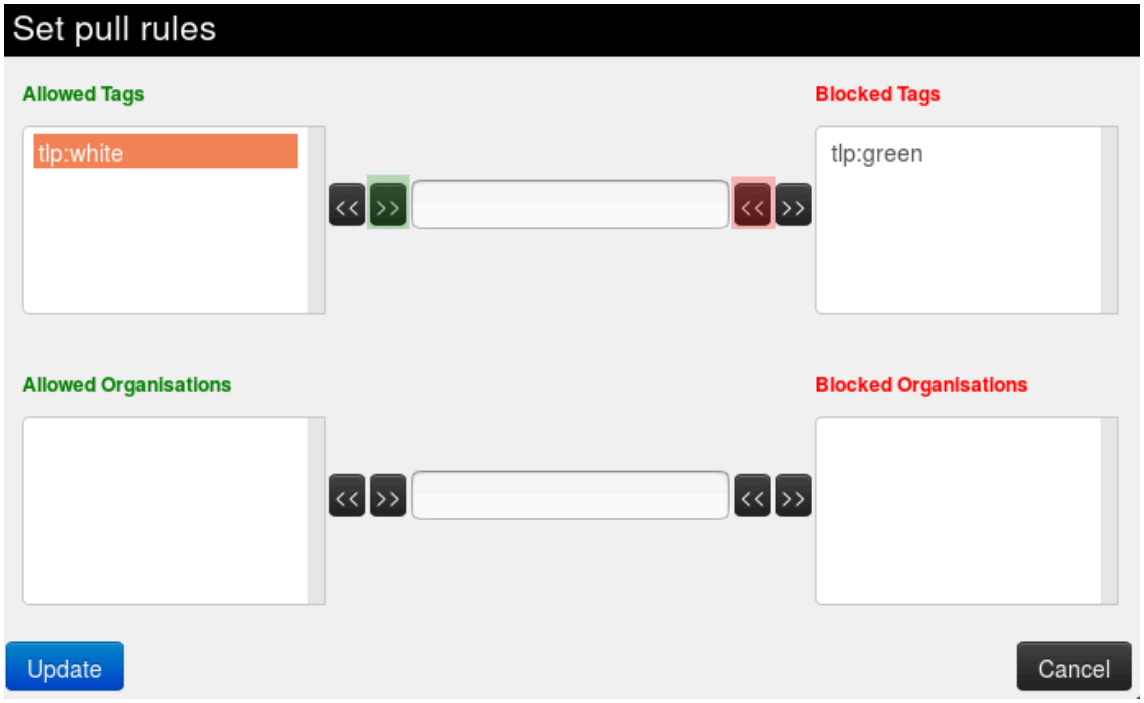


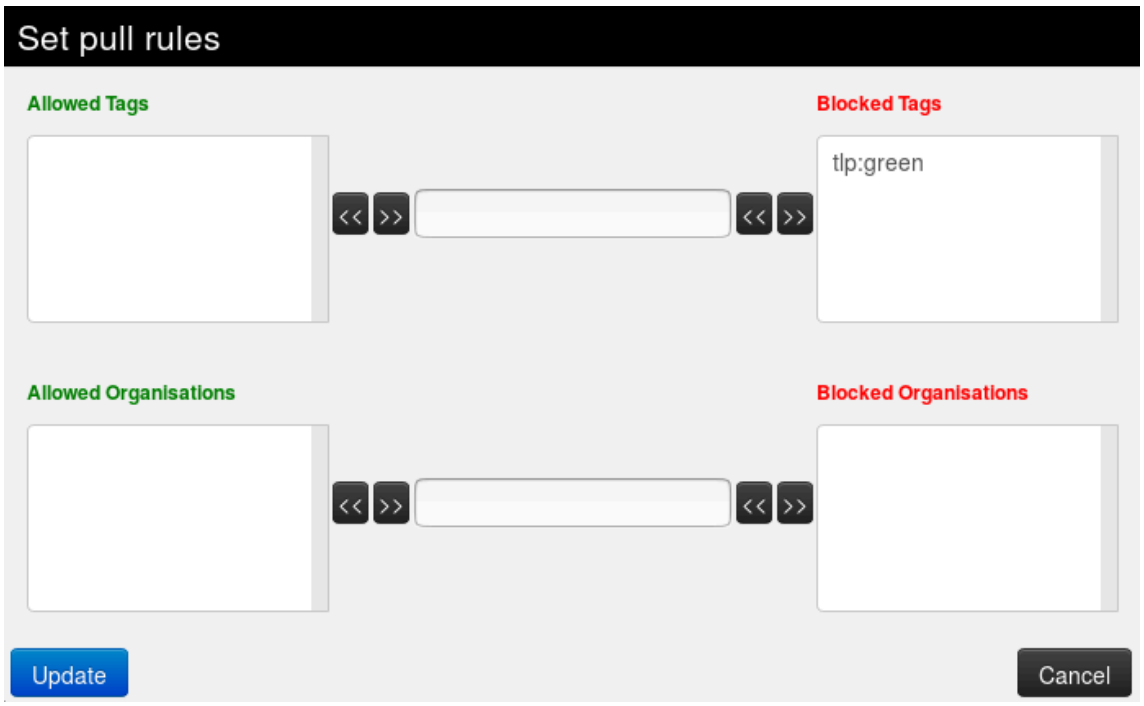
To add a tag (or organisation respectively), first type it into the top middle (bottom middle for organisation) text field. Then use the arrows that point to the outside to add it to the allowed or blocked tags (or organisations respectively) list.





To remove a tag (or organisation respectively), select it in the list and click on the arrow pointing to the inside.





33.1.6 Importing feeds

To import a new feed, select the Import Feeds from JSON option on the side menu. Paste MISP feed metadata JSON into the text box and click the Add button. Double check values in ID fields that are specific to an instance including the feed ID (field id), event ID (field event_id), organisation ID (field orgc_id), sharing group ID (field sharing_group_id) and tag ID (field tag_id).

33.1.7 Analyze feed overlap

To analyze feed data overlap, select the Feed overlap analysis matrix option on the side menu. This presents overlap matrix obtained by analyzing cached feed data.

Feed overlap analysis matrix

	F1	F2	F9	F10	F20	F25	F26	F31
Feed #1 CIRCL	-	1%	0%	0%	0%	0%	0%	0%
Feed #2 Botvrij.eu	30%	-	0%	0%	0%	0%	0%	0%
Feed #9 Tor exit nodes	0%	0%	-	95%	4%	0%	0%	6%
Feed #10 Tor ALL nodes	0%	0%	22%	-	1%	0%	0%	1%
Feed #20 pop3gropers	0%	0%	0%	0%	-	0%	0%	0%
Feed #25 OpenPhish url list	0%	0%	0%	0%	0%	-	0%	0%
Feed #26 firehol_level1	1%	0%	0%	0%	0%	0%	-	0%
Feed #31 blocklist.de	0%	0%	0%	0%	1%	0%	0%	-

Figure 33.5: Overlap analysis matrix

33.1.8 Exporting feeds

To export all feed metadata, select the Export Feeds settings option on the side menu. The server will push a JSON file for download.

33.2 Feed correlation

If an indicator from a feed matches an indicator within a MISP event, it will show up as “Feed hits” in the event overview. The correlation will not show up in the correlation graph of the event.

34. Collections

A **Collection** is a named, distributable container that bundles a set of **events** and/or **galaxy clusters** into one shareable unit. Collections let you organise data — whether it is shared by the community or produced during your own research — into buckets based on a common theme: a campaign, an intrusion set, a piece of research, and so on. Collections are first-class citizens in MISP: each has its own UUID and obeys the **same distribution and sharing rules as events**, so a collection can be shared and synchronised across instances just like the intelligence it groups.

Collections are managed from *Global Actions* → *List Collections*, and creating or editing one requires a role with modify permission.

> **Note**
>
> A collection groups **events and galaxy clusters** — nothing else. It is distinct from two similarly named features:
>
> - A **Sharing Group** is an access-control list (which organisations may see data), not a content bundle — a collection *references* a sharing group when its distribution is set to "Sharing group".
> - A **Tag Collection** is a preset bundle of *tags* applied together to events, unrelated to this feature.

34.1 What a collection contains

A collection has a small set of metadata plus a list of element references:

Field	Purpose
Name	A human-readable label for the collection.
Type	The kind of grouping: *campaign*, *intrusion set*, *named threat*, *research*, or *other*.
Description	Free text describing what the collection is for.
Distribution	How widely the collection may be shared (see below).
Sharing group	The sharing group to use when distribution is set to "Sharing group".

Each **element** in a collection is a *reference* (by UUID) to an event or a galaxy cluster, optionally with a short per-element description. The collection stores the reference, not a copy — so the underlying event or cluster keeps its own ownership and distribution, and an element can be removed from a collection without touching the event itself. The same element cannot be added to the same collection twice.

34.2 Creating and populating a collection

From *List Collections* choose to create a new collection and fill in its name, type, description and distribution. Once it exists you can add elements to it in two ways:

- **From the collection** — open the collection and add an element by giving its type (Event or Galaxy Cluster) and UUID.
- **From an event or cluster** — while viewing an event or a galaxy cluster, use the "add to collection" action to attach it to one of your organisation's existing collections. If you have not created a collection yet, MISP prompts you to create one first.

The collection view lists all of its elements and shows the collection's metadata; the index lists your collections with an element count, type, distribution and timestamps, and lets you edit or delete each. Deleting a collection removes its element references but leaves the referenced events and clusters untouched.

34.3 Distribution and sharing

Collections use the standard event distribution levels:

Level	Meaning
0	Your organisation only
1	This community only
2	Connected communities
3	All communities
4	Sharing group

When the level is *Sharing group*, MISP checks that you are allowed to use the chosen sharing group. Visibility follows the same logic as events: you can see a collection if your organisation owns it, if it is distributed to your community or wider, or if it belongs to a sharing group you are part of.

Note that a collection's distribution governs visibility of the **collection object itself** — the list and its metadata. The events and clusters it references each retain their own distribution, so being able to see a collection does not automatically grant access to every element inside it.

34.4 Synchronising collections between instances

Collections synchronise between MISP instances much like events do, but only when **both** instances support the feature and the relevant per-server toggle is enabled. On the sync server configuration (see Sharing) there are two dedicated options:

- **Push collections** — send your eligible collections to the remote instance.
- **Pull collections** — retrieve collections from the remote instance.

During synchronisation MISP negotiates collection support with the peer, exchanges a compact `{uuid: modified}` index to find what is new or newer, and transfers only the differences. Distribution degrades by one step on each outbound hop, exactly as for events. Collections received from a remote instance are marked as *locked* so that a locally created collection is never silently overwritten by an inbound copy; between two synced copies, the one with the more recent modification time wins. Editing a collection's elements updates its modification time so the change propagates on the next sync. There is no separate revision history — the modification timestamp and the locked flag are what keep copies consistent.

34.5 API

Collections are available over the REST API under the `/collections/` and `/collectionElements/` controllers, using the standard MISP JSON envelope:

Endpoint	Purpose
`GET /collections/index[.json]`	List collections (the REST view includes their elements).
`GET /collections/view/{id or uuid}`	Retrieve a single collection and its elements.
`POST /collections/add`	Create a collection (optionally attaching elements in the same call).
`POST /collections/edit/{id or uuid}`	Edit a collection.
`POST /collections/delete/{id or uuid}`	Delete a collection and its element references.
`POST /collectionElements/add/{collection_id}`	Add an element to a collection.

Endpoint	Purpose
`GET /collectionElements/index/{collection_id}`	List a collection's elements.
`POST /collectionElements/delete/{element_id}`	Remove an element from a collection.

Listing and viewing require only a logged-in user (subject to visibility); creating, editing and deleting require modify permission. The synchronisation endpoints used between instances require the sync permission. See the Automation & API chapter for authentication and the request/response envelope.

35. Cryptographic Keys & Event Signing

When an event is shared across a chain of MISP instances, each hop can, in principle, relay or modify it before passing it on. **Event signing** lets the originator of an event constrain who is allowed to do that. By attaching one or more **cryptographic keys** to an event and putting it into **protected mode**, the creator declares: *only instances that can sign updates with one of these keys may propagate or change this event.* Every hop then verifies the signature before accepting the event, giving tamper-evidence and origin authentication along the whole sharing path.

This is a security feature for high-assurance sharing. It does not replace distribution levels or sharing groups (which control *who can see* an event) — it controls *who can author and relay* updates to a protected event as it travels.

> **Note**
>
> Event signing uses **PGP** keys. Both ends of every synchronisation hop must run a MISP version that understands protected events (MISP 2.4.156 or newer — so any 2.5.x instance qualifies). An older instance simply will not be offered protected events.

35.1 How it works

- Each **cryptographic key** attached to an event is a **PGP public key** (identified by its fingerprint). The set of keys on an event is the list of authorised signers.
- An event's own key list travels **with the event** during synchronisation, so every receiving instance knows which keys are authorised. Key revocations propagate too: a key marked revoked upstream is marked revoked locally, and keys removed upstream are removed locally.
- When a **protected** event is transferred, the sending instance signs the payload with its instance key and includes a detached PGP signature in the transfer. The receiving instance verifies that signature against the event's authorised key list *before* accepting the add or edit. If verification fails, the event is **rejected** — it is never silently accepted in a weakened, unprotected form.
- An instance will only relay a protected event onward if it can itself sign for it (i.e. it holds a private key matching one of the event's public keys). If it cannot, the event simply stops at that hop rather than being passed on unsigned. This is what "cannot reliably be propagated beyond the reach of those that can sign" means in practice.

35.1.1 The instance signing key

An instance signs with its configured **GnuPG key** — the same key used for encrypted/signed email notifications (the `GnuPG.email` / `GnuPG.password` settings, or the key in the instance's GPG homedir when `GnuPG.download_gpg_from_homedir` is enabled). The instance's **public** key is published at `…/gpg.asc`. An instance can only participate in protected sharing if it has a GnuPG key configured; without one, signing fails and protected events cannot be pushed or served.

35.2 Protecting an event

The typical workflow, performed by a user in the event's creator organisation:

1. **Create the event** as usual.
2. **Attach the authorised signing keys.** On the event, add each hop's PGP *public* key. Your own instance's key can be added with a single click ("use the instance's signing key"); partner instances are added by pasting their published `gpg.asc` public key. Only a member of the event's creator organisation (or a site admin) may add or remove an event's keys.

3. **Switch the event to protected mode** using the *Protect* action. This flags the event as protected and, because it is a significant change, **un-publishes it** — so you must **re-publish** it for the change to take effect and propagate.

You can later **Unprotect** the event to remove the restriction (this also un-publishes it, requiring a re-publish).

> ⚠ Warning
>
> Once an event is protected, it — and its subsequent modifications — will be **rejected by MISP instances you synchronise with unless the hop through which the event is propagated can sign with one of the listed keys**. Make sure every instance that legitimately needs to relay or update the event has a key on the list, or the event will not reach where you expect.

35.3 Verification on synchronisation

Signature verification happens at every stage of sync, and failure always means the update is blocked:

- **Pushing** a protected event to a remote: MISP signs the payload with the instance key and sends the signature. If the remote is too old to understand protected events, or the instance has no valid signing key, the push is aborted.
- **Receiving** a pushed protected event (over the REST API): the receiver requires a valid signature matching the event's key list, and rejects the add/edit otherwise. (Site-admin and internal-sync trust paths are exempt from this check.)
- **Pulling** a protected event from a non-internal remote: the puller validates the remote's signature before importing; a bad or missing signature means the event is **not imported** and the failure is logged.
- **Serving** a pull: MISP signs the served event with the instance key. A protected event whose key list does **not** include this instance's key is omitted from what the instance serves — propagation halts here rather than continuing unsigned.

All signature validation failures are written to the audit log, so you can see when and why a protected event was rejected.

35.4 Permissions and the server-signing endpoint

- Attaching or removing an event's keys, and toggling protection, require the `perm_add` permission (plus creator-organisation ownership of the event).
- MISP also exposes a **server-signing** endpoint, `POST /cryptographicKeys/serverSign`, which signs an arbitrary payload with the instance key and returns the signature. It is gated by a dedicated **Server Signing** role permission (`perm_server_sign`) and is intended for internal services that need to produce instance-signed data.

35.5 API

Endpoint	Purpose
`POST /cryptographicKeys/add/Event/{event_id}`	Attach a PGP public key to an event.
`POST /cryptographicKeys/delete/{id}`	Remove a key from an event.
`GET /cryptographicKeys/view/{id}`	View a stored key.
`GET /cryptographicKeys/index`	List cryptographic keys.
`POST /cryptographicKeys/serverSign`	Sign a payload with the instance key (`perm_server_sign`).
`POST /events/protect/{id}`	Put an event into protected mode.
`POST /events/unprotect/{id}`	Remove protected mode from an event.

During synchronisation the detached signature is carried in the `x-pgp-signature` HTTP header; REST clients pushing a protected event must include it. See the Automation & API chapter for authentication and the request/response envelope, and Sharing for how synchronisation between instances is configured.

V Automation & Integration

36 Automation and MISP API 257
36.1 The MISP REST API 257
36.2 General 257
36.3 Automation using PyMISP 260
36.4 Status Codes 260
36.5 Error Handling 261
36.6 Search 261
36.7 Events management 262
36.8 Tag management 267
36.9 Attribute management 269
36.10 Objects management 277
36.11 Server management 278
36.12 Sightings 278
36.13 User management 278
36.14 Discussion API 281
36.15 Organisation management 281
36.16 Special Cases 283
36.17 RPZ export 285
36.18 Text export 286
36.19 RESTful searches with XML result export 287
36.20 Export attributes of event with specified type as XML 288
36.21 Filtering event metadata 289
36.22 Download attachment or malware sample . . . 289
36.23 Download malware sample by hash 289
36.24 Upload malware samples using the “Upload Sample” API 290
36.25 Proposals API 290
36.26 Sharing groups 291
36.27 Enable, disable and fetching feeds via the API . 291
36.28 Sightings API 292
36.29 Warninglists API 295

37 Attribute statistics API 297

38 Additional statistics 299

39 MISP modules 301
39.1 Description 301

40 PyMISP - Python Library to Access MISP 305

41 ZeroMQ - MISP Publish/Subscribe . 311
41.1 MISP ZeroMQ 311

42 Modules 323
42.1 Module families 323
42.2 Running the module server 323
42.3 Enabling and configuring modules in MISP . . . 323
42.4 Using modules 323
42.5 Module endpoints 324

43 External Connectors 325
43.1 How MISP connects to other tools 325
43.2 Notable integrations 325
43.3 Worked example: MISP to Microsoft Graph Security 326

36. Automation and MISP API

Almost everything you can do in the MISP web interface you can also do programmatically through its **REST API**: creating and updating events and attributes, searching, importing and exporting in many formats, and administering the instance. Automation is authenticated with an API key rather than an interactive login, so scripts and tools can talk to MISP directly. Your personal API keys are managed under **Global Actions → My Profile → Auth Keys**.

36.1 The MISP REST API

MISP ships a complete, self-documenting API description. Rather than relying on a hand-maintained endpoint list, use these built-in references on your own instance:

- **OpenAPI / Swagger reference** — a browsable OpenAPI 3.0 specification of the API is served at **`/api/openapi`** (rendered with Redoc). It is the authoritative, versioned reference for the endpoints and their request/response schemas for your exact MISP version.
- **Built-in REST client** — the interactive REST client at **`/api/rest`** lets you build, send and inspect API calls from within MISP itself, using your session for authentication. It shows the exact headers, URL and body for a call (handy for turning a UI action into a script) and keeps a history of your requests.
- **PyMISP** — for Python, the PyMISP library wraps the API; see that chapter.

The rest of this chapter is a set of practical recipes for the most common calls. Where an endpoint's full parameter set matters, cross-check it against `/api/openapi` for your version, as the API evolves between releases.

36.2 General

36.2.1 Automation URL

The documentation will include a default MISP URL in the examples. Don't forget to replace it with your MISP URL.

Default MISP URL in the documentation:

```
https://<misp url>/
```

36.2.2 Automation key

The authentication of the automation is performed via a secure key. Make sure you keep that key secret, as it gives access to your MISP data! Manage your keys under **Global Actions > My Profile > Auth Keys**.

Pass the key in an `Authorization` header on every request. URL-based authentication (putting the key in the URL) is no longer supported.

The authorization is performed by using the following header:

```
Authorization: YOUR API KEY
```

36.2.2.1 Creating an automation key (using advanced authkeys)

Using the menu, go to Global Actions > My Profile and click "Auth keys" to show the auth keys view.

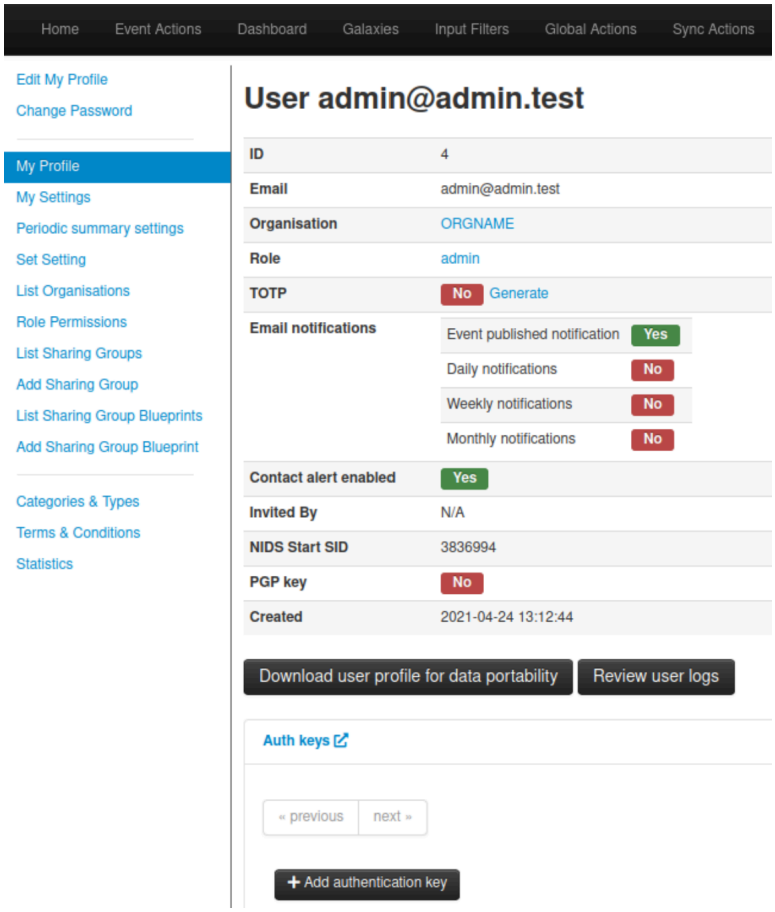
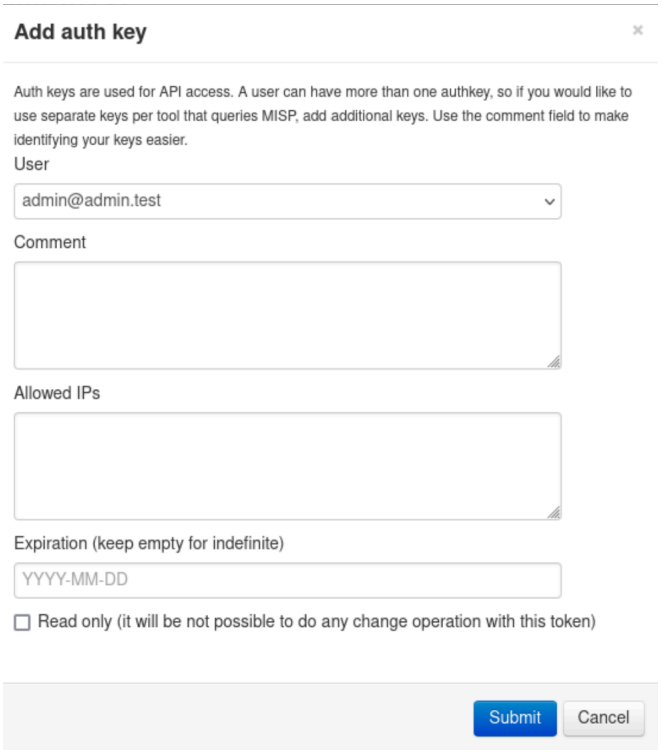


Figure 36.1: Screenshot of My Profile view with Auth keys expanded

The following form will be displayed:



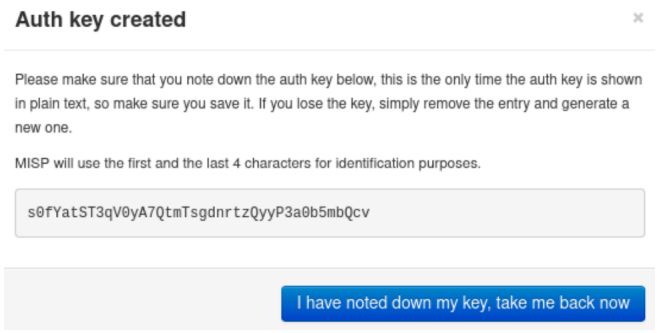
You can add an optional comment to indicate what the key will be used for.

You can also limit the usage of the key to specific IPs or subnets (one per line), by adding them in the Allowed IPs field. On some instances it is mandatory to set an IP allowlist. When adding subnets, please note that you need to use the format network_ip/subnet_mask .

You can optionally set an expiration time for the key.

Finally, it is also possible to make this key read-only, meaning that it will not be possible to do any changes on this instance using this automation key.

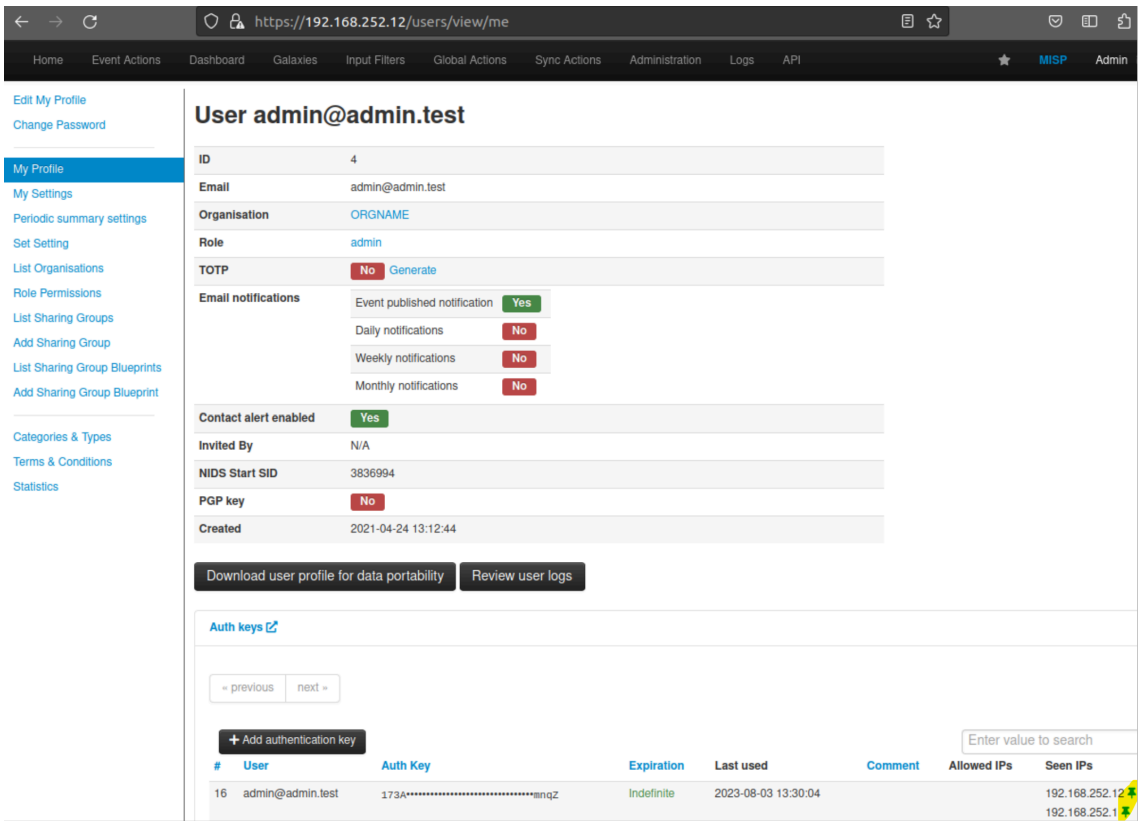
After clicking submit you will get a confirmation that the auth key was created, the key will be shown only one time.



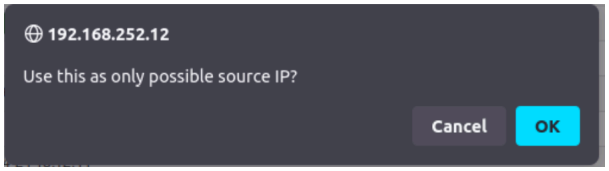
The same fields are available when editing an automation key.

36.2.2.2 Pinning an allowed IP for an automation key (using advanced authkeys)

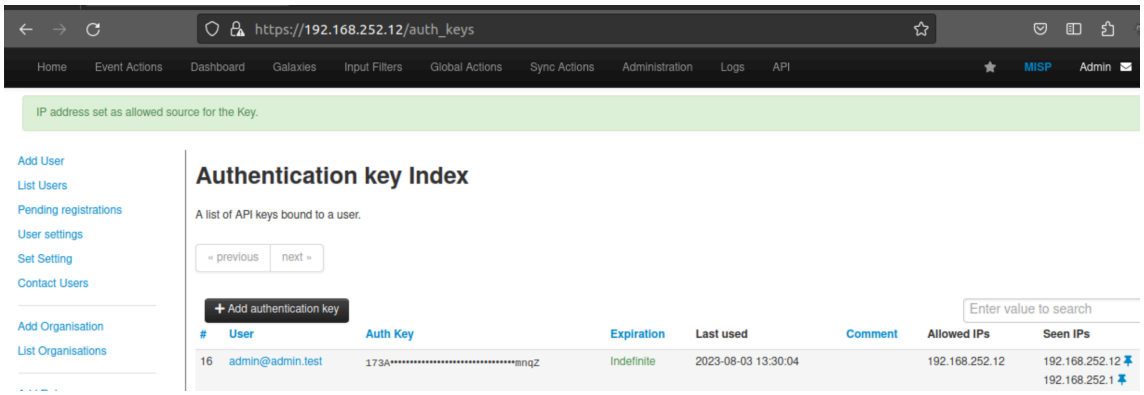
MISP will keep track of the unique IPs that were seen for a specific automation key. You can easily limit future usage of an automation key to one of the IPs that was seen in the past. To do so, using the menu, go to Global Actions > My Profile and click “Auth keys” to show the auth keys view. If the automation key was used in the past, you will see the “Seen IPs” listed per key. Click on the pin button next to the IP you want to limit usage to.



You will get a pop up requesting confirmation that you want to pin this IP for the key:



After confirmation, if all goes well, you will get a confirmation that the allowed IP was set for the automation key:



36.2.3 Accept and Content-Type headers

When performing your request, depending on the type of request, you might need to explicitly specify in what content type you want to get your results. This is done by setting one of the below Accept headers:

```
Accept: application/json
Accept: application/xml
```

When submitting data in a POST, PUT or DELETE operation you also need to specify in what content-type you encoded the payload. This is done by setting one of the below Content-Type headers:

```
Content-Type: application/json
Content-Type: application/xml
```

Example:

```
curl --header "Authorization: YOUR API KEY" --header "Accept: application/json" --
header "Content-Type: application/json" https://<misp url>/
```

By appending .json or .xml the content type can also be set without the need for a header.

36.3 Automation using PyMISP

PyMISP is a Python library to access MISP platforms via their REST API.

PyMISP allows you to fetch events, add or update events/attributes, add or update samples or search for attributes.

PyMISP is available including a documentation with various examples.

36.4 Status Codes

The API uses standard HTTP status codes:

- **200 OK** — the request succeeded.
- **403 Forbidden** — authentication failed (missing/invalid API key) or the key's role lacks permission for the action; also returned for a read-only key attempting a write.
- **404 Not Found** — the endpoint or object does not exist (or is not visible to you).
- **405 Method Not Allowed** — the HTTP method is not accepted for that endpoint.
- **500 Internal Server Error** — an error occurred server-side; check the application logs.

Error responses carry a JSON body with `name`, `message` and the requested `url` (see the example below).

36.5 Error Handling

36.5.1 Wrong endpoint chosen

36.5.1.1 Example

```
curl --header "Authorization: YOUR API KEY " --header "Accept: application/json" --
header "Content-Type: application/json" http://10.50.13.60/servers/gaaa
```

```
{"name":"Not Found","message":"Not Found","url":"\/servers\/gaaa"}
```

36.6 Search

It is possible to search in the database for a list of attributes or events based on a list of criteria.

To return attributes or events in a desired format, use the following URL and header settings:

URL: YOUR_MISP_URL/attributes/restSearch YOUR_MISP_URL/events/restSearch

Headers: Accept: application/json Content-type: application/json Authorization: YOUR_API_KEY

The next feature to take care of then is the body of the query. This is where you are going to put your filters.

As an example, if we want to export all the IP addresses that have a TLP marking and not marked as TLP:red, you can find below the corresponding filters to use: ~~json { “returnFormat”: “json”, “type”: { “OR”: [“ip-src”, “ip-dst”] }, “tags”: { “NOT”: [“tlp:red”], “OR”: [“tlp:%”] } }~~

Find below a non exhaustive list of parameters that can be used to filter data in your search (some parameters specific to given export formats are not mentioned): - **returnFormat**: Set the return format of the search. Can be passed as the first parameter after `restSearch` or via the JSON payload. The formats supported by `/events/restSearch` are: `attack`, `attack-sightings`, `cache`, `context`, `context-markdown`, `count`, `csv`, `hashes`, `hosts`, `json`, `kunai`, `netfilter`, `opendata`, `openioc`, `rpz`, `snort`, `stix`, `stix-json`, `stix2`, `suricata`, `text`, `xml`, `yara` and `yara-json`. The exact set can vary by version — see `/api/openapi` for the authoritative list. - **limit**: Limit the number of results returned, depending on the scope (for example 10 attributes or 10 full events). - **page**: If a limit is set, sets the page to be returned. page 3, limit 100 will return records 201->300). - **value**: Search for the given value in the attributes’ value field. - **type**: The attribute type, any valid MISP attribute type is accepted. - **category**: The attribute category, any valid MISP attribute category is accepted. - **org**: Search by the creator organisation by supplying the organisation identifier. - **tags**: To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a ‘!’. - **quickfilter**: Enabling this (by passing “1” as the argument) will make the search ignore all of the other arguments, except for the auth key and value. MISP will return an xml / json (depending on the header sent) of all events that have a sub-string match on value in the event info, event orgc, or any of the attribute value1 / value2 fields, or in the attribute comment. - **from**: Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event. - **to**: Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event. - **eventid**: The events that should be included / excluded from the search - **withAttachments**: If set, encodes the attachments / zipped malware samples as base64 in the data field within each attribute - **metadata**: Only the metadata (event, tags, relations) is returned, attributes and proposals are omitted. - **uuid**: Restrict the results by uuid. - **publish_timestamp**: Restrict the results by the timestamp of the last publishing of the event. The input can be a timsetamp or a short-hand time description (7d or 24h for example). You can also pass a list with two values to set a time range (for example [“14d”, “7d”]). - **last**: (Deprecated synonym for publish_timestamp) Restrict the results by the timestamp of the last publishing of the event. The input can be a timestamp or a short-hand time description (7d or 24h for example). You can also pass a list with two values to set a time range (for example [“14d”, “7d”]). - **timestamp**: Restrict the results by the timestamp (last edit). Any event with a timestamp newer than the given timestamp will be returned. In case you are dealing with / attributes as scope, the attribute’s timestamp will be used for the lookup. The input can be a timestamp or a short-hand time description (7d or 24h for example). You can also pass a list with two values to set a time range (for example [“14d”, “7d”]). - **published**: Set whether published or unpublished events should be returned. Do not set the parameter if you want both. - **enforceWarninglist**: Remove any attributes from the result that would cause a hit on a warninglist entry. - **to_ids**: By default (0) all attributes are returned that match the other filter parameters, regardless of their to_ids setting. To restrict the returned data set to to_ids only attributes set this parameter to 1. You can only use the special “exclude” setting

to only return attributes that have the to_ids flag disabled. - **deleted**: Default value 0. If set to 1, only deleted attributes will be returned. If set to [0,1] , both deleted and non-deleted attributes wil be returned. - **includeEventUuid**: Instead of just including the event ID, also include the event UUID in each of the attributes. - **event_timestamp**: Only return attributes from events that have received a modification after the given timestamp. The input can be a timestamp or a short-hand time description (7d or 24h for example). You can also pass a list with two values to set a time range (for example ["14d", "7d"]). - **sgReferenceOnly**: If this flag is set, sharing group objects will not be included, instead only the sharing group ID is set. - **eventinfo**: Filter on the event's info field. - **searchall**: Search for a full or a substring (delimited by % for substrings) in the event info, event tags, attribute tags, attribute values or attribute comment fields. - **attackGalaxy**: Select the ATT&CK matrix like galaxy to use when using returnFormat = attack. Defaults to the Mitre ATT&CK library via mitre-attack-pattern. - **order**: Only available for /events/restSearch. Sorts the results, can be useful in combination with limit. The accepted values to sort on are "Event.[any_field_in_the_events_table] [asc|desc]". You can also sort on multiple columns, see example below. ~ { "returnFormat": "json", "limit": "3", "order": "Event.attribute_count desc, Event.date desc" } ~

36.7 Events management

36.7.1 /events

36.7.1.1 Accepted Methods

- GET
- POST
- PUT
- DELETE

36.7.1.2 Description

Receive, update or delete Events. There is also a good amount of special output formats that can be triggered.

36.7.2 GET /events

36.7.2.1 Description

Receive events based on criteria

36.7.2.2 URL Arguments

- event_id: Event id to receive
- event_uuid : Event uuid to receive

36.7.2.3 Output


```
[{"id":"1","org_id":"1","date":"2014-12-10","info":"OSINT - F-Secure W32\/Regin, Stage
#1","uuid":"54884656-2da8-4625-bf07-43ef950d210b","published":true,"analysis":"2","attribute_count":"39"
{"id":"1","name":"ORGNAME"},"Orgc":{"id":"2","name":"CIRCL"},"EventTag":
[{"id":"1","event_id":"1","tag_id":"1","Tag":{"id":"1","name":"Type:OSINT","colour":"#1eed40","exportable
{"id":null,"name":null}}]
```


36.7.2.4 Example

```
curl --header "Authorization: YOUR API KEY" --header "Accept: application/json" --
header "Content-Type: application/json" https://<misp url>/
```

36.7.3 POST /events

36.7.3.1 Example

```
curl -i -H "Accept: application/json" -H "content-type: application/json" -H
"Authorization: YOUR API KEY" --data "@event.json" -X POST http://10.50.13.60/events
```

That is how an event JSON object should look like ~~json {“Event”:{“date”:“2015-01-01”,“threat_level_id”:“1”,“info”:“testevent”,“published”:false,“analysis”:“0”,“distribution”:“0”,“Attribute”:[{“type”:“domain”,“category”:“Network activity”,“to_ids”:false,“distribution”:“0”,“comment”:“”,“value”:“test.com”}]}}~~

36.7.4 DELETE /events

36.7.4.1 Description

Delete events based on criteria

36.7.4.2 URL Arguments

- event_id: Event id to receive
- event_uuid : Event uuid to receive

36.7.4.3 Output

```
{
    "name": "Event deleted.",
    "message": "Event deleted.",
    "url": "\/events\/delete\/1"
}
```

36.7.4.4 Example

curl –header “Authorization: YOUR API KEY” –header “Accept: application/json” –header “Content-Type: application/json” https:/// curl –header “Authorization: YOUR API KEY” –header “Accept: application/json” –header “Content-Type: application/json” -X “DELETE” http://10.50.13.60/events/1

36.7.5 GET /events/index

36.7.5.1 Description

Return the event index. - Warning, there’s a limit on the number of results

36.7.5.2 Output

```
[{"id":"1","org_id":"1","date":"2014-12-10","info":"OSINT - F-Secure W32\/Regin, Stage
#1","uuid":"54884656-2da8-4625-bf07-43ef950d210b","published":true,"analysis":"2","attribute_count":"39"
{"id":"1","name":"ORGNAME"},"Orgc":{"id":"2","name":"CIRCL"},"EventTag":
[{"id":"1","event_id":"1","tag_id":"1","Tag":{"id":"1","name":"Type:OSINT","colour":"#1eed40","exportabl
{"id":null,"name":null}}]
```

36.7.5.3 Example

```
curl --header "Authorization: YOUR API KEY " --header "Accept: application/json" --
header "Content-Type: application/json" http://10.50.13.60/events/index
```

36.7.6 POST /events/AddTag

Add a tag or a tag collection to an existing event.

36.7.6.1 URL Arguments

- event (id)
- tag (id, name or collection_[collection_id])

Examples:

```
curl \
 -d '{"event":"1210","tag":"tlp:green"}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.223/events/addTag
```

```
curl \
 -d '{"event":"1210","tag":"383"}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.223/events/addTag
```

```
curl \
 -d '{"event":"1210","tag":"collection_1"}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.223/events/addTag
```

36.7.7 POST /events/removeTag

Remove a tag from an existing event. Note that removing a tag collection in one go is not possible.

36.7.7.1 URL Arguments

- event (id)
- tag (id, name).

Examples:

```
curl \
 -d '{"event":"1210","tag":"tlp:amber"}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.223/events/removeTag
```

```
curl \
 -d '{"event":"1210","tag":"987"}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.223/events/removeTag
```

36.7.8 GET /events/pushEventToZMQ/

36.7.8.1 Description

Will push an Event to ZMQ

36.7.8.2 URL Arguments

- event_id

36.7.8.3 Example

```
curl --header "Authorization: YOUR API KEY" --header "Accept: application/json" --
header "Content-Type: application/json" https://<misp url>events/pushEventToZMQ/223
```

36.7.9 GET /events/nids NIDS rules export

Automatic export of all network related attributes is available under the Snort or Suricata rule format. Only published events and attributes marked as IDS Signature are exported.

You can configure your tools to automatically download the following file:

```
https://<misp url>/events/nids/suricata/download
https://<misp url>/events/nids/snort/download
```

The full API syntax is as follows:

```
https://<misp     url>/events/nids/[format]/download/[eventid]/[frame]/[tags]/[from]/
[to]/[last]
```

format The export format, can be "suricata" or "snort" eventid Restrict the download to a single event frame Some commented out explanation framing the data. The reason to disable this would be if you would like to concatenate a list of exports from various select events in order to avoid unnecessary duplication of the comments. tags To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'. You can also chain several tag commands together with the '&&' operator. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead). For example, to include tag1 and tag2 but exclude tag3 you would use:

```
https://<misp url>/events/nids/snort/download/false/false/tag1&&tag2&&!tag3
```

from Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event. to Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event. last Events published within the last x amount of time, where x can be defined in days, hours, minutes (for example 6d or 12h or 30m). This filter will use the published timestamp of the event. The keywords false or null should be used for optional empty parameters in the URL.

An example for a Suricata export for all events excluding those tagged tag1, without all of the commented information at the start of the file would look like this:

```
https://<misp url>/events/nids/suricata/download/null/true/!tag1
```

Administration is able to maintain an allowedlist containing host, domain name and IP numbers to exclude from the NIDS export.

36.7.10 GET /events/hids Hash - HIDS database export

Automatic export of MD5/SHA1 checksums contained in file-related attributes. This list can be used to feed forensic software when searching for suspicious files. Only published events and attributes marked as IDS Signature are exported.

You can configure your tools to automatically download all the MD5 hashes from MISP:

```
https://<misp url>/events/hids/md5/download
```

Or the SHA1 hashes:

```
https://<misp url>/events/hids/sha1/download
```

The API's full format is as follow:

```
https://<misp url>/events/hids/[format]/download/[tags]/[from]/[to]/[last]
```

format The export format, can be "md5" or "sha1" tags To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'. You can also chain several tag commands together with the '&&' operator. Please be aware the colons (:) cannot be used in the tag search. Use

semicolons instead (the search will automatically search for colons instead). For example, to include tag1 and tag2 but exclude tag3 you would use:

```
https://<misp url>/events/hids/md5/download/tag1&&tag2&&!tag3
```

from Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event. to Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event. last Events published within the last x amount of time, where x can be defined in days, hours, minutes (for example 5d or 12h or 30m). This filter will use the published timestamp of the event. The keywords false or null should be used for optional empty parameters in the URL.

For example, to only show sha1 values from events tagged tag1, use:

```
https://<misp url>/events/hids/sha1/download/tag1
```

36.7.11 GET /events/stix STIX export

You can export MISP events in MITRE's STIX format (to read more about STIX). The STIX XML export is currently very slow and can lead to timeouts with larger events or collections of events. The STIX JSON return format does not suffer from this issue.

Usage of the API:

```
https://<misp url>/events/stix/download
```

Search parameters can be passed to the function via URL parameters or by POSTing an xml or json object (depending on the return type). The following parameters can be passed to the STIX export tool: id, withAttachments, tags. Both id and tags can use the && (and) and ! (not) operators to build queries. Using the URL parameters, the syntax is as follows:

```
https://<misp    url>/events/stix/download/[id]/[withAttachments]/[tags]/[from]/[to]/
[last]
```

id The event's ID withAttachments Encode attachments where applicable tags To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'. You can also chain several tag commands together with the '&&' operator. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead). For example, to include tag1 and tag2 but exclude tag3 you would use:

```
https://<misp url>/events/stix/download/false/true/tag1&&tag2&&!tag3
```

from Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event. to Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event. last Events published within the last x amount of time, where x can be defined in days, hours, minutes (for example 5d or 12h or 30m). This filter will use the published timestamp of the event. You can post an XML or JSON object containing additional parameters in the following formats.

If you use JSON query objects:

```
https://<misp url>/events/stix/download.json
```

```
{"request":                        {"id":["!51","!62"],"withAttachment":false,"tags":
["APT1","!OSINT"],"from":false,"to":"2015-02-15"}}
```

If you use XML query objects:

```
https://<misp url>/events/stix/download
```

```
<request><id>!51</id><id>!62</id><withAttachment>false</withAttachment><tags>APT1</
tags><tags>!OSINT</tags><from>false</from><to>2015-02-15</to></request>
```

36.7.11.1 Various ways to narrow down the search results of the STIX export

For example, to retrieve all events tagged "APT1" but excluding events tagged "OSINT" and excluding events #51 and #62 without any attachments: https:///events/stix/download/!51&&!62/false/APT1&&!OSINT/2015-02-15

To export the same events using a POST request use: https:///events/stix/download.json
Together with this JSON object in the POST message:

```
{"request": {"id":["!51","!62"],"tags":["APT1","!OSINT"],"from":"2015-02-15"}}
```

XML is automatically assumed when using the STIX export:

```
https://<misp url>/events/stix/download
```

The same search could be accomplished using the following POSTed XML object (note that ampersands need to be escaped, or alternatively separate id and tag elements can be used):

```
<request><id>!51</id><id>!62</id><tags>APT1</tags><tags>!OSINT</
tags><from>2015-02-15</from></request>
```

36.8 Tag management

36.8.1 POST /tags/add

36.8.1.1 Description

Add a tag on the instance

36.8.1.2 URL Arguments

Mandatory

- name

Optional

- colour : A valid hexadecimal colour, for example #51961a, if not set, a random colour is chosen
- exportable : whether the tag is exported when synchronising with other instances, default true
- hide_tag : if set, the tag will not be selectable, default false
- org_id : if set, only users from this organisation will be able to add the tag to objects
- user_id : if set, only this user will be able to add the tag to objects

36.8.1.3 Example

```
curl \
 -d '{"name":"tag_only_name"}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.223/tags/add
```

```
curl \
 -d '{"name":"tag_doc","colour":"#51961a","exportable":1,"hide_tag":0,"org_id":1}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.223/tags/add
```

36.8.2 POST /tags/attachTagToObject

36.8.2.1 Description

Attaches a tag to an object by a given UUID. Note that adding a tag collection via this endpoint is not possible. Please refer to /events/addTag and /attributes/addTag for that functionality.

This endpoint exists for convenience reasons and might be slightly less performant than /events/addTag and /attributes/addTag.

36.8.2.2 URL Arguments

- UUID
- tag (as id or name)

36.8.2.3 Response

```
{
    "name": "Tag tlp3Awhite(7) successfully attached to Attribute(153).",
    "message": "Tag tlp3Awhite(7) successfully attached to Attribute(153).",
    "url": "\/tags\/attachTagToObject"
}
```

36.8.2.4 Example

```
curl --header "Authorization: YOUR API KEY " --header "Accept: application/
json" --header "Content-Type: application/json" -X POST http://10.50.13.60/tags/
attachTagToObject/5a0d68b3-6da0-4ced-8233-77bb950d210f/tlp3Awhite
```

```
curl --header "Authorization: YOUR API KEY" --header "Accept:
application/json" --header "Content-Type: application/json" -X POST -
d '{"uuid":"5a0d68b3-6da0-4ced-8233-77bb950d210f","tag":"tlp:white"}' https://<misp
url>/tags/attachTagToObject
```

```
curl \
 -d '{"uuid":"e76949e6-5ccb-4483-bef2-0e4cac73d236","tag":"6"}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.223/tags/attachTagToObject
```

36.8.3 POST /tags/removeTagFromObject

36.8.3.1 Description

Removes a tag from an object (attribute or event) with given UUID.

This endpoint exists for convenience reasons and might be slightly less performant than /events/removeTag and /attributes/removeTag.

36.8.3.2 URL Arguments

- UUID
- tag (as id or name)

36.8.3.3 Response

```
{
    "name": "Tag tlp3Awhite(7) successfully removed from Attribute(153).",
    "message": "Tag tlp3Awhite(7) successfully removed from Attribute(153).",
    "url": "\/tags\/removeTagFromObject"
}
```

36.8.3.4 Example

```
curl --header "Authorization: YOUR API KEY " --header "Accept: application/
json" --header "Content-Type: application/json" -X POST http://10.50.13.60/tags/
removeTagFromObject/5a0d68b3-6da0-4ced-8233-77bb950d210f/tlp3Awhite
```

```
curl \
 -d '{"uuid":"85752e06-7644-40c8-8190-f8bbe9e7b2c7","tag":"tlp:white"}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.223/tags/removeTagFromObject
```

```
curl \
 -d '{"uuid":"85752e06-7644-40c8-8190-f8bbe9e7b2c7","tag":7}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.223/tags/removeTagFromObject
```

36.8.4 GET /tags/tagStatistics/

36.8.4.1 Description

Will give an overview of the used attribute tags

36.8.4.2 Output

```
{
    "tags": {
        "Type:OSINT": "1",
        "tlp:white": "1",
        "osint:source-type=\"technical-report\"": "1",
        "misp-galaxy:threat-actor=\"Lazarus Group\"": "1",
        "misp-galaxy:rat=\"FALLCHILL\"": "1"
    },
    "taxonomies": []
}
```

36.8.4.3 Example

```
curl --header "Authorization: YOUR API KEY " --header "Accept: application/json" --
header "Content-Type: application/json" -X GET http://10.50.13.60/tags/tagStatistics/
```

36.9 Attribute management

36.9.1 POST /attributes/add/

Adds an Attribute to an event

36.9.1.1 URL Arguments

- event id

36.9.1.2 Output

36.9.1.3 Example

```
curl --header "Authorization: YOUR API KEY" --header "Accept:
application/json" --header "Content-Type: application/json" -X POST -d
'{"event_id":"3542","value":"1.2.3.4","category":"Network activity","type":"ip-dst"}'
https://<misp url>/attributes/add/3542
```

36.9.2 GET /attributes

Get an attribute

36.9.2.1 URL Arguments

- attribute uuid

36.9.2.2 URL Attributes

36.9.2.3 Output


```
{"Attribute":
{"id":"39","event_id":"1","object_id":"0","object_relation":null,"category":"Payload
installation","type":"md5","to_ids":true,"uuid":"548847db-060c-4275-a0c7-15bb950d210b","timestamp":"1418
samples collected.","deleted":false,"disable_correlation":false,"value":"049436bb90f71cf38549817d9b90e2da
bf07-43ef950d210b"}}
```


36.9.2.4 Example

```
curl --header "Authorization: YOUR API KEY " --header "Accept:
application/json" --header "Content-Type: application/json" http://10.50.13.60/
attributes/548847db-060c-4275-a0c7-15bb950d210b
```

36.9.3 POST /attributes/delete/

36.9.3.1 Description

Delete attributes.

36.9.3.2 URL Arguments

- attribute uuid
- attribute id
- attribute id/1 <– hard delete

36.9.3.3 Output


```
{"message":"Attribute deleted."}
```


36.9.3.4 Example

```
curl -X POST --header "Authorization: YOUR API KEY" --header "Accept: application/json"
--header "Content-Type: application/json" https://<misp url>/attributes/delete/12345
```

```
curl -X POST --header "Authorization: YOUR API KEY " --header "Accept: application/
json" --header "Content-Type: application/json" http://10.50.13.60/attributes/
delete/548847db-060c-4275-a0c7-15bb950d210b
```

Hard delete:

```
curl -X POST --header "Authorization: YOUR API KEY" --header "Accept: application/json"
--header "Content-Type: application/json" https://<misp url>/attributes/delete/12345/1
```

36.9.4 POST /attributes/addTag

Add a tag or a tag collection to an existing attribute.

36.9.4.1 URL Arguments

- attribute (id)
- tag (id, name or collection_[collection_id])

Examples:

```
curl \
 -d '{"attribute":"256919","tag":"tlp:green"}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.223/attributes/addTag
```

```
curl \
 -d '{"attribute":"256919","tag":"987"}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.223/attributes/addTag
```

```
curl \
 -d '{"attribute":"256919","tag":"collection_1"}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.223/attributes/addTag
```

36.9.5 POST /attributes/removeTag

Remove a tag from an existing event. Note that removing a tag collection in one go is not possible.

36.9.5.1 URL Arguments

- attribute (id)
- tag (id, name).

Examples:

```
curl \
 -d '{"event":"1210","tag":"tlp:amber"}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.223/events/removeTag
```

```
curl \
 -d '{"event":"1210","tag":"987"}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.223/events/removeTag
```

36.9.6 GET /attributes/attributeStatistics

36.9.6.1 Description

Will give an overview of the used attribute types

36.9.6.2 Output

```
{
    "attachment": "1",
    "comment": "1",
    "filename": "2",
    "float": "2",
    "ip-dst": "90",
    "ip-dst|port": "3",
    "link": "3",
    "md5": "16",
    "port": "3",
    "sha1": "2",
    "sha256": "2",
    "size-in-bytes": "1",
    "ssdeep": "2"
}
```

36.9.6.3 Example

```
curl --header "Authorization: YOUR API KEY " --header "Accept: application/
json" --header "Content-Type: application/json" -X GET http://10.50.13.60/attributes/
attributeStatistics/
```

36.9.7 GET /attributes/describeTypes Describe types API

MISP can procedurally describe all attribute types and attribute categories it currently supports including the category - type mappings. To access this information simply send a GET request to:

36.9.7.1 Example

```
https://<misp url>/attributes/describeTypes
```

Depending on the headers passed the returned data will be a JSON object or an XML, with 3 main sections: types, categories, category_type_mappings.

36.9.8 POST /attributes/restSearch

Do not use this function with GET!

36.9.8.1 Parameters

- **returnFormat**: The format to return data in. Allowed formats:
  - **attack-sightings**: Returns ATTA&CK Sightings in json format for attributes with mitre-attack-pattern galaxies attached. For further details on the ATT&CK Sightings, please visit the related MITRE website page.
  - **cache**: Hashes the attributes and returns them as txt. A hashing algorithm can be chosen by also adding the hash_type parameter. Supported hashing algorithms can be found on the PHP website.
  - **count**: Returns the attribute count as txt.
  - **csv**
  - **hashes**: Returns hash attributes in txt format. For composite attributes, only the hash part is returned.
  - **json**
  - **netfilter**: Returns netfilter rules for IPs. Action can be set with the netfilter_action parameter. The default action is DROP.

  - **opendata**: Please refer to the related MISP project blog post.
  - **openioc**
  - **rpz**
  - **snort**
  - **suricata**
  - **text**: Returns only the attribute values in text format.
  - **xml**
  - **yara**:
  - **yara-json**
- **value**: Search for the given value in the attributes' value field.
- **type**: The attribute type, any valid MISP attribute type is accepted.
- **category**: The attribute category, any valid MISP attribute category is accepted.
- **org**: Search by the creator organisation by supplying the organisation identifier.
- **tags**: Include or exclude attributes with certain tags. See example below. It is strongly recommended to specifically exclude the tags you want to avoid, even if the tags should be exclusive, for example tlp:red and tlp:green.

```
{
    "returnFormat": "json",
    "tags": {
        "NOT": [
            "tlp:red"
        ],
        "OR": [
            "tlp:green"
        ]
    }
}
```

- **from**: Will return attributes from events with the date set to a date after the one specified in the from field (format: 2015-02-15).
- **to**: Will return attributes with the date set to a date before the one specified in the to field (format: 2015-02-15).
- **last**: ***Deprecated!!!*** (synonym for publish_timestamp) Restrict the results by the timestamp of the last publication of the event. Any attribute with a last publication timestamp newer than the given timestamp will be returned. The input can be a timestamp or a short-hand time description (7d or 24h for example). You can also pass a list with two values to set a time range (for example ["14d", "7d"]).
- **eventid**: The events that should be included / excluded from the search.
- **withAttachments**: If set, encodes the attachments / zipped malware samples as base64 in the data field within each attribute
- **uuid**: Restrict the results by uuid.
- **publish_timestamp**: Restrict the results by the timestamp of the last publication of the event. Any attribute with a last publication timestamp newer than the given timestamp will be returned. The input can be a timestamp or a short-hand time description (7d or 24h for example). You can also pass a list with two values to set a time range (for example ["14d", "7d"]).
- **published**: Set whether published or unpublished events should be returned. Do not set the parameter if you want both.
- **timestamp**: ***Deprecated!!!*** (synonym for attribute_timestamp) Restrict the results by the timestamp (last edit). Any attribute with a timestamp newer than the given timestamp will be returned. The input can be a timestamp or a short-hand time description (7d or 24h for example). You can also pass a list with two values to set a time range (for example ["14d", "7d"]).
- **enforceWarninglist**: Remove any attributes from the result that would cause a hit on a warninglist entry.
- **to_ids**: By default (0) all attributes are returned that match the other filter parameters, regardless of their to_ids setting. To restrict the returned data set to to_ids only attributes set this parameter to 1. You can only use the special "exclude" setting to only return attributes that have the to_ids flag disabled.
- **deleted**: Default value 0. If set to 1, only deleted attributes will be returned. If set to [0,1] , both deleted and non-deleted attributes wil be returned.

- **includeEventUuid**: Instead of just including the event ID, also include the event UUID in each of the attributes.
- **event_timestamp**: Only return attributes from events that have received a modification after the given timestamp. The input can be a timestamp or a short-hand time description (7d or 24h for example). You can also pass a list with two values to set a time range (for example ["14d", "7d"]).
- **threat_level_id**: Only return attributes of events with the given threat level id(s). 1 = High, 2 = Medium, 3=Low, 4 = Undefined. See example below.

```
{
    "returnFormat": "json",
    "threat_level_id": [1,2]
}
```

- **includeEventTags**: If set to 1, the event tags of the event the attribute belongs to will be added to the attribute.
- **limit**: Limit the number of results returned, for example 10 attributes.
- **page**: If a limit is set, sets the page to be returned. page 3, limit 100 will return records 201->300).
- **requested_attributes**: Only for CSV export. Choose the fields you want in the csv output. Available fields are (*non-exhaustive list, more fields can be available depending on the values of other parameters*):
  - uuid
  - event_id
  - category
  - type
  - value
  - comment
  - to_ids
  - date
  - object_relation
  - attribute_tag
  - object_uuid
  - object_name
  - object_meta_category
  - event_info. Only available if includeContext parameter is set to 1.
  - event_member_org. Only available if includeContext parameter is set to 1.
  - event_source_org. Only available if includeContext parameter is set to 1.
  - event_distribution. Only available if includeContext parameter is set to 1.
  - event_threat_level_id. Only available if includeContext parameter is set to 1.
  - event_analysis. Only available if includeContext parameter is set to 1.
  - event_date. Only available if includeContext parameter is set to 1.
  - event_tag. Only available if includeContext parameter is set to 1.
  - event_timestamp. Only available if includeContext parameter is set to 1.
- **includeContext**: Adds extra event level context to the output. For each attribute more details are added to the Event object in the output. Please note that this significantly bloats the output data. Example below.

```
"Event": {
          "id": "31",
          "orgc_id": "1",
          "org_id": "1",
          "date": "2021-03-11",
          "threat_level_id": "1",
          "info": "Correlation 2",
          "published": true,
          "uuid": "0bfe7bf3-f793-4761-8450-8b30ca9d9964",
          "analysis": "0",
          "timestamp": "1616972381",
          "distribution": "1",
          "publish_timestamp": "1616972392",
          "sharing_group_id": "0",
```

```
          "extends_uuid": "",
          "Tag": [],
          "Orgc": {
            "id": "1",
            "name": "SHARINGORG",
            "uuid": "26867ddf-5a9b-4af0-b552-e4020a913b95",
            "local": true
          }
        }
```


- **headerless**: Only for CSV export. The CSV created when this setting is set to true will not contain the header row.
- **includeWarninglistHits**: Adds a warnings block to an attribute if it has warninglist hits. See example below.


```
"warnings": [
          {
            "match": "10.0.0.0/8",
            "value": "10.0.0.1",
            "warninglist_name": "List of RFC 5735 CIDR blocks",
            "warninglist_id": "46"
          },
          {
            "match": "10.0.0.0/8",
            "value": "10.0.0.1",
            "warninglist_name": "List of RFC 1918 CIDR blocks",
            "warninglist_id": "44"
          }
        ]
```


- **object_relation**: Search on the object_relation field of attributes. You can search for 'malware-sample' attributes of file objects for example. Searching for multiple values at the same time is possible as well.


```
{
    "returnFormat": "json",
    "object_relation": ["malware-sample", "institution-name"]
}
```


- **includeSightings**: Adds a list of sightings for attributes that have sightings. See example below.


```
"Sighting": [
          {
            "id": "1",
            "attribute_id": "29",
            "event_id": "31",
            "org_id": "1",
            "date_sighting": "1617017091",
            "uuid": "48d21518-6b2a-4615-8c4e-91fbe4f08fe7",
            "source": "",
            "type": "0",
            "attribute_uuid": "b3c25257-7f47-41af-a29b-89188e583b5c",
            "Organisation": {
              "id": "1",
              "uuid": "26867ddf-5a9b-4af0-b552-e4020a913b95",
              "name": "SHARINGORG"
            }
          }
        ]
```

- **includeCorrelations**: Adds a list of correlated attributes for attributes that have correlations. See example below.


```
"RelatedAttribute": [
          {
            "id": "31",
            "event_id": "30",
            "object_id": "0",
            "object_relation": null,
            "category": "Network activity",
            "type": "ip-dst",
            "uuid": "f3b54c94-89ff-4fcf-9f47-52f70c6540b8",
            "timestamp": "1616961683",
            "distribution": "5",
            "sharing_group_id": "0",
            "to_ids": false,
            "comment": "",
            "value": "10.0.0.1",
            "Event": {
              "id": "30",
              "uuid": "8cca9f2f-9281-49fd-9b30-e16a8dbf6855",
              "threat_level_id": "1",
              "analysis": "0",
              "info": "Correlation 1",
              "extends_uuid": "",
              "distribution": "1",
              "sharing_group_id": "0",
              "published": false,
              "date": "2021-03-11",
              "orgc_id": "1",
              "org_id": "1"
            }
          }
        ]
```


- **includeDecayScore**: If set to 1, decay score information will be included for attributes that are affected by decaying. See example below. Note that includeEventTags will be set to 1 automatically if includeDecayScore is true.


```
"decay_score": [
          {
            "score": 77.40239901751683,
            "base_score": 80,
            "decayed": false,
            "DecayingModel": {
              "id": "2",
              "name": "NIDS Simple Decaying Model"
            }
          }
        ],
```


- **decayingModel**: Allows you to set the decaying model(s) to use to calculate the decay score. You can use a model that is not enabled. The value should be set to the id of the model. If this value is not set, a decay score entry will be added for all enabled decaying models that apply to the attribute type.
- **excludeDecayed**: Filter out all expired IOCs. Note that includeDecayScore will be set to 1 automatically if excludeDecayed is true.
- **modelOverrides**: JSON that can be used to modify Model parameters on-the-fly. Example can be found beow.

```
{
    "type": "ip-src",
    "tags": ["tlp:%","phishing:%"],
    "includeDecayScore": 1,
    "excludeDecayed": 1,
    "modelOverrides": {
        "threshold": 30
    }
    "decayingModel": [84, 12],
}
```

- **includeFullModel**: If set to 1, includes the full decaying model details instead of just the id and name.
- **score**: Overrides the model threshold value with the one you set. This means attributes for which the decay score calculated for all relevant models is lower than this value, will be considered decayed.
- **attribute_timestamp**: Restrict the results by the timestamp (last edit). Any attribute with a timestamp newer than the given timestamp will be returned. The input can be a timestamp or a short-hand time description (7d or 24h for example). You can also pass a list with two values to set a time range (for example [“14d”, “7d”]).
- **first_seen**: Restrict the results by the first_seen timestamp of the attribute. Any attribute with a first_seen timestamp newer than the given timestamp will be returned. The input can be a timestamp or a short-hand time description (7d or 24h for example). You can also pass a list with two values to set a time range (for example [“14d”, “7d”]).
- **last_seen**: Restrict the results by the last_seen timestamp of the attribute. Any attribute with a first_seen timestamp newer than the given timestamp will be returned. The input can be a timestamp or a short-hand time description (7d or 24h for example). You can also pass a list with two values to set a time range (for example [“14d”, “7d”]).
- **searchall**: Search for a full or a substring (delimited by % for substrings) in the attribute tags, attribute values or attribute comment fields.

36.9.8.2 Example

```
curl \
 -d '{"returnFormat":"json","value":"foobar"}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.220/attributes/restSearch
```


```
{"response": {"Attribute": [{"id":"44","event_id":"30","object_id":"0","object_relation":null,"category":
{"org_id":"1","distribution":"1","id":"30","info":"Correlation
1","orgc_id":"1","uuid":"8cca9f2f-9281-49fd-9b30-e16a8dbf6855"}}]}}
```


36.10 Objects management

36.10.1 POST /objects/delete/[object_id]/[hard_delete]

36.10.1.1 Description

Delete an object.

36.10.1.2 URL Arguments

- object_id : mandatory
- hard_delete : if set to 1 or true, the object will be hard deleted. Defaults to false.

36.10.1.3 Example

```
curl \
 -d '{}' \
 -H "Authorization: YOUR API KEY" \
 -H "Accept: application/json" \
 -H "Content-type: application/json" \
 -X POST https://192.168.0.223/objects/delete/18732/1
```

36.11 Server management

36.11.1 GET /servers/getVersion

Returns the MISP version and the calling key's capabilities. The response also carries the **recommended PyMISP version** for this instance (`pymisp_recommended_version`), which supersedes the deprecated `/servers/getPyMISPVersion` endpoint.

36.11.1.1 Result

```
{
  "version": "2.5.42",
  "pymisp_recommended_version": "2.5.34.1",
  "perm_sync": true,
  "perm_sighting": true,
  "perm_galaxy_editor": true,
  "perm_analyst_data": true,
  "uuid": "5bfba94a-d0a4-4ad0-8d23-73d4011fb691",
  "request_encoding": ["gzip"],
  "filter_sightings": true,
  "collection_sync": true
}
```

36.11.1.2 Example

```
curl --header "Authorization: YOUR API KEY" --header "Accept: application/json"
https://<misp url>/servers/getVersion
```

36.12 Sightings

36.12.1 POST /sightings/add/

- attribute_id
- attribute_uuid

The different sightings types are:

```
0 => 'sighting',
1 => 'false-positive',
2 => 'expiration'
```

36.13 User management

MISP allows administrators to create and manage users via its REST API

```
https://<misp url>/admin/users/view/[user id]
```

36.13.1 POST /admin/users/add

To create a new user, send a POST request to:
####Sample input

```
{
    "email":"andras.iklody@circl.lu",
    "org\_id":1,
    "role\_id":1
}
```

To view the mandatory and optional fields, use a GET request on the above URL.
####Sample output

```
{
    "name": "\/admin\/users\/add API description",
   "description": "POST a User object in JSON format to this API to create a new user.",
    "mandatory_fields": [
        "email",
        "org_id",
        "role_id"
    ],
    "optional_fields": [
        "password",
        "external_auth_required",
        "external_auth_key",
        "enable_password",
        "nids_sid",
        "server_id",
        "gpgkey",
        "certif_public",
        "autoalert",
        "contactalert",
        "disabled",
        "change_pw",
        "termsaccepted",
        "newsread"
    ],
    "url": "\/admin\/users\/add"
}
```

36.13.2 POST admin/users/edit/

To edit an existing user send a POST request to:

```
https://<misp url>/admin/users/edit/[user id]
```

Only the fields POSTed will be updated, the rest is left intact. To view all possible parameters, simply send a GET request to the above URL.

36.13.3 POST admin/users/delete/

You can also delete users by POSTing to the below URL, but keep in mind that disabling users (by setting the disabled flag via an edit) is always preferred to keep user associations to events intact.

36.13.3.1 Parameters

- [user id]

36.13.3.2 Example

```
https://<misp url>/admin/users/delete/[user id]
```

36.13.4 GET admin/users

36.13.4.1 Description

Will output all users

36.13.4.2 Output


```
[
    {
        "User": {
            "id": "1",
            "password": "*****",
            "org_id": "1",
            "server_id": "0",
            "email": "admin@admin.test",
            "autoalert": false,
            "authkey": "*****",
            "invited_by": "0",
            "gpgkey": null,
            "certif_public": "",
            "nids_sid": "4000000",
            "termsaccepted": true,
            "newsread": "0",
            "role_id": "1",
            "change_pw": "0",
            "contactalert": false,
            "disabled": false,
            "expiration": null,
            "current_login": "1515752313",
            "last_login": "1515748671",
            "force_logout": false,
            "date_created": null,
            "date_modified": null,
            "org_ci": "ORGNAME"
        },
        "Role": {
            "id": "1",
            "name": "admin",
            "perm_auth": true
        },
        "Organisation": {
            "id": "1",
            "name": "ORGNAME"
        }
    }
]
```


36.13.4.3 Example

```
curl --header "Authorization: YOUR API KEY " --header "Accept: application/json" --
header "Content-Type: application/json" -X GET http://10.50.13.60/admin/users
```

36.13.5 GET admin/users/view/

36.13.5.1 Description

Will return a single user. To view a user simply send a GET request.

36.13.5.2 Parameters

- id

36.13.5.3 Output


```
{
    "User": {
        "id": "1",
        "password": "*****",
        "org_id": "1",
        "server_id": "0",
        "email": "admin@admin.test",
        "autoalert": false,
        "authkey": "YOUR API KEY",
        "invited_by": "0",
        "gpgkey": null,
        "certif_public": "",
        "nids_sid": "4000000",
        "termsaccepted": true,
        "newsread": "0",
        "role_id": "1",
        "change_pw": "0",
        "contactalert": false,
        "disabled": false,
        "expiration": null,
        "current_login": "1515752313",
        "last_login": "1515748671",
        "force_logout": false,
        "orgAdmins": []
    }
}
```


36.13.5.4 Example

```
curl --header "Authorization: YOUR API KEY " --header "Accept: application/json" --
header "Content-Type: application/json" -X GET http://10.50.13.60/admin/users/view/1
```

36.13.6 POST admin/users/add/

36.14 Discussion API

If you would like to fetch a discussion thread including all of its posts, simply send a GET request to:

```
https://<misp url>/threads/view/<thread id>
```

Using the following headers:

```
Authorization: [Your auth key]
Content-type: application/json
Accept: application/json
```

To get all posts related to an event simply send a GET request to:

```
https://<misp url>/threads/viewEvent/<event id>
```

36.15 Organisation management

MISP allows administrators to create and manage organisations via its REST API
The API is available in JSON format so make sure you use the following headers:

```
Authorization: [Your auth key]
Content-type: application/json
Accept: application/json
```

To fetch all organisations send a GET request to:

```
https://<misp url>/organisations
```

To view an individual organisation, send a get request to:

```
https://<misp url>/organisations/view/id
```

The management of users happens via three apis:

```
https://<misp url>/admin/organisations/add
https://<misp url>/admin/organisations/edit/[org id]
https://<misp url>/admin/organisations/delete/[org id]
```

To delete an organisation simply send a POST or DELETE request to the above URL.

For creating or modifying an organisation, simply POST a JSON containing the relevant fields to the appropriate API. The only mandatory field is the organisation name, with a host of optional parameters

An example for a simple organisation object:

```
{
    "name": "Blizzard",
    "nationality": "US"
}
```

Not setting a field will assume the default settings for the given field in the case of a new organisation whilst it would retain the existing setting for existing organisations. The above example would create the following object in MISP:

```
{
    "Organisation": {
        "id": "1108",
        "name": "Blizzard",
        "alias": "",
        "anonymise": false,
        "date_created": "2017-01-22 17:32:29",
        "date_modified": "2017-01-22 17:32:29",
        "description": "",
        "type": "",
        "nationality": "US",
        "sector": "",
        "created_by": "1",
        "uuid": "5884de9d-04f0-4d7d-bf15-0b88c0a83865",
        "contacts": "",
        "local": true,
        "landingpage": ""
    }
}
```

To query the add or edit APIs for the valid parameters, simply send a GET requests to either API. The result currently looks like this (which might change when new fields are added):

```
{
    "name": "\/admin\/organisations\/add API description",
    "description": "POST an Organisation object in JSON format to this API to create
a new organisation.",
    "mandatory_fields": [
        "name"
    ],
    "optional_fields": [
```

```
        "anonymise",
        "description",
        "type",
        "nationality",
        "sector",
        "uuid",
        "contacts",
        "local"
    ],
    "url": "\/admin\/organisations\/add"
}
```

36.16 Special Cases

> ⚠ Warning
>
> The positional-URL export endpoints described in this section (for example `/events/xml/download/...`, `/events/csv/...`, `/events/text/...`, `/events/rpz/...` and `restSearch` with parameters baked into the URL path) are **legacy**. They still resolve for backward compatibility, but the canonical way to export data is a POST to `/events/restSearch` or `/attributes/restSearch` with a JSON body and a `returnFormat` (see Search). Prefer restSearch for new integrations.

36.16.1 XML Export

An automatic export of all events and attributes (except file attachments) is available under a custom XML format.

You can configure your tools to automatically download the following file:

```
https://<misp url>/events/xml/download
```

If you only want to fetch a specific event append the eventid number:

```
https://<misp url>/events/xml/download/1
```

You can post an XML or JSON object containing additional parameters in the JSON query format or XML query format. Query parameters provide a way to filter the output to specific parameters.

36.16.1.1 JSON query format

The URL is appended with json:

```
https://<misp url>/events/xml/download.json
```

The query parameters can be the following:

```
{"request":         {"eventid":["!51","!62"],"withAttachment":false,"tags":["APT1","!
OSINT"],"from":false,"to":"2015-02-15"}}
```

36.16.1.2 XML query format

The URL is path is:

```
https://<misp url>/events/xml/download
```

The query parameters can be the following:

```
<request><eventid>!
51</eventid><eventid>!62</eventid><withAttachment>false</withAttachment><tags>APT1</
tags><tags>!OSINT</tags><from>false</from><to>2015-02-15</to></request>
```

36.16.1.3 XML download and URL parameters

The XML download also accepts two additional the following optional parameters in the url:

```
https://<misp     url>/events/xml/download/[eventid]/[withattachments]/[tags]/[from]/
[to]/[last]
```

eventid Restrict the download to a single event withattachments A boolean field that determines whether attachments should be encoded and a second parameter that controls the eligible tags. tags To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'. You can also chain several tag commands together with the '&&' operator. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead). For example, to include tag1 and tag2 but exclude tag3 you would use:

```
https://<misp url>/events/xml/download/false/true/tag1&&tag2&&!tag3
```

from Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event. to Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event. last Events published within the last x amount of time, where x can be defined in days, hours, minutes (for example 5d or 12h or 30m). This filter will use the published timestamp of the event. The keywords false or null should be used for optional empty parameters in the URL. Also check out the User Guide to read about the REST API.

36.16.2 CSV export

An automatic export of attributes is available as CSV. Only attributes that are flagged "to_ids" will get exported.

You can configure your tools to automatically download the following file:

```
https://<misp url>/events/csv/download
```

This will download all the valid attributes in your MISP instance (might take some time).

You can also configure your tools to download the attributes from a specific event. Here is the old legacy CSV export that will work like exporting all attributes:

```
https://<misp url>/events/csv/download/<event-id>
```

You can specify additional flags for CSV exports as follows:

POST to: https:///events/csv/download

Headers: Authorization: Content-type: application/json

Body: ~~json {"parameter1":"value1", "parameter2":1, "parameter3":["value3", "value4", "!value5"]}~~

eventid Restrict the download to a single event ignore Setting this flag to true will include attributes that are not marked "to_ids". tags Simply add a list of tags that should be included or negated (by prepending the tag name with a "!"). Any event with a negated tag will be ignored, even if an included tag is matching. An example is included further down. category The attribute category, any valid MISP attribute category is accepted. type The attribute type, any valid MISP attribute type is accepted. includeContext Include the event data with each attribute. from Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event. to Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event. last Events published within the last x amount of time, where x can be defined in days, hours, minutes (for example 5d or 12h or 30m). This filter will use the published timestamp of the event. For example, to only download a csv generated of the "domain" type and the "Network activity" category attributes all events except for the one and further restricting it to events that are tagged "tag1" or "tag2" but not "tag3", only allowing attributes that are IDS flagged use the following syntax:

POST to: https:///events/csv/download

Headers: Authorization: Content-type: application/json

Body: ~~json {"tags":["tag1", "tag2", "!tag3"], "category":"Network activity", "type": "domain"}~~

Alternatively you can fall back to the deprecated syntax of passing parameters in a GET request via the URL, however this is discouraged: https:///events/csv/download/[eventid]/[ignore]/[tags]/[category]/[type]/[includeContext]/[from]/[to]/[last] If you use the deprecated URL parameter method, keep in mind that the keywords false or null should be used for optional empty parameters. To export the attributes of all events that are of the type "domain", use the following syntax: https:///events/csv/download/false/false/false/false/domain

36.16.2.1 Update 2.4.82

Since version 2.4.82, the new export format allows to select more columns using the following query format:

```
https://<misp-instance>/events/csv/download/<event-
id>?attributes=timestamp,type,uuid,value
```

The order of columns will be honoured including those related to object level information.

To select object level columns, simply prepend the given object column's name by object_, such as:

```
https://<misp-instance>/events/csv/download/<event-
id>?attributes=timestamp,type,uuid,value&object_attributes=uuid,name
```

The following columns will be returned (all columns related to objects will be prefixed with object_):

```
timestamp,type,uuid,value,object_uuid,object_name
```

includeContext option includes the tags for the event for each line.

36.17 RPZ export

You can export RPZ zone files for DNS level firewall by using the RPZ export functionality of MISP. The file generated will include all of the IDS flagged domain, hostname and IP-src/IP-dst attribute values that you have access to.

It is possible to further restrict the exported values using the following filters:

tags To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'. You can also chain several tag commands together with the '&&' operator. Please be aware the colons (:) cannot be used in the tag search when passed through the URL. Use semicolons instead (the search will automatically search for colons instead). id The event's ID from Events with the date set to a date after the one specified in the from field (format: 2015-02-03) to Events with the date set to a date before the one specified in the to field (format: 2015-02-03) MISP will inject header values into the zone file as well as define the action taken for each of the values that can all be overwritten. By default these values are either the default values shipped with the application, or ones that are overwritten by your site administrator. The values are as follows:

Value name	Default value
RPZ_policy	DROP
RPZ_walled_garden	127.0.0.1
RPZ_serial	$date00
RPZ_refresh	2h
RPZ_retry	30m
RPZ_expiry	30d
RPZ_minimum_ttl	1h
RPZ_ttl	1w
RPZ_ns	localhost.
RPZ_ns_alt	
RPZ_email	root.localhost

To override the above values, either use the URL parameters as described below:

```
https://<misp     url>/attributes/rpz/download/[tags]/[eventId]/[from]/[to]/[policy]/
[walled_garden]/[ns]/[ns_alt]/[email]/[serial]/[refresh]/[retry]/[expiry]/[minim
um_ttl]/[ttl]
```

Or POST an XML or JSON object with the above listed options:

```
<request><tags>OSINT&&!OUTDATED</tags><policy>walled-garden</
policy><walled_garden>teamliquid.net</walled_garden><refresh>5h</refresh></request>
```

```
{"request": {"tags": ["OSINT", "!OUTDATED"], "policy": "walled-garden",
"walled_garden": "teamliquid.net", "refresh": "5h"}
```

36.18 Text export

An export of all attributes of a specific type to a plain text file. By default only published and IDS flagged attributes are exported.

You can configure your tools to automatically download the following files:

```
https://<misp url>/attributes/text/download/md5
https://<misp url>/attributes/text/download/sha1
https://<misp url>/attributes/text/download/sha256
https://<misp url>/attributes/text/download/filename
https://<misp url>/attributes/text/download/filename|md5
https://<misp url>/attributes/text/download/filename|sha1
https://<misp url>/attributes/text/download/filename|sha256
https://<misp url>/attributes/text/download/ip-src
https://<misp url>/attributes/text/download/ip-dst
https://<misp url>/attributes/text/download/hostname
https://<misp url>/attributes/text/download/domain
https://<misp url>/attributes/text/download/email-src
https://<misp url>/attributes/text/download/email-dst
https://<misp url>/attributes/text/download/email-subject
https://<misp url>/attributes/text/download/email-attachment
https://<misp url>/attributes/text/download/url
https://<misp url>/attributes/text/download/http-method
https://<misp url>/attributes/text/download/user-agent
https://<misp url>/attributes/text/download/regkey
https://<misp url>/attributes/text/download/regkey|value
https://<misp url>/attributes/text/download/AS
https://<misp url>/attributes/text/download/snort
https://<misp url>/attributes/text/download/pattern-in-file
https://<misp url>/attributes/text/download/pattern-in-traffic
https://<misp url>/attributes/text/download/pattern-in-memory
https://<misp url>/attributes/text/download/yara
https://<misp url>/attributes/text/download/vulnerability
https://<misp url>/attributes/text/download/attachment
https://<misp url>/attributes/text/download/malware-sample
https://<misp url>/attributes/text/download/link
https://<misp url>/attributes/text/download/comment
https://<misp url>/attributes/text/download/text
https://<misp url>/attributes/text/download/other
https://<misp url>/attributes/text/download/named pipe
https://<misp url>/attributes/text/download/mutex
https://<misp url>/attributes/text/download/target-user
https://<misp url>/attributes/text/download/target-email
https://<misp url>/attributes/text/download/target-machine
https://<misp url>/attributes/text/download/target-org
https://<misp url>/attributes/text/download/target-location
https://<misp url>/attributes/text/download/target-external
```

To restrict the results by tags, use the usual syntax. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead). To get ip-src values from events tagged tag1 but not tag2 use:

```
https://<misp url>/attributes/text/download/ip-src/tag1&&
```

It is possible to restrict the text exports on additional flags. The first allows the user to restrict based on event ID, whilst the second is a boolean switch allowing non IDS flagged attributes to be exported. Additionally, choosing "all" in the type field will return all eligible attributes.

```
https://<misp   url>/attributes/text/download/[type]/[tags]/[event_id]/[allowNonIDS]/
[from]/[to]/[last]
```

type The attribute type, any valid MISP attribute type is accepted. tags To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'. You can also chain several tag commands together with the '&&' operator. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead). allowNonIDS Include attributes that would normally be excluded due to the IDS flag not being set or due to being allowlisted from Set the lowest "date" field value that should be included in the export (format YYYY-MM-DD) to Set the highest "date" field value that should be included in the export (format YYYY-MM-DD) last Set the timeframe of the export based on the "timestamp" value. The parameter uses a time + metric notation (valid examples: "2w", "60m", "24h") For example, to include tag1 and tag2 but exclude tag3 you would use:

```
https://<misp url>/attributes/text/download/all/tag1&&tag2&&!tag3
```

event_id Restrict the results to the given event IDs. allowNonIDS Allow attributes to be exported that are not marked as "to_ids". from Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event. to Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event. last Events published within the last x amount of time, where x can be defined in days, hours, minutes (for example 5d or 12h or 30m). This filter will use the published timestamp of the event. The keywords false or null should be used for optional empty parameters in the URL.

For example, to retrieve all attributes for event #5, including non IDS marked attributes too, use the following line:

```
https://<misp url>/attributes/text/download/all/null/5/true
```

36.19 RESTful searches with XML result export

It is possible to search the database for attributes based on a list of criteria.

To return an event with all of its attributes, relations, shadowAttributes, use the following syntax:

```
https://<misp  url>/events/restSearch/download/[value]/[type]/[category]/[org]/[tag]/
[quickfilter]/[from]/[to]/[last]/[eventid]/[withAttachments]/[metadata]/[uuid]
```

value Search for the given value in the attributes' value field. type The attribute type, any valid MISP attribute type is accepted. category The attribute category, any valid MISP attribute category is accepted. org Search by the creator organisation by supplying the organisation idenfitier. tags To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'. You can also chain several tag commands together with the '&&' operator. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead). For example, to include tag1 and tag2 but exclude tag3 you would use:

```
https://<misp url>/events/restSearch/download/null/null/null/null/tag1&&tag2&&!tag3
```

quickfilter Enabling this (by passing "1" as the argument) will make the search ignore all of the other arguments, except for the auth key and value. MISP will return an xml / json (depending on the header sent) of all events that have a sub-string match on value in the event info, event orgc, or any of the attribute value1 / value2 fields, or in the attribute comment. from Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event. to Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event. last Events published within the last x amount of time, where x can be defined in days, hours, minutes (for example 5d or 12h or 30m). This filter will use the published timestamp of the event. eventid The events that should be included / excluded from the search withAttachments Include the attachments/encrypted samples in the export metadata Only fetch the event metadata (event data, tags, relations) and skip the attributes limit Limit the number of results returned; use together with page. page If a limit is set, sets the page to be returned, starting at 1; page 3, limit 100 will return records 201-

>300). When requesting a page beyond the number of available pages, the returned results list will be empty. The keywords false or null should be used for optional empty parameters in the URL.

For example, to find any event with the term "red october" mentioned, use the following syntax (the example is shown as a POST request instead of a GET, which is highly recommended):

POST to:

```
https://<misp url>/events/restSearch/download
```

POST message payload (XML):

```
  <request><value>red october</value><searchall>1</searchall><eventid>!15</eventid></
request>
```

POST message payload (JSON):

```
{"request": {"value":"red october","searchall":1,"eventid":"!15"}}
```

To just return a list of attributes, use the following syntax:

value Search for the given value in the attributes' value field. type The attribute type, any valid MISP attribute type is accepted. category The attribute category, any valid MISP attribute category is accepted. org Search by the creator organisation by supplying the organisation identifier. tags To include a tag in the results just write its names into this parameter. To exclude a tag prepend it with a '!'. You can also chain several tag commands together with the '&&' operator. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead). from Events with the date set to a date after the one specified in the from field (format: 2015-02-15). This filter will use the date of the event. to Events with the date set to a date before the one specified in the to field (format: 2015-02-15). This filter will use the date of the event. last Events published within the last x amount of time, where x can be defined in days, hours, minutes (for example 5d or 12h or 30m). This filter will use the published timestamp of the event. eventid The events that should be included / excluded from the search. uuid The returned events must include an attribute with the given UUID, or alternatively the event's UUID must match the value(s) passed. The keywords false or null should be used for optional empty parameters in the URL.

```
https://<misp    url>/attributes/restSearch/download/[value]/[type]/[category]/[org]/
[tag]/[from]/[to]/[last]/[eventid]/[withattachments]/[uuid]
```

Value, type, category and org are optional. It is possible to search for several terms in each category by joining them with the '&&' operator. It is also possible to negate a term with the '!' operator. Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead). For example, in order to search for all attributes created by your organisation that contain 192.168 or 127.0 but not 0.1 and are of the type ip-src, excluding the events that were tagged tag1 use the following syntax:

```
https://<misp  url>/attributes/restSearch/download/192.168&&127.0&&!0.1/ip-src/false/
CIRCL/!tag1
```

You can also use search for IP addresses using CIDR. Make sure that you use '|' (pipe) instead of '/' (slashes). Please be aware the colons (:) cannot be used in the tag search. Use semicolons instead (the search will automatically search for colons instead). See below for an example:

```
https://<misp url>/attributes/restSearch/download/192.168.1.1|16/ip-src/null/CIRCL
```

36.20 Export attributes of event with specified type as XML

If you want to export all attributes of a pre-defined type that belong to an event, use the following syntax:

```
https://<misp url>/attributes/returnAttributes/download/[id]/[type]/[sigOnly]
```

sigOnly is an optional flag that will block all attributes from being exported that don't have the IDS flag turned on. It is possible to search for several types with the '&&' operator and to exclude values with the '!' operator. For example, to get all IDS signature attributes of type md5 and sha256, but not filename|md5 and filename|sha256 from event 25, use the following:

```
https://<misp    url>/attributes/returnAttributes/download/25/md5&&sha256&&!filename/
true
```

36.21 Filtering event metadata

As described in the REST section, it is possible to retrieve a list of events along with their metadata by sending a GET request to the /events API. However, this API in particular is a bit more versatile. You can pass search parameters along to search among the events on various fields and retrieve a list of matching events (along with their metadata). Use the following URL:

```
https://<misp url>/events/index
```

POST a JSON object with the desired lookup fields and values to receive a JSON back. An example for a valid lookup:

```
Authorization: <your API key>
Accept: application/json
Content-type: application/json
```

Body:

```
{"searchinfo":"Locky", "searchpublished":1, "searchdistribution":0}
```

The list of valid parameters: searchpublished: Filters on published or unpulished events [0,1] - negatable searchinfo: Filters on strings found in the event info - negatable searchtag: Filters on attached tag names - negatable searcheventid: Filters on specific event IDs - negatable searchthreatlevel: Filters on a given event threat level [1,2,3,4] - negatable searchdistribution: Filters on the distribution level [0,1,2,3] - negatable searchanalysis: Filters on the given analysis phase of the event [0,1,2,3] - negatable searchattribute: Filters on a contained attribute value - negatable searchorg: Filters on the creator organisation - negatable searchemail: Filters on the creator user's email address (admin only) - negatable searchDatefrom: Filters on the date, anything newer than the given date in YYYY-MM-DD format is taken - non-negatable searchDateuntil: Filters on the date, anything older than the given date in YYYY-MM-DD format is taken - non-negatable

36.22 Download attachment or malware sample

If you know the attribute ID of a malware-sample or an attachment, you can download it with the following syntax:

```
https://<misp url>/attributes/downloadAttachment/download/[Attribute_id]
```

36.23 Download malware sample by hash

You can also download samples by knowing its MD5 hash. Simply pass the hash along as a JSON/XML object or in the URL (with the URL having overruling the passed objects) to receive a JSON/XML object back with the zipped sample base64 encoded along with some contextual information.

You can also use this API to get all samples from events that contain the passed hash. For this functionality, just pass the "allSamples" flag along. Note that if you are getting all samples from matching events, you can use all supported hash types (md5, sha1, sha256) for the lookup.

You can also get all the samples from an event with a given event ID, by passing along the eventID parameter. Make sure that either an event ID or a hash is passed along, otherwise an error message will be returned. Also, if no hash is set, the allSamples flag will get set automatically.

https:///attributes/downloadSample/[hash]/[allSamples]/[eventID]

POST message payload (XML):

```
<request><hash>7c12772809c1c0c3deda6103b10fdfa0</hash><allSamples>1</
allSamples><eventID>13</eventID></request>
```

POST message payload (json):

```
{"request": {"hash": "7c12772809c1c0c3deda6103b10fdfa0", "allSamples": 1, "eventID":
13}}
```

A description of all the parameters in the passed object:

hash A hash in MD5 format. If allSamples is set, this can be any one of the following: md5, sha1, sha256. allSamples If set, it will return all samples from events that have a match for the hash provided above. eventID If set, it will only fetch data from the given event ID.

36.24 Upload malware samples using the "Upload Sample" API

```
https://<misp url>/events/upload_sample/[Event_id]
```

This API will allow you to populate an event that you have modify rights to with malware samples (and all related hashes). Alternatively, if you do not supply an event ID, it will create a new event for you.

The files have to be base64 encoded and POSTed as explained below. All samples will be zipped and password protected (with the password being "infected"). The hashes of the original file will be captured as additional attributes.

For sample upload (for objects in general) there is no check for duplicates.

The event ID is optional. MISP will accept either a JSON or an XML object posted to the above URL.

The general structure of the expected objects is as follows:

```
{"request":  {"files":  [{"filename":  filename1,  "data":  base64encodedfile1},
{"filename": filename2, "data": base64encodedfile2}],
   "optional_parameter1", "optional_parameter2", "optional_parameter3"}}
```

JSON:

```
{"request":{"files":  [{"filename":  "test1.txt",  "data":  "dGVzdA=="},  {"filename":
"test2.txt", "data": "dGVzdDI="}], "distribution": 1, "info" : "test", "event_id": 15}}
```

XML:

```
<request><files><filename>test3.txt</filename><data>dGVzdA==</data></
files><files><filename>test4.txt</filename><data>dGVzdDI=</data></files><info>test</
info><distribution>1</distribution><event_id>15</event_id></request>
```

The following optional parameters are expected:

event_id The Event's ID is optional. It can be either supplied via the URL or the POSTed object, but the URL has priority if both are provided. Not supplying an event ID will cause MISP to create a single new event for all of the POSTed malware samples. You can define the default settings for the event, otherwise a set of default settings will be used. distribution The distribution setting used for the attributes and for the newly created event, if relevant. [0-3] to_ids You can flag all attributes created during the transaction to be marked as "to_ids" or not. category The category that will be assigned to the uploaded samples. Valid options are: Payload delivery, Artifacts dropped, Payload Installation, External Analysis. info Used to populate the event info field if no event ID supplied. Alternatively, if not set, MISP will simply generate a message showing that it's a malware sample collection generated on the given day. analysis The analysis level of the newly created event, if applicable. [0-2] threat_level_id: The threat level ID of the newly created event, if applicatble. [0-3] comment This will populate the comment field of any attribute created using this API. The **threat_level_id** is mapped as such:

```
0 = high
1 = medium
2 = low
3 = undefined
```

36.25 Proposals API

You can interact with the proposals via the API directly since version 2.3.148.

HTTP	URL	Explanation	Expected Payload	Response
GET	/shadow_attributes/view/[proposal_id]	View a proposal	N/A	ShadowAttribute object
POST	/shadow_attributes/add/[event_id]	Propose a new attribute to an event	ShadowAttribute object	ShadowAttribute object
POST	/shadow_attributes/edit/[attribute_id]	Propose an edit to an attribute	ShadowAttribute object	ShadowAttribute object
POST	/shadow_attributes/accept/[proposal_id]	Accept a proposal	N/A	Message
POST	/shadow_attributes/discard/[proposal_id]	Discard a proposal	N/A	Message

When posting a shadow attribute object, use the following format
JSON:

```
{"request": {"ShadowAttribute": {"value": "5.5.5.5", "to_ids": false, "type": "ip-
dst", "category": "Network activity"}}}
```

XML:

```
<request><ShadowAttribute><value>5.5.5.5</value><to_ids>0</to_ids><type>ip-src</
type><category>Network activity</category></ShadowAttribute></request>
```

None of the above fields are mandatory, but at least one of them has to be provided.

36.26 Sharing groups

MISP allows sharing groups to be retrieved via the API.

```
https://<misp url>/sharing_groups/index.json
```

Based on the API key used, the list of visible sharing groups will be returned in a JSON file. The JSON includes the organization parts of a given sharing group along with the associated server.

36.27 Enable, disable and fetching feeds via the API

The MISP feeds can be enabled via the API.
A feed can be enabled by POSTing on the following URL (feed_id is the id of the feed):

```
/feeds/enable/feed_id
```

A feed can be disabled by POSTing on the following URL (feed_id is the id of the feed):

```
/feeds/disable/feed_id
```

All feeds can cached via the API:

```
/feeds/cacheFeeds/all
```

or you can replace `all` by the feed format to fetch like `misp` or `freetext`. `all` can be replaced with the `id` value of the feed to fetch a specific feed.
To fetch a feed or all feeds:

```
/feeds/fetchFromFeed/feed_id
/feeds/fetchFromAllFeeds
```

This API can be also used to download feeds at regular interval via cronjobs or alike.

36.28 Sightings API

MISP allows Sightings data to be conveyed in several ways.

The most basic way is to POST a blank message to the Sightings API with the attribute ID or attribute UUID. This will create a sightings entry with the creation of the entry as the timestamp for the organisation of the authenticated user.

```
https://<misp url>/sightings/add/[attribute_id]
https://<misp url>/sightings/add/[attribute_uuid]
```

Alternatively, it is possible to POST a JSON object and gain additional granularity. The following fields are recognised by the API:

id The attribute's ID uuid The attribute's UUID value Will create a sighting for any attribute with the given value or for composite attributes, for the value matching any element of the attribute value values Expects a list, MISP will create sightings for any attribute matching any of the given values or for composite attributes, for any of the values matching any element of the attribute value timestamp Unix timestamp of the sighting, overrides the current time Some examples:

To create a sighting for attribute #9001:

```
{"id":"9001"}
```

To create a sighting for any attribute with the value being teamliquid.net or 173.231.136.216 with the time of sighting being :

```
{"values":["teamliquid.net", "173.231.136.216"], "timestamp":1460558710}
```

It is also possible to POST a STIX indicator with sighting data to the following URL (keep in mind that the content type has to be XML):

```
https://<misp url>/sightings/add/stix
```

MISP will use the sightings related observables to gather all values and create sightings for each attribute that matches any of the values. If no related observables are provided in the Sighting object, then MISP will fall back to the Indicator itself and use its observables' values to create the sightings. The time of the sighting is the current time, unless the timestamp attribute is set on the Sightings object, in which case that is taken.

An example STIX sightings document:

```
<stix:STIX_Package
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:stix="http://stix.mitre.org/stix-1"
    xmlns:indicator="http://stix.mitre.org/Indicator-2"
    xmlns:stixCommon="http://stix.mitre.org/common-1"
    xmlns:cybox="http://cybox.mitre.org/cybox-2"
    xmlns:AddressObject="http://cybox.mitre.org/objects#AddressObject-2"
    xmlns:DomainNameObj="http://cybox.mitre.org/objects#DomainNameObject-1"
    xmlns:cyboxVocabs="http://cybox.mitre.org/default_vocabularies-2"
    xmlns:stixVocabs="http://stix.mitre.org/default_vocabularies-1"
    xmlns:example="http://example.com/"
    xsi:schemaLocation="
    http://stix.mitre.org/stix-1 ../stix_core.xsd
    http://stix.mitre.org/Indicator-2 ../indicator.xsd
          http://cybox.mitre.org/objects#DomainNameObject-1   http://cybox.mitre.org/
XMLSchema/objects/Domain_Name/1.0/Domain_Name_Object.xsd
    http://stix.mitre.org/common-1 http://stix.mitre.org/XMLSchema/common/1.1.1/stix_
common.xsd
                        http://cybox.mitre.org/default_vocabularies-2      ../cybox/
cybox_default_vocabularies.xsd
    http://stix.mitre.org/default_vocabularies-1 ../stix_default_vocabularies.xsd
                  http://cybox.mitre.org/objects#AddressObject-2     ../cybox/objects/
```

```
Address_Object.xsd"
    id="example:STIXPackage-33fe3b22-0201-47cf-85d0-97c02164528d"
    timestamp="2014-05-08T09:00:00.000000Z"
    version="1.1.1"
    >
    <stix:STIX_Header>
        <stix:Title>Example watchlist that contains IP information.</stix:Title>
         <stix:Package_Intent xsi:type="stixVocabs:PackageIntentVocab-1.0">Indicators
- Watchlist</stix:Package_Intent>
    </stix:STIX_Header>
    <stix:Indicators>
                                <stix:Indicator    xsi:type="indicator:IndicatorType"
id="example:Indicator-2e20c5b2-56fa-46cd-9662-8f199c69d2c9"
timestamp="2014-05-08T09:00:00.000000Z">
                 <indicator:Type  xsi:type="stixVocabs:IndicatorTypeVocab-1.1">Domain
Watchlist</indicator:Type>
                             <indicator:Observable   id="example:Observable-87c9a5bb-
d005-4b3e-8081-99f720fad62b">
              <cybox:Object id="example:Object-12c760ba-cd2c-4f5d-a37d-18212eac7928">
                     <cybox:Properties xsi:type="DomainNameObj:DomainNameObjectType"
type="FQDN">
               <DomainNameObj:Value condition="Equals" apply_condition="ANY">malicious1.example.com##comma
DomainNameObj:Value>
                   </cybox:Properties>
              </cybox:Object>
          </indicator:Observable>
          <indicator:Sightings>
            <indicator:Sighting timestamp="2014-05-08T09:00:00.000000Z">
              <indicator:Source>
                <stixCommon:Identity>
                  <stixCommon:Name>FooBar Inc.</stixCommon:Name>
                </stixCommon:Identity>
              </indicator:Source>
              <indicator:Related_Observables>
                <indicator:Related_Observable>
                                                              <stixCommon:Observable
id="example:Observable-45b3acdf-1888-4bcc-89a9-6d9f8116fede">
              <cybox:Object id="example:Object-a3d36250-42fa-4653-9172-87b87598390c">
                     <cybox:Properties xsi:type="DomainNameObj:DomainNameObjectType"
type="FQDN">
                   <DomainNameObj:Value>malicious2.example.com</DomainNameObj:Value>
                     </cybox:Properties>
                   </cybox:Object>
                 </stixCommon:Observable>
                </indicator:Related_Observable>
              </indicator:Related_Observables>
            </indicator:Sighting>
          </indicator:Sightings>
        </stix:Indicator>
    </stix:Indicators>
</stix:STIX_Package>
```

POSTing this as the message's body to MISP will sight any attributes visible to the user with he value "malicious2.example.com". For composite types, a match on a component will also trigger a sighting (so for example for attributes of type domain|ip a domain match would be sufficient).

If no Related observables are set in the Sighting itself, MISP will fall back to the observable directly contained in the indicator. So in the following example:

```
<stix:STIX_Package
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:stix="http://stix.mitre.org/stix-1"
```

```
    xmlns:indicator="http://stix.mitre.org/Indicator-2"
    xmlns:stixCommon="http://stix.mitre.org/common-1"
    xmlns:cybox="http://cybox.mitre.org/cybox-2"
    xmlns:AddressObject="http://cybox.mitre.org/objects#AddressObject-2"
    xmlns:DomainNameObj="http://cybox.mitre.org/objects#DomainNameObject-1"
    xmlns:cyboxVocabs="http://cybox.mitre.org/default_vocabularies-2"
    xmlns:stixVocabs="http://stix.mitre.org/default_vocabularies-1"
    xmlns:example="http://example.com/"
    xsi:schemaLocation="
    http://stix.mitre.org/stix-1 ../stix_core.xsd
    http://stix.mitre.org/Indicator-2 ../indicator.xsd
          http://cybox.mitre.org/objects#DomainNameObject-1   http://cybox.mitre.org/
XMLSchema/objects/Domain_Name/1.0/Domain_Name_Object.xsd
    http://stix.mitre.org/common-1 http://stix.mitre.org/XMLSchema/common/1.1.1/stix_
common.xsd
                        http://cybox.mitre.org/default_vocabularies-2      ../cybox/
cybox_default_vocabularies.xsd
    http://stix.mitre.org/default_vocabularies-1 ../stix_default_vocabularies.xsd
                http://cybox.mitre.org/objects#AddressObject-2    ../cybox/objects/
Address_Object.xsd"
    id="example:STIXPackage-33fe3b22-0201-47cf-85d0-97c02164528d"
    timestamp="2014-05-08T09:00:00.000000Z"
    version="1.1.1"
    >
    <stix:STIX_Header>
        <stix:Title>Example watchlist that contains IP information.</stix:Title>
         <stix:Package_Intent xsi:type="stixVocabs:PackageIntentVocab-1.0">Indicators
- Watchlist</stix:Package_Intent>
    </stix:STIX_Header>
    <stix:Indicators>
                                <stix:Indicator    xsi:type="indicator:IndicatorType"
id="example:Indicator-2e20c5b2-56fa-46cd-9662-8f199c69d2c9"
timestamp="2014-05-08T09:00:00.000000Z">
                <indicator:Type  xsi:type="stixVocabs:IndicatorTypeVocab-1.1">Domain
Watchlist</indicator:Type>
                            <indicator:Observable   id="example:Observable-87c9a5bb-
d005-4b3e-8081-99f720fad62b">
             <cybox:Object id="example:Object-12c760ba-cd2c-4f5d-a37d-18212eac7928">
                    <cybox:Properties xsi:type="DomainNameObj:DomainNameObjectType"
type="FQDN">
              <DomainNameObj:Value condition="Equals" apply_condition="ANY">malicious1.example.com##comma
DomainNameObj:Value>
                  </cybox:Properties>
             </cybox:Object>
          </indicator:Observable>
          <indicator:Sightings>
            <indicator:Sighting timestamp="2014-05-08T09:00:00.000000Z">
              <indicator:Source>
                <stixCommon:Identity>
                  <stixCommon:Name>FooBar Inc.</stixCommon:Name>
                </stixCommon:Identity>
              </indicator:Source>
            </indicator:Sighting>
          </indicator:Sightings>
        </stix:Indicator>
    </stix:Indicators>
</stix:STIX_Package>
```

MISP would create sightings for attributes matching any of the following: malicious1.example.com, malicious2.example.com, malicious3.example.com

36.29 Warninglists API

36.29.1 GET warninglists/index

36.29.1.1 Description

Return the index of warninglists enabled on the MISP instance

36.29.1.2 Parameters

- id

36.29.1.3 Output


```
...
{"Warninglists":[{"Warninglist":{"id":"17","name":"List of known Office 365 URLs and
IP address ranges","type":"string","description":"Office 365 URLs and IP address ranges","version":"2017021
src, ip-dst, domain|ip, hostname"}},{"Warninglist":{"id":"16","name":"List of known
google domains","type":"string","description":"Event contains one or more entries of
known google domains","version":"3","enabled":true,"warninglist_entry_count":"665","valid_attributes":"do
hostname, domain|ip"}},{"Warninglist":{"id":"15","name":"List of hashes for EICAR test
virus","type":"string","description":"Event contains one or more entries based on
hashes for EICAR test virus","version":"1","enabled":true,"warninglist_entry_count":"15","valid_attributes
sha1, sha256, sha512, filename|md5, filename|sha1, filename|sha256,
filename|sha512"}},{"Warninglist":{"id":"14","name":"Top 1000 website from
Alexa","type":"string","description":"Event contains one or more entries from the top
1000 of the most used website (Alexa).","version":"20170212","enabled":true,"warninglist_entry_count":"100
domain"}},{"Warninglist":{"id":"13","name":"TLDs as known by
IANA","type":"string","description":"Event contains one or more TLDs as attribute with
an IDS flag set","version":"2","enabled":true,"warninglist_entry_count":"1290","valid_attributes":"hostnar
domain, domain|ip"}},{"Warninglist":{"id":"12","name":"Second level TLDs as known by
Mozilla Foundation","type":"string","description":"Event contains one or more second
level TLDs as attribute with an IDS flag set","version":"2","enabled":true,"warninglist_entry_count":"6462"
domain, domain|ip"}},{"Warninglist":{"id":"11","name":"List of RFC 5735 CIDR
blocks","type":"cidr","description":"Event contains one or more entries part of the
RFC 5735 CIDR blocks - Special Use IPv4 Addresses","version":"2","enabled":true,"warninglist_entry_count":"
src, ip-dst, domain|ip"}},{"Warninglist":{"id":"10","name":"List of RFC 3849 CIDR
blocks","type":"cidr","description":"Event contains one or more entries part of the
IPv6 documentation prefix (RFC 3849)","version":"2","enabled":true,"warninglist_entry_count":"1","valid_at
src, ip-dst, domain|ip"}},{"Warninglist":{"id":"9","name":"List of RFC 1918 CIDR
blocks","type":"cidr","description":"Event contains one or more entries part of the
RFC 1918 CIDR blocks","version":"2","enabled":true,"warninglist_entry_count":"3","valid_attributes":"ip-
src, ip-dst, domain|ip"}},{"Warninglist":{"id":"8","name":"List of known IPv6 public
DNS resolvers","type":"string","description":"Event contains one or more public IPv6
DNS resolvers as attribute with an IDS flag set","version":"20160803","enabled":true,"warninglist_entry_cou
{"Warninglist":{"id":"7","name":"List of known IPv4 public DNS
resolvers","type":"string","description":"Event contains one or more public IPv4 DNS
resolvers as attribute with an IDS flag set","version":"20160803","enabled":true,"warninglist_entry_count":
{"Warninglist":{"id":"6","name":"List of RFC 5771 multicast CIDR
blocks","type":"cidr","description":"Event contains one or more entries part of the
RFC 5771 multicast CIDR blocks","version":"2","enabled":true,"warninglist_entry_count":"16","valid_attribu
src, ip-dst, domain|ip"}},{"Warninglist":{"id":"5","name":"List of known microsoft
domains","type":"string","description":"Event contains one or more entries of known
microsoft domains","version":"1","enabled":true,"warninglist_entry_count":"152","valid_attributes":"domai
hostname, domain|ip"}},{"Warninglist":{"id":"4","name":"List of IPv6 link local
blocks","type":"cidr","description":"Event contains one or more entries part of the
IPv6 link local prefix (RFC 4291)","version":"1","enabled":true,"warninglist_entry_count":"1","valid_attri
src, ip-dst, domain|ip"}}
....
```

36.29.1.4 Example

```
curl   --header "Authorization: YOUR API KEY " --header "Accept: application/json"
--header "Content-Type: application/json" -X "GET" https://10.50.13.60/warninglists/
index
```

36.29.2 GET warninglists/view/1

36.29.2.1 Description

Return the a warninglist by id

36.29.2.2 Parameters

- id

36.29.2.3 Output

```
to long
```

36.29.2.4 Example

```
curl   --header "Authorization: YOUR API KEY " --header "Accept: application/json"
--header "Content-Type: application/json" -X "GET" https://10.50.13.60/warninglists/
view/17
```

37. Attribute statistics API

If you are interested in the attribute type or attribute category data distribution on your instance, MISP offers an API that will create an aggregates list. To access the API, simple sent a GET request to:

```
https://<misp url>/attributes/attributeStatistics/[context]/[percentage]
```

Where the following parameters can be set:

Context Set whether you are interested in the type or category statistics of your instance. This parameter can be either set to "type" or "category", with type being the default setting if the parameter is not set. Percentage An optional field, if set, it will return the results in percentages instead of the count. The results are always returned as JSON.

Sample output of the types in percentages from CIRCL's MISP instance:


```
{
    "AS": "0.015%",
    "attachment": "0.177%",
    "btc": "0.005%",
    "campaign-name": "0.005%",
    "comment": "1.47%",
    "domain": "15.992%",
    "domain|ip": "0.005%",
    "email-attachment": "0.207%",
    "email-dst": "0.121%",
    "email-src": "0.192%",
    "email-subject": "0.146%",
    "filename": "3.698%",
    "filename|md5": "0.349%",
    "filename|sha1": "0.894%",
    "filename|sha256": "0.652%",
    "hostname": "17.558%",
    "http-method": "0.045%",
    "ip-dst": "7.087%",
    "ip-src": "2.707%",
    "link": "5.748%",
    "malware-sample": "0.702%",
    "malware-type": "0.005%",
    "md5": "21.064%",
    "mutex": "0.278%",
    "named pipe": "0.03%",
    "other": "1.495%",
    "pattern-in-file": "0.192%",
    "pattern-in-memory": "0.303%",
    "pattern-in-traffic": "0.051%",
    "regkey": "0.126%",
    "regkey|value": "0.187%",
    "sha1": "8.921%",
    "sha256": "5.597%",
    "snort": "0.045%",
    "target-machine": "0.248%",
    "target-org": "0.01%",
    "target-user": "0.106%",
    "text": "0.934%",
    "threat-actor": "0.005%",
    "url": "2.258%",
    "user-agent": "0.081%",
```

```
    "vulnerability": "0.182%",
    "whois-registrant-email": "0.01%",
    "x509-fingerprint-sha1": "0.01%",
    "yara": "0.086%"
}
```

38. Additional statistics

Additional statistics are available as JSON which are the statistics also usable via the user interface. A ".json" can be appended to the following URLs:

```
- https://<misp url>/users/statistics/tags.json
- https://<misp url>/users/statistics.json
- https://<misp url>/users/statistics/attributehistogram.json
- https://<misp url>/users/statistics/orgs.json
```

An example output of https:///users/statistics.json:

```
{
    "stats": {
        "event_count": 5233,
        "event_count_month": 21,
        "attribute_count": 645498,
        "attribute_count_month": 723,
        "correlation_count": 207152,
        "proposal_count": 48944,
        "user_count": 1073,
        "org_count": 587,
        "thread_count": 191,
        "thread_count_month": 0,
        "post_count": 337,
        "post_count_month": 0
    }
}
```

39. MISP modules

39.1 Description

It is possible call misp-modules directly from API. If the module needs credentials, API will get the information directly from MISP configuration.

39.1.1 GET /modules/

Retrieve a list of all modules enabled.

39.1.1.1 Example

```
curl --header "Authorization: <APIKEY> " --header "Accept: application/json" --header
"Content-Type: application/json" -X GET http://<MISP>/modules/
```

39.1.1.2 Output


```
[
  {
    "name": "passivetotal",
    "type": "expansion",
    "mispattributes": {
      "input": [
        "hostname",
        "domain",
        "ip-src",
        "ip-dst"
      ],
      "output": [
        "ip-src",
        "ip-dst",
        "hostname",
        "domain"
      ]
    },
    "meta": {
       "description": "PassiveTotal expansion service to expand values with multiple
Passive DNS sources",
      "config": [
        "username",
        "password"
      ],
      "author": "Alexandre Dulaunoy",
      "version": "0.1"
    }
  },
  {
    "name": "sourcecache",
    "type": "expansion",
    "mispattributes": {
      "input": [
        "link"
```

```
      ],
      "output": [
        "link"
      ]
    },
    "meta": {
       "description": "Module to cache web pages of analysis reports, OSINT sources.
The module returns a link of the cached page.",
      "author": "Alexandre Dulaunoy",
      "version": "0.1"
    }
  },
  {
    "name": "dns",
    "type": "expansion",
    "mispattributes": {
      "input": [
        "hostname",
        "domain"
      ],
      "output": [
        "ip-src",
        "ip-dst"
      ]
    },
    "meta": {
       "description": "Simple DNS expansion service to resolve IP address from MISP
attributes",
      "author": "Alexandre Dulaunoy",
      "version": "0.1"
    }
  }
]
```


39.1.2 POST /modules/queryEnrichment

Call any enabled module.

39.1.2.1 Example

Content of dns.json ~json { “hostname”: “www.foo.be”, “module”: “dns” } ~

Query using MISP API

```
curl --header "Authorization: <APIKEY> " --header "Accept: application/json" --
header "Content-Type: application/json" --data @dns.json -X POST http://<MISP>/modules/
queryEnrichment
```

The output will be following JSON:


```
{
  "results": [
    {
      "types": [
        "ip-src",
        "ip-dst"
      ],
      "values": [
        "188.65.217.78"
      ]
    }
```

```
  ]
}
```

40. PyMISP - Python Library to Access MISP

PyMISP is a Python library for accessing MISP platforms through their REST API.

PyMISP lets you fetch events, add or update events and attributes, add or update samples, and search for attributes.

You need Auth Key access on your MISP instance to use PyMISP.

40.0.1 Capabilities

- Add, get, update, publish, delete events
- Add or remove tags
- Add file attributes: hashes, registry key, patterns, pipe, mutex
- Add network attributes: IP dest/src, hostname, domain, url, UA, ...
- Add Email attributes: source, destination, subject, attachment, ...
- Upload/download samples
- Update sightings
- Proposals: add, edit, accept, discard
- Full text search and search by attributes
- Get STIX event
- Export statistics And more. See `api.py` for the full list.

40.0.2 Installation

You can install PyMISP either with pip or by retrieving the latest version from the GitHub repository.

40.0.2.1 Install from pip

```
pip install pymisp
```

40.0.2.2 Install the latest version from the repository

```
git clone https://github.com/MISP/PyMISP.git && cd PyMISP
python setup.py install
```

Note that you will also need to install requests if you don't have it already.

40.0.3 Getting started

You now need to get your automation key. You can find it on the automation page:

```
https://<misp url>/events/automation
```

or on your profile

```
https://<misp url>/users/view/me
```

If you did not install from the repository, you can still clone it to access the examples: git clone https://github.com/MISP/PyMISP.git

To use these examples, create a file named `keys.py` in the `examples` folder and add the URL of your MISP instance together with your automation key. cd examples cp keys.py.sample keys.py vim keys.py

Once you have done that, you are ready to start.

This is how **keys.py** looks:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

misp_url = 'https://<your MISP URL>/'
misp_key = 'Your MISP auth key' # The MISP auth key can be found on the MISP web
interface under the automation section
misp_verifycert = True
```

40.0.4 Using PyMISP

To better understand how to use PyMISP, let us look at one of the existing examples: `add_named_attribute.py`. This script allows us to add an attribute to an existing event while knowing only its type, because the category is determined automatically. python #!/usr/bin/env python # -- *coding: utf-8 --*

from pymisp import PyMISP from keys import misp_url, misp_key import argparse First of all, it is obvious that we need to import PyMISP. Then we also need to know both the instance with which we will work and the API key to use: Both should be stored in the keys.py file. Finally we import argparse library so the script can handle arguments. ~~python # For python2 & 3 compat, a bit dirty, but it seems to be the least bad one try: input = raw_input except NameError: pass~~ Just a few lines to be sure that python 2 and 3 are supported ~~python def init(url, key): return PyMISP(url, key, True, 'json', debug=True)~~ This function will create a PyMISP object that will be used later to interact with the MISP instance. As seen in the api.py, a PyMISP object need to know both the URL of the MISP instance and the API key to use. It can also take additional and not mandatory data, such as the use or not of SSL or the name of the export format. ~~python if **name** == '**main**': parser = argparse.ArgumentParser(description='Create an event on MISP.') parser.add_argument("-e", "--event", type=int, help="The id of the event to update.") parser.add_argument("-t", "--type", help="The type of the added attribute") parser.add_argument("-v", "--value", help="The value of the attribute") args = parser.parse_args()~~ Then the function starts by preparing the awaited arguments: * event: The event that will get a new attribute * type: The type of the attribute that will be added. See here for more information * value: The value of the new attribute ~~python misp = init(misp_url, misp_key)~~ Thanks to the previously created function, we create a PyMISP object. ~~python event = misp.get_event(args.event) event = misp.add_named_attribute(event, args.type, args.value)~~ In order to add the new argument, we first need to fetch the event in the MISP database using the get_event function which only need the event_id. Then only once we have it, we can call the function add_named_attribute that will add the argument. ~~python print(event)~~ Finally the new event is printed, so we can check that the attribute was correctly added, and that a category was attached to it automatically.

40.0.5 Existing examples

As the name implies you will find several example scripts in the examples folder. For each you can get help if you do `scriptname.py -h`

Let us have a look at some of these examples:

40.0.5.1 add_named_attribute.py

Allow to add an argument to an existing event by giving only the type of the attribute. The category will be set with a default value.

Arguments: * **event**: The id of the event to update. * **type**: The type of the added attribute. * **value**: The value of the attribute.

40.0.5.2 add_user.py

Allow to add a user by giving the mandatory fields as entries.

Arguments: * **email**: Email linked to the account. * **org_id**: Organisation linked to the user. * **role_id**: Role linked to the user.

40.0.5.3 add_user_json.py

Add the user described in the given json. If no file is provided, returns a json listing all the fields used to describe a user.

Arguments: * **json_file**: The name of the json file describing the user you want to create.

40.0.5.4 create_events.py

Allow a user to create a new event on the MISP instance.

Arguments: * **distrib**: The distribution setting used for the attributes and for the newly created event, if relevant. [0-3]. * **info**: Used to populate the event info field if no event ID supplied. * **analysis**: The analysis level of the newly created event, if applicable. [0-2] * **threat**: The threat level ID of the newly created event, if applicable. [1-4]

40.0.5.5 del.py

Delete an event or an attribute from a MISP instance. The event has the priority: if both are set, only the event will be deleted.

Arguments: * **event**: Event ID to delete. * **attribute**: Attribute ID to delete.

40.0.5.6 delete_user.py

Delete the user with the given id. Keep in mind that disabling users (by setting the disabled flag via an edit) is always preferred to keep user associations to events intact.

Arguments: * **user_id**: The id of the user you want to delete.

40.0.5.7 edit_user.py

Edit the email of the user designed by the user_id.

Arguments: * **user_id**: The name of the json file describing the user you want to modify. * **email**: Email linked to the account.

40.0.5.8 edit_user_json.py

Edit the user designed by the user_id. If no file is provided, returns a json listing all the fields used to describe a user.

Arguments: * **user_id**: The name of the json file describing the user you want to modify. * **json_file**: The name of the json file describing your modifications.

40.0.5.9 get.py

Get an event from a MISP instance in json format.

Arguments: * **event**: Event ID to get. * **output**: Output file

40.0.5.10 last.py

Download latest events from a MISP instance. A output file can be created to store these events.

Arguments: * **last**: can be defined in days, hours, minutes (for example 5d or 12h or 30m). * **output**: Output file

40.0.5.11 searchall.py

Get all the events matching a value.

Arguments: * **search**: String to search. * **quiet**: Only display URLs to MISP * **output**: Output file

40.0.5.12 sharing_groups.py

Get a list of the sharing groups from the MISP instance. No argument.

40.0.5.13 sighting.py

Add sighting.

Arguments: * **json_file**: The name of the json file describing the attribute you want to add sighting to.

40.0.5.14 stats.py

Output attributes statistics from a MISP instance. No argument.

40.0.5.15 suricata.py

Download Suricata events.

Arguments: * **all**: Download all suricata rules available. * **event**: Download suricata rules from one event.

40.0.5.16 tags.py

Get tags from MISP instance. No argument.

40.0.5.17 tagstatistics.py

Get statistics from tags.

Arguments: * **percentage**: An optional field, if set, it will return the results in percentages, otherwise it returns exact count. * **namesort**: An optional field, if set, values are sort by the namespace, otherwise the sorting will happen on the value.

40.0.5.18 up.py

Update an existing event regarding the data inside a given json file.

Arguments: * **event**: Event ID to modify. * **input**: Input file

40.0.5.19 upload.py

Send malware sample to MISP.

Arguments: * **upload**: File or directory of files to upload. * **event**: Not supplying an event ID will cause MISP to create a single new event for all of the POSTed malware samples. * **distrib**: The distribution setting used for the attributes and for the newly created event, if relevant. [0-3]. * **ids**: You can flag all attributes created during the transaction to be marked as "to_ids" or not. * **categ**: The category that will be assigned to the uploaded samples. Valid options are: Payload delivery, Artefacts dropped, Payload Installation, External Analysis. * **info**: Used to populate the event info field if no event ID supplied. * **analysis**: The analysis level of the newly created event, if applicable. [0-2] * **threat**: The threat level ID of the newly created event, if applicable. [1-4] * **comment**: Comment for the uploaded file(s).

40.0.5.20 users_list.py

Get a list of the sharing groups from the MISP instance. No argument.

40.0.6 Going further

40.0.6.1 feed-generator

It is used to generate the CIRCL OSINT feed. This script export the events as json, based on tags, organisation, events, ... It automatically update the dumps and the metadata file.

Here is an example of a config file: url = '' key = '' ssl = True outputdir = 'output' # filters = {'tag' : 'tlp : white|feed-export|!privint', 'org':'CIRCL'} filters = {}

valid_attribute_distribution_levels = ['0', '1', '2', '3', '4', '5']

```
#### Consuming feed

As the feed is a simple set of MISP json files, the files can be easily imported
directly into any MISP instance. The script below processes the manifest file of
an OSINT
feed and reimport them in a MISP directly.

~~~~python
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from pymisp import PyMISP
import requests

url = 'https://www.circl.lu/doc/misp/feed-osint/'
osintcircl = requests.get('{}manifest.json'.format(url))
```

```
misp = PyMISP('http://misp.test/', 'key', False, 'json')
for uri in osintcircl.json():
        req = requests.get('{}{}.json'.format(url,uri))
        misp.add_event(req.json())
```

40.0.6.2 ioc-2-misp

Allow to import OpenIOC files into MISP easily. It is also possible to set specific tags on these events.

40.0.6.3 Situational Awareness

- attribute_treemap.py generate a tree-map showing the distribution of the attributes on the MISP instance.
- tags_* : these functions help having statistics and graphs about the tag repartition.

40.0.6.4 Simple example on fetching the last events

> **i** Note
>
> The example below is **executed live against a MISP instance when the book is built**, so its output always reflects the current PyMISP/API behaviour. (The old `download_last()` helper is deprecated — modern PyMISP uses `search()`.) Results are *frozen* into `_freeze/` so CI can render without a live instance.

```
import os, urllib3
from pymisp import PyMISP
urllib3.disable_warnings()

misp = PyMISP(os.environ["MISP_URL"], os.environ["MISP_KEY"], ssl=False)  # ssl=True
in prod

# Modern equivalent of last.py: fetch the most recent TLP:CLEAR events
events = misp.search(controller="events", tags=["tlp:clear"], limit=5, pythonify=True)
for e in events:
    print(f"[{e.id}] {e.Orgc.name}: {e.info[:60]} - {len(e.attributes)} attributes")
```

```
[1779] CIRCL: Police.CH - Erpresserische Kryptowährung-Adressen - 467 attributes
[1780] Organisation Centre for Cyber security Belgium: CustomerLoader: a new malware
distributing a wide variety of - 58 attributes
[1786] CIRCL: Pandora analysis (NEW ORDER LIST GREEN VALLEY CORP.xlam) - 0 attributes
[1787] CIRCL: CISA - MAR-10459736.r1.v1 - WHIRLPOOL Variant - 1 attributes
[1788] ESET: MoustachedBouncer: Espionage against foreign diplomats in Be - 4 attributes
```

41. ZeroMQ - MISP Publish/Subscribe

41.1 MISP ZeroMQ

MISP includes a flexible publish-subscribe model to allow real-time integration of MISP activities (event publication, attribute creation or removal, sighting). The MISP ZeroMQ plugin operates at global level in MISP which means standard distribution rules don't apply and every activity will be published within the ZeroMQ pub-sub channels.

MISP ZeroMQ functionality can be used for various models of integration or to extend MISP functionalities:

- real-time search of indicators into a SIEM
- automatic expansion
- dashboard activities
- logging mechanisms
- continuous indexing
- custom software or scripting

The following notification topic channels exist and can be included in the MISP ZeroMQ pub-sub:

- `misp_json` - a full event serialisation, emitted when an event is published
- `misp_json_event` - event created, updated or deleted
- `misp_json_object` - object created or updated
- `misp_json_object_reference` - object reference created or deleted
- `misp_json_attribute` - attribute created, updated or deleted
- `misp_json_tag` - tag created, updated or attached/detached
- `misp_json_sighting` - sighting added to an attribute or an event
- `misp_json_warninglist` - warninglist created or updated
- `misp_json_conversation` - discussion (conversation) messages
- `misp_json_workflow` - workflow execution notifications
- `misp_json_user` - user login, creation or update
- `misp_json_organisation` - organisation creation or update
- `misp_json_audit` - audit-log entries
- `misp_json_self` - keep-alive messages sent every minute

Each channel maps to a matching `Plugin.ZeroMQ_*_notifications_enable` setting, so a site admin decides per channel which activity is published (see below). Every message carries an `action` field (for example `add`, `edit`, `delete`) describing what happened.

41.1.1 MISP ZeroMQ configuration

To enable MISP ZeroMQ, set `Plugin.ZeroMQ_enable` to `true` under *Administration* → *Server Settings & Maintenance* → *Plugin settings* (the *Plugin* tab).

The ZeroMQ server ships with MISP and runs from the MISP Python virtualenv. Its dependencies (`pyzmq`, `redis`) are installed with the rest of MISP's Python packages, so on a standard install you do not need to install them by hand. If they are missing, install them into the MISP virtualenv rather than system-wide:

```
sudo -H -u www-data /var/www/MISP/venv/bin/pip install pyzmq redis
```

If you have problems and the plugin does not start, the logfile may be helpful.

```
sudo cat /var/www/MISP/app/tmp/logs/mispzmq.error.log
```

Optional	Plugin.ZeroMQ_enable	true	Enables or disables the pub/sub feature of MISP. Make sure that you install the requirements for the plugin to work. Refer to the installation instructions for more information.	
Optional	Plugin.ZeroMQ_host	127.0.0.1	The host that the pub/sub feature will use.	Value not set.
Optional	Plugin.ZeroMQ_port	50000	The port that the pub/sub feature will use.	Value not set.
Optional	Plugin.ZeroMQ_username		The username that client need to use to connect to ZeroMQ.	Value not set.
Optional	Plugin.ZeroMQ_password File only Redacted	*****	The password that client need to use to connect to ZeroMQ.	Value not set.
Optional	Plugin.ZeroMQ_redis_host	127.0.0.1	Location of the Redis db used by MISP and the Python PUB script to queue data to be published.	
Optional	Plugin.ZeroMQ_redis_port	6379	The port that Redis is listening on.	
Optional	Plugin.ZeroMQ_redis_password File only Redacted	*****	The password, if set for Redis.	
Optional	Plugin.ZeroMQ_redis_database	1	The database to be used for queuing messages for the pub/sub functionality.	
Optional	Plugin.ZeroMQ_redis_namespace	mispq	The namespace to be used for queuing messages for the pub/sub functionality.	Value not set.
Optional	Plugin.ZeroMQ_include_attachments	false	Enable this setting to include the base64 encoded payloads of malware-samples/attachments in the output.	Value not set.
Optional	Plugin.ZeroMQ_event_notifications_enable	false	Enables or disables the publishing of any event creations/edits/deletions.	Value not set.
Optional	Plugin.ZeroMQ_object_notifications_enable	false	Enables or disables the publishing of any object creations/edits/deletions.	Value not set.
Optional	Plugin.ZeroMQ_object_reference_notifications_enable	false	Enables or disables the publishing of any object reference creations/deletions.	Value not set.
Optional	Plugin.ZeroMQ_attribute_notifications_enable	false	Enables or disables the publishing of any attribute creations/edits/soft deletions.	Value not set.
Optional	Plugin.ZeroMQ_tag_notifications_enable	false	Enables or disables the publishing of any tag creations/edits/deletions as well as tags being attached to / detached from various MISP elements.	Value not set.
Optional	Plugin.ZeroMQ_sighting_notifications_enable	false	Enables or disables the publishing of new sightings to the ZMQ pubsub feed.	Value not set.
Optional	Plugin.ZeroMQ_user_notifications_enable	false	Enables or disables the publishing of new/modified users to the ZMQ pubsub feed.	Value not set.
Optional	Plugin.ZeroMQ_organisation_notifications_enable	false	Enables or disables the publishing of new/modified organisations to the ZMQ pubsub feed.	Value not set.
Optional	Plugin.ZeroMQ_audit_notifications_enable	true	Enables or disables the publishing of log entries to the ZMQ pubsub feed. Keep in mind, this can get pretty verbose depending on your logging settings.	
Optional	Plugin.ZeroMQ_warninglist_notifications_enable	false	Enables or disables the publishing of new/modified warninglist to the ZMQ pubsub feed.	Value not set.

Figure 41.1: ZeroMQ configuration

Each notification channel has its own `Plugin.ZeroMQ_<type>_notifications_enable` toggle (`attribute`, `audit`, `event`, `object`, `object_reference`, `organisation`, `sighting`, `tag`, `user`, `warninglist`), so the MISP site admin can decide exactly which types of messages are published. Related settings control the binding (`ZeroMQ_host`, `ZeroMQ_port`), optional authentication (`ZeroMQ_username`, `ZeroMQ_password`), the backing Redis (`ZeroMQ_redis_host`, `ZeroMQ_redis_port`, `ZeroMQ_redis_database`, `ZeroMQ_redis_namespace`, `ZeroMQ_redis_password`) and whether attachments are included (`ZeroMQ_include_attachments`).

By default, the ZMQ pub-sub channel is available to localhost only on TCP port 50000. The binding of the pub-sub channel can be updated in the configuration interface as shown above

41.1.2 MISP ZeroMQ debugging and testing

In the diagnostic section, ZeroMQ service can be started and stopped. There is a status check that can be run to give information about the number of events processed by the service.

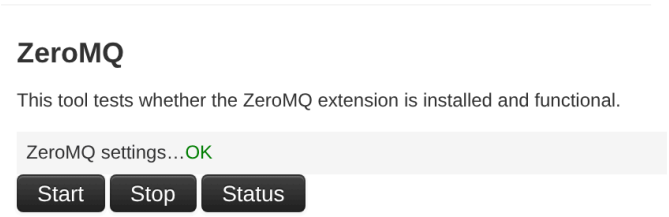


Figure 41.2: ZeroMQ diagnostics

41.1.3 Testing with sub.py tool

A simple command line tool is included with MISP to connect to the MISP ZeroMQ channel and get the notifications:

```
python3 sub.py --help
usage: sub.py [-h] [-s] [-p PORT] [-r HOST] [-o ONLY] [-t SLEEP]

Generic ZMQ client to gather events, attributes and sighting updates from a
MISP instance

optional arguments:
  -h, --help            show this help message and exit
  -s, --stats           print regular statistics on stderr
  -p PORT, --port PORT  set TCP port of the MISP ZMQ (default: 50000)
  -r HOST, --host HOST  set host of the MISP ZMQ (default: 127.0.0.1)
  -o ONLY, --only ONLY  set filter (misp_json, misp_json_event,
                        misp_json_attribute or misp_json_sighting) to limit the
                        output for a specific type; may be repeated (default: no
                        filter)
  -t SLEEP, --sleep SLEEP
                        sleep time (default: 0.1)
```

The `sub.py` will output the JSON objects for the subscribed topic, by default, all the topic channels are dumped:

```
misp@cpeb:/var/www/MISP/tools/misp-zmq$ python3 -u sub.py  | jq .
....
{
  "uptime": 50,
  "status": "And when you're dead I will be still alive."
}
{
  "uptime": 60,
  "status": "And believe me I am still alive."
}
{
  "uptime": 70,
  "status": "I'm doing science and I'm still alive."
}
{
  "uptime": 80,
  "status": "I feel FANTASTIC and I'm still alive."
}
{
  "uptime": 90,
  "status": "While you're dying I'll be still alive."
}
{
  "Sighting": {
    "uuid": "592d9588-fda0-490f-bf6e-4e56950d210f",
    "source": "",
    "type": "0",
    "date_sighting": 1496159624,
    "org_id": "2",
    "event_id": "8102",
    "attribute_id": "1044812"
  }
}
{
  "Attribute": {
    "id": "1044802",
```

```
    "value2": "",
    "value1": "1.2.3.4",
    "uuid": "592d8494-7120-4760-b5e2-4858950d210f",
    "batch_import": "0",
    "comment": "",
    "value": "1.2.3.4",
    "type": "ip-dst",
    "to_ids": 0,
    "timestamp": 1496155284,
    "distribution": "5",
    "sharing_group_id": 0,
    "deleted": "0",
    "disable_correlation": "0",
    "event_id": "8100",
    "category": "Network activity"
  }
}
....
```

41.1.4 Notification Schemas

Each notification channel uses a slightly different JSON schema. Consult this section to identify which MISP components exist in a channel:

41.1.4.1 misp_json - events published

When an event is published to ZMQ (which is different from being published in MISP) the ZMQ notification will contain the MISP event data along with all its component children. These components include: * A list of attributes * A list of objects, which contain their own lists of attributes * A list of related events - added when attributes in separate events correlate * Any galaxies that this event belongs to * A list of tags that apply to the event

Example:

```
{
    "Event": {
        "id": "625",
        "orgc_id": "2",
        "org_id": "1",
        "date": "2017-05-24",
        "threat_level_id": "3",
        "info": "M2M - Fwd: IMG_3428.pdf",
        "published": false,
        "uuid": "59259036-fcd0-4749-8a6c-4d88950d210f",
        "attribute_count": "7",
        "analysis": "1",
        "timestamp": "1505755565",
        "distribution": "3",
        "proposal_email_lock": false,
        "locked": false,
        "publish_timestamp": "1505416766",
        "sharing_group_id": "0",
        "disable_correlation": false,
                "Org":  {"id":  "1",  "name":  "MISP",  "uuid":  "56ef3277-1ad4-42f6-
b90b-04e5c0a83832"},
                            "Orgc":   {"id":   "2",   "name":   "CIRCL",   "uuid":
"55f6ea5e-2c60-40e5-964f-47a8950d210f"},
        "Attribute": [
            {
                "id": "157835",
                "type": "attachment",
```

```
            "category": "Artifacts dropped",
            "to_ids": false,
            "uuid": "59259037-1014-4669-96b1-46af950d210f",
            "event_id": "625",
            "distribution": "5",
            "timestamp": "1495633975",
            "comment": "IMG_3428.pdf",
            "sharing_group_id": "0",
            "deleted": false,
            "disable_correlation": false,
            "object_id": "0",
            "object_relation": null,
            "value": "tmpzuni0skf",
            "ShadowAttribute": []
        }
    ],
    "ShadowAttribute": [],
    "RelatedEvent": [],
    "Galaxy": [],
    "Object": [
        {
            "id": "1",
            "name": "http-request",
            "meta-category": "network",
            "description": "A single HTTP request header",
            "template_uuid": "b4a8d163-8110-4239-bfcf-e08f3a9fdf7b",
            "template_version": "1",
            "event_id": "625",
            "uuid": "59c0016c-0984-4779-9688-05b8c0a83832",
            "timestamp": "1505755500",
            "distribution": "5",
            "sharing_group_id": "0",
            "comment": "",
            "deleted": false,
            "ObjectReference": [],
            "Attribute": [
                {
                    "id": "164371",
                    "type": "http-method",
                    "category": "Network activity",
                    "to_ids": false,
                    "uuid": "59c0016c-a744-440d-ad92-05b8c0a83832",
                    "event_id": "625",
                    "distribution": "5",
                    "timestamp": "1505760143",
                    "comment": "",
                    "sharing_group_id": "0",
                    "deleted": false,
                    "disable_correlation": false,
                    "object_id": "1",
                    "object_relation": "method",
                    "value": "POST",
                    "ShadowAttribute": []
                },
                {
                    "id": "164372",
                    "type": "uri",
                    "category": "Network activity",
                    "to_ids": true,
                    "uuid": "59c0016c-fac0-4055-9f3d-05b8c0a83832",
```

```
                        "event_id": "625",
                        "distribution": "5",
                        "timestamp": "1505755500",
                        "comment": "",
                        "sharing_group_id": "0",
                        "deleted": false,
                        "disable_correlation": false,
                        "object_id": "1",
                        "object_relation": "uri",
                        "value": "\/test.php",
                        "ShadowAttribute": []
                    }
                ]
            }
        ],
         "Tag": [{"id": "2", "name": "tlp:white", "colour": "#ffffff", "exportable":
true, "hide_tag": false}]
    }
}
```

41.1.4.2 misp_json_attribute - attribute updated or created

The attributes appear to have the most diversity depending on the action applied to them.

When an attribute gets created, the attribute gets sent out via ZMQ. The parent event id is included in the attribute JSON, but there is no extra event metadata like there is when an attribute is deleted or modified.

Create Example:

```
{
    "Attribute": {
        "to_ids": "1",
        "timestamp": 1505235275,
        "distribution": "5",
        "deleted": "0",
        "disable_correlation": "0",
        "event_id": "625",
        "category": "Network activity",
        "type": "domain",
        "value": "microsoft.net",
        "comment": "",
        "batch_import": "0",
        "uuid": "59b8114b-1c80-4149-be3a-03e9c0a83832",
        "sharing_group_id": 0,
        "value1": "microsoft.net",
        "value2": "",
        "id": "164363"
    }
}
```

Edited attribute notifications send metadata about their parent events and information about the attribute's sharing group, attribute-level tags, and sightings data. It's important to note that only the new value of the edited attribute is sent along the ZMQ channel. In order to see the difference between the new and old values, you'd need to have a copy of the old attribute value stored somewhere. If you do, you can use the attribute's `uuid` key (which never changes) to correlate the new and old values.

Edit Example:

```
// microsoft.net --> microsoft.com
{
    "Attribute": {
        "id": "164363",
```

```
        "event_id": "625",
        "category": "Network activity",
        "type": "domain",
        "value1": "microsoft.com",
        "value2": "",
        "to_ids": "1",
        "uuid": "59b8114b-1c80-4149-be3a-03e9c0a83832",
        "timestamp": 1505235283,
        "distribution": "5",
        "sharing_group_id": 0,
        "comment": "",
        "deleted": false,
        "disable_correlation": false,
        "value": "microsoft.com",
        "batch_import": "0"
    },
    "Event": {
        "id": "625",
        "org_id": "1",
        "date": "2017-05-24",
        "info": "M2M - Fwd: IMG_3428.pdf",
        "user_id": "1",
        "uuid": "59259036-fcd0-4749-8a6c-4d88950d210f",
        "published": false,
        "analysis": "1",
        "attribute_count": "5",
        "orgc_id": "2",
        "timestamp": "1505235275",
        "distribution": "3",
        "sharing_group_id": "0",
        "proposal_email_lock": false,
        "locked": false,
        "threat_level_id": "3",
        "publish_timestamp": "1505233367",
        "disable_correlation": false
    },
    "SharingGroup": {
        "id": null,
        "name": null,
        "releasability": null,
        "description": null,
        "uuid": null,
        "organisation_uuid": null,
        "org_id": null,
        "sync_user_id": null,
        "active": null,
        "created": null,
        "modified": null,
        "local": null,
        "roaming": null
    },
    "AttributeTag": [],
    "Sighting": []
},
```


When an attribute gets deleted, the `deleted` key gets set to `1`, and the attribute's event metadata gets sent alongside it.

Delete Example:


```
{
    "Attribute": {
```

```
        "id": "164362",
        "event_id": "625",
        "category": "Network activity",
        "type": "domain",
        "value1": "microsoft.com",
        "value2": "",
        "to_ids": true,
        "uuid": "59b81121-f4b4-4ed3-aa43-03eac0a83832",
        "timestamp": 1505235262,
        "distribution": "5",
        "sharing_group_id": 0,
        "comment": "",
        "deleted": 1,
        "disable_correlation": false,
        "value": "microsoft.net"
    },
    "Event": {
        "id": "625",
        "org_id": "1",
        "date": "2017-05-24",
        "info": "M2M - Fwd: IMG_3428.pdf",
        "user_id": "1",
        "uuid": "59259036-fcd0-4749-8a6c-4d88950d210f",
        "published": false,
        "analysis": "1",
        "attribute_count": "5",
        "orgc_id": "2",
        "timestamp": "1505235233",
        "distribution": "3",
        "sharing_group_id": "0",
        "proposal_email_lock": false,
        "locked": false,
        "threat_level_id": "3",
        "publish_timestamp": "1505233367",
        "disable_correlation": false
    }
}
```


41.1.4.3 misp_json_sighting - sighting added to an attribute or an event

The message sent for sightings is fairly simple, with the type of sighting (0 = Addition, 1 = False Positive), the date (in seconds-since-epoch format), the id of the attribute it applies to and the id of the attribute's parent event.

Addition Example:


```
{
    "Sighting": {
        "type": "0",
        "attribute_id": "164373",
        "event_id": "625",
        "org_id": "1",
        "date_sighting": 1505767537,
        "source": "",
        "uuid": "59c03071-f480-4311-a710-03edc0a83832",
        "id": "1"
    }
}
```


False Positive Example:

```
{
    "Sighting": {
        "type": "1",
        "attribute_id": "164373",
        "event_id": "625",
        "org_id": "1",
        "date_sighting": 1505767543,
        "source": "",
        "uuid": "59c03077-d560-4a8b-b841-05b8c0a83832",
        "id": "2"
    }
}
```

41.1.4.4 misp_json_user - user updates or creation

An update is sent through ZMQ when users log in. There are actually two messages for this - both being fairly sparse. The `current_login` message contains who logged in and at what time (in seconds-since-epoch format) the login happened. The `last_login` message contains who just logged in, what time the previous login occurred (last_login) and the time the record was modified, which should be the same as current_login time.

Login Example:

```
{
    "User": {
        "id": "1",
        "last_login": "1000046766",
        "date_modified": 1000060160
    }
}
{
    "User": {
        "id": "1",
        "current_login": 1000060160,
        "date_modified": 1000060160
    }
}
```

When a user gets created, all of the information about the user (id, email, base64 encoded GnuPG key, role, etc.) gets sent to ZMQ. If this information is modified, the same JSON will be sent along the ZMQ channel, with updated values. For example, if the below user is disabled, the same JSON will be sent, but the `disabled` key will be set to `"1"`

User Creation and User Edit Example:

```
{
    "User": {
        "server_id": 0,
        "autoalert": "1",
        "invited_by": "1",
        "nids_sid": 5976699,
        "termsaccepted": 0,
        "role_id": "3",
        "change_pw": 1,
        "contactalert": "1",
        "disabled": "0",
        "current_login": "0",
        "last_login": "0",
        "force_logout": "0",
        "email": "user@testemail.com",
        "enable_password": "0",
        "org_id": "1",
```

```
        "authkey": "__<redacted>__",
        "gpgkey": "__<redacted>__",
        "notify": "1",
        "date_created": 1000000000,
        "date_modified": 1000000000,
        "newsread": 0,
        "certif_public": "",
        "id": "4"
    }
}
```

41.1.4.5 misp_json_organisation - organisation updates or creation

Org notifications are sent when Orgs are updated and created, but not when they are deleted. They are generally the same, with the additional fields `created_by` and `date_created` being present when an Org is created.

Creation Example:

```
{
    "Organisation": {
        "created_by": "1",
        "local": "1",
        "name": "test",
        "uuid": "59c0367d-fe8c-42a4-9db2-03ecc0a83832",
        "description": "Test",
        "nationality": "Not specified",
        "sector": "",
        "type": "",
        "contacts": "",
        "logo": {
            "name": "",
            "type": "",
            "tmp_name": "",
            "error": 4,
            "size": 0
        },
        "date_created": "2017-09-18 23:11:28",
        "date_modified": "2017-09-18 23:11:28",
        "id": "10"
    }
}
```

Edit Example:

```
{
    "Organisation": {
        "local": "1",
        "name": "test",
        "uuid": "59c0367d-fe8c-42a4-9db2-03ecc0a83832",
        "description": "Alternate Test",
        "nationality": "Not specified",
        "sector": "",
        "type": "",
        "contacts": "",
        "logo": {
            "name": "",
            "type": "",
            "tmp_name": "",
            "error": 4,
            "size": 0
        },
```

```
        "id": "10",
        "date_modified": "2017-09-18 23:11:37"
    }
}
```

41.1.4.6 misp_json_self - keep-alive messages sent every minute

Only really useful to ensure the ZMQ server is running. And for a bit of humor

```
{
    "status": "I'm doing science and I'm still alive.",
    "uptime": 9170
}
```

41.1.5 Tips for Building a Subscriber

1. `misp_json_attribute` notifications are sent when attributes are created, deleted, and edited
   - Check the `deleted` key to identify if an attribute has been created or deleted
   - If an attribute has been edited, the new value will be sent out via ZMQ, but the `uuid` key will remain the same. Use this to determine if an attribute has existed before or not
2. Some compound attribute types have component types that don't exist outside of them
   - For example, MISP doesn't have a single `ip` attribute type except in the `domain|ip` type
   - If you're going to split up and resubmit these attributes, you may have to modify these component types so MISP will recognize them (e.g. domain|ip -> domain, ip-dst)

42. Modules

MISP modules extend MISP with expansion, import, export and action services without touching the core. They are self-contained Python 3 programs that run as a separate service exposing a small REST interface, which MISP calls when it needs them. This keeps the module ecosystem — well over a hundred modules — decoupled from the MISP application, so it can be installed, updated and scaled independently. The modules are maintained in the misp-modules repository.

42.1 Module families

Modules come in five families, each hooking into a different part of MISP:

- **Enrichment (expansion)** — take an attribute (or object) and look up extra information about it, adding new attributes/objects to the event. This is the largest family (passive DNS, VirusTotal, Shodan, WHOIS, sandbox lookups, and many more).
- **Enrichment (hover)** — a lightweight variant of expansion that shows a lookup result on the fly when you hover over (or click) a value, without adding anything to the event.
- **Import** — parse an external format or source into MISP attributes/objects, surfaced under the event's **Populate from...** menu.
- **Export** — serialise event data into an external format, surfaced under the event's **Download as...** menu.
- **Action** — perform an action in response to something happening in MISP (for example sending a notification), typically triggered from a workflow.
- **Cortex** — query Cortex analysers, treated as its own family alongside the native enrichment modules.

42.2 Running the module server

The modules run as a separate service (`misp-modules`), usually in a Python virtualenv on the MISP host (or a dedicated host), listening on a local port that MISP connects to. Follow the repository's installation instructions:

- Install in a Python virtualenv (recommended)
- Install on RHEL-based distributions

42.3 Enabling and configuring modules in MISP

Once the module service is running, enable and point MISP at it under **Administration → Server Settings → Plugin**. Each family has its own settings:

- Turn the family on with its `*_services_enable` setting (`Enrichment_services_enable`, `Import_services_enable`, `Export_services_enable`, `Action_services_enable`, `Cortex_services_enable`).
- Tell MISP where the module service is with the matching `*_services_url` and `*_services_port` settings (the enrichment and Cortex families each have their own URL/port).
- For hover lookups, enable `Enrichment_hover_enable` (and, optionally, `Enrichment_hover_popover_only` to require a click rather than an automatic hover).

Individual modules are then toggled with their own `Plugin.<Family>_<name>_enabled` setting — for example `Plugin.Enrichment_dns_enabled` — so you can enable only the modules you need.

42.4 Using modules

Where a module surfaces depends on its family (see also enrichment in the Using the System chapter):

- **Expansion / Cortex** — from an event, use **Enrich Event** to run the applicable modules over the whole event, or the per-attribute enrichment (asterisk) and Cortex (eye) actions on a single attribute.

- **Hover** — hover over a value (or click its magnifier) to see the lookup result inline.
- **Import** — pick the module under the event's **Populate from…** menu.
- **Export** — pick the module under the event's **Download as…** menu.

42.5 Module endpoints

MISP talks to the module service through a couple of endpoints, which you can also call directly:

- `GET /modules/index` — list the modules MISP currently sees as enabled, with the attribute types they accept.
- `POST /modules/queryEnrichment` — run an enrichment module against a piece of data (see the automation chapter).

43. External Connectors

MISP is designed to sit at the centre of a security stack, not in isolation. Threat intelligence stored in MISP is meant to flow *out* to detection and response tooling (SIEM, EDR/XDR, firewalls, IDS), and context is meant to flow *in* from those same tools and from other intelligence sources. This chapter is an index of the ways MISP connects to external systems, with one worked example.

> **Tip**
>
> The authoritative, continuously updated catalogue of tools that talk to MISP lives on the MISP project website and under the MISP GitHub organisation. When in doubt about the current state of a specific integration, check there — third-party connectors evolve independently of MISP releases.

43.1 How MISP connects to other tools

There is rarely just one way to integrate; pick the mechanism that fits the other system.

Mechanism	Use it for	Covered in
REST API	Any programmatic access — query, create, export, orchestrate	Automation & API
PyMISP	Python integrations, scripts and glue code	PyMISP
MISP-to-MISP sync	Sharing with other MISP instances and communities	Sharing
STIX 1 / STIX 2 & TAXII 2.x	Interop with STIX/TAXII-based platforms	Sharing, Automation
Enrichment / import / export modules	Pluggable enrichment and format conversion	Misp-modules
ZeroMQ pub/sub	Real-time streaming of MISP activity to other systems	ZeroMQ
Feeds	Consuming external indicator feeds into MISP	Managing feeds

Most third-party "connectors" are built on top of one of these — typically the REST API (via PyMISP) for pulling indicators out, or STIX/TAXII for standards-based exchange.

43.2 Notable integrations

The following are widely used integrations. Consult the linked upstream projects for current setup instructions and version support — they are maintained outside the MISP core release.

- **SIEM / log analytics** — Splunk, Elastic/ELK, IBM QRadar and similar platforms typically ingest MISP indicators through the REST API or a dedicated add-on. Several are listed in the MISP tools catalogue.
- **Microsoft Sentinel & Microsoft Defender** — export IoCs through the Microsoft Graph Security API using the sample script (worked example below).
- **SOAR / case management** — TheHive and Cortex integrate closely with MISP for alerting, observable enrichment and case handling.

- **STIX / TAXII platforms** — MISP exports and imports STIX 1 and STIX 2 (via the misp-stix library) and can push to TAXII 2.x servers; see the Sharing chapter's TAXII section.
- **Trust & directory** — Cerebrate distributes organisation and sharing-group information across a community; see Sharing.

43.3 Worked example: MISP to Microsoft Graph Security

The **MISP to Microsoft Graph Security Script** lets you export custom threat indicators (IoCs) from MISP to Microsoft products such as **Microsoft Sentinel** and **Microsoft Defender**, through the Microsoft Graph Security API.

The sample and its most current instructions live in the Microsoft Graph security-api-solutions repository (short link). The summary below reflects that sample at time of writing; treat the upstream README as authoritative.

43.3.1 Prerequisites

- Python 3.x.
- A Microsoft account or an Office 365 for business account able to register an application in Azure Active Directory.

43.3.2 Getting started

1. Clone the sample repository.
2. Change into the `security-api-solutions/Samples/MISP` directory.
3. Install the dependencies: `pip3 install requests requests-futures pymisp`.
4. Register an application in the Azure portal under *App registrations → New registration*. Record the **Application (client) ID**, create a **client secret** under *Certificates & secrets*, and under *API permissions* grant the Microsoft Graph application permission **ThreatIndicators.ReadWrite.OwnedBy**.
5. Fill in `config.py` with your tenant, client ID and client secret, and your MISP connection details.
6. Run the script: `PYTHONHASHSEED=0 python3 script.py`.

43.3.3 Key configuration options (`config.py`)

- **Target product** — `targetProduct = "Azure Sentinel"` or `targetProduct = "Microsoft Defender ATP"`.
- **MISP connection** — `misp_domain`, `misp_key` (use a dedicated sync/automation key — see Advanced AuthKeys), and `misp_verifycert` (keep `True` in production with a valid certificate).
- **Event filters** — `misp_event_filters` is a list of filter dictionaries applied as an *intersection*; within a single filter, a list of values is an *OR*. An empty list (`misp_event_filters = []`) exports all events. The filter keys mirror MISP's REST search parameters (`type_attribute`, `category`, `tags`, `date_from`/`date_to`, `last`, `published`, and so on).
- **Action** — `alert` (default), `allow`, or `block`, controlling how the indicator is treated in the target product.
- **Expiry** — `days_to_expire` sets how long indicators live in the Graph Security API (default 30).

43.3.4 Reading and troubleshooting

- List indicators already pushed: `python3 script.py -r`.
- Every request is logged as a JSON file under `logs/`; failed requests can be reviewed with `cat logs/*_error_*`.
- Schedule regular runs with cron, for example every Sunday at 02:00:

```
0 2 * * Sun /home/user/misp-graph-script/python3 script.py
```

> **Note**
>
> This worked example is adapted from the upstream Microsoft sample and is included to show the *shape* of a REST/PyMISP-based connector. For the exact, current steps — including any changes to Azure app registration or the script itself — always follow the upstream README.

VI Administration

44 Administration 329
44.1 Users 329
44.2 Organisations 334
44.3 Roles 338
44.4 Tools 341
44.5 Server settings and maintenance 341
44.6 Blocklists and block rules 343
44.7 Import Regexp 348
44.8 Managing the Signature allowedlist 348
44.9 Managing correlations 349
44.10 Logs 350
44.11 Background Processing 351
44.12 Other 2.5 administration features 353
44.13 Various administration tips & tricks 354
44.14 Jobs 359
44.15 MISP Backup 360
44.16 MISP Restore 360
44.17 Command line interface (CLI) commands 360
44.18 Administration of TOTP/HOTP 364

45 Workflows & Blueprints 367
45.1 Concepts 367
45.2 Enabling workflows 368
45.3 The workflow editor 368
45.4 Ad-hoc workflows 368
45.5 Workflow Blueprints 369
45.6 Debugging and logs 369
45.7 Permissions and execution context 369
45.8 API and CLI 369

46 Updating Python Dependencies . . . 371
46.1 Installation 371
46.2 Updating MISP and its dependencies 371

44. Administration

This chapter covers the administration of a MISP instance: managing users, organisations and roles; the server settings and diagnostics tool; blocklists, correlation management and logging; background processing; and the command-line maintenance tools. Most of the actions described here require site-admin privileges, though organisation admins can perform a subset of them within their own organisation.

44.1 Users

As an admin (not to be confused with Org Admin), you can set up new accounts for users, edit user profiles, delete them, or just have a look at all the viewers' profiles. Organisation admins (Org Admin) are restricted to executing these actions exclusively within their own organisation's users only.

44.1.1 Adding a new user

To add a new user, click on the Add User button in the administration menu to the left and populate the fields available the loaded view:

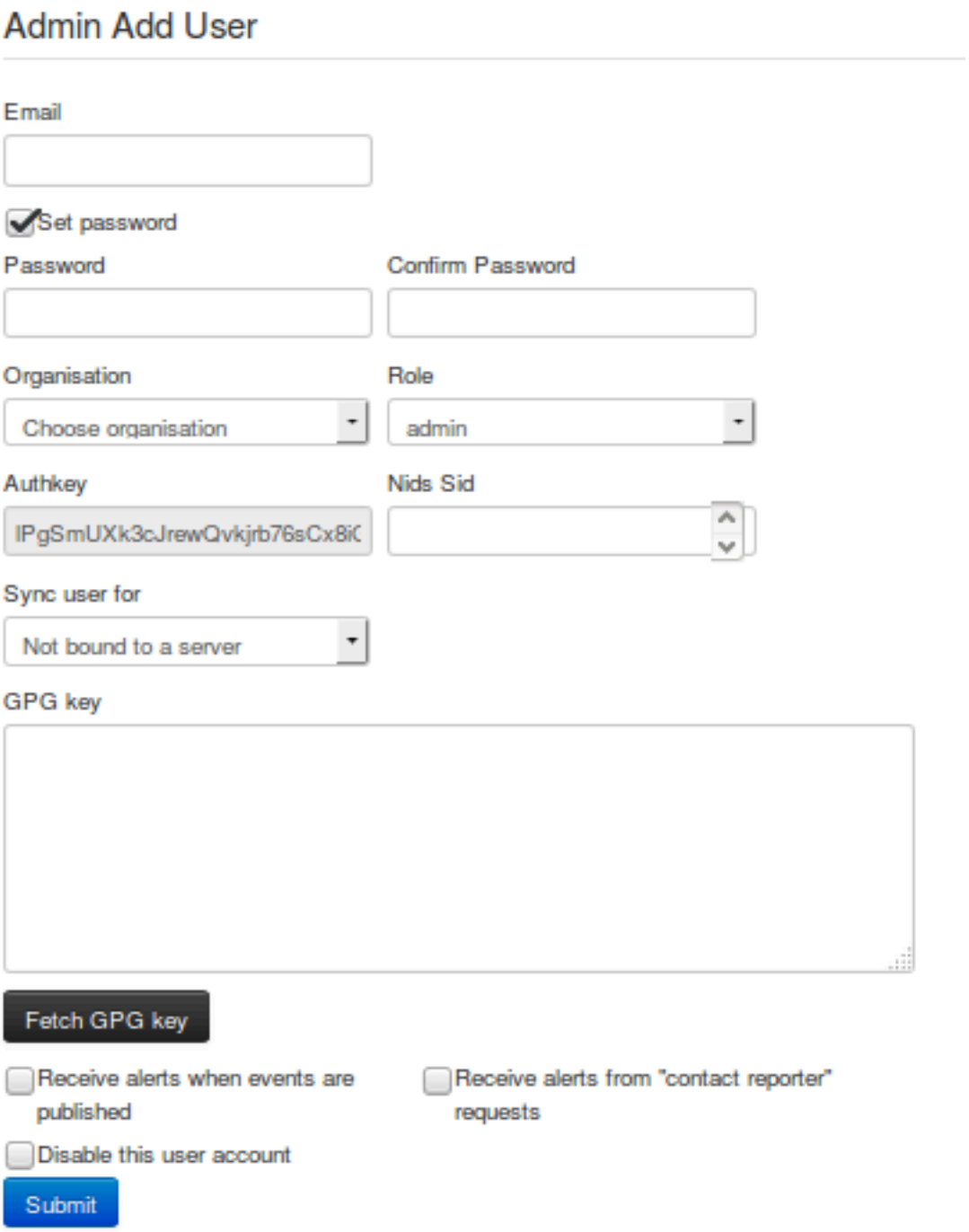


Figure 44.1: Fill this form out to add a new user. Keep in mind that the drop-down menu titled "Role" controls privileges the user will have.

- **Email:** The user's e-mail address, this will be used as his/her login name and as an address to send all automated e-mails as well as e-mails sent by contacting the user as the reporter of an event.
- **Set password:** Tick the box if you want to define a temporary user-password for the user. If you don't, you should use the action button 'reset password' in the 'List Users' view to generate one and send it by email to the user.
- **Password:** *This textbox is displayed only when 'Set password' is ticked.* A Temporary password for the user that he/she should change after the first login. Ensure that password pass the MISP password policy.

- **Confirm Password:** *This textbox is displayed only when 'Set password' is ticked.* This should be an exact copy of the Password field.
- **Organisation:** A drop-down list enables you to choose an organisation for the user. To learn more about organisation, click here.
- **Roles:** A drop-down list allows you to select a role-group that the user should belong to. Roles define user privileges attributed to the user. To learn more about roles, click here.
- **Authkey:** This is assigned automatically and is the unique authentication key of said user (he/she will be able to reset this and receive a new key). It is used for exports and for connecting one server to another, but it requires the user to be assigned to a role that has auth permission enabled.
- **NIDS SID:** Network Intrusion Detection System (NIDS) Signature ID (SID). Snort rules exported by the created user will have the offset defined in the user profile and each rule generated during the export will receive an incrementing SID starting with the user's offset. If no SID offset is specified a default, randomized value will be set.
- **Sync user for:** If this option is set, the selected server's push rules will also be applied when this local user is used to pull from a remote instance. This option is available for admin, Org Admin and Sync user role.
- **Gpgkey:** The key used to encrypt e-mails sent through the system.
- **Fetch GnuPG key:** Fetch GnuPG public key.
- **Receive alerts when events are published:** This option will subscribe the new user to automatically generated e-mails whenever an event is published.
- **Receive alerts from "contact reporter" requests:** This option will subscribe the new user to e-mails that are generated when another user tries to get in touch with an event's reporting organisation that matches that of the new user.
- **Disable this user account:** Tick it if you want to disable this user account. (preferred to removing an account)

44.1.2 Listing all users

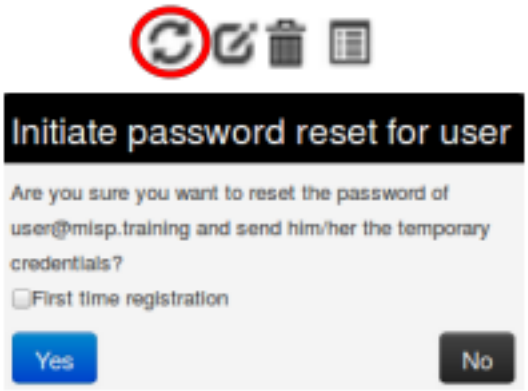
To list all current users of the system, just click on List Users under the administration menu to the left. A view will load containing a list of all users and the following columns of information:

ID	Org	Role	Email	NIDS SID	Last Login	Created	Last API Access	External Auth	Actions
576	Sample Organisation	Sync user	sync@sample.test	7165264	Never	2026-07-08 07:42:56			
575	Sample Organisation	Read Only	readonly@sample.test	5005074	Never	2026-07-08 07:42:56			
574	Sample Organisation	Publisher	publisher@sample.test	5298936	Never	2026-07-08 07:42:56			
573	Sample Organisation	User	analyst@sample.test	4604616	2026-07-08 09:49:35	2026-07-08 07:42:56			
572	Sample Organisation	Org Admin	org.admin@sample.test	2300256	Never	2026-07-08 07:35:10			

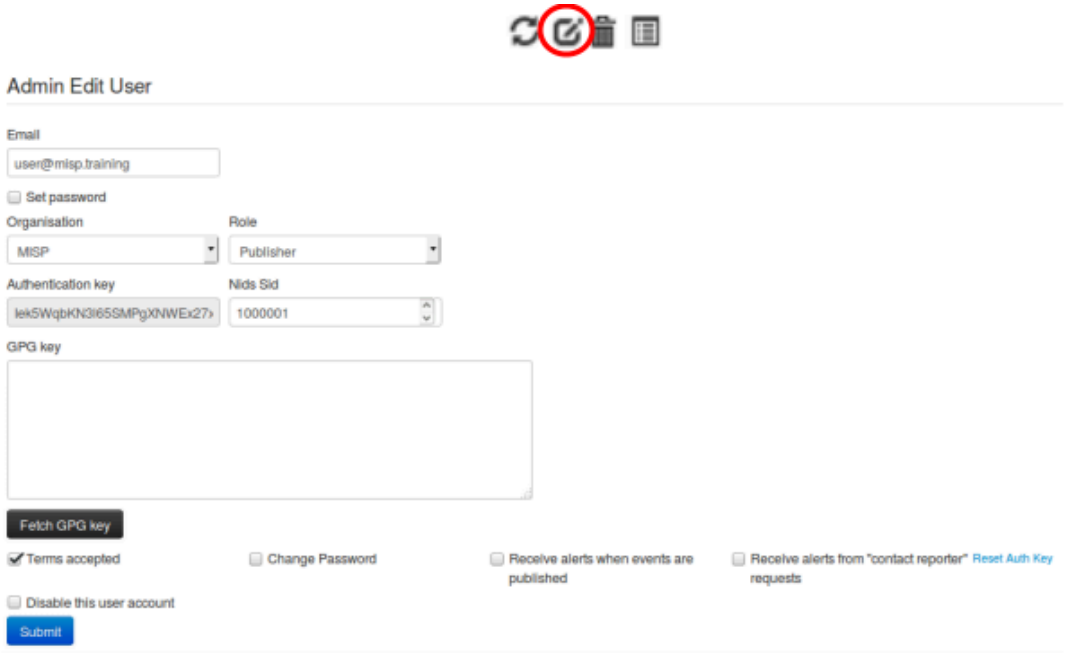
Figure 44.2: View, Edit or Delete a user using the action buttons to the right.

- **Id:** The user's automatically assigned ID number.
- **Org:** The organisation that the user belongs to.
- **Email:** The e-mail address (and login name) of the user.
- **Authkey:** Unique authentication key of the user.
- **Autoalert:** Shows whether the user has subscribed to auto-alerts and is continuing to receive mass-emails regarding newly published events that he/she is eligible for.
- **Contactalert:** Shows whether the user has the subscription to contact reporter e-mails directed at his/her organisation is turned on or off.
- **Gpgkey:** Shows whether the user has entered a GnuPG key yet.
- **Nids Sid:** Displays the currently assigned NIDS ID.
- **Termsaccepted:** This flag indicates whether the user has accepted the terms of use or not.
- **Last login:** Date of last login.
- **Disabled:** Displays the user status. Enabled or disabled.
- **Action Buttons:** There are 4 options available: reset the password, edit the user, delete the user or display a user's information. These options are also available on the left menu.

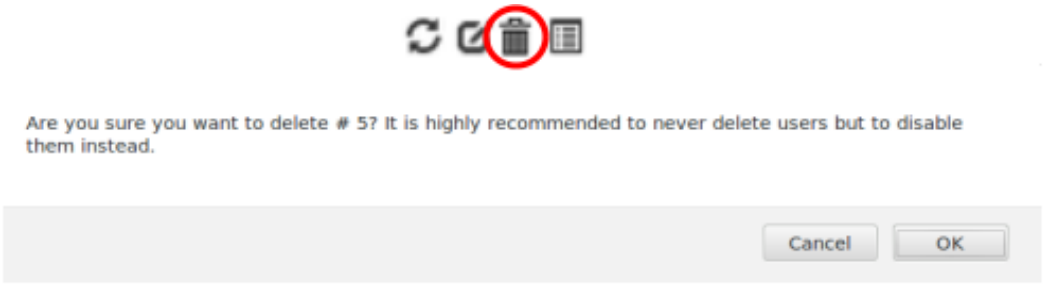
- **Reset Password:** Use this action to reset a password. If you've created a new user without A password, tick the 'First time registration' checkbox to send a welcome message. Otherwise a reset



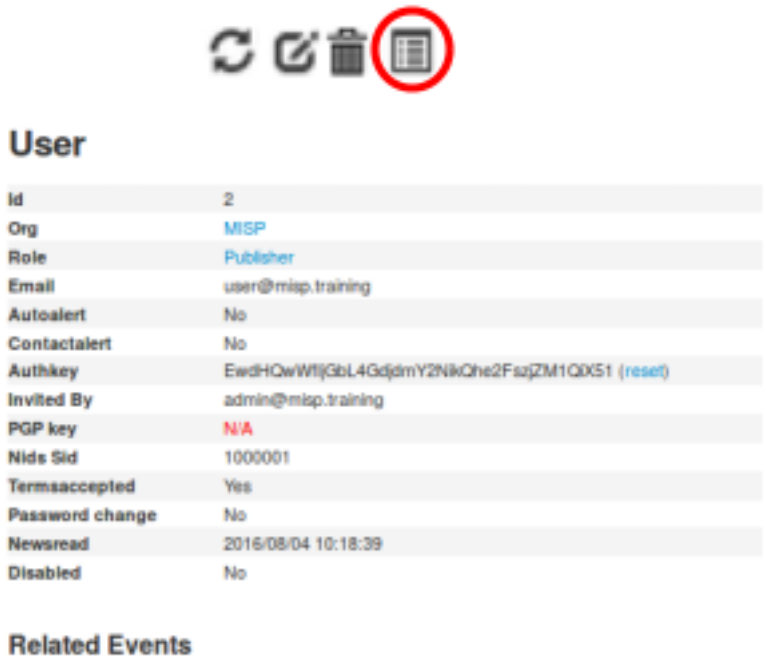
password message will be sent.
- **Edit the user:** Same options of create user's view. Only a few options are available here: * **Terms accepted:** Indicates whether the user has accepted the terms of use already or not. * **Change Password:** Setting this flag will require the user to change password after the next login. * **Reset Auth Key:** Use this link for generate a new AuthKey.



- **Delete the user:** If you want to delete a user. (Note: disabling is the preferred method)



– **Display the user:** Display all user's information.



44.1.3 Contacting a user

Site admins can use the "Contact users" feature to send all or individual user an e-mail. Users that have a GnuPG key set will receive their e-mails encrypted. When clicking this button on the left, you'll be presented with a form that allows you to specify the type of the e-mail, who it should reach and what the content is using the following options:

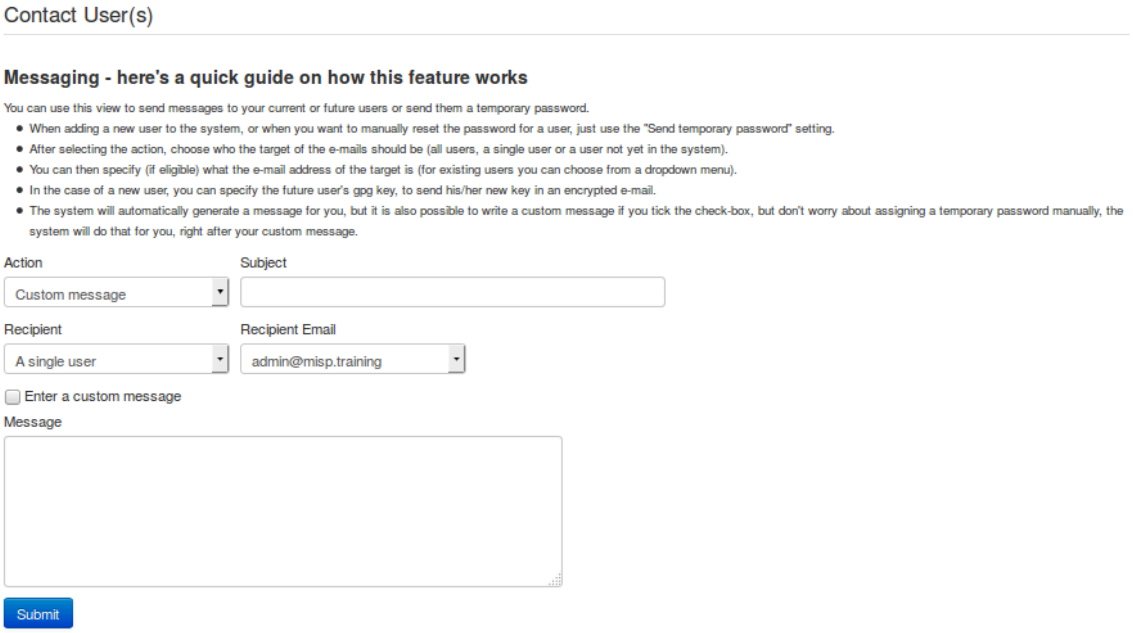


Figure 44.3: Contact your users here.

- **Action:** This defines the e-mail type, which can be a custom message or a password reset. Password resets automatically include a new temporary password at the bottom of the message and will automatically change the user's password accordingly.
- **Subject:** In the case of a custom e-mail, you can enter a subject line here.
- **Recipient:** The recipient toggle lets you contact all your users, a single user (which creates a second drop-down list with all the e-mail addresses of the users) and potential future users (which opens up a text field for the e-mail address and a text area field for a GnuPG public key).
- **Custom message checkbox:** This is available for password resets and for welcome messages. You can either write your own message (which will be appended with a temporary key and the signature), or let the system generate one automatically.

Keep in mind that all e-mails sent through this system, in addition to your own message, will be signed in the name of the instance's host organisation's support team, the e-mail will also include the e-mail address of the instance's support (if the contact field is set in the bootstrap file), and will include the instance's GnuPG signature for users that have a GnuPG key set (and thus are eligible for an encrypted e-mail).

:warning: GnuPG instance key is the GnuPG key used by the MISP instance and which is only used to sign notification. The GnuPG key used in the MISP instance must not be used anywhere else and should not be valuable.

44.2 Organisations

Each users belongs to an organisation. As admin, you can manage these organisations.

44.2.1 Adding a new organisation

To add a new organisation, click on the "Add Organisation" button in the administration menu to the left and fill out the following fields in the view that is loaded:

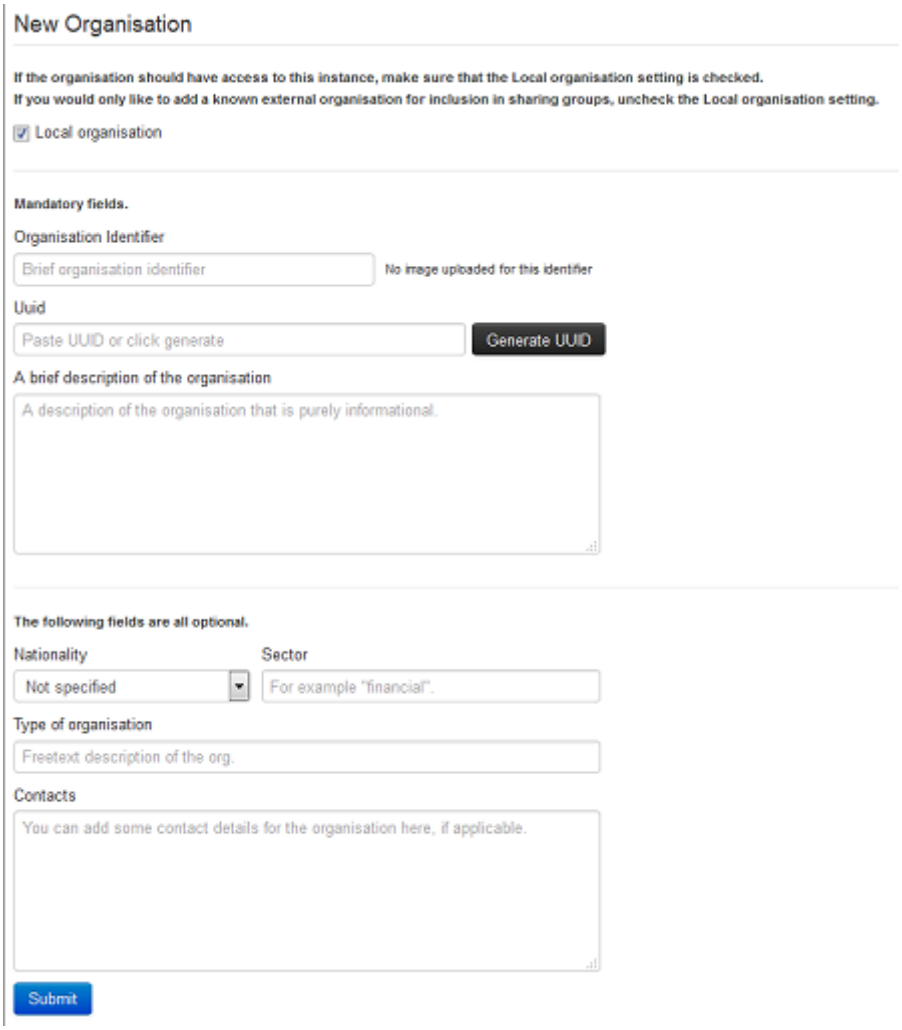


Figure 44.4: Fill this form out to add a new organisation.

- **Local organisation:** If the organisation should have access to this instance, tick the checkbox. If you would only like to add a known external organisation for inclusion in sharing groups, uncheck it.
- **Organisation Identifier:** Name your organisation. If you want to add a picture, you should add a file on the webserver using the 'Server Settings menu'. Picture should have the same name. To learn more about server settings menu, click here.
- **Uuid:** Unique identifier. If you want to share organisation between MISP multi-instance, use the same Uuid.
- **A brief description of the organisation:** A word for describing the organisation.
- **Nationality:** A drop-down list for selecting the country of organisation.
- **Sector:** Define the sector of organisation (financial, transport, telecom...)
- **Type of organisation:** Define the type of the organisation.
- **Contacts:** You can add some contact details for the organisation.

44.2.2 Listing all organisations

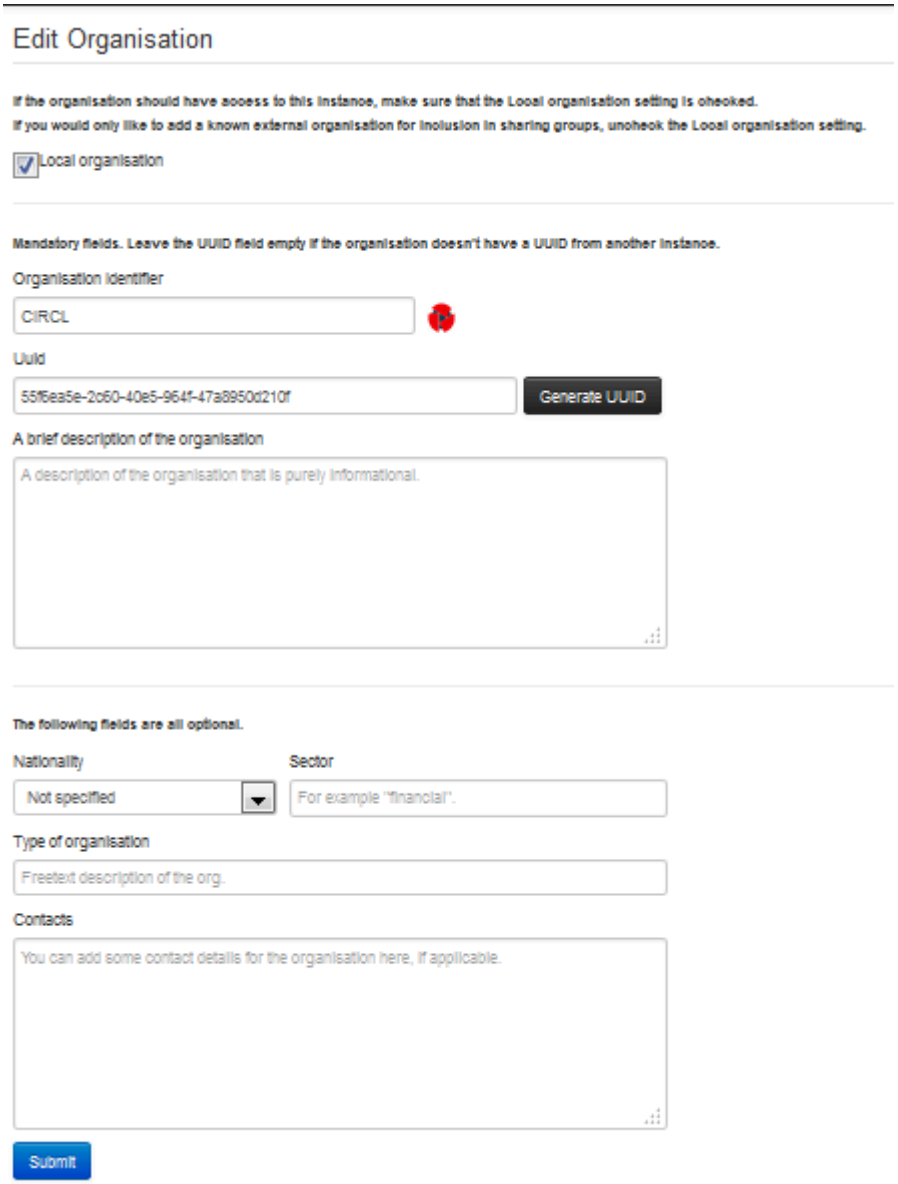
To list all current organisations of the system, just click on List Organisations under the administration menu to the left. There are 3 tabs in this view to filter local organisations, remote organisations or both. The default view displays local organisations. For all views the following columns of information are available:

ID	Name	UUID	Description	Nationality	Sector	Type	Contacts	Added by	Local	Users	Restrictions	Actions
21	abuse.ch	9b086132-8588-49ed-97fd-8578a777822c		Switzerland	Government			admin@admin.test	✔	0		
439		79610c5c-cf3f-48fd-9d6c-cb3b59836cbf		Austria	National	National CSIRT	team@cert.at	admin@admin.test	✔	0		
437	cert.ee	69f98dea-c539-4aa2-a7e4-24a4b559d2dd		Estonia		National CSIRT		admin@admin.test	✔	0		
440	cert.pl	5db98ff5-1554-4d88-88d5-4c43950d2111						admin@admin.test	✔	0		
438	CERT.PT	58e3bcc4-a394-443f-aed2-3065c0a84401		Portugal			cert@cert.pt	admin@admin.test	✔	0		
436	cert.ro	9ef27f29-8080-4879-8627-771f3ae79240						admin@admin.test	✔	0		
2		55f6ea5e-2c60-40e5-964f-47a8950d210f		Luxembourg	GSIRT			admin@admin.test	✔	1		
147	ENISA	a67afdb8-fa0a-4b40-b891-016e6e8af298		International	Government			admin@admin.test	✔	0		

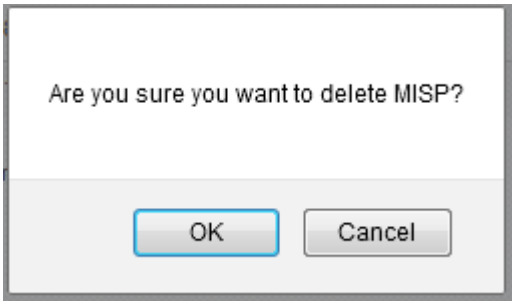
Figure 44.5: List of organisations.

- **Id:** The organisation's automatically assigned ID number.
- **Logo:** Picture of the organisation.
- **Name:** Name of the organisation.
- **Uuid:** Unique identifier of organisation. Share this Uuid when using it between MISP's multi-instance.
- **Description:** Description of the organisation.
- **Nationality:** Country of the organisation.
- **Sector:** Sector defined for the organisation.
- **Type:** Type of organisation.
- **Contacts:** Contacts of organisation.
- **Added by:** Login of the user who added the organisation
- **Local:** Flag defined if the organisation is local or remote.
- **Users:** The amount of users on this instance belonging to the organisation.
- **Actions:** There are 3 options available: edit, delete or display an organisation's information. These options are also available on the left menu when you are on the display view.

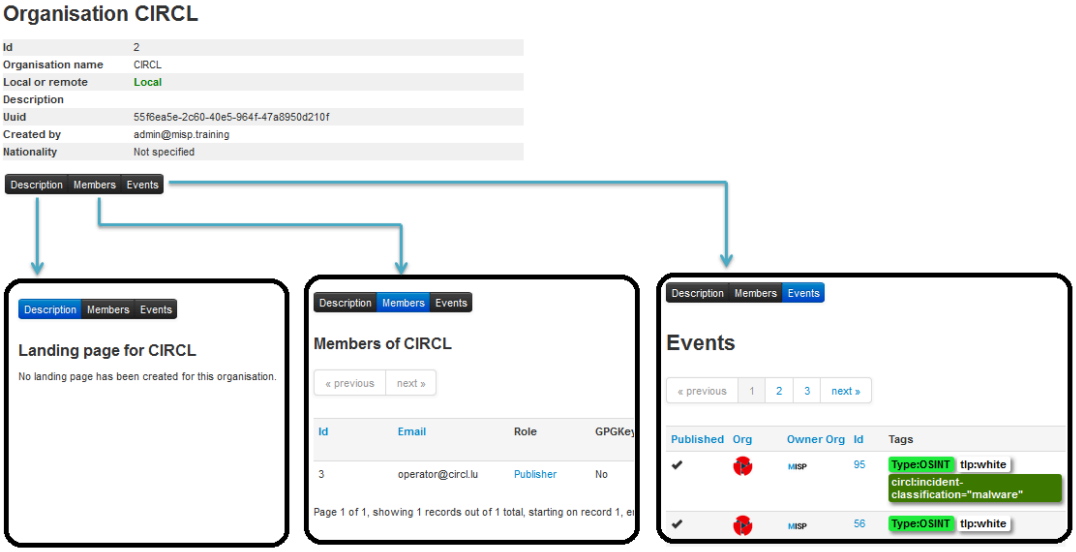
– **Edit Organisation:** Same options of create organisation's view.



– **Delete Organisation:** Use this option for deleting organisation.



– **View Organisation:** Use this option to display information about the selected organisation. In this view, you can display the user belongs to this organisation and events published by organisation.



44.2.3 Merge organisations

Merge Organisation menu is available only in the organisation view, under the left menu. Merging one organisation into another will transfer all users and data from one organisation to a different one. The organisation of which the users and data will be transferred is displayed on the left, the target organisation is displayed on the right.

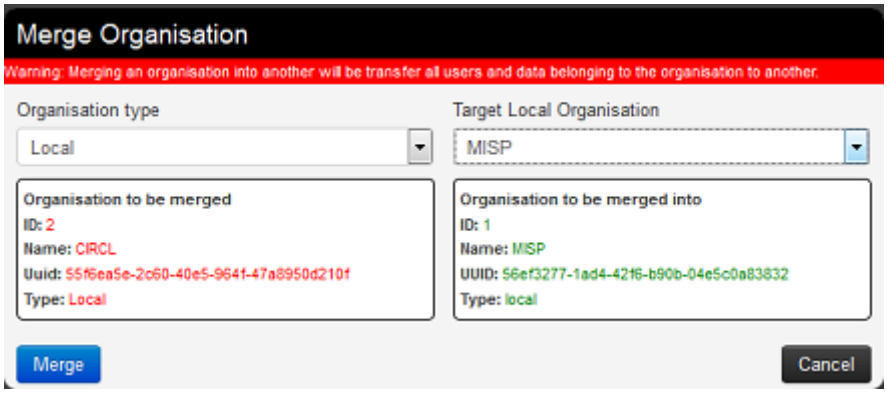


Figure 44.6: Merge organisations.

44.3 Roles

Privileges are assigned to users through **roles**. A role combines a base **permission level** (what the user can do with events) with a set of individual permission flags (access to specific features).

The base permission level is chosen from a single **Permissions** dropdown with four options:

- **Read Only:** Browse the events the user's organisation has access to, without making any changes.
- **Manage Own Events:** Create, modify and delete the user's own events, but not publish them.
- **Manage Organisation Events:** Create events, and modify or delete any event belonging to the user's organisation.
- **Manage and Publish Organisation Events:** All of the above, plus publishing the organisation's events.

On top of the base level, the following individual permissions can be granted (each is a checkbox on the role form):

Permission	Grants
Site Admin	Unrestricted access to all data and functionality on the instance.
Org Admin	Limited administration — create and manage the users of the own organisation.
Sync Actions	Use the account as a synchronisation user; its authentication key can be given to a remote instance's administrator to enable sync.
Internal Sync Actions	Internal sync strategy — distribution levels are not downgraded on pull (for connections within a trusted community).
Authoritative Sync Actions	Treat the source as authoritative; local data is mirrored to match the source (including deletions).
Audit Actions	Access the audit logs of the user's organisation.
Auth key access	Use an authentication key for API access.
Regex Actions	Edit the regular-expression replacement rules (dangerous — a bad expression can cause data loss or infinite loops).
Tagger	Attach and detach existing tags to and from events and attributes.
Tag Editor	Create new tags.
Template Editor	Create and modify event templates.
Sharing Group Editor	Create and modify sharing groups.
Delegations Access	Create delegation requests for own-organisation events.
Sighting Creator	Submit sightings (feedback on attributes).
Object Template Editor	Create and modify MISP object templates.
Galaxy Editor	Create and modify galaxies and galaxy clusters.
Decaying Model Editor	Create and modify decaying models.
ZMQ publisher	Publish data to the ZeroMQ pub/sub channel.
Kafka publisher	Publish data to Kafka.
Warninglist Editor	Manage warninglists.
View Feed Correlations	View feed correlations (carries a performance cost).
Analyst Data Creator	Create and modify analyst data (notes, opinions, relationships).
Skip OTP Reqs	Exempt the account from one-time-password requirements (for service accounts); has no effect unless OTP is required instance-wide.
Server Signing	Access the cryptographic server-signing endpoint.

MISP ships with a set of default roles — **admin** (full site admin), **Org Admin**, **User** (the default role for new users: manage organisation events without publishing), **Publisher**, **Sync user** and **Read Only** — which you can use as they are or as a starting point for your own.

44.3.1 Adding a new role

When creating a new role, you will have to enter a name for the role to be created and set up permissions (as described above) using the drop-down menu and related check-boxes.

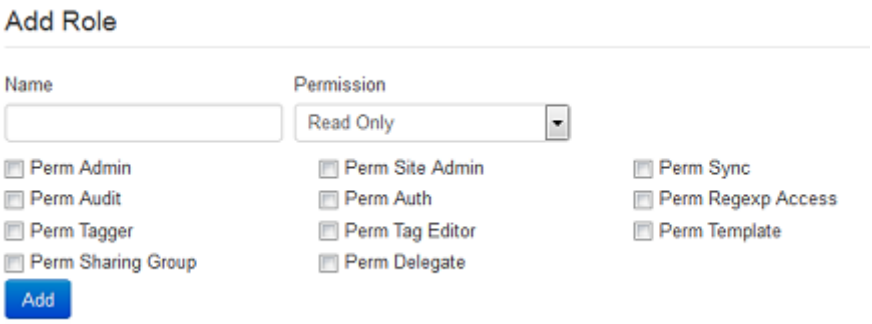


Figure 44.7: Add a new role.

44.3.2 Listing roles

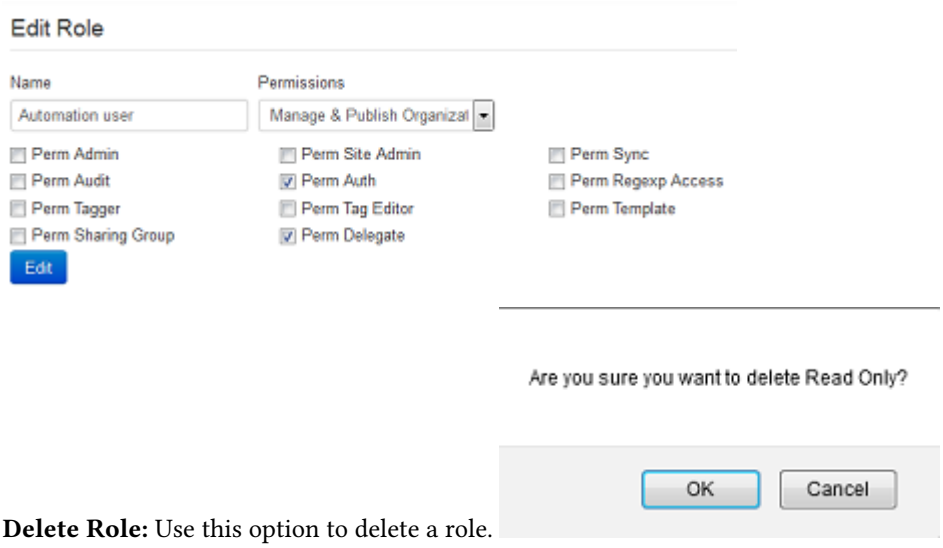
By clicking on the List Roles button, you can view a list of all currently registered roles and their enabled permissions. In addition, you can find buttons that allow you to edit and delete said roles. Keep in mind that you will need to first remove every member from a role before you can delete it.



Figure 44.8: You can Edit or Delete roles using the action buttons to the right in each row. Keep in mind that in order to Delete a role, all members of a Role must be removed from said role before it can be deleted.

- **Id:** The role's automatically assigned ID number.
- **Name:** The name of role.
- **Permission:** One of the 4 permissions: Read Only, Manage My Own Events, Manage Organization Events, Manage & Publish Organisation Events.
- **Permission flags:** A column for each individual permission (see the table above), showing whether the role grants it.
- **Action Buttons:** There are 2 options available: Edit Role or Delete it.

– **Edit Role:** Same options of create role's view.



– **Delete Role:** Use this option to delete a role.

44.4 Tools

MISP has a couple of administrative tools that help administrators keep their instance up to date and healthy. The list of these small tools can change rapidly with each new version, but they should be self-explanatory. Be sure to check this section after each upgrade to a new version, just in case there's a new upgrade script in there - though if this is the case it will be mentioned in the upgrade instructions.

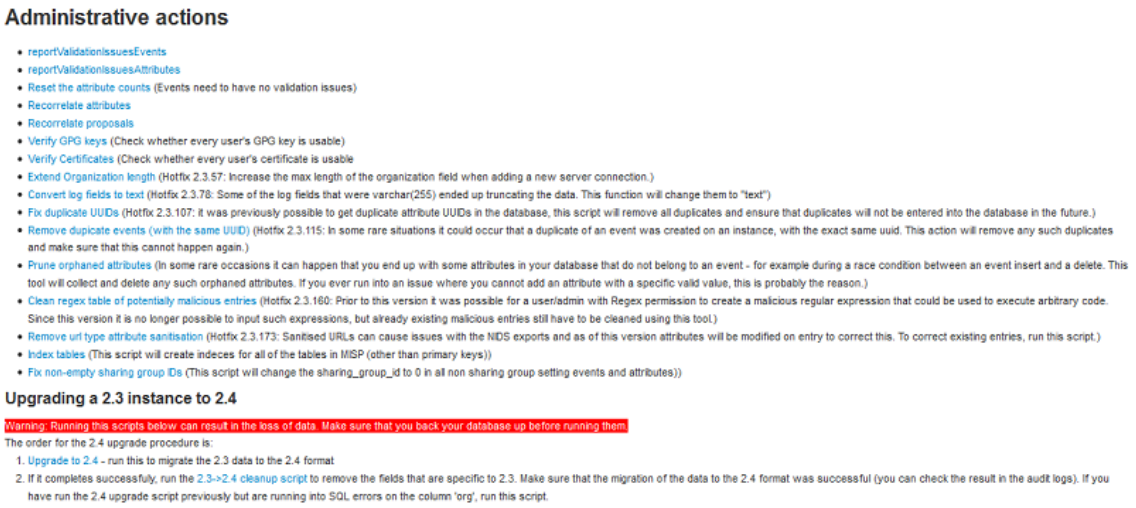


Figure 44.9: Administrative Tools.

44.5 Server settings and maintenance

Since version 2.3, MISP has a settings and diagnostics tool that allows site-admins to manage and diagnose their MISP installation. You can access this by navigating to Administration - Server settings & Maintenance.

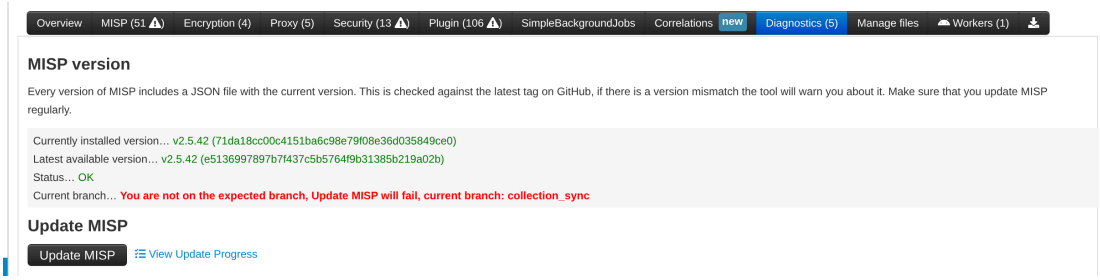


Figure 44.10: Server settings overview with all of the tabs explained.

The settings and diagnostics tool is split up into several aspects, all accessible via the tabs on top of the tool. For any unset or incorrectly set setting, or failed diagnostic a number next to the tab name will indicate the number and severity of the issues. If the number is written with a red font, it means that the issue is critical. First, let's look at the various tabs: * **Overview**: General overview of the current state of your MISP installation * **MISP settings**: Basic MISP settings. This includes the way MISP handles the default settings for distribution settings, whether background jobs are enabled, etc * **GnuPG settings**: GnuPG related settings. * **Proxy settings**: HTTP proxy related settings. * **Security settings**: Settings controlling brute-force protection and the application's salt key. * **Misc settings**: Settings controlling debug options, please ensure that debug is always disabled on a production system. * **Diagnostics**: The diagnostics tool checks if all directories that MISP uses to store data are writeable by the apache user. Also, the tool checks whether the STIX libraries and GnuPG are working as intended. * **Workers**: Shows the background workers (if enabled) and shows a warning if they are not running. Admins can also restart the workers here. * **Download report**: Download a report of all the settings visible in the tool, in JSON format.

Priority	Setting	Value	Description	Error Message
Critical	MISP.baseurl	http://localhost:5007	The base url of the application (in the format https://www.mymispinstance.com or https://myserver.com/misp). Several features depend on this setting being correctly set to function.	
Critical	MISP.external_baseurl	http://localhost:5007	The base url of the application (in the format https://www.mymispinstance.com) as visible externally/by other MISPs. MISP will encode this URL in sharing groups when including itself. If this value is not set, the baseurl is used as a fallback.	
Critical	MISP.live	true	Unless set to true, the instance will only be accessible by site admins.	
Critical	MISP.language	eng	Select the language MISP should use. The default is english.	
Critical	MISP.enable_themes CLI only	true	Enable themes for users of the instance. Currently this is used to allow users to opt-in to a the various preview/beta modes.	
Critical	MISP.correlation_engine	Default Correlation Engine	Choose which correlation engine to use. MISP defaults to the default engine, maintaining all data in the database whilst enforcing ACL rules on any non site-admin user. This is recommended for any MISP instance with multiple organisations. If you are an endpoint MISP, consider switching to the much leaner and faster No ACL engine.	
Critical	MISP.correlation_limit	20	Set a value for the maximum number of correlations a value should have before MISP will refuse to correlate it (extremely over-correlating values are rarely useful from a correlation perspective).	

Figure 44.11: The settings tabs explained.

Each of the setting pages is a table with each row representing a setting. Coloured rows indicate that the setting is incorrect / not set and the colour determines the severity (red = critical, yellow = recommended, green = optional). The columns are as follows: * **Priority**: The severity of the setting. * **Setting**: The setting name. * **Value**: The current value of the setting. * **Description**: A description of what the setting does. * **Error Message**: If the setting is incorrect / not set, this field will let the user know what is wrong.

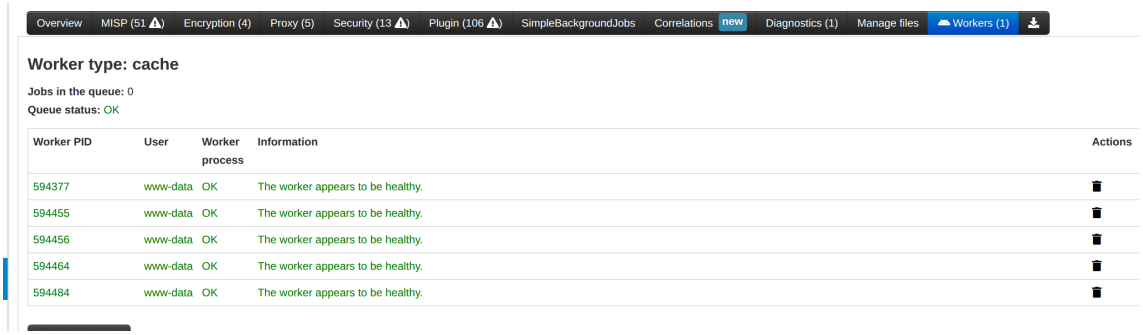


Figure 44.12: The workers tab.

The workers tab shows the workers that MISP can use. In MISP 2.5 the background worker system is **SimpleBackgroundJobs**, managed by Supervisor and backed by Redis; the legacy CakeResque worker system has been removed. You can restart the workers from this page with the "restart all workers" button, but starting and stopping them is normally handled by Supervisor (the `misp-workers` service installed by the setup scripts).

The tab is grouped into one section per queue — MISP has 6 queues: `default`, `email`, `cache`, `prio`, `update` and `scheduler`. Each section reports **Jobs in the queue** and an overall **Queue status**, followed by a table of the worker processes currently monitoring that queue:

- **Worker PID**: The process ID of the running worker.
- **User**: The system user the worker runs as (typically `www-data`).
- **Worker process**: `OK` when the process is alive.
- **Information**: A health message (for example, "The worker appears to be healthy.").
- **Actions**: Stop an individual worker.

44.5.1 Worker types

Each worker monitors one Redis-backed queue:

- **default**: General-purpose background jobs that are not handled by a more specific queue.
- **email**: Sending notification and alert e-mails.
- **cache**: Building caches — feed caches, warninglist caches and export caches.
- **prio**: High-priority, often user-triggered jobs (for example publishing an event) so they are not held up behind long-running tasks.
- **update**: Running MISP updates and database schema migrations.
- **scheduler**: Enqueues recurring/scheduled jobs onto the other queues at the configured times.

44.5.2 Restarting or starting the workers

Even if the workers are stopped, any actions queued for them are simply on hold — nothing is lost. Starting the workers again resumes any pending operations.

Under Supervisor the workers start automatically (including on boot); you can also restart them from the **Workers** tab in the UI. From the command line, control the worker processes with `supervisorctl` and inspect the queues with the `Worker` console shell (see Managing the background workers):

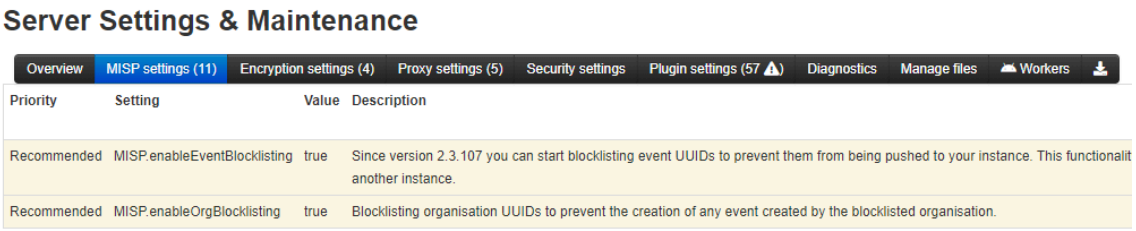
```
sudo supervisorctl status misp-workers:*
sudo supervisorctl restart misp-workers:*
```

44.6 Blocklists and block rules

It is possible to block certain events or organisations from ever being added to the system. Administrators can add, edit or delete blocklisted items. The appropriate pages are linked in the Administration menu.

44.6.1 Event blocklist

Blocklisting an event prevents the event from being added on the instance. Blocklisting an existing event will not result in the event being removed. The event will still be editable as well. Blocklisting events functionality is enabled by default. If blocklisting events is enabled, deleted events will automatically be added to the event blocklist. Enabling/disabling event blocklisting can be done using the MISP settings view.



44.6.1.1 Blocklisting an event

The blocklist event screen can be accessed through the main administration menu. You can enter the UUID of one event or a list of event UUIDs (one per line). If the optional fields creating organisation, event info or comment are filled in, their values will be added for all added UUIDs.

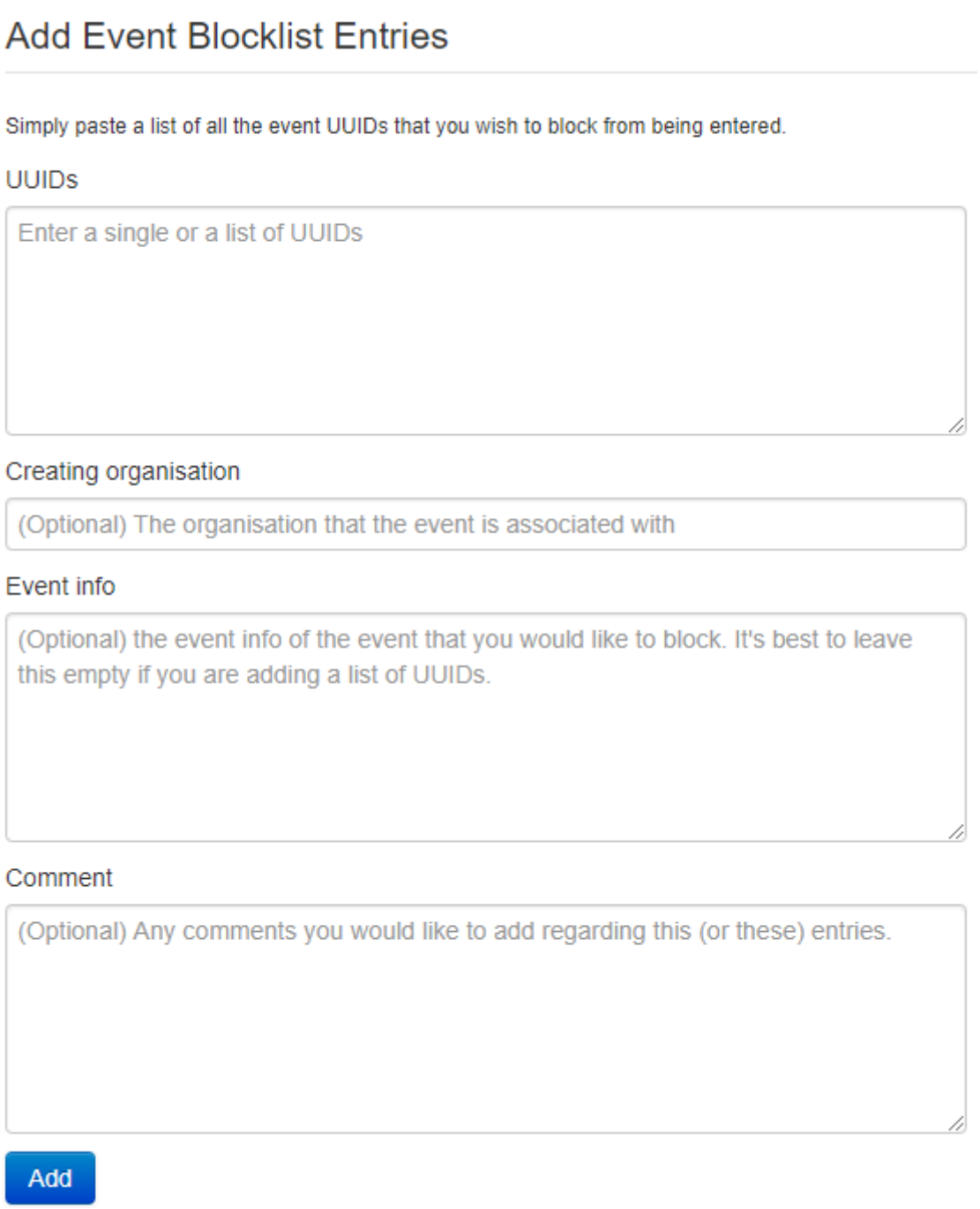
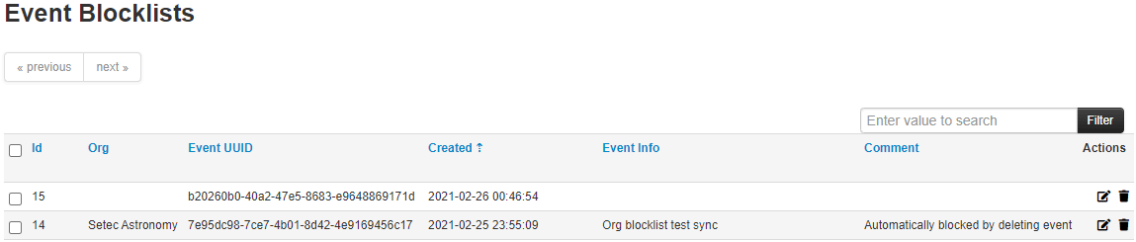


Figure 44.13: Add event blocklist entries view

44.6.1.2 Viewing event blocklist entries

The list of blocklisted events can be accessed through the main administration menu. You can delete a blocklist entry or access the edit screens for specific blocklisted events from here.



44.6.2 Event block rules

Event block rules allow you to add a simple tag filter to block events from being added or synced.

An example of a rule can be found below:

```
{
"tags": ["tag1", "tag2"]
}
```

The rule will block: - Syncing of events with “tag1” or “tag2” - Direct adding of events with “tag1” or “tag2” in one go, for example using /events/add

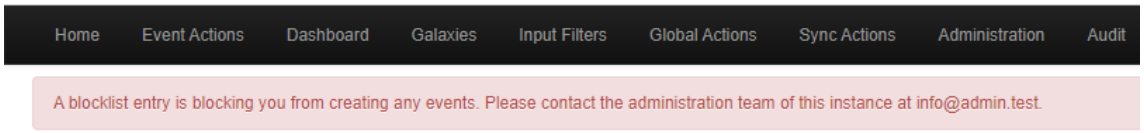
The rule will not block: - The adding of “tag1” or “tag2” to an existing event through non syncing actions, for example by adding it via the graphical user interface.

It is not possible to add more complex rules with boolean logic (NOT, AND).

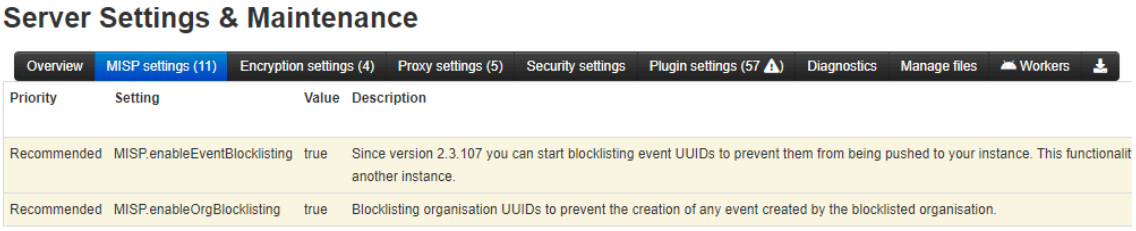
Event block rules are managed under **Administration → Event Block Rules** and are stored as a single JSON rule.

44.6.3 Organisation blocklist

Blocklisting an organisation prevents the creation of any event by the blocklisted organisation. It does not prevent a local user from the blocklisted organisation from logging in or viewing data.



When syncing, events created by blocklisted organisations will not be added to the instance. Updates will also not propagate. A user from a blocklisted organisation can still edit an event from the blocklisted organisation locally though. Blocklisting organisations functionality is enabled by default. Enabling/disabling organisation blocklisting can be done using the MISP settings view.



44.6.3.1 Blocklisting an organisation

The blocklist organisation screen can be accessed through the main administration menu. You can enter the UUID of one organisation or a list of organisations UUIDs (one per line). If the optional fields organisation name or comment are filled in, their values will be added for all added UUIDs.

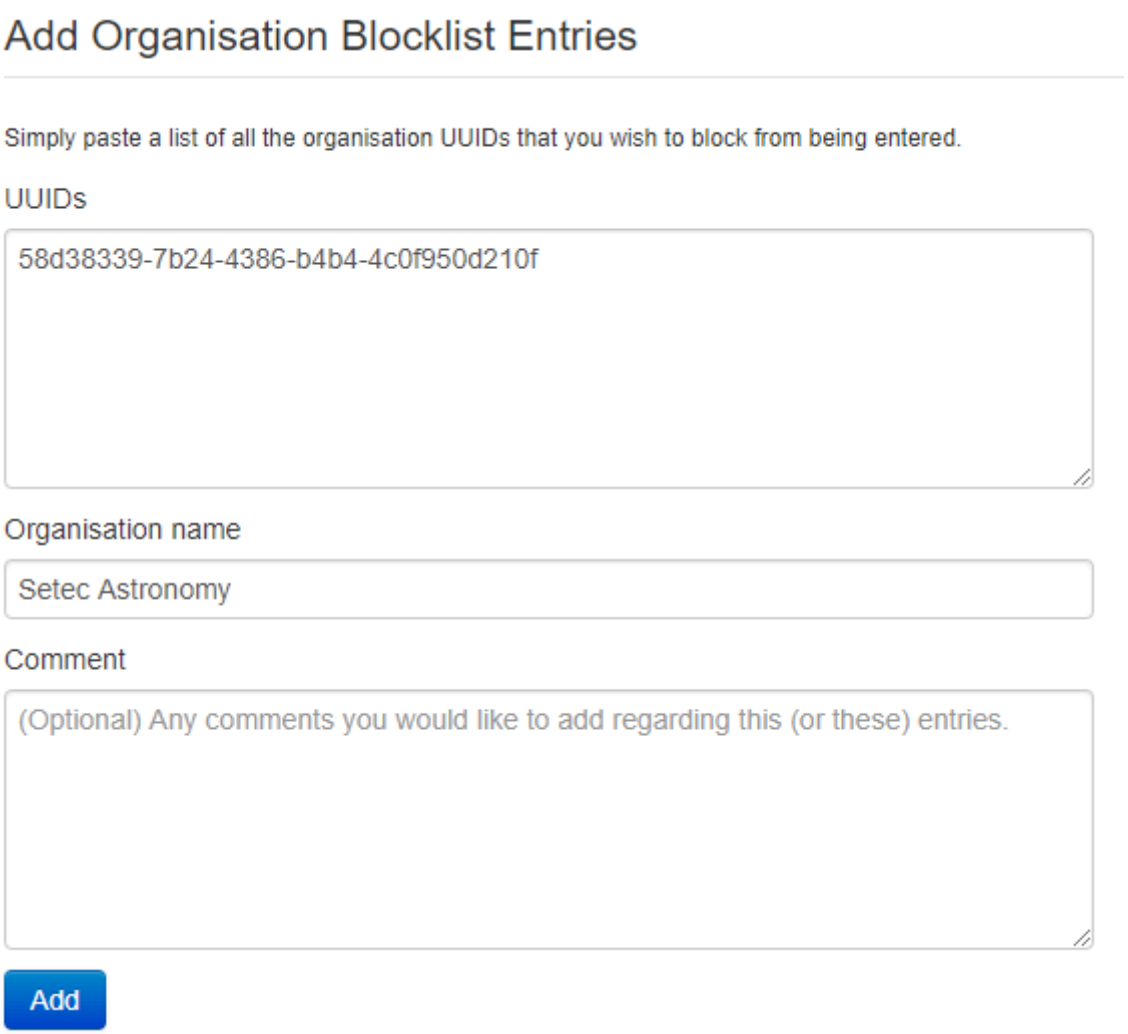
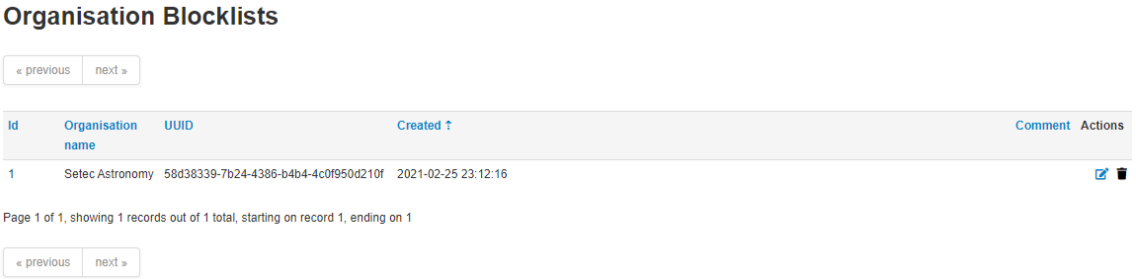


Figure 44.14: Blocklist organisation view

44.6.3.2 Viewing organisation blocklist entries

The list of blocklisted organisations can be accessed through the main administration menu. You can delete a blocklist entry or access the edit screens for specific blocklisted organisations from here.



44.6.4 Sighting blocklist

Blocklisting an organisation for sightings prevents any sightings created by that organisation from being stored on the instance (locally or via sync) — useful for filtering out a noisy or untrusted source of sighting feedback. Sighting blocklisting is enabled by default (`MISP.enableSightingBlocklisting`) and is managed from the administration menu; entries are the UUIDs of the organisations to block.

44.6.5 Galaxy cluster blocklist

Blocklisting a galaxy cluster prevents that cluster (identified by its UUID) from being captured or synchronised onto the instance — for example to keep a deprecated or unwanted cluster out of your data. There is no on/off setting for this blocklist; it is always active. Entries are galaxy-cluster UUIDs, with optional cluster info, creator organisation and comment.

44.6.6 Analyst data blocklist

Blocklisting analyst data prevents specific notes, opinions or relationships (identified by their UUIDs) from being captured or synchronised onto the instance. As with the galaxy-cluster blocklist there is no enable/disable setting. Entries are analyst-data UUIDs, with optional info, creator organisation and comment.

44.7 Import Regexp

The system allows administrators to set up rules for regular expressions that will automatically alter newly entered or imported events.

44.7.1 The purpose of Import Regexp entries

They can be used for several things, such as unifying the capitalisation of file paths for more accurate event correlation or to automatically censor the usernames and use system path variable names (changing C:.exe to %APPDATA%.exe). The second use is blocking, if a regular expression is entered with a blank replacement, any event info or attribute value containing the expression will not be added. Please make sure the entered regexp expression follows the preg_replace pattern rules as described here

44.7.2 Adding and modifying entries

Administrators can add, edit or delete regular expression rules, these “expressions” are made up of a regex pattern that the system searches for and a replacement for the detected pattern.

Id ↓	Regexp	Replacement	Type
1	/.:.ProgramData./i	%ALLUSERSPROFILE%\\	ALL
2	/.:.Documents and Settings.All Users./i	%ALLUSERSPROFILE%\\	ALL
3	/.:.Program Files.Common Files./i	%COMMONPROGRAMFILES%\\	ALL
4	/.:.Program Files (x86).Common Files./i	%COMMONPROGRAMFILES(x86)%\\	ALL
5	/.:.Users.(\w+).AppData.Local.Temp./i	%TEMP%\\	ALL
6	/.:.ProgramData./i	%PROGRAMDATA%\\	ALL
7	/.:.Program Files./i	%PROGRAMFILES%\\	ALL

Figure 44.15: Add, edit or remove Regexp entries that will affect all newly created attributes here.

44.8 Managing the Signature allowedlist

The signature allowedlist view, accessible through the administration menu on the left, allows administrators to create and maintain a list of addresses that are allowlisted from ever being added to the NIDS signatures. Addresses listed here will be commented out when exporting the NIDS list.

44.8.1 Allowlisting an address

While in the allowedlist view, click on New Allowedlist on the left to bring up the “add allowedlist” view to add a new address.

44.8.2 Managing the list

When viewing the list of allowlisted addresses, the following data is shown: The ID of the allowlist entry (assigned automatically when a new address is added), the address itself that is being allowlisted and a set of controls allowing you to delete the entry or edit the address.

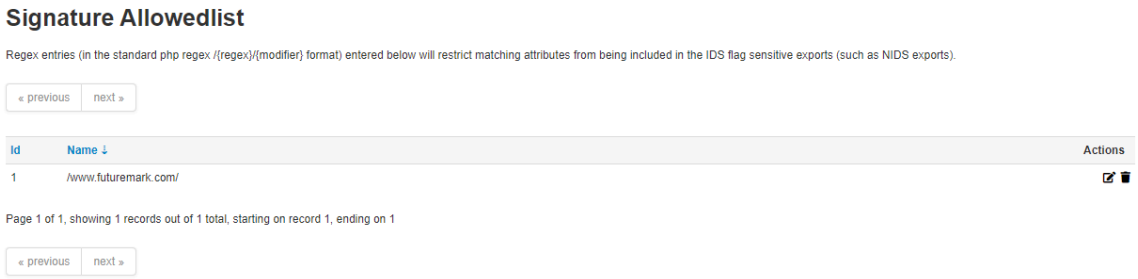


Figure 44.16: You can edit or delete currently allowlisted addresses using the action buttons on this list.

44.9 Managing correlations

MISP automatically correlates attributes that share the same value across events. Administrators have several tools to tune what correlates.

44.9.1 Correlation exclusions

Correlation exclusions let you exclude certain **values** from the correlation engine. Values can be exact 1:1 matches or substring searches denoted with a leading and/or trailing `%`:

- `https://www.google.com/%` matches anything starting with `https://www.google.com/`
- `%google.com%` matches anything that contains `google.com`

After adding an exclusion, new values that match it will no longer correlate. To remove correlations that already exist, run the cleaner via the **Clean up correlations** button.

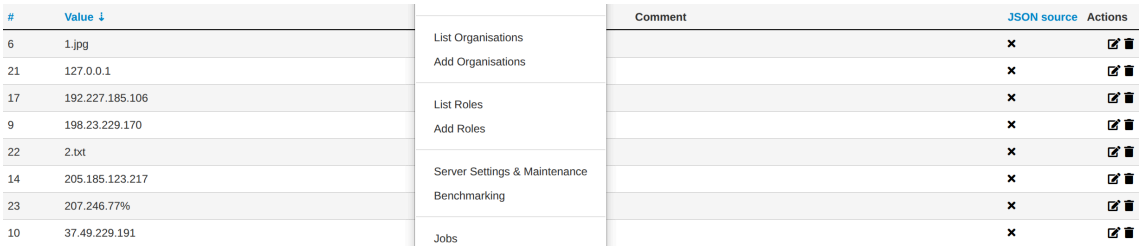


Figure 44.17: Index view of correlation exclusions, showing leading- and trailing-wildcard examples.

44.9.2 Correlation rules

Where exclusions suppress a specific value everywhere, **correlation rules** prevent whole **events** from correlating with one another based on a selector. A rule matches by creator organisation, owner organisation, event ID, or a substring of the event info, and stops the selected events from correlating. Executing a rule can also retroactively remove correlations that already exist between the matching events. Correlation rules are managed under **Administration → Correlation rules**.

44.9.3 Over-correlating values

Some values (for example a very common string) correlate so widely that the matches become meaningless and expensive to compute. MISP automatically stops correlating a value once its correlation count exceeds `MISP.correlation_limit` (default 20). You can review the affected values — and which ones are currently over-correlating — under **Administration → Over-correlating values**.

44.9.4 Top correlations

Administration → Top Correlations shows a ranked list of the values that correlate the most. It is a good starting point for finding candidates for exclusions or over-correlation suppression.

44.9.5 Correlation engine

The correlation engine is selected with the `MISP.correlation_engine` setting (Default, NoAcl or OnDemand). On very large instances the *OnDemand* engine computes correlations only when an event is viewed, trading immediate correlation for a lower write-time cost.

44.10 Logs

Users with audit permission can browse the actions that MISP records. In 2.5 the top-menu **Logs** entry groups four views:

- **Application Logs** — the legacy application log: logins and logouts, failed authentication attempts, and a range of application events.
- **Audit Logs** — the detailed audit trail of data changes (see below); shown once the audit log is enabled.
- **Access Logs** — a per-request HTTP access log (see below); site admins only.
- **Search Logs** — a filtered search over the application log.

44.10.1 Audit logs

The audit log records every change to the data model — additions, edits, soft and hard deletes, tagging, galaxy attachment, publishing, and more — across events, attributes, objects, users, organisations, roles, servers and dozens of other models. Crucially it stores the **field-level changes** (old and new values) for each entry, with secrets such as passwords and API keys redacted.

Enable it with the **`MISP.log_new_audit`** setting (Administration → Server Settings → MISP tab); the audit index warns you if it is not yet enabled. Browse it under **Logs → Audit Logs**, filtering by user, organisation, model, action, event, IP, request type and date. Site admins see everything, organisation admins see their own organisation, and regular users see only their own actions. The per-event **View Event History** action is backed by the same audit log.

Id ↑	Created	IP	Email	Org	Model	Model ID	Action	Title	Change
2335267	2026-07-08 07:42:57	127.0.0.1	admin@admin.test	Iglocska	User	576	add	User (576): sync@sample.test	server_id () => (0), autoalert () => (0), invited_by () => (1), nids_sid () => (7165264), termsaccepted () => (1), newsread () => (0), role_id () => (5), change_pw () => (0), contactalert () => (0), current_login () => (0), last_login () => (0), force_logout () => (0), external_auth_required () => (0), last_api_access () => (0), notification_daily () => (0), notification_weekly () => (0), notification_monthly () => (0), email () => (sync@sample.test), org_id () => (873), date_created () => (1783496576), last_pw_change () => (1783496576), authkey (*****) => (*****), date_modified () => (1783496576)
2335265	2026-07-08 07:42:56	127.0.0.1	admin@admin.test	Iglocska	User	575	add	User (575): readonly@sample.test	server_id () => (0), autoalert () => (0), invited_by () => (1), nids_sid () => (5005074), termsaccepted () => (1), newsread () => (0), role_id () => (6), change_pw () => (0), contactalert () => (0), current_login () => (0), last_login () => (0), force_logout () => (0), external_auth_required () => (0), last_api_access () => (0), notification_daily () => (0), notification_weekly () => (0), notification_monthly () => (0), email () => (readonly@sample.test), org_id () => (873), date_created () => (1783496576), last_pw_change () => (1783496576), authkey (*****) => (*****), date_modified () => (1783496576)
2335263	2026-07-08 07:42:56	127.0.0.1	admin@admin.test	Iglocska	User	574	add	User (574): publisher@sample.test	server_id () => (0), autoalert () => (0), invited_by () => (1), nids_sid () => (5298936), termsaccepted () => (1), newsread () => (0), role_id () => (4), change_pw () => (0), contactalert () => (0), current_login () => (0), last_login () => (0), force_logout () => (0), external_auth_required () => (0), last_api_access () => (0), notification_daily () => (0), notification_weekly () => (0), notification_monthly () => (0), email () => (publisher@sample.test), org_id () => (873), date_created () => (1783496576), last_pw_change () => (1783496576), authkey (*****) => (*****), date_modified () => (1783496576)
2335261	2026-07-08 07:42:56	127.0.0.1	admin@admin.test	Iglocska	User	573	add	User (573): analyst@sample.test	server_id () => (0), autoalert () => (0), invited_by () => (1), nids_sid () => (4604616), termsaccepted () => (1), newsread () => (0), role_id () => (3), change_pw () => (0), contactalert () => (0), current_login () => (0), last_login () => (0), force_logout () => (0), external_auth_required () => (0), last_api_access () => (0), notification_daily () => (0), notification_weekly () => (0), notification_monthly () => (0), email () => (analyst@sample.test), org_id () => (873), date_created () => (1783496576), last_pw_change () => (1783496576), authkey (*****) => (*****), date_modified () => (1783496576)

Figure 44.18: The application log index (Logs → Application Logs).

44.10.2 Access logs

The access log records one row per HTTP request — the user, organisation, controller/action, URL, response code, duration, memory and query counts, and (optionally) the client IP, request body and full SQL query log. It is written when request logging is enabled via **`MISP.log_paranoid`** (all requests), **`MISP.log_paranoid_api`** (authenticated API requests only), or per-user monitoring. Related

settings control the verbosity — `MISP.log_client_ip`, `MISP.log_paranoid_include_post_body` and `MISP.log_paranoid_include_sql_queries`. Browse it under **Logs → Access Logs** (site admins only), with per-request drill-downs into the stored body and query log.

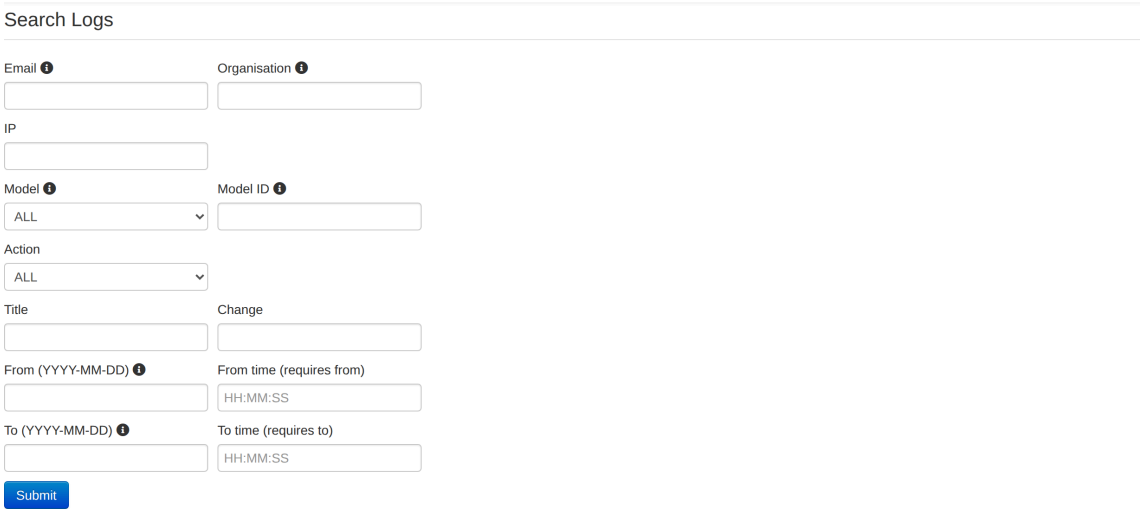


Figure 44.19: Searching the application log (Logs → Search Logs).

44.11 Background Processing

If enabled, MISP can delegate a lot of the time intensive tasks to the background workers. These will then be executed in sequence, allowing the users of the instance to keep using the system without a hiccup and without having to wait for the process to finish. It also allows for certain tasks to be scheduled and automated.

44.11.1 Command line tools for the background workers

In MISP 2.5 the background workers run under **SimpleBackgroundJobs** (Supervisor + Redis); the old CakeResque commands no longer apply. The worker processes are controlled by Supervisor:

```
sudo supervisorctl status misp-workers:*
sudo supervisorctl restart misp-workers:*
```

To inspect the queues and jobs, use the `Worker` console shell from `/var/www/MISP/app/Console`:

- **showQueues**: Show the configured queues and how many jobs are waiting in each.
- **flushQueue [queue]**: Remove all queued jobs from the given queue.
- **showJobStatus [worker]**: Show the status of the jobs handled by the workers.

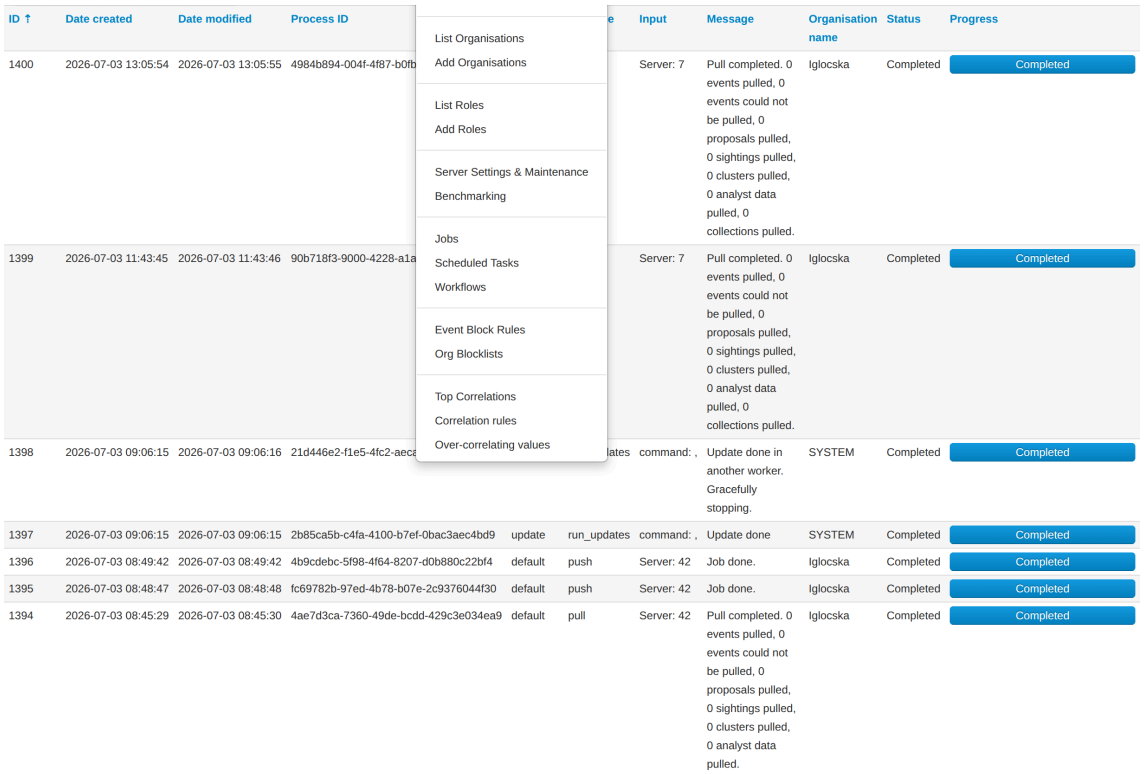
```
sudo -u www-data /var/www/MISP/app/Console/cake Worker showQueues
```

44.11.2 Monitoring the Background Processes

The "Jobs" menu item within the Administration menu allows site admins to get an overview of all of the current and past scheduled jobs. Admins can see the status of each job, and what the queued job is trying to do. If a job fails, it will try to set an error message here too. The following columns are shown in the jobs table:

- **Id**: The job's ID (this is the ID of the job's metadata stored in the default datastore, not to be confused with the process ID stored in the redis database and used by the workers)
- **Process**: The process's ID.
- **Worker**: The name of the worker queue. If background jobs are enabled there are 6 queues (default, email, cache, prio, update and scheduler); the scheduler queue should never show up in the jobs table.
- **Job Type**: The name of the queued job.

- **Input**: Shows a basic input handled by the job - such as “Event:50” for a publish email alert job for event 50.
- **Message**: This will show what the job is currently doing or alternatively an error message describing why a job failed.
- **Org**: The string identifier of the organisation that has scheduled the job.
- **Status**: The status reported by the worker.
- **Retries**: Currently unused, it is planned to introduced automatic delayed retries for the background processing and thus add resilience.
- **Progress**: A progress bar showing how the job is coming along.



ID ↑	Date created	Date modified	Process ID			Input	Message	Organisation name	Status	Progress
1400	2026-07-03 13:05:54	2026-07-03 13:05:55	4984b894-004f-4f87-b0fb			Server: 7	Pull completed. 0 events pulled, 0 events could not be pulled, 0 proposals pulled, 0 sightings pulled, 0 clusters pulled, 0 analyst data pulled, 0 collections pulled.	Iglocska	Completed	Completed
1399	2026-07-03 11:43:45	2026-07-03 11:43:46	90b718f3-9000-4228-a1a			Server: 7	Pull completed. 0 events pulled, 0 events could not be pulled, 0 proposals pulled, 0 sightings pulled, 0 clusters pulled, 0 analyst data pulled, 0 collections pulled.	Iglocska	Completed	Completed
1398	2026-07-03 09:06:15	2026-07-03 09:06:16	21d446e2-f1e5-4fc2-aeca		lates	command: ,	Update done in another worker. Gracefully stopping.	SYSTEM	Completed	Completed
1397	2026-07-03 09:06:15	2026-07-03 09:06:15	2b85ca5b-c4fa-4100-b7ef-0bac3aec4bd9	update	run_updates	command: ,	Update done	SYSTEM	Completed	Completed
1396	2026-07-03 08:49:42	2026-07-03 08:49:42	4b9cdebc-5f98-4f64-8207-d0b880c22bf4	default	push	Server: 42	Job done.	Iglocska	Completed	Completed
1395	2026-07-03 08:48:47	2026-07-03 08:48:48	fc69782b-97ed-4b78-b07e-2c9376044f30	default	push	Server: 42	Job done.	Iglocska	Completed	Completed
1394	2026-07-03 08:45:29	2026-07-03 08:45:30	4ae7d3ca-7360-49de-bcdd-429c3e034ea9	default	pull	Server: 42	Pull completed. 0 events pulled, 0 events could not be pulled, 0 proposals pulled, 0 sightings pulled, 0 clusters pulled, 0 analyst data pulled.	Iglocska	Completed	Completed

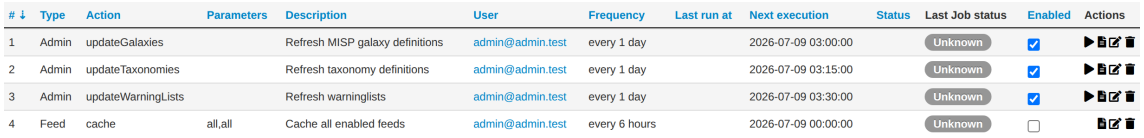
Figure 44.20: Site administrators can monitor the process of all queued jobs here.

44.11.3 Scheduling recurring tasks

Site administrators can schedule recurring maintenance tasks under **Administration → Scheduled Tasks**. In 2.5 this is a reworked, cron-style scheduler driven by the Supervisor-managed `scheduler` worker (it requires that worker to be running, or the page shows a warning). You add tasks explicitly rather than editing a fixed list, choosing a **type** and the matching action:

- **Server** — `pull`, `push` or `cache` against a synchronisation server.
- **Feed** — `fetch` or `cache` a feed (or all feeds).
- **Workflow** — run an ad-hoc workflow.
- **Periodic Summary** — send the periodic summary notification.
- **TAXII** — push to a TAXII server.
- **Admin** — the JSON/definition updates: `updateGalaxies`, `updateTaxonomies`, `updateWarningLists`, `updateNoticeLists` and `updateObjectTemplates`.

Each task row shows its type/action and parameters, the owning user, the frequency (every N seconds/minutes/hours/days), the last and next run times, the status of its last job, and an enable/disable toggle. Rows can be forced to run immediately, edited, deleted, or have their job logs inspected.



#	Type	Action	Parameters	Description	User	Frequency	Last run at	Next execution	Status	Last Job status	Enabled	Actions
1	Admin	updateGalaxies		Refresh MISP galaxy definitions	admin@admin.test	every 1 day		2026-07-09 03:00:00		Unknown	☑	
2	Admin	updateTaxonomies		Refresh taxonomy definitions	admin@admin.test	every 1 day		2026-07-09 03:15:00		Unknown	☑	
3	Admin	updateWarningLists		Refresh warninglists	admin@admin.test	every 1 day		2026-07-09 03:30:00		Unknown	☑	
4	Feed	cache	all,all	Cache all enabled feeds	admin@admin.test	every 6 hours		2026-07-09 00:00:00		Unknown	☐	

Figure 44.21: Site administrators can schedule recurring tasks on this page.

For anything more elaborate, prefer real cron jobs built from the console commands (see Automating certain console tasks).

44.12 Other 2.5 administration features

MISP 2.5 ships a number of features that administrators configure or manage. Each is summarised here with where to find it.

44.12.1 Advanced authentication keys

With the `Security.advanced_authkeys` setting enabled, each user can hold multiple API keys, each with its own comment, expiry and optional IP allowlist. Keys are stored hashed and are shown only once, at creation — they cannot be recovered afterwards. Manage them under **Administration → List Auth Keys**.

44.12.2 Decaying models

Decaying models score how relevant an indicator remains over time. Administrators create and edit models, and map them to attribute types, under **Global Actions → List Decaying Models**, and can experiment with the **Decaying Models Tool**. The *Decaying Model Editor* role permission grants access.

44.12.3 Workflows

Workflows attach trigger-based automation graphs to MISP events — for example blocking or enriching an event on publish. They are disabled by default; enable them with `Plugin.Workflow_enable` and manage them under **Administration → Workflows**.

44.12.4 Cerebrate

Cerebrate is a companion platform for managing organisations and sharing communities. Connect a Cerebrate node under **Sync Actions → Cerebrates** to pull organisation and sharing-group definitions into your instance.

44.12.5 TAXII servers

MISP can push data to TAXII 2.x collections. Configure the server endpoints under **Sync Actions → TAXII Servers**.

44.12.6 Dashboards

The **Dashboard** (top menu) is a configurable set of widgets — recent events, statistics, trends and more — with importable and exportable templates.

44.12.7 Benchmarking

When enabled with `Plugin.Benchmarking_enable`, MISP records per-user and per-endpoint execution times and query load, viewable under **Administration → Benchmarking**. It is a diagnostic aid; leave it off during normal operation.

44.13 Various administration tips & tricks

44.13.1 Setting a Publish Alert Filter

To regulate the reception of e-mail from MISP it is possible to create filters. Each individual user account can apply such filter.

The filter can be configured by the user but also by the organization administrator.

After login goto Administration -> Set User Setting:

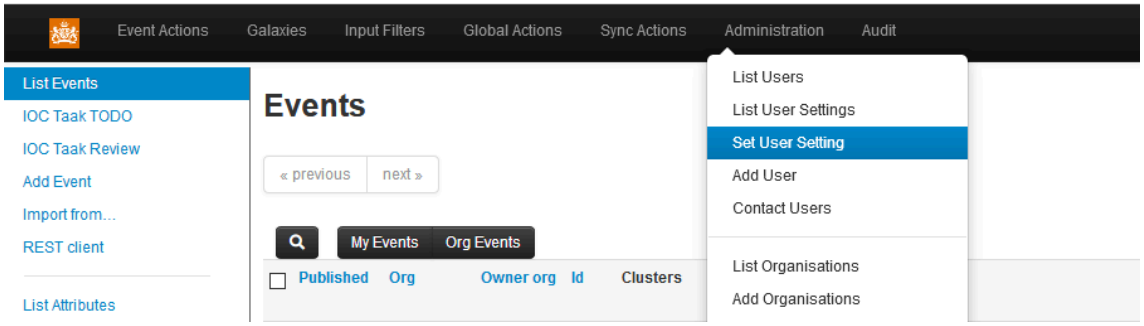


Figure 44.22: Set User settings

A new screen appears. Make sure the “Setting” drop down box shows “publish_alert_filter”:

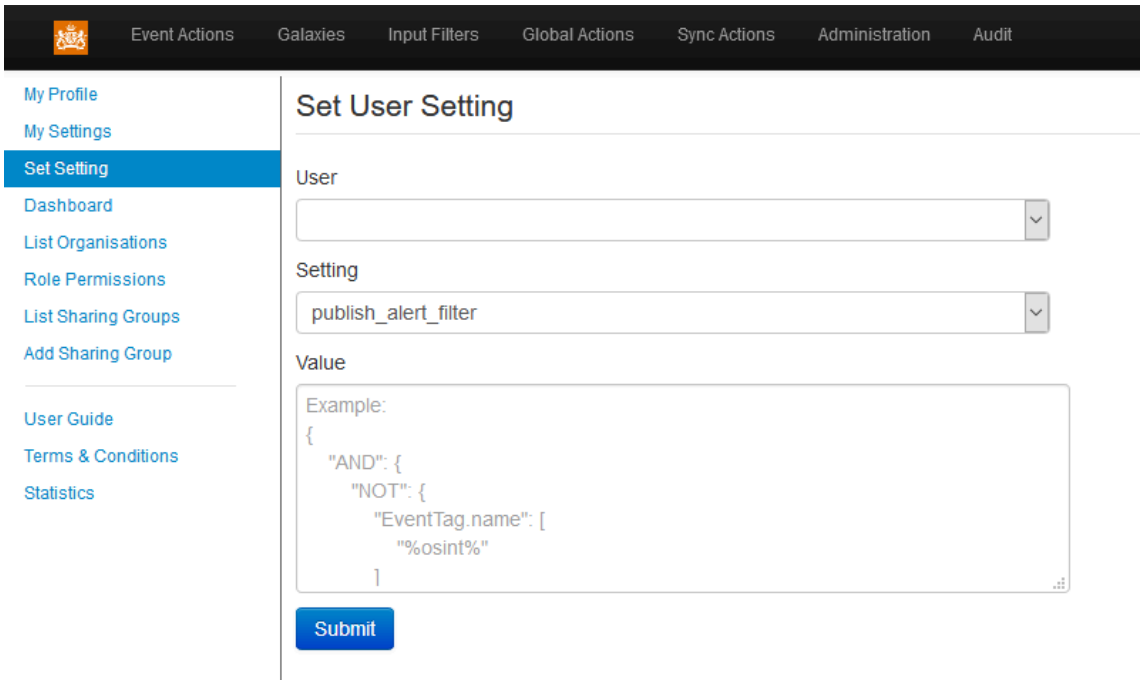


Figure 44.23: Set User settings

The text field “Value” contains the filter, which needs to be provided in JSON format. Important JSON-objects which can be used here go by the name AND“, “OR” and “NOT”. These should be structured in a logical tree.

The filtering can be applied to tags, the publishing organization and the threat level. Valid filters:

- AttributeTag.name
- EventTag.name
- Tag.name (checks against both event and attribute tags)
- Orgc.uuid (creator org uuid)
- Orgc.name (creator org name)

- ThreatLevel.name

In the following example, all notifications will be filtered which carry 'tlp.white' and 'tlp.green' in the name of the tag:

```
{

    "NOT": {

        "Tag.name" : [ "tlp.white", "tlp.green" ]

    }

}
```

The publish_alert_filter setting allows one filter definition to be active.

After applying the configuration, the filter will show up in the "My Settings" menu:

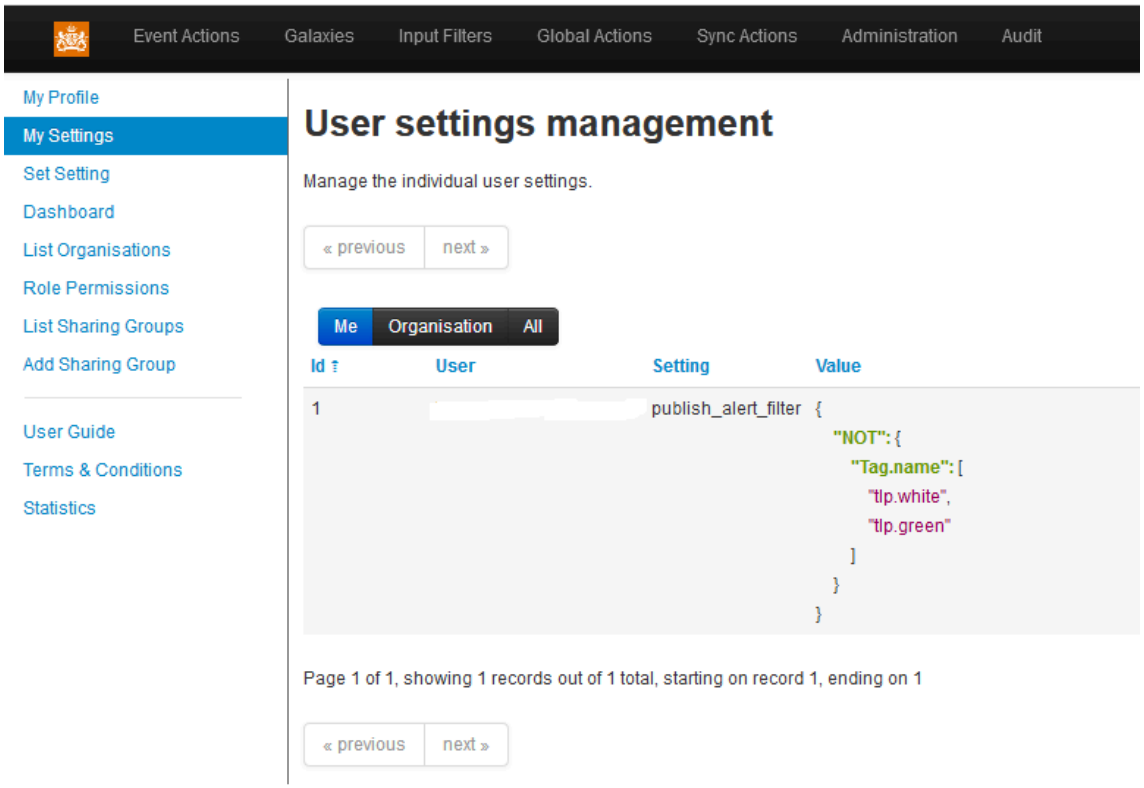


Figure 44.24: Set User settings

44.13.2 Default sharing level

Choose your default sharing level to match your usage scenario for MISP. The setting is named *default_event_distribution* and the values can be:

- Your organisation only (default)
- This community only
- Connected communities
- All communities

You can also set a default distribution level for attributes contained in an event with *default_attribute_distribution*, and it has the same values as the default sharing level for events plus an additional one that allows attributes to inherit the sharing level of the event.

44.13.3 Adding organisation logos

You can add a logo for your organisations in MISP by uploading them via the tab **Manage files** under the **Administration** menu & **Server Settings** sub-menu. The filename must be exactly the same as the organisation name that you will use in MISP. It is recommended to use PNG files of 48x48 pixels.

44.13.4 The scheduler worker is not starting

Under SimpleBackgroundJobs the scheduler runs as one of the Supervisor-managed workers. If it is not running, check `sudo supervisorctl status misp-workers:*` and the Supervisor logs, and confirm the `SimpleBackgroundJobs` settings (Redis host/port/database and the Supervisor host/port/credentials) under **Administration → Server Settings → Diagnostics**.

44.13.5 How to redirect HTTP to HTTPS

Here is a sample configuration for Apache webserver.

```
<VirtualHost *:80>
        ServerAdmin misp@misp.misp
        ServerName misp.misp.misp
        ServerAlias misp-int.misp.misp

        Redirect permanent / https://misp.misp.misp

        LogLevel warn
        ErrorLog /var/log/apache2/misp.local_error.log
        CustomLog /var/log/apache2/misp.local_access.log combined
        ServerSignature Off
</VirtualHost>

<VirtualHost *:443>
        ServerAdmin misp@misp.misp
        ServerName misp.misp.misp
        ServerAlias misp-int.misp.misp

        DocumentRoot /var/www/MISP/app/webroot
        <Directory /var/www/MISP/app/webroot>
                Options -Indexes
                AllowOverride all
                Order allow,deny
                allow from all
        </Directory>

        SSLEngine On
        SSLCertificateFile /etc/ssl/misp.misp.misp/misp.crt
        SSLCertificateKeyFile /etc/ssl/misp.misp.misp/misp.key
        SSLCertificateChainFile /etc/ssl/misp.misp.misp/mispCA.crt

        LogLevel warn
        ErrorLog /var/log/apache2/misp.local_error.log
        CustomLog /var/log/apache2/misp.local_access.log combined
        ServerSignature Off
</VirtualHost>
```

Taken from Koen Van Impe's blog

44.13.6 Increase max size of Samples / other files

Trying to upload a large samples (>50M) might cause the following error:

```
[!] 500 Server Error: Internal Server Error
```

Or will give you an error page in browser.

The error logs on the system will display the following:

```
PHP Warning:  POST Content-Length of 57526024 bytes exceeds the limit of 8388608 bytes
in Unknown on line 0, referer: https://XYZ/attributes/add_attachment/1948
```

And / Or

```
PHP Fatal error:  Allowed memory size of 134217728 bytes exhausted (tried to allocate
76705009 bytes) in /var/www/MISP/app/Lib/cakephp/lib/Cake/Network/CakeRequest.php on
line 996
```

To fix that you need to adjust the php settings:

```
vi /etc/php/8.3/apache2/php.ini
```

Increase to the following values (or more if you want to)

```
; Maximum size of POST data that PHP will accept.
; Its value may be 0 to disable the limit. It is ignored if POST data reading
; is disabled through enable_post_data_reading.
; http://php.net/post-max-size
post_max_size = 256M
[…]
; Maximum amount of memory a script may consume (128MB)
; http://php.net/memory-limit
memory_limit = 1024M
```

And then restart apache2

```
service apache2 restart
```

44.13.7 Support & feature requests

The preferred method for support & feature requests is to use the GitHub ticketing system.

If you want to discuss something related to MISP, want some help from the community, etc… You have the MISP Users mailing list and the MISP developers mailing list.

A number of companies offer custom development, consulting, and support around MISP, please check the support page of the MISP Project website.

44.13.8 More information in the notification emails about new events

The setting MISP.extended_alert_subject allows you to have an extended subject. One word of warning though. If you're using encryption : the subject will not be encrypted. Be aware that you might leak some sensitive information this way. Below is an example how the two subject types look like. First with the option disabled, then with the option enabled.

```
Event 7 - Low - TLP Amber
Event 8 - OSINT - Dissecting  XXX… - Low - TLP Amber
```

Taken from Koen Van Impe's blog

44.13.9 Get top API users

Per-request data is captured in the **access log** (`access_logs` table) when request logging is enabled — see Access logs. The easiest way to see the busiest API users is **Administration → Access Logs**, filtered by user and date. To query it directly, for example for the top users over the last day:

```
SELECT u.email, COUNT(*) AS requests
FROM access_logs a
JOIN users u ON u.id = a.user_id
WHERE a.created > (NOW() - INTERVAL 1 DAY)
GROUP BY a.user_id
ORDER BY requests DESC;
```

The `ip` column is stored in binary form, so filter on it with `a.ip = INET6_ATON('203.0.113.5')`. If you only have `MISP.log_auth` enabled (without the paranoid access log), successful API authentications instead land in the legacy `logs` table under the `auth` action.

44.13.10 MISP Logs

By default, MISP has several layers of logs that can be used to trouble-shoot and monitor the system. Let's walk through each of the available logs:

- **Apache access logs**: Rotating logs generated by apache, logging each request, by default (on Ubuntu) they are found in /var/log/apache2/misp.local_access.log. The location can be changed via the apache conf file
- **Apache error logs**: Rotating logs generated by apache, logging error messages, by default (on Ubuntu) they are found in /var/log/apache2/misp.local_error.log. This error log file will generally not be used by MISP, however, if there is a PHP level error that prevents MISP from functionining you might have relevant entries here.
- **MISP error log**: Generated by MISP, logging any exceptions that occur during usage. These can be found in /var/www/MISP/app/tmp/logs/error.log (assuming default installation path). If you see errors in here and are stuck with an issue let us know via GitHub!
- **MISP debug log**: Generated by MISP, any debug messages and Notice level messages will be sent to this file. Generally less interesting, but can be helpful during debugging sessions. It should not be necessary to monitor this under normal usage. The file can be found in /var/www/MISP/app/tmp/logs/debug.log (assuming default installation path).
- **Background worker logs**: Generated by the SimpleBackgroundJobs workers (managed by Supervisor). Job output and worker exceptions are written to /var/www/MISP/app/tmp/logs/misp-workers.log and /var/www/MISP/app/tmp/logs/misp-workers-errors.log. These replace the old CakeResque `resque-*` logs, which are no longer written in MISP 2.5.

44.13.11 Logging of failed authentication attempts

By default, MISP logs failed login and authentication attempts in the application log. To view them, log in as a site admin and navigate to **Logs → Application Logs** (or **Logs → Search Logs**).

There are two types of entries that will be interesting if you are looking for failed authentication attempts, entries of action "auth_fail" (for failed attempts to authenticate via the API key or the external authentication system) and login_fail (for failed login attempts via the login page).

You can also search for any such entries using **Logs → Search Logs**, simply choose the desired action from the two listed above and hit search. (The `Security.log_each_individual_auth_fail` setting controls whether every failure is logged or throttled.)

What is logged:

```
+----------------+------------+---------------------------+----------+
| Auth method    | Action     | Failed credentials logged | IP       |
+----------------+------------+---------------------------+----------+
| Webform        | login_fail | None                      | Optional |
| API            | auth_fail  | API key                   | Optional |
| Webform        | auth_fail  | External auth key         | Optional |
+----------------+------------+---------------------------+----------+
```

In order to enable IP logging for any logged request in MISP, navigate to Administration - Server settings - MISP settings and enable the MISP.log_client_ip setting.

It is also possible to enable full logging of API and external authentication requests using the MISP.log_auth setting in the same location, but keep in mind that this is highly verbose and will log every request made. In addition to the information above, all accessed resource URLs are also logged.

44.13.12 Clearing expired sessions

By default the garbage collection of sessions is disabled in PHP. It is possible to enable it, but it's not recommended and as such MISP provides a manual way of clearing the sessions.

Navigate to the diagnostics screen of MISP (Administration - Server settings - Diagnostics) and near the bottom of the page there will be a counter showing the count of currently stored expired sessions. Simply purge them by clicking the applicable button when the number grows too large.

44.13.13 Troubleshooting MISP not connecting to redis but redis-cli working

Make sure the Redis connection settings MISP uses (host, port, database and password under the **MISP** and **SimpleBackgroundJobs** settings) match your running Redis, and that Redis is reachable from the web and CLI users. On an IPv6-enabled host, if `localhost` resolves to the IPv6 loopback (`::1`) first and Redis only listens on IPv4, use an explicit `127.0.0.1` instead. You can confirm reachability with `redis-cli -h 127.0.0.1 -p 6379 ping`.

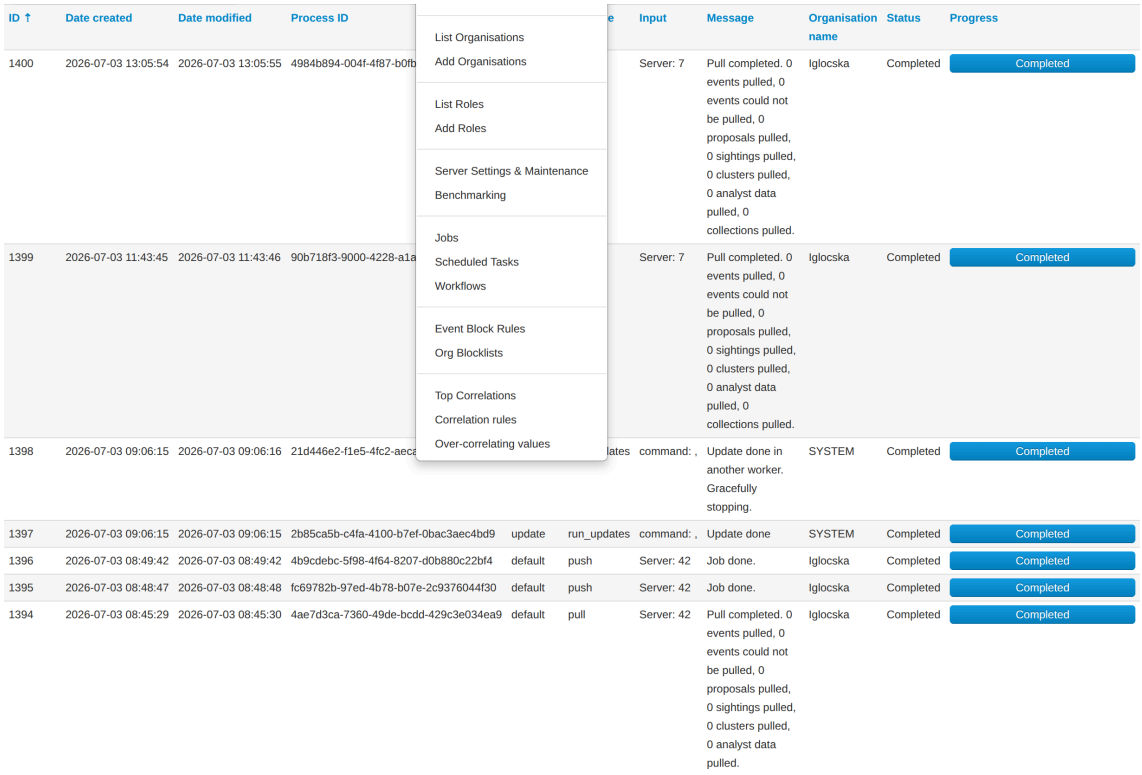
44.13.14 Errors about fields or tables

If you have errors with fields or tables that you can see in the error.log or in the page (if you enabled *debug* or *site_admin_debug* settings), an easy fix to make most of them go away is to use the **clean cache** feature on the *server settings* menu, *diagnostics* tab. An example of error message:

```
Error: [PDOException] SQLSTATE[42S22]: Column not found: 1054 Unknown column 'Task.job_id' in 'field list'
```

44.14 Jobs

The Jobs tab gives you an overview on any currently running jobs or jobs that were previously completed and their status.



ID ↑	Date created	Date modified	Process ID			Input	Message	Organisation name	Status	Progress
1400	2026-07-03 13:05:54	2026-07-03 13:05:55	4984b894-004f-4f87-b0fb			Server: 7	Pull completed. 0 events pulled, 0 events could not be pulled, 0 proposals pulled, 0 sightings pulled, 0 clusters pulled, 0 analyst data pulled, 0 collections pulled.	Iglocska	Completed	Completed
1399	2026-07-03 11:43:45	2026-07-03 11:43:46	90b718f3-9000-4228-a1a			Server: 7	Pull completed. 0 events pulled, 0 events could not be pulled, 0 proposals pulled, 0 sightings pulled, 0 clusters pulled, 0 analyst data pulled, 0 collections pulled.	Iglocska	Completed	Completed
1398	2026-07-03 09:06:15	2026-07-03 09:06:16	21d446e2-f1e5-4fc2-aec		lates	command: ,	Update done in another worker. Gracefully stopping.	SYSTEM	Completed	Completed
1397	2026-07-03 09:06:15	2026-07-03 09:06:15	2b85ca5b-c4fa-4100-b7ef-0bac3aec4bd9	update	run_updates	command: ,	Update done	SYSTEM	Completed	Completed
1396	2026-07-03 08:49:42	2026-07-03 08:49:42	4b9cdebc-5f98-4f64-8207-d0b880c22bf4	default	push	Server: 42	Job done.	Iglocska	Completed	Completed
1395	2026-07-03 08:48:47	2026-07-03 08:48:48	fc69782b-97ed-4b78-b07e-2c9376044f30	default	push	Server: 42	Job done.	Iglocska	Completed	Completed
1394	2026-07-03 08:45:29	2026-07-03 08:45:30	4ae7d3ca-7360-49de-bcdd-429c3e034ea9	default	pull	Server: 42	Pull completed. 0 events pulled, 0 events could not be pulled, 0 proposals pulled, 0 sightings pulled, 0 clusters pulled, 0 analyst data pulled.	Iglocska	Completed	Completed

Figure 44.25: Running Jobs

Typically this is one of the places you would turn to even some background process might not complete as expected to get an indication on any issues related to user initiated Jobs.

For ease of use, you can filter the Jobs by queue (All, Default, Prio, Email, Cache).

You can also purge the entries, either only those with a completed status or all of them. This is not automated and needs to be done manually.

44.14.1 Scheduled Tasks

The **Scheduled Tasks** page lets you run a small set of definition-update tasks on a recurring schedule; it is described under Scheduling recurring tasks above.

The built-in scheduler is intentionally limited. For anything beyond the handful of supported update actions, prefer a real scheduler such as your system's crontab, built from the console commands. As a rule of thumb: if you can click on it, MISP can automate it.

44.15 MISP Backup

Currently there exists this backup script simply called misp-backup.sh

All you need is to copy the the sample config and make sure it is correct. Then launch the script.

```
cd /var/www/MISP/tools/misp-backup
sudo -u www-data cp misp-backup.conf.sample misp-backup.conf
sudo ./misp-backup.sh
```

Script output:

```
/var/www/MISP/tools/misp-backup ⌜◿ 2.4 ● $ sudo ./misp-backup.sh
File ./misp-backup.conf exists.
copy of org images and other custom images
MySQL Dump
/var/www/MISP/tools/misp-backup
MISP Backup Completed, OutputDir: /opt/backup
FileName: MISP-Backup-20181128_163215.tar.gz
FullName: /opt/backup/MISP-Backup-20181128_163214.tar.gz
```

44.16 MISP Restore

In a similar fashion you can restore your MISP instance with the **misp-restore.sh** script. Read the script for details.

44.17 Command line interface (CLI) commands

The below info is also available in the MISP GUI. Go to event actions -> automation -> bottom of the page

44.17.1 Administering MISP via the CLI

44.17.1.1 Get Setting

```
MISP/app/Console/cake Admin getSetting [setting]
```

44.17.1.2 Set Setting

```
MISP/app/Console/cake Admin setSetting [setting] [value]
```

44.17.1.3 Get Authkey

```
MISP/app/Console/cake Admin getAuthkey [email]
```

44.17.1.4 Reset Authkey

```
MISP/app/Console/cake Authkey [email] [api_key | optional]
```

44.17.1.5 Set Baseurl

```
MISP/app/Console/cake Baseurl [baseurl]
```

44.17.1.6 Change Password

```
MISP/app/Console/cake User change_pw [User ID or e-mail address] [new_password] [--
no_password_change]
```

If –no_password_change is used, the user will not be required to change their password after their first login with the set password.

44.17.1.7 Clear Bruteforce Entries

```
MISP/app/Console/cake Admin clearBruteforce [user_email]
```

44.17.1.8 Run Database Update

```
MISP/app/Console/cake Admin runUpdates
```

44.17.1.9 Update All JSON Structures

```
MISP/app/Console/cake Admin updateJSON
```

44.17.1.10 Update Galaxy Definitions

```
MISP/app/Console/cake Admin updateGalaxies
```

44.17.1.11 Update Taxonomy Definitions

```
MISP/app/Console/cake Admin updateTaxonomies
```

44.17.1.12 Enable all tags of a taxonomy

```
MISP/app/Console/cake Admin enableTaxonomyTags [taxonomy_id]
```

44.17.1.13 Update Object Templates

```
MISP/app/Console/cake Admin updateObjectTemplates
```

44.17.1.14 Update Warninglists

```
MISP/app/Console/cake Admin updateWarningLists
```

44.17.1.15 Update Noticelists

```
MISP/app/Console/cake Admin updateNoticeLists
```

44.17.1.16 Update MISP

```
MISP/app/Console/cake Admin updateMISP
```

44.17.1.17 Set Default Role

```
MISP/app/Console/cake Admin setDefaultRole [role_id]
```

44.17.1.18 Get IPs For User ID

```
MISP/app/Console/cake User user_ips [user_id]
```

44.17.1.19 Get User ID For User IP

```
MISP/app/Console/cake User ip_user [ip]
```

44.17.1.20 Get / Set The Live State

```
MISP/app/Console/cake Admin live [true|false]
```

With no argument this prints the current live state; passing `false` puts the instance into maintenance mode (blocking logins for non-site-admins).

44.17.1.21 Check Redis Connectivity

```
MISP/app/Console/cake Admin redisReady
```

44.17.1.22 Lint The Configuration

```
MISP/app/Console/cake Admin configLint
```

Reports settings whose current value fails validation.

44.17.1.23 Run Schema Diagnostics

```
MISP/app/Console/cake Admin schemaDiagnostics
```

Reports critical differences between the live database schema and the expected `db_schema.json`.

44.17.1.24 Optimise Database Tables

```
MISP/app/Console/cake Admin optimiseTables
```

44.17.1.25 Run A Security Audit

```
MISP/app/Console/cake Admin securityAudit
```

44.17.1.26 Clean The Caches

```
MISP/app/Console/cake Admin cleanCaches
```

44.17.1.27 Re-encrypt Stored Secrets

```
MISP/app/Console/cake Admin reencrypt [--old <key>] [--new <key>]
```

Re-encrypts stored authentication keys and sensitive settings under a new `Security.encryption_key`.

44.17.1.28 Migrate To Advanced Auth Keys

```
MISP/app/Console/cake Admin updateToAdvancedAuthKeys
```

44.17.2 Automating certain console tasks

If you would like to automate tasks such as caching feeds or pulling from server instances, you can do it using the following command line tools. Simply execute the given commands via the command line / create cron jobs easily out of them.

44.17.2.1 PullAll

```
MISP/app/Console/cake Server pullAll [user_id] [full|update]
```

44.17.2.2 Pull

```
MISP/app/Console/cake Server pull [user_id] [server_id] [full|update]
```

44.17.2.3 Push

```
MISP/app/Console/cake Server push [user_id] [server_id]
```

44.17.2.4 List Feeds

```
MISP/app/Console/cake Server listFeeds
```

44.17.2.5 Cache Feeds For Quick Lookups

```
MISP/app/Console/cake Server cacheFeed [user_id] [feed_id|all|csv|text|misp]
```

44.17.2.6 Fetch Feeds As Local Data

```
MISP/app/Console/cake Server fetchFeed [user_id] [feed_id|all|csv|text|misp]
```

44.17.2.7 Run Enrichment

```
MISP/app/Console/cake Event enrichment [user_id] [event_id] [json_encoded_module_list]
```

44.17.2.8 Test Server

```
MISP/app/Console/cake Server test [server_id]
```

44.17.2.9 List Servers

```
MISP/app/Console/cake Server listServers
```

44.17.3 Managing the background workers

Under SimpleBackgroundJobs the worker processes are managed by Supervisor, so the old `Admin startWorker` / `restartWorker` / `restartWorkers` / `killWorker` commands are no-ops — they print *"This*

*method does nothing when SimpleBackgroundJobs are enabled."*Use `supervisorctl` to control the processes and the `Worker` shell to inspect the queues.

44.17.3.1 Control The Worker Processes

```
sudo supervisorctl status misp-workers:*
sudo supervisorctl restart misp-workers:*
```

44.17.3.2 Show Queues

```
MISP/app/Console/cake Worker showQueues
```

44.17.3.3 Flush A Queue

```
MISP/app/Console/cake Worker flushQueue [queue_name]
```

44.17.3.4 Show Job Status

```
MISP/app/Console/cake Worker showJobStatus [worker]
```

44.17.3.5 List Workers

```
MISP/app/Console/cake Admin getWorkers [all|dead]
```

44.18 Administration of TOTP/HOTP

MISP 2.4.172 introduced multi-factor authentication (TOTP/HOTP) support.

Before using or testing this feature, please note that it is extremely important to make sure your server has correct time syncing set up, since the TOTP tokens are time based. If you are alread using e-mail OTP, you can leave this on. The two multi-factor authentication methods can co-exist, users that have TOTP/HOTP set up, will no longer be able to use e-mail OTP. Those that do not have it set, will still be prompted for it in that case.

After updating your MISP, make sure you have installed the required php dependencies by using the top menu to go to Administration > Server Settings & Maintenance > Diagnostics.

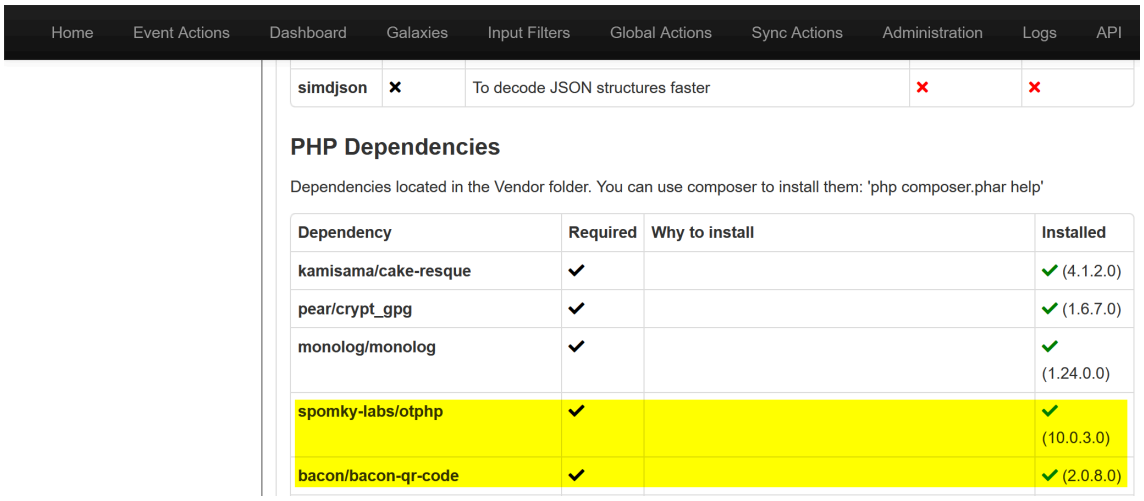
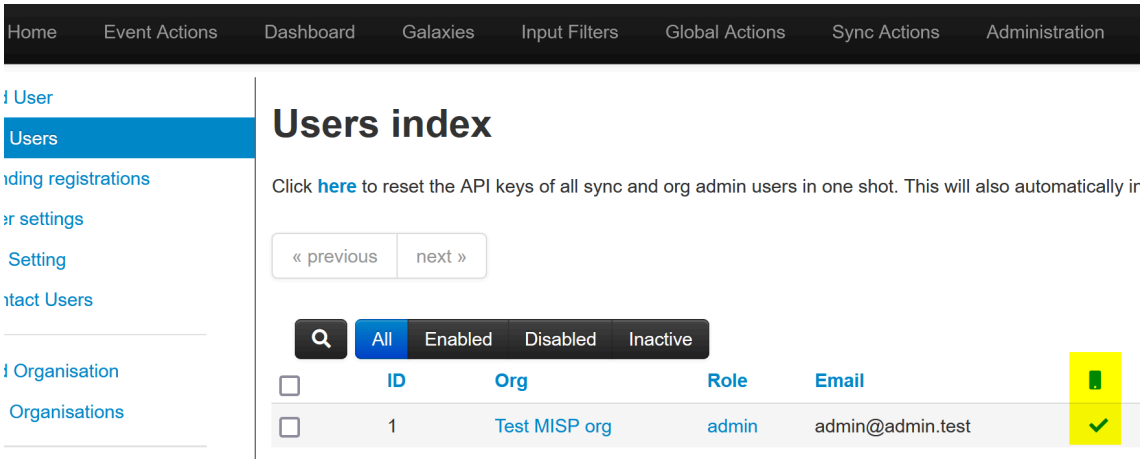


Figure 44.26: Screenshot of diagnostics page showing required php libraries for TOTP/HOTP

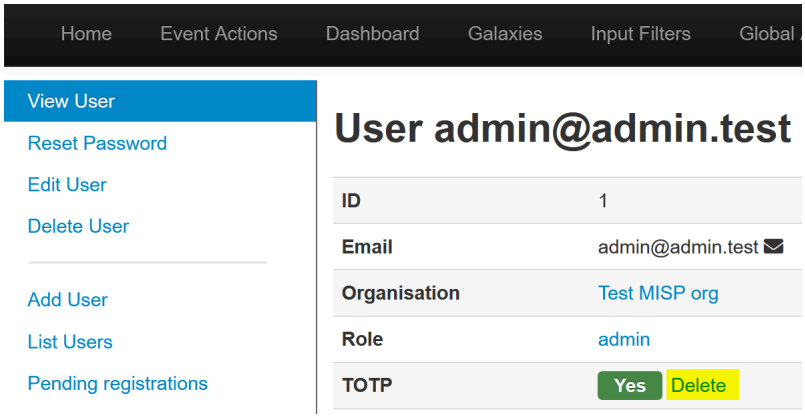
If you do not have them installed yet, you can run the equivalent of the below command for your setup / OS to install them:

```
sudo -u www-data sh -c "cd /var/www/MISP/app;php composer.phar update"
```

You can see which users have TOTP/HOTP configured in the users index:



As a site admin or org admin (users can't do this themselves), you can delete TOTP/HOTP for a user from the view user page, by clicking the TOTP Delete button.



44.18.1 Mandating TOTP/HOTP usage

You can mandate the usage of TOTP/HOTP by setting the Security.otp_required setting to true. Users will then be prompted to set up TOTP/HOTP when trying to access a page, if they haven't done so yet.

From the command line you can run the equivalent of the below command for your setup, to configure this:

```
sudo -u www-data /var/www/MISP/app/Console/cake Admin setSetting Security.otp_required
true
```

44.18.2 Transitioning from e-mail OTP to TOTP/HOTP

If you are currently using e-mail OTP on your instance, you have the option to enable TOTP/HOTP (by installing the required php dependencies) and giving your users a transition period to set up their TOTP (e-mail OTP will still work during this period), before mandating TOTP.

44.18.3 How to use TOTP/HOTP

For information on how to use this feature from a normal user perspective, please refer to the using the system section.

45. Workflows & Blueprints

Workflows let a MISP instance react automatically to things that happen inside it. When an event is saved, published, or about to be deleted — and at many other hook points — MISP can run a small graph of nodes that inspects the data and takes action: enrich it, tag it, publish it, push it to an external system, or even **block** the operation from completing. Workflows turn MISP from a passive store into an event-driven automation platform.

> ⚠ Warning
>
> Workflows are an **experimental** feature and are **site-admin only**. They are powerful — a blocking workflow can prevent events from being saved or published — so treat them with the same care as any automation that sits in a critical path, and test thoroughly before enabling one on a production instance.

45.1 Concepts

A workflow is a directed graph made of three kinds of node:

- **Trigger** — the single entry point. Each workflow is bound to exactly one trigger, which is a hook point in MISP core (for example *event before-save*, *event publish*, *attribute after-save*).
- **Logic** — branching and filtering nodes. `IF` nodes fork the flow into a *then* and an *else* path based on a condition; *filter* nodes narrow the set of data flowing downstream.
- **Action** — nodes that *do* something: enrich, tag, publish, add analyst data, push to ZMQ, call a webhook, send a Teams/Telegram message, stop execution, and many more.

Data enters at the trigger, flows through the logic and action nodes along the graph, and each node operates on a shared "roaming" copy of the data. The graph must be acyclic and have exactly one trigger.

45.1.1 Blocking vs non-blocking

This is the most important distinction to understand.

- A **non-blocking** trigger fires *after* the core operation and runs asynchronously as a background job. The workflow's outcome does not affect the operation that triggered it — it has already happened.
- A **blocking** trigger fires *before* (or as part of) the core operation and runs **synchronously**. Its result is returned to the core code, which can then decide whether to proceed. A blocking workflow that reaches a **Stop execution** node causes the core operation to be **aborted**.

Only a handful of triggers are blocking. In MISP 2.5 these are:

- **event before-save** — can prevent an event from being saved,
- **event before-delete** — can prevent an event from being deleted,
- **event publish** — can prevent an event from being published,
- **shadow-attribute before-save** — can prevent a proposal from being saved,
- **user before-save** — can prevent a user from being saved,
- **enrichment before-query** — can prevent an enrichment query.

Every other trigger (attribute after-save, event after-save, sighting after-save, tag attached, object after-save, and so on) is non-blocking and runs in the background.

> **Note**
>
> The editor warns you if you place a blocking node (such as *Stop execution*) on a path fed by a non-blocking trigger — in that context it cannot actually block anything, it will just stop that branch of the workflow.

45.2 Enabling workflows

The feature has three prerequisites, all enforced when you open the Workflows UI:

1. **Background jobs** must be enabled (`MISP.background_jobs`) — non-blocking workflows run as background jobs.
2. **The workflow plugin** must be enabled (`Plugin.Workflow_enable`), under *Administration → Server Settings & Maintenance → Plugin settings*.
3. **Redis** must be reachable — it stores the trigger routing and the set of enabled modules.

In addition, **each trigger is disabled by default** and has its own setting (`Plugin.Workflow_triggers_<trigger-id>`). A trigger only actually runs a workflow when it is both **enabled** and **listened to** by a saved workflow. This double gate means simply creating a workflow does not put it live — you also enable its trigger.

45.3 The workflow editor

Workflows are built in a drag-and-drop visual **editor**. You start from a trigger and connect logic and action nodes onto the canvas; each node has a small configuration form for its parameters (conditions, hash paths to the field being tested, tags to apply, webhook URLs, and so on). Some parameters support Jinja templating for dynamic values.

The editor validates the graph as you build it — checking that it is acyclic, that outputs are connected sensibly, and surfacing warnings such as blocking-node-on-non-blocking-path. You can also **test a single module in isolation** against sample input from the editor, which is the quickest way to confirm a node behaves as you expect before wiring it into a live workflow.

Supporting pages let you browse and manage the building blocks:

- **Triggers** — the list of available triggers, filterable by scope, whether they are enabled, and whether they are blocking.
- **Modules** — the catalogue of logic and action modules, where you enable or disable each module.
- **Workflows** — your saved workflows.

45.3.1 Modules must be enabled

Not every module is available out of the box. Only a small default set is enabled — the common `IF` conditions (generic, distribution, published, organisation, tag), the concurrent-task node, *Stop execution*, *Webhook* and *Push to ZMQ*. Any other logic or action module (for example *Enrich event*, *Attach warninglist*, *Tag operation*, *Attach decay score*, *Add analyst data*, the messaging integrations) must be **enabled in the module index** before it can be used in a workflow.

Some action modules — such as sending mail, Jinja rendering, and service-provided actions — depend on a working misp-modules service and degrade gracefully if it is unavailable.

45.4 Ad-hoc workflows

Besides trigger-bound workflows, MISP supports **ad-hoc workflows** that you run manually rather than from a core hook. An ad-hoc workflow collects its input on demand — passed event IDs, a REST search, or roaming data supplied to it — and is useful for one-off curation runs or for testing a graph against real data. An action module also lets one workflow **run another** ad-hoc workflow.

45.5 Workflow Blueprints

A **Workflow Blueprint** is a reusable, saved fragment of a node graph. Blueprints are *building blocks for the editor*, not live automations: you drop a blueprint onto the canvas to reuse a proven pattern rather than rebuilding it node by node.

MISP ships a set of **default blueprints** covering common curation recipes — for example normalising TLP/PAP tags, assigning a country galaxy from enrichment, or removing the IDS flag on a warninglist hit. You can:

- **Import** a blueprint from pasted JSON or an uploaded file,
- **Export** a blueprint to JSON to share it, and
- **Update** the default blueprints from the upstream `misp-workflow-blueprints` collection (matched by UUID, upgrading those with a newer upstream version).

Each blueprint shows a flowchart preview so you can see its shape before using it. Blueprints you create or edit are always saved as non-default.

45.6 Debugging and logs

Every workflow run is counted, and **errors are always logged** to the audit log and to `app/tmp/logs/workflow-execution.log`. For deeper inspection, each workflow has a **debug toggle**. With debugging on, MISP writes verbose, per-node execution messages and can POST execution snapshots (`init`, per-module `exec`, `end`) to an external debug receiver configured via `Plugin.Workflow_debug_url`. You can also **execute a whole workflow manually** with a supplied payload to observe its behaviour end-to-end.

45.7 Permissions and execution context

All workflow and blueprint actions are **restricted to site administrators** — there is no finer-grained role permission for workflows. Note also that a workflow **executes as a system (host-organisation / admin) user**, not as the user whose action triggered it; the initiating user is recorded separately for logging and for UI toast notifications. Keep this in mind when a workflow reads or modifies data — it is not limited by the triggering user's distribution visibility.

45.8 API and CLI

Both workflows and blueprints are fully accessible over the REST API under the `/workflows/` and `/workflowBlueprints/` controllers (index, add, edit, delete, view, import/export, and the editor support actions). The `view` action can additionally emit the graph as Graphviz `dot` or **Mermaid** with `?format=dot` / `?format=mermaid`.

On the command line, the `cake Workflow` shell provides the entry points the background workers use — `executeWorkflowForTrigger` (run a trigger's workflow), `executeAdHocWorkflow` (run an ad-hoc workflow), and `walkGraph` (execute a concurrent sub-branch as its own job). In normal operation these are invoked by MISP's background job system rather than by hand.

46. Updating Python Dependencies

MISP requires several Python libraries for its full feature set to work correctly.
These features include the import and export tools, binary extraction from attachments, and PyMISP.

46.1 Installation

MISP uses virtual environments to make Python installation and maintenance easier.
If you use the installation scripts to set up MISP on your machine, or start from the automatically generated virtual machine, you already have the latest required Python libraries installed in a virtual environment.
If you are using an older MISP version, you may want to create the virtual environment manually.

46.1.0.1 Set Up the Virtual Environment

```
# Create a python3 virtualenv
sudo -H -u www-data virtualenv -p python3 /var/www/MISP/venv

# Make pip happy
sudo mkdir /var/www/.cache
sudo chown www-data:www-data /var/www/.cache
```

If you already have a venv directory, you can skip this step.

46.2 Updating MISP and its dependencies

Keeping MISP as up to date as possible is the safest way to avoid most potential issues.
You can do this either from the Update button in the diagnostic tool in the MISP UI or from the command line.

46.2.1 Updating MISP core

To update MISP dependencies, first pull the latest MISP version so you also get the latest submodule references.

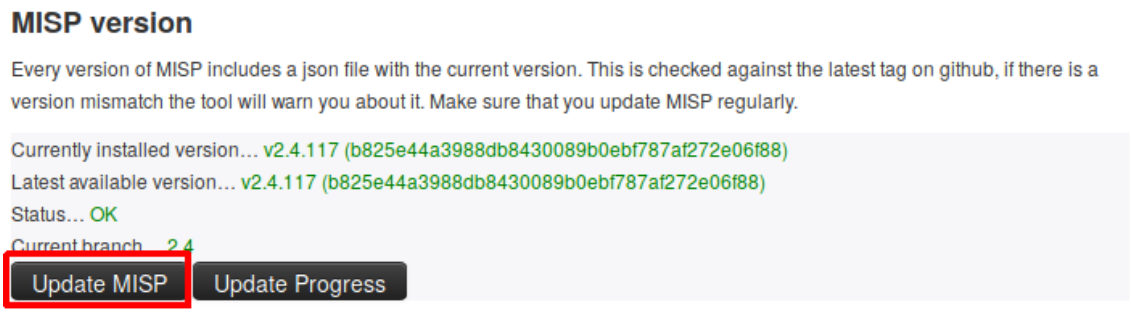


Figure 46.1: Update MISP from the diagnostic tool in the UI

OR ~~bash sudo -H -u www-data git pull origin 2.5~~
Once you have the latest MISP update, you can start updating the Python libraries.

46.2.2 Updating the python dependencies

MISP includes several submodules that support additional functionality, so it is important to keep those dependencies up to date. ~~bash sudo -H -u www-data git submodule update --init --recursive~~

46.2.2.1 Updating python dependencies

You can check the status of all Python libraries required by MISP by using the diagnostic tool again.

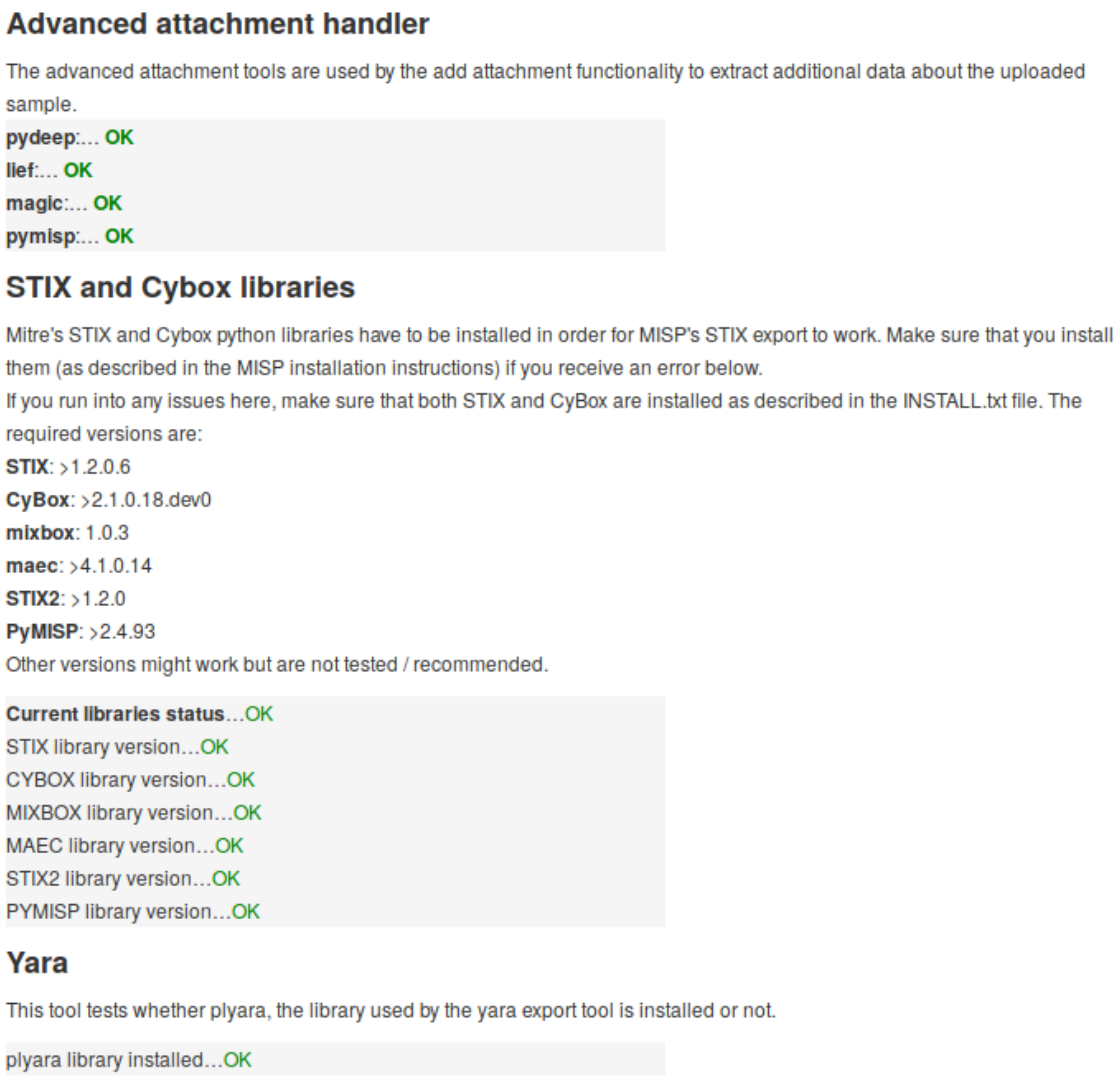


Figure 46.2: Python libraries status

If something is wrong, updating the corresponding library should resolve the diagnostic warning.

The simplest and recommended way is to install the pinned set from MISP's top-level `requirements.txt`, which covers PyMISP, misp-stix, the attachment handlers and the other libraries in one go:

```
sudo -H -u www-data /var/www/MISP/venv/bin/pip install -U -r /var/www/MISP/
requirements.txt
```

If you prefer to update libraries individually:

```
# Update PyMISP
cd /var/www/MISP/PyMISP
sudo -H -u www-data /var/www/MISP/venv/bin/pip install -U .
```

```
# Advanced attachment handling
sudo -H -u www-data /var/www/MISP/venv/bin/pip install -U lief python-magic
# pydeep2 - WARNING: requires libfuzzy-dev
sudo -H -u www-data /var/www/MISP/venv/bin/pip install -U pydeep2

# STIX 1 and 2 conversion is provided by a single library, misp-stix
sudo -H -u www-data /var/www/MISP/venv/bin/pip install -U misp-stix

# Yara
sudo -H -u www-data /var/www/MISP/venv/bin/pip install -U plyara
```

> **Note**
>
> The old per-package STIX 1 stack (`python-stix`, `python-cybox`, `python-maec`, `mixbox`) is legacy. MISP 2.5 uses **misp-stix** (`misp_stix_converter`) for both STIX 1 and STIX 2 conversion, and it is included in `requirements.txt`.

If you want to use or update the ZeroMQ functionality, the `pyzmq` library is also part of `requirements.txt`; to update it on its own:

```
sudo -H -u www-data /var/www/MISP/venv/bin/pip install -U pyzmq
```

46.2.2.2 Updating MISP modules

Another set of dependencies you may want to update are MISP modules.

MISP modules have their own dependencies that need to be up-to-date as well as the modules scripts themselves.

Note that the following instructions consider your MISP modules are installed in the default path where we install them on our virtual machine or following the install script. Please change the path accordingly if needed.

```
# Change here the path if needed
cd /usr/local/src/misp-modules
# Update misp-modules requirements
sudo -H -u www-data /var/www/MISP/venv/bin/pip install -U -r REQUIREMENTS
# Update misp-modules scripts
sudo -H -u www-data /var/www/MISP/venv/bin/pip install -U .
```

You will then need to restart the modules, please refer to the documentation.

VI Reference & FAQ

50 FAQ 385

51 Frequently Asked Questions 387
51.1 General questions 387
51.2 Specific questions 387
51.3 Usage questions 392
51.4 Permission issues 392
51.5 RHEL/CentOS SELinux debug 393
51.6 When to update MISP? 394
51.7 OS Upgrades 394
51.8 Hardening 395
51.9 Maintenance mode 395
51.10 Update MISP fails 396
51.11 Uninstalling MISP 399
51.12 Updating PyMISP to incorporate newer versions of the MISP object templates 399
51.13 How to disable freetext/custom/user-created tags and only allow certain tags 400
51.14 How to enable the csv import module? 400
51.15 Why do I see 'The request has been black-holed' when I submit forms? 400
51.16 Importing large feeds creates PHP Fatal error . 400
51.17 I deleted the admin user by mistake 400
51.18 config.php is not writeable 401
51.19 How to debug misp-dashboard 401
51.20 How to update object templates? 401
51.21 What to do if my REST client is throwing SSL errors when trying to query my MISP instance? 401
51.22 What to do if my REST client cannot reach the host, despite me being able to issue requests using Curl / Postman / etc.? 401
51.23 How would one set up a sharing group with a remote org, where we only share a mutual community instance (i.e. we both have sync users on that instance). On our local instance, they exist as a remote org (from events that have synced from their instance via our shared community instance). 402
51.24 Is it possible to propose objects to an event? . . 402
51.25 How to use the enforceWarninglist parameter in REST search? 402
51.26 Column not found issue 402
51.27 WatchList Customization 403
51.28 How to add a galaxy to an event via PyMISP . . 403
51.29 How to block attachment (and malware-sample) to be ingested 403
51.30 What are the required steps after a MISP installation to have a properly running instance? 403

52 Developer FAQ 405
52.1 Main Developer Resources 405
52.2 About the codebase 405

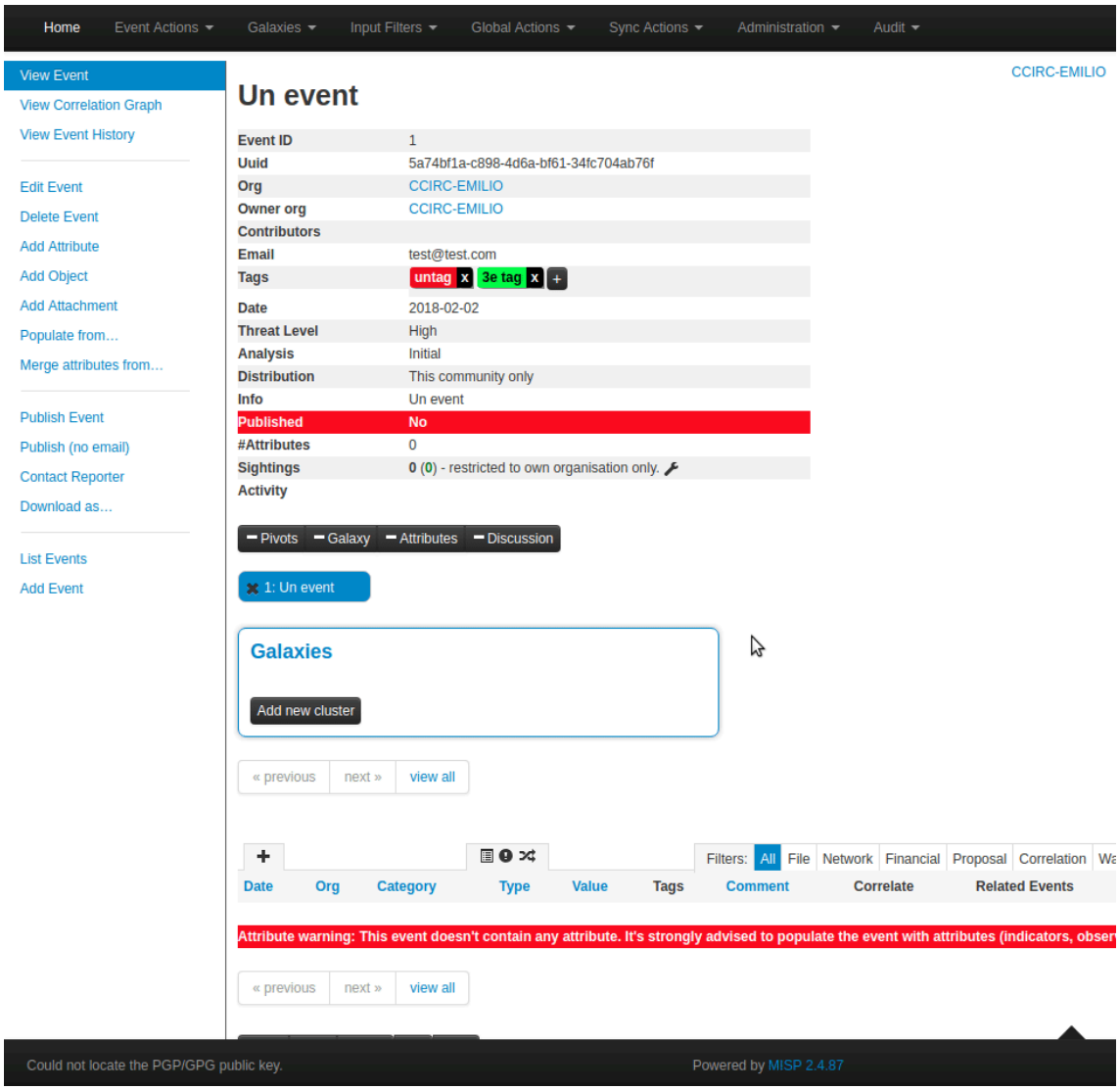
53 User Stories 407

54 User Personas 417
54.1 Primary personas 417
54.2 Secondary personas 419
54.3 Other personas 425

47. Keyboard Shortcuts

MISP ships with keyboard shortcuts on the busiest pages. To see which shortcuts are available on the page you are currently on, press the small triangle at the bottom-right of the window (or press `?`), which opens the *Keyboard shortcuts for this page* helper.

Demo:



47.1 Default shortcuts

The following shortcuts are available everywhere:

Key	Action
`l`	Go to the event list (`/events/index`)
`e`	Go to the add-event page (`/events/add`)
`?`	Show the keyboard-shortcuts helper for this page

On the **event view** page (`/events/view`) these additional shortcuts apply:

Key	Action
`t`	Open the tag-selection modal
`i`	Open the *Populate from…* import choice modal
`f`	Open the freetext import modal
`a`	Add an attribute
`o`	Add an object
`s`	Focus the attribute filter bar

On the **event index** page (`/events/index`), `s` focuses the event filter bar.

> **Note**
>
> The default set is defined in the shipped source and may grow between releases — always check the on-page helper (`?`) for the authoritative list on your version.

47.2 Customising the shortcuts

Since MISP 2.5 the shortcuts are defined in a single JavaScript definition file rather than in per-page JSON files. To add or change a shortcut, edit `app/webroot/js/keyboard-shortcuts-definition.js` and extend the `getShortcutsDefinition()` function, which receives the current `controller` and `action` and returns the list of shortcuts for that page:

```
function getShortcutsDefinition(controller, action) {
    var shortcuts = [
        {
            "key": "l",
            "description": "Go to event list",
            "action": function () {
                document.location.href = baseurl + '/events/index';
            }
        }
        // ... other global shortcuts ...
    ];

    if (controller === 'events' && action === 'view') {
        shortcuts.push({
            "key": "t",
            "description": "Open the tag selection modal",
            "action": function () {
                $('.addTagButton').first().click();
            }
        });
    }
    return shortcuts;
}
```

Each shortcut is an object with a `key`, a human-readable `description` (shown in the on-page helper), and an `action` — a JavaScript function that runs when the key is pressed. To scope a shortcut to a specific page, add it inside a matching `controller`/`action` branch; to make it global, add it to the top-level `shortcuts` array. There is currently no in-GUI editor for the shortcuts — the definition file is the single source of truth, so keep local changes under version control so an upgrade does not silently overwrite them.

48. Translations - i18n & l10n

49. MISP and internationalization (i18n)

MISP is internationalized (i18n) and localized (l10n) using the standard gettext workflow provided by CakePHP. Every user-facing string in the interface is wrapped in a translation function, extracted into a translation template (`default.pot`), and translated per language under `app/Locale/<lang>/LC_MESSAGES/default.po`. MISP can be translated into more than 140 languages this way.

Translations are **crowd-sourced through Crowdin** — you no longer edit `.po`/`.pot` files by hand or open pull requests with translated files. The two audiences below cover almost everything:

- **Translators** contribute and review translations online on Crowdin (no MISP checkout or local tooling required).
- **Developers** make new strings translatable in the codebase so that Crowdin can pick them up.

> **i** Note
>
> Crowdin replaced the older Transifex-based workflow. The project lives at crowdin.com/project/misp. Crowdin is an independent service with its own privacy policy.

49.1 Translating MISP (for translators)

You can help correct, improve, or complete the translation of MISP into your native language directly from a web browser:

1. Open the MISP project on Crowdin: crowdin.com/project/misp.
2. Log in (or create a free Crowdin account) and pick your target language.
3. Translate strings in the online editor. Crowdin shows you the surrounding context, existing suggestions, and a translation memory so recurring phrases stay consistent.

If you are new to Crowdin, read their introduction for volunteer translators.

> **!** Important
>
> Only **reviewed** translations are shipped with MISP. Once a string is translated it still needs to be proofread/approved by a reviewer before it is exported into the codebase and distributed in the next release. If you would like to become a reviewer for your language, reach out to the MISP team.

49.1.1 Style

Please follow whatever is the purest and most intelligible form of the written language being translated. Where a technical term has an established local form, prefer it; where the English term is what practitioners actually use, keeping it is fine — consistency within a language matters more than a strict rule.

49.2 Enabling a language on your instance

Translations reach an instance through the release cycle: reviewed strings on Crowdin are periodically exported to `app/Locale/<lang>/LC_MESSAGES/default.po`, compiled, and shipped with MISP. To select the interface language on a running instance, a site administrator sets the **`MISP.language`** server setting (default `eng`) under *Administration* → *Server Settings & Maintenance* → *MISP settings*. The dropdown is populated from the locales present in `app/Locale/`, so only languages that ship with your version are selectable. Changing it clears the relevant caches automatically.

49.3 Making MISP translatable (for developers)

New user-facing strings only become translatable if they are wrapped in CakePHP's `__()` translation function. When you add or change a view, wrap the strings so that they are extracted into `default.pot` and appear on Crowdin. See the CakePHP documentation on i18n & l10n for the underlying mechanism.

A few conventions make the extracted strings far easier to translate.

49.3.1 Use named placeholders, not `%s`

This is hard to translate because the translator gets no context for the substituted value:

```
<p><?php echo __('Are you sure you want to delete Proposal #') . $id . '?' ?></p>
```

Prefer the format-string notation with a **named** placeholder — the generated `msgid` carries the variable name, which is far more meaningful to a translator than a bare `%s`, and it lets the translator reorder the sentence freely:

```
<p><?php echo __('Are you sure you want to delete Proposal #{$id}?'); ?></p>
```

49.3.2 Keep HTML out of translatable strings

Move HTML tags out of the sentence — and out of the PHP call where possible — so translators translate language, not markup. Instead of:

```
<p><?php echo __('<h1>Are you sure you want to:<br />Delete Proposal #%s?', $id); ?></
h1></p>
```

write:

```
<p><h1><?php echo __('Are you sure you want to:%sDelete Proposal #{$id}?', '<br />'); ?
></h1></p>
```

49.3.3 Wrap button labels and form labels

Untranslated labels are a common oversight. This label is not translatable:

```
echo $this->Form->button('Submit', array('class' => 'btn btn-primary'));
```

Wrap it:

```
echo $this->Form->button(__('Submit'), array('class' => 'btn btn-primary'));
```

The same applies to form inputs — this:

```
echo $this->Form->input('sharing_group_id', array(
    'options' => array($sharingGroups),
    'label' => 'Sharing Group',
));
```

becomes:

```
echo $this->Form->input('sharing_group_id', array(
    'options' => array($sharingGroups),
    'label' => __('Sharing Group'),
));
```

49.4 Issues

Sometimes a phrase is impossible to translate cleanly, or you notice bad formatting or awkward sentence segmentation in the source string. In that case, open an issue on GitHub so the source string can be fixed — that fixes it for every language at once.

49.5 Translating the website or this book

Interested in translating the MISP website or the MISP Book as well? Those are separate from the software translation on Crowdin — reach out to the MISP team and we will help you get set up.

49.6 Reach out to the community

Want to chat with other MISP contributors and translators? Join the MISP Gitter channel.

50. FAQ

51. Frequently Asked Questions

The following page hosts some frequently asked questions as noticed in our issues and gitter channels.

51.1 General questions

51.1.1 Where can I get support?

If you have feature requests or you found a bug you can open a ticket on MISP's GitHub repository issue tracker.

If you want to discuss something related to MISP or want help from the MISP community, join the appropriate MISP Gitter channel:

- MISP Developer Room Dev discussions
- MISP Support Room OMGoo! My MISP doesn't work discussions
- MISP Sharing Room Threat Intelligence Sharing discussions
- misp-cloud Room Using MISP in the clouds discussions

If you want commercial support, feel free to look at the MISP professional services.

51.1.2 What are the hardware requirements?

From a hardware perspective, MISP's requirements are quite humble, a web server with 2+ cores and 8-16 GB of memory should be plenty, though more is always better of course. A lot of it depends on the data set and the number of users you are dealing with.

We recommend a standard LAMP stack (PHP 8.x) on a current Ubuntu 24.04 LTS or Debian 12/13. For details on the exact dependencies please refer to the installation guides as well as the requirements for the MISP modules.

During a Hackathon a small tool called MISP-Sizer was conceived. It will give you a **very rough** idea on what requirements are if you have a bigger installation. source-code is here. The reference for sizing a MISP instance is Sizing your MISP instance.

51.1.3 How to monitor MISP?

There are many ways to monitor MISP via the MISP API or relying on external tools. The following documentation is available for different monitoring tools:

- Using Cacti, a blog post with the instruction is available.
- Using Munin, misp-monitor for instructions.
- Using Nagios, Monitoring MISP with Nagios
- Using OpenNMS, a blog post with the instructions is available.
- Live monitoring of MISP usage via the httpd logs.

51.2 Specific questions

51.2.1 Can I configure MISP encrypted notification emails to contain more information in the subject?

The setting 'MISP.extended_alert_subject' allows you to have an extended subject. /! Beware if you're using encryption: the subject will not be encrypted. Be aware that you might leak some sensitive information this way. Below is an example how the two subject types look like. First with the option disabled, then with the option enabled.

```
Event 7 - Low - TLP Amber
Event 8 - OSINT - Dissecting  XXX... - Low - TLP Amber
```

(Source: Getting started with MISP)

51.2.2 Why are there malicious files in my MISP directory?

It is entirely normal to find malicious files in the directory where MISP—a threat intelligence platform—stores evidence, including malware or other malicious files. Because MISP is designed to collect, analyze, and share threat data, these malicious files are intentionally present as part of the analytical process. If your security tools are triggering alerts on these files, we recommend excluding the `./app/files` directory from detection to prevent unnecessary disruptions to MISP's normal operations.

51.2.3 How can I restart the workers?

The workers can be restarted from the web interface:

```
administration -> server settings -> workers -> restart all
```

In MISP 2.5 the workers run under SimpleBackgroundJobs and are managed by Supervisor, so from the command line you restart them with:

```
sudo supervisorctl restart misp-workers:*
```

51.2.4 How can I redirect HTTP to HTTPs?

```
<VirtualHost *:80>
        ServerAdmin misp@misp.misp
        ServerName misp.misp.misp
        ServerAlias misp-int.misp.misp

        Redirect permanent / https://misp.misp.misp

        LogLevel warn
        ErrorLog /var/log/apache2/misp.local_error.log
        CustomLog /var/log/apache2/misp.local_access.log combined
        ServerSignature Off
</VirtualHost>

<VirtualHost *:443>
        ServerAdmin misp@misp.misp
        ServerName misp.misp.misp
        ServerAlias misp-int.misp.misp

        DocumentRoot /var/www/MISP/app/webroot
        <Directory /var/www/MISP/app/webroot>
                Options -Indexes
                AllowOverride all
                Order allow,deny
                allow from all
        </Directory>

        SSLEngine On
        SSLCertificateFile /etc/ssl/misp.misp.misp/misp.crt
        SSLCertificateKeyFile /etc/ssl/misp.misp.misp/misp.key
        SSLCertificateChainFile /etc/ssl/misp.misp.misp/mispCA.crt

        LogLevel warn
        ErrorLog /var/log/apache2/misp.local_error.log
        CustomLog /var/log/apache2/misp.local_access.log combined
```

```
        ServerSignature Off
</VirtualHost>
```

Source: Getting started with MISP

51.2.5 When I try to access my new installation, I am redirected to localhost:8443 and get an error.

By default, MISP runs on a local instance and is setup for local access upon installation. This allows you to setup security and customizations before making it available elsewhere. If you would like to access the MISP instance from a remote host (including another VM host/client), assign an IP to the MISP host and point your browser accordingly. Please refer to how to set the baseurl from command line. Once the update is done, you should then be able to access the instance using the set IP/DNS name.

51.2.6 How can I define the default sharing level?

MISP allows you to define the group of people with whom you want to share your threat data. If you do not set it to your preferred default then it's likely that at one given moment you'll make an error and share your intel with the wrong group. Defining the sharing level is done with the setting default_event_distribution in the configuration file. There are three levels:

```
0 : Your organisation only (default)
1 : This community only
2 : Connected communities
3 : All communities
```

You can set a similar configuration setting for the attributes. The setting default_attribute_distribution has the same values as default_event_distribution. Additionally it has the value event which allows the attribute to get the setting from the event to which it belongs.

Source: Getting started with MISP

51.2.7 How can I add an organisation logo and/or foter logo?

MISP can be made more appealing to the eye by adding some graphics.

As Org.- or Site-admin navigate to *Administration -> List organisations* and edit the corresponding organization. Within this editor you will be able to update the logo.

Other ways to achieve this, would be:

Set your organisation logo by adding an image (.png) that has the same name as your organisation in the directory */var/www/MISP/app/webroot/img/orgs/*.

Yet another way of doing this is by logging into your MISP instance with Admin rights, navigate to *Administration -> Server Settings*, tab -> *Manage files.*

You can add a footer logo. Add an image to the directory */var/www/MISP/app/webroot/img/custom/* and define the footer logo in the config file (config.php) or in *Adminitration -> Server Settings... -> MISP settings* (search for: "footer_logo") point to the location on-disk of the image.

Partial source: Getting started with MISP

51.2.8 How can I import data directly from PDF reports?

/! This section needs review, verification and eventual amendments to make sure it works.

The simplest option is to select the text in the report and paste it into the free-text import form. For automated extraction, use the MISP import modules — for example the OCR, PDF or CSV import modules — enabled under **Administration → Server Settings → Plugin settings**.

Another option is the new OCR import module that can be used via the import modules. You will need to install the OCR software tesseract.

51.2.9 I have many failed jobs when doing email notification. What should I do?

This is most probably due to some encryption failing for some users. We strongly advise to review the current PGP keys and to ensure that they keys are not expired or perhaps not supported anymore (weak keys). The keys can be reviewed at the following location in MISP:

```
https://<YOUR MISP URL>/users/verifyGPG
```

51.2.10 I have issues with pushing events

- What does the 'Connection test' for the specific server report? (*Sync Actions -> List Servers*)
- Is the event you assume to push/pull ready to be published?
- Is the distribution level set not too restrictive?
- Have you enabled push in the servers config you want to push to?
- Do you have any limitations to the push rules e.g. limited to a certain TLP Level tag or other?
- What is written in your job log?

  https:///jobs/index

Have a look at: /var/www/MISP/app/tmp/logs and /var/log/apache2/misp (or the relevant apache log folder of the instance in cause)

51.2.11 I have many users or API accesses, what's the best PHP session handler?

We strongly recommend production-level MISP installations to rely on PHP session in Redis. As Redis is already part of a standard MISP setup, we recommend to enable the redis session handling.

To configure the redis session handling in PHP, edit :

```
session.save_handler = redis
session.save_path = "tcp://127.0.0.1:6379
```

51.2.12 Is there TAXII support?

Yes. MISP 2.5 can push events to TAXII 2.x collections directly: configure a TAXII server under **Sync Actions → TAXII Servers** (see the Synchronisation and Sharing chapter).

For TAXII 1.x, a separate server implementation is available at https://github.com/MISP/MISP-Taxii-Server, which receives STIX files into its inbox and uploads them to MISP.

51.2.13 Wipe MISP data - Remove all data

If you need to start from scratch with your MISP database and remove all data you can use the `misp-wipe` script provided in the `tools/` folder.

51.2.14 Constantly acknowledging my self-signed certificate drives me nuts

You want to add it in 2 places: Your browser(s) and your OS.

The following steps can be performed on the CLI to install the Certificate:

```
sudo mkdir -m 0755 /usr/local/share/ca-certificates/MISP
sudo cp /etc/ssl/private/misp.local.crt /usr/local/share/ca-certificates/MISP
sudo chmod 0644 /usr/local/share/ca-certificates/MISP/misp.local.crt
sudo update-ca-certificates
```

For the Chrome Browser:

1. Visit: "Advanced Settings" -> chrome://settings/?search=Manage+certificates
2. Scroll down to: Manage Certificates (click)
3. Select: "Authorities"
4. Click: "Import"
5. Browse to your .crt file and import it.
6. On the next screen tick: "Trust this certificate for identifying websites"
7. Done, enjoy the new gained quality of life

Note: Chrome might expect a Subject Alternative Name make sure you created your certificate with '-extension san'.

To allow insecure localhost connections enable this option: chrome://flags/#allow-insecure-localhost
Sources: CLI and Chrome/Chrome insecure localhost
For the Firefox Browser

51.2.15 How can I change the theme?

MISP uses bootstrap.css the specific CSS file can be found on a typical MISP install at `/var/www/MISP/app/webroot/css/bootstrap.css`.

You can customize this for your own needs. There are also pre-made boostrap themes which you can use as-is or build upon.

Before making any changes, confirm the version of boostrap currenlty used by running `head -5 /var/www/MISP/app/webroot/css/bootstrap.css`. You can find themes on sites like Bootswatch.

To replace the current theme with a theme you found on bootsplash, run: `sudo -u www-data wget https://bootswatch.com/2/readable/bootstrap.css -O /var/www/MISP/app/webroot/css/bootstrap.css` , replacing the URL as needed.

Some bootswatch themes applied on MISP: * https://i.imgur.com/usONTLk.png * https://i.imgur.com/5XMjB7o.png * https://i.imgur.com/5gc57VU.png * https://i.imgur.com/4AJCPgf.png * https://i.imgur.com/JuMGm8U.png * https://i.imgur.com/v1Wu6xW.png

51.2.16 How can I deal with a MISP instance that has pulled in feeds over and over into new events, generating hundreds of GBs of junk correlations, rendering the instance unusable?

Step 1: ensure that all your CSV/freetext source_format feeds are using the fixed event setting. If you want to make sure this is the case, you can run this SQL query instead of doing it manually:

```
UPDATE feeds SET fixed_event = 1 WHERE source_format="csv" OR source_format="freetext";
```

Step 2: purge all of your correlations (this will make the next steps much faster), for which you have two methods at your disposal: - either go to your administration -> server settings -> MISP tab and set `MISP.completely_disable_correlation` to true - via MYSQL run `TRUNCATE correlations;`

Step 3: purge all of your feed data that have been pulled into multiple events. The easiest way of doing this: check which feeds are enabled (ignore misp source format feeds, they are not causing issues) and note down the IDs. Afterwards, use the CLI cleanup tool to remove all the feed events:
`/var/www/MISP/app/Console/cake Admin purgeFeedEvents [user_id] [feed_id]`

Execute this for each feed that you had enabled, replacing user_id with your admin user's ID and feed_id with the individual feed IDs on your list.

Step 4: recorrelate your data, depending on which method you've used in *Step 2* you have two options: - either go to your *Administration -> Server Settings... -> MISP...* tab and set `MISP.completely_disable_correlation` to *false* - recorrelate your current data-set via the recorrelate attributes tool on `/pages/display/administration`

51.2.17 I have a long list of events that I want to delete via the API, do I really have to loop through each and issue a delete to /events/delete?

No, the delete action also accepts a list of IDs when it comes to bulk event deletions.

Simply POST your ID list to `/events/delete` in the following format:

```
{
  "id": [1,3,5,7,9]
}
```

51.2.18 I can no longer log in. How do I reset the admin password?

You can reset the password via the console. See Issue #1160
`/var/www/MISP/app/Console/cake Password [email] [password]`

51.2.19 How do I set the baseurl from the command line?

You can change the baseurl via the console.

```
sudo -u www-data /var/www/MISP/app/Console/cake Baseurl [baseurl]
```

You can confirm the baseurl is updated correctly by checking the config.php file

```
grep baseurl /var/www/MISP/app/Config/config.php
```

51.3 Usage questions

51.3.1 How can I see all the deleted events in a MISP instance?

You can use the logging system for this, to see all deleted events, simply go to *Audit -> Search Logs* and use the following parameters:

```
model: Event
action: delete
```

This will list all event deletions. To find out more about what a particular deleted event was, simply grab the ID from the above search results and search for:

```
model: Event
action: add
model_id: <Event ID retrieved from the listing of all event deletions>
```

To do the same via the API, first search for the deletions:

```
POST request:
  url: https://url.of.your.misp/logs/index
  headers:
    Authorization: <your_api_key>
    Accept: application/json
    Content-type: application/json
  Body:
    {
      "model": "Event",
      "action": "delete"
    }
```

Then find the individual event's metadata that was deleted

```
POST request:
  url: https://url.of.your.misp/logs/index
  headers:
    Authorization: <your_api_key>
    Accept: application/json
    Content-type: application/json
  Body:
    {
      "model": "Event",
      "action": "add",
      "model_id": "<Event ID retrieved from the query before>"
    }
```

51.4 Permission issues

If you have any permission issues, please set the permissions to something sane first.

51.4.1 RHEL/CentOS

There are a plethora of issues that might arise when using SELinux when it comes to permissions. First, please familiarize yourself with the basics of SELinux. RedHat has a comprehensive SELINUX USER'S AND ADMINISTRATOR'S GUIDE.

For file system permissions, refer to the install guide first.

Another way to see what SELinux might not be happy about is to use **ausearch**. This assumes Audit is enabled.

```
# Just php-fpm
sudo ausearch -c 'php-fpm' --message AVC
# All messages
sudo ausearch --message AVC
```

51.4.2 Redis Connection problems

If you have the following in **error.log**

```
2019-05-08 10:16:05 Error: [RedisException] Permission denied
Request URL: /events/view/1
Stack Trace:
#0 /var/www/MISP/app/Model/AppModel.php(1776): Redis->connect('127.0.0.1', 6379)
#1 /var/www/MISP/app/Model/Feed.php(329): AppModel->setupRedis()
#2    /var/www/MISP/app/Model/Event.php(2073):    Feed->attachFeedCorrelations(Array,
Array, Array, false)
#3  /var/www/MISP/app/Controller/EventsController.php(1547):  Event->fetchEvent(Array,
Array)
#4 [internal function]: EventsController->view('1')
#5             /var/www/MISP/app/Lib/cakephp/lib/Cake/Controller/Controller.php(499):
ReflectionMethod->invokeArgs(Object(EventsController), Array)
#6   /var/www/MISP/app/Lib/cakephp/lib/Cake/Routing/Dispatcher.php(193):   Controller-
>invokeAction(Object(CakeRequest))
#7   /var/www/MISP/app/Lib/cakephp/lib/Cake/Routing/Dispatcher.php(167):   Dispatcher-
>_invoke(Object(EventsController), Object(CakeRequest))
#8 /var/www/MISP/app/webroot/index.php(92): Dispatcher->dispatch(Object(CakeRequest),
Object(CakeResponse))
#9 {main}
```

This means that apache/php-fpm cannot connect over the network (localhost included).

Fix:

```
sudo setsebool -P httpd_can_network_connect on
# Perhaps a reload is not needed, but good practice wants us to test it anyways.
sudo systemctl restart php-fpm.service
sudo systemctl restart httpd.service
```

51.5 RHEL/CentOS SELinux debug

More often than not there might be issues with SELinux when not configured correctly. The below will give you pointers where to look and how to figure out what is wrong.

You can investigate SELinux issues without any tools by opening the audit log it generates. This log is found at /var/log/audit/audit.log. However, unless you know exactly what to look for and have a lot of free time, you're going to find it difficult making sense of the log.

Install some handy tools:

```
# Note: This will pull in some X tools, you have been warned
sudo yum install setroubleshoot setools
```

We now have a tool called sealert that analyzes the audit log used by SELinux. Sealert will scan the log file and will then generate a report containing all discovered SELinux issues. In this overview of what went wrong you will see suggestions on how to fix them after the issue detected.

To run sealert from the command-line, we need to point it to the SELinux audit log.

```
sudo sealert -a /var/log/audit/audit.log
```

51.5.1 Clearing the audit logs

It is not recommended to clear the audit logs as they might contain information needed in the future for troubleshooting or security investigations. However, if that is not the case, just empty the audit log:

```
# > /var/log/audit/audit.log
```

Partial source
StackExchange
Gentoo Wiki

51.6 When to update MISP?

One question might be how often to update MISP. You can update MISP as ofte as you like. If you see the follwing:

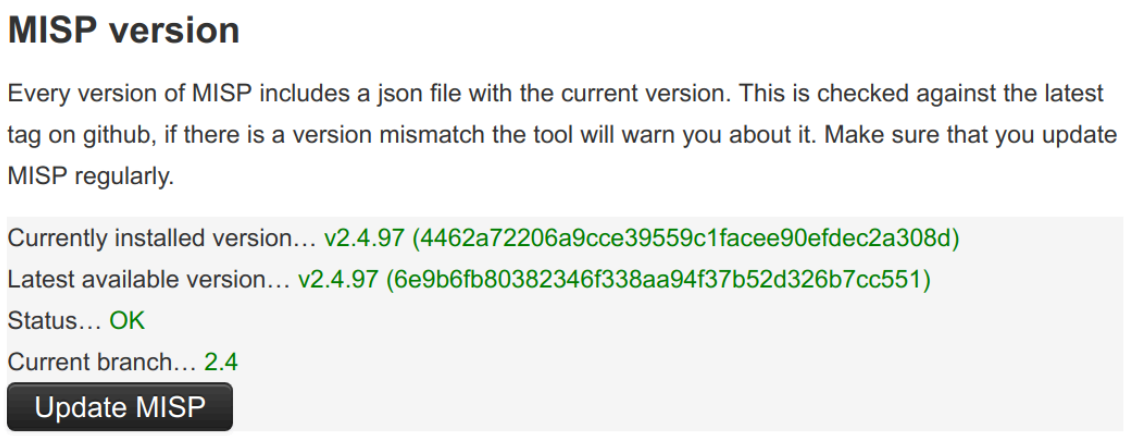


Figure 51.1: MISP Update

This means that the main repository has an update available.

If you want to play it safer or want to integrate it in your Weekly/Bi-Monthly update routine you can track our Changelog a more up to date version is available here

Further on we do regular tagged releases. (Approximately once per month.) The releases happen either if a milestone has been hit for a certain feature/improvement/fix or for any security related matters.

Thus you have the choice of either tracking 2.4 which is on a rolling release schedule, or track the tagged releases.

51.6.1 How to switch from tagged releases and back?

This can be achieved with the following git commands:

```
$ cd /var/www/MISP # aka. $PATH_TO_MISP
$ sudo -H -u www-data git checkout tags/$(git describe --tags `git rev-list --tags --
max-count=1`)
```

51.7 OS Upgrades

In theory all should "just work"(tm), but in practice the following dependencies might make your install unstable and need a little though before just doing the updates.

- php/pear

- python
- apache
- init scheme/scripts
- mariadb/mysql
- redis
- git

51.7.1 PHP

This is probably the most likely one that might get you into trouble. The following happened on a Debian Testing lately. During the upgrade php got upgraded to php-7.3 and seemingly some php-7.2 dependencies were deinstalled and the system now had 2 concurrent versions of php installed. The fix was to remove any *libapache2-mod-php7.2* packages and make sure that *apt remove libapache2-mod-php7.3* was installed. Most certainly you need to add symbolic links to */etc/apache2/mods-enabled* to make php7.3 work. Then double check if all the php dependencies are install, refer to the install documents.

The same for pear, where we mostly use 2 (bundled) packages: Console Command Line, Crypt GPG. If you upgrade from a very old and out of date version of MISP this might raise issues.

php.ini might also become problematic if you just erase the recommended defaults.

51.7.2 Python3

If you use python2 for MISP, please read the install docs about MISP being Python 3 only. Currently Python3.6 is minimum. It is known working on 3.7 with some minor difficulties (see PyMISP issues). The biggest issue is certainly with PyMISP doig unexpected things when python might be updated. Using a virtualenv, whilst not always ideal for all setups, will at least make sure that problems are contained a little more.

51.7.3 Apache

Mostly config issues might be a show stopper. And major version updates where some underlying config might need to be changed.

51.7.4 init/systemd

MISP launches a couple of things on boot. Changing what handles boot behavious might have an impact.

51.7.5 MariaDB/MySQL/redis

Similar to apache, most importantly always take good care that the DB engine is not all of a sudden changed without you noticing it. From minor to major updates, rarely things might need to be adapted.

51.7.6 git

Currently (as of v2.4.108) the git-cli command is used in MISP core. In very rare cases where the expected output changes, this might be an issue. Included here more as an FYI then anything else.

51.8 Hardening

51.8.1 How do I harden my MISP instance?

You can check the hardening section in the install guide.

51.9 Maintenance mode

51.9.1 Is there a MISP maintenance mode?

Yes, you want to flip your instances “Live-mode”. This wants to be done on the CLI if you experience issues:

```
$PATH_TO_MISP/app/Console/cake Admin live 0
```

Other related MISP Settings

Optional MISP.maintenance_message Great things are happening! MISP is undergoing maintenance, but will return shortly. You can contact the administration at $email or call CIRCL. The message that users will see if the instance is not live.

Critical MISP.live true Unless set to true, the instance will only be accessible by site-admins.

51.10 Update MISP fails

If your MISP instance is outdated, meaning ONLY the core, not the modules or dashboard or python modules, you well see the following.

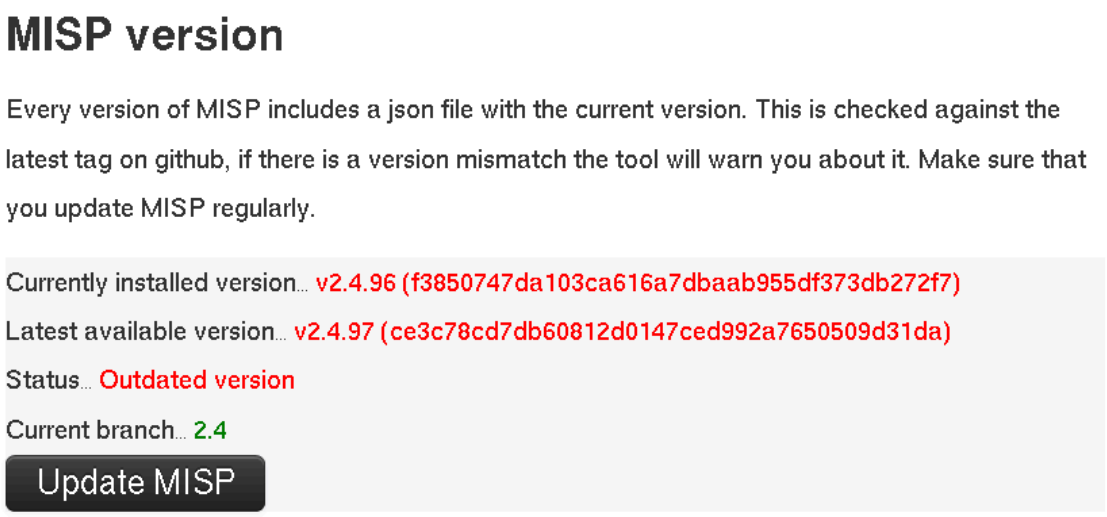


Figure 51.2: MISP outdated

Once you click on update MISP you will be asked confirmation.

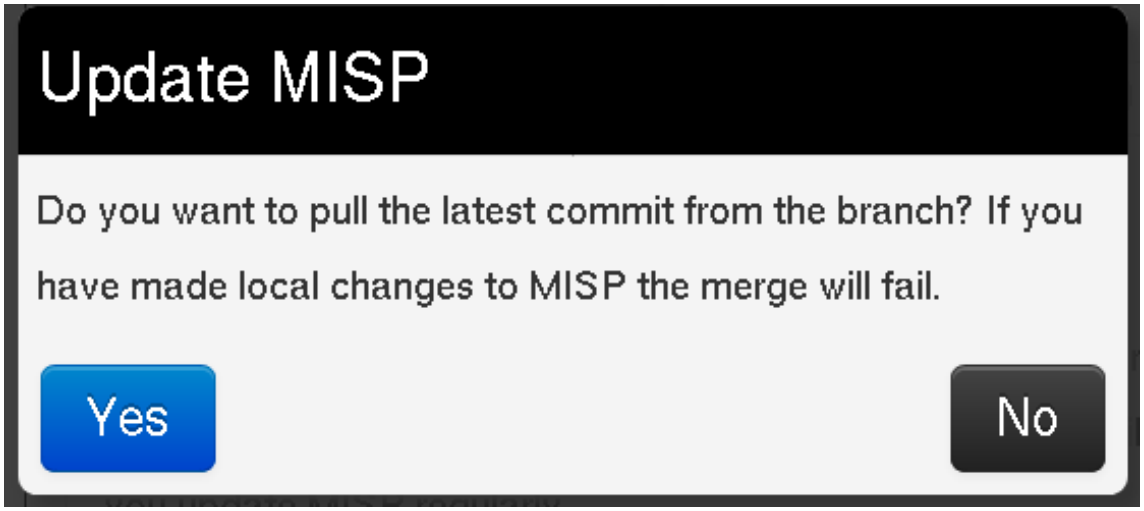


Figure 51.3: MISP Update Yes/No

If you are not on a branch, the UI will tell you this, the update will fail.



Figure 51.4: not on branch

If you cannot write the **.git** files and directory as the user running the web server (and thus PHP), the update will fail. The following diagnostic check will let you know if you can update or not.

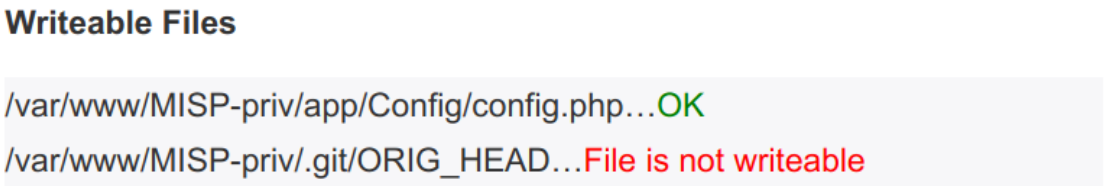


Figure 51.5: .git not writeable

In case you get a file not found on **.git/ORIG_HEAD**, this means that you have never updated your MISP OR you have installed git from an archive file (like .zip/.tar.gz or similar) Try to click update MISP and see what happens.

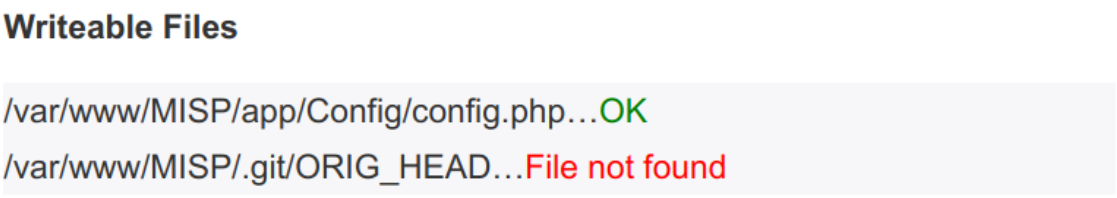


Figure 51.6: ORIG_HEAD file not found

51.10.1 What can go wrong if I update MISP?

In theory nothing. We put great effort into protecting the integrity of the data stored in your MISP instance. DB upgrades happen upon login or on reload once you have update the repository. You cannot "break" anything by clicking **Update MISP** worse case it will complain about something and you will certainly find the answer on this page.

IF not, please open an issue on GitHub or come to our gitter chat to see if the community can help.

51.10.2 error: pathspec 'app/composer.json' did not match any file(s) known to git

This is **not** an error and can be ignore. Nothing will be impacted by this.



Figure 51.7: pathspec

51.10.3 MISP modules “Connection refused”

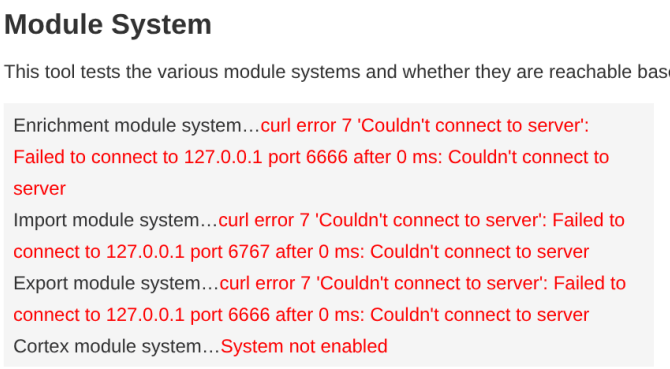


Figure 51.8: MISP Modules

If you get have a **Connection refused state** on your modules one of the following might be true.

- You have no misp-modules not installed
- They are instaled but not running
- Something completly different

If they are not installed, check out this section of the INSTALL guide of misp-modules.

In case they are not running, try this on the console:

```
sudo -u www-data /var/www/MISP/venv/bin/misp-modules -l 127.0.0.1 -s &
```

OR if you were foolish enough to not install in a Python virtualenv:

```
sudo -u www-data misp-modules -l 127.0.0.1 -s &
```

> ⚠ Warning
>
> Running misp-modules like this will certainly kill it once you quit the session. Make sure it is in your **/etc/rc.local** or some other init script that gets run on boot.

51.11 Uninstalling MISP

There is no official procedure to uninstalling a MISP instance.

If you want to re-use a machine where MISP was installed, wipe the machine and do a fresh install. Consider the data in your MISP instance as potentially confidential and if you synchronized with other instances, be respectful and wipe it clean.

51.12 Updating PyMISP to incorporate newer versions of the MISP object templates

In some cases, for instance if a newer version of a MISP object is present on the server but not yet on PyMISP, you want to reflect the current state in your PyMISP installation.

In order to do so, perform the following steps. It fetches the latest object templates and installs PyMISP again:

```
git clone https://github.com/MISP/PyMISP.git
cd PyMISP/pymisp/data
git submodule update --init
cd misp-objects
git pull origin master
```

```
cd ../../../
sudo pip3 install -I .
```

51.13 How to disable freetext/custom/user-created tags and only allow certain tags

Remove the "tag editor" from the permissions that you grant to users. Set all tags that you do not want to "hidden". There is a server setting to treat all incoming tags as hidden by default: `MISP.incoming_tags_disabled_by_default`

Important Make sure that you don't remove "tag editor" from sync users, or you'll be stripping tags from synchronized data.

51.14 How to enable the csv import module?

First you have to enable the import services: double-click on "false" in the very first line and change it to "true".

In Server Settings & Maintenance -> Plugin Settings -> Import -> set "Plugin.Import_csvimport_enabled" to true. Afterwards you'll find the csvimport from within the newly created event: "Populate from..."

Don't use from the main site ("Import from...").

51.15 Why do I see 'The request has been black-holed' when I submit forms?

That's a security measure for form tampering protection.

All forms have a timeout (~15min) and all of them can only be submitted once. If you use your browser's "back" button and resubmit the form MISP will consider it as a potential attempt at form tampering.

51.16 Importing large feeds creates PHP Fatal error

When importing a large feed like the CIRCL feed, the job reaches 99% and then fails. The log file records:

```
PHP Fatal error: Allowed memory size of 536870912 bytes exhausted (tried to allocate
1941504 bytes) in /var/www/MISP/app/Model/Feed.php on line 691
```

In this case you will need to increase the memory_limit option in `php.ini` file

51.17 I deleted the admin user by mistake

Now, I only have Org Admin.

You have several options:

1. Delete the org admin. MISP automatically creates a new default site-admin user if no users are found in the db (mysql: truncate users;)
2. Upgrade a user to a site-admin, such as an org admin user:

```
SELECT id, email from users;
```

Note down the ID you want to upgrade. Let's say this is 2 for the example's sake.

```
SELECT id, name from roles;
```

Note down the role ID you want to upgrade. Let's say this is 1 for the example's sake.

```
UPDATE users set role_id = 1 where id = 2;
```

51.18 config.php is not writeable

```
Warning: app/Config/config.php is not writeable. This means that any setting changes
made here will NOT be saved.
```

According to the install guide, make sure to:

```
chown -R apache:apache /var/www/MISP
find /var/www/MISP -type d -exec chmod g=rx {} \;
chmod -R g+r,o= /var/www/MISP
```

If it still doesn't work, make sure SELinxu is not enabled or modify the rule set:

```
chcon -t httpd_sys_rw_content_t /var/www/MISP/app/files
chcon -t httpd_sys_rw_content_t /var/www/MISP/app/files/terms
chcon -t httpd_sys_rw_content_t /var/www/MISP/app/files/scripts/tmp
chcon -t httpd_sys_rw_content_t /var/www/MISP/app/Plugin/CakeResque/tmp
chcon -R -t httpd_sys_rw_content_t /var/www/MISP/app/tmp
chcon -R -t httpd_sys_rw_content_t /var/www/MISP/app/webroot/img/orgs
chcon -R -t httpd_sys_rw_content_t /var/www/MISP/app/webroot/img/custom
```

51.19 How to debug misp-dashboard

This is the full chain from MISP to the live dashboard and some tips to find out which link is faulty.

1. MISP Ensure that ZMQ is installed and enabled with the correct settings
2. MISP ZMQ You can use MISP/tools/misp-zmq/sub.py which will subscribe to the ZMQ and print the data
3. ZMQ_subscriber You can change the logging level from logging.INFO to logging.DEBUG and look in the logs for the string Pushed: * Or add a print statement in the put_in_redis_list function
4. ZMQ_dispatcher Look in the logs for the string Handling *
5. Server (Flask) and Browser (live Dashboard) Open the Web developer Network tab in your browser and look for the url /_logs with Content-Type: text/event-stream;

51.20 How to update object templates?

`git submodule update` in your MISP directory (or via the diagnostic page) and just click "Update Objects" in List Object Templates.

51.21 What to do if my REST client is throwing SSL errors when trying to query my MISP instance?

The REST client will use the framework's certificate store to validate the contacted host. If your root CA / self-signed certificate is not known by the certificate store, the request will fail. You can skip the SSL validation altogether using the "Skip SSL validation" checkbox.

51.22 What to do if my REST client cannot reach the host, despite me being able to issue requests using Curl / Postman / etc.?

The REST client issues instructions to your MISP server to contact a remote host (most commonly itself). Always consider how your MISP server can address itself when using the REST client, by default it will prepend the requested relative path in the URL field with the instance's baseurl.

If your MISP cannot reach itself via the baseurl the request will fail. You can use the "Use full path - disclose my API key" checkbox along with the full URL in the URL field to instruct MISP to use another path than what it would construct using the baseurl.

51.23 How would one set up a sharing group with a remote org, where we only share a mutual community instance (i.e. we both have sync users on that instance). On our local instance, they exist as a remote org (from events that have synced from their instance via our shared community instance).

It is not possible to do that. Keep in mind that if you are both on a mutual community instance, someone is in charge of that instance that will have database and admin level access. They would be able to inspect the data you exchange on their community instance with one another, so MISP will block any attempt to share with them.

If you really want to go through the community instance to exchange with them, you explicitly have to include the host organisation of the community instance (they would get access if they wanted to anyway, this way we can ensure that you are clear about that):

- You are org a on instance A.
- Your partner that you want to share with is org b on instance B.
- You have no way of directly reaching org b, but you both have access to instance C, which is run by org c (the sharing instance)
- In order to reach org b, you have two options for sharing groups, depending on whether you want to be able to push to them or want to rely on them pulling data from the community instance:

```
SG Option 1 (push all the way to B)
  orgs: a, b, c
  instances, A, B, C
```

```
SG Option 2 (b has to pull from C):
  orgs: a, b, c
  instances: A, C
```

51.24 Is it possible to propose objects to an event?

This is not possible yet. What you can do at the moment: Create a new event and extend it with the other (foreign) event.

51.25 How to use the enforceWarninglist parameter in REST search?

If you would like to export IoCs, for example into a suricata rule and exclude all values matching your warning lists, you can use the following:

```
{
    "returnFormat": "suricata",
    "published": 0,
    "enforceWarninglist": 1
}
```

Keep in mind that unpublished events need the `"published": 0` parameter in order to be exported.

51.26 Column not found issue

When a user attempts to add an object to an event and the following error is received (Level 1 debug enabled):

```
SQLSTATE[42S22]: Column not found: 1054 Unknown column 'Event.org_id' in 'where clause'
```

One potential resolution is to upgrade MISP to 2.4.107.

51.26.0.1 Symptoms

Users with the site-admin role are able to add objects to events without any error. This error was encountered when a user belonged to every role **except** site-admin.

51.27 WatchList Customization

51.27.1 How to create a customized WatchList.

WatchLists are stored within folder under /var/www/MISP/app/files/warninglists/lists Every folder contains a list.json file. Create a new folder and copy and modify an existing list (or create a new one from scratch). Ensure the "name" value within the file is unique. Increment the version number when the file is changed.

Within the MISP GUI, go to WarningLists and "Update WarningLists".

The new WarningList will now show up. In case of errors, check the permissions on the list.json and it's folder.

To modify the list or to add entries to it, go back to the file via the CLI, modify the file and reload it via the GUI ("Update WarningLists".

51.28 How to add a galaxy to an event via PyMISP

A galaxy can be assigned like a tag. You can use the add tag function and copy the full conntector-tag. Example `misp-galaxy:ransomware="Locky"`, which can be found in `/galaxy_clusters/view/`

51.29 How to block attachment (and malware-sample) to be ingested

For that, we'll use the Import Regex feature in MISP.

1. Got to `/regexp/admin_index` (In the UI under Input Fulters -> Import Regexp)
2. Create a new import Regex `/admin/regexp/add`
   - Regexp: `/^.$/i`
     - Note: If you want to only allow specific file extensions, you can use a regex a similar regex (in this case, we only allow .png attachment): `/^.*\.(?!png)[^.]+$/i`
   - Replacement: `[Leave the field empty]` (Leaving the field empty will prevent that value to be saved)
   - Select the checkboxes `attachment` and `malware-sample`
3. From that point onward, users will still be able to add attachement but they won't be accepted by the system anymore.
   - Note: If you allowed specific file extensions, only file ending with the provided extension will be accepted. Please, bear in mind that it only looks at the filename and skip MIME checks alltogether.

51.30 What are the required steps after a MISP installation to have a properly running instance?

- First login with the installation credentials and change the password immediately (especially if your instance is publicly accessible)
- Set the base_url to the hostname of your machine (apache virtualhost name)
- Create a new organisation which will be the host organisation running the MISP instance
- Set the new organisation in `MISP.host_org_id` to replace the default one
- Set messages like `MISP.footermidleft` and alike to a proper message to help your users
- Create a new user as `admin` role with the new organisation
- Log with the new user, if successful, remove the default user used during the installation such as `admin@admin.test`
- Select and enable required taxonomies for your sharing community
- Select and enable the external feeds (as caching only if you don't want full events but you can get the full feeds too)
- Select and enable the warning-list (if you don't know what to enable, select all)
- Add the remote MISP instances where you have access to (either caching only or full pull if you want the complete events)

52. Developer FAQ

52.1 Main Developer Resources

The main developer resources can be found on GitHub in the MISP Wiki.
The following pages are worth inspecting closer in case you want to actively develop for MISP:

- The real FAQ
- Contributor Overview
- Some objectives of MISP
- Various deployment tools
- MISP Code of Conduct
- UI coloring scheme
- Notes on MISP and STIX 2
- Commit Messages Best Practices
- Internationalization (i18n)

Our gitter channel is a welcome place to ask other community developers in case you are stuck.

52.2 About the codebase

A few orientation points for new contributors to MISP core (2.5):

- MISP core is a **CakePHP 2** application (`app/Controller`, `app/Model`, `app/View`), running on **PHP 8.x**. The data model is captured in `db_schema.json` and the attribute type system in `describeTypes.json`.
- Background processing runs on **SimpleBackgroundJobs** (Supervisor + Redis) — the legacy CakeResque worker system has been removed. See the administration chapter.
- Much of MISP's content (galaxies, objects, taxonomies, warninglists, noticelists, event/object templates) lives in companion repositories pulled in as submodules under `app/files/`.
- The REST API is described by an OpenAPI specification served at `/api/openapi`.

53. User Stories

User story	Example workflow
As a lead threat intelligence analyst, I want to lead a team focused on hunting down threats so that I can prevent attacks against ICT infrastructures and organizations	• Monitor what teams are up to in real-time using the Live Dashboard
As a threat analyst, I want to research, analyze and reverse engineer malware so that I can know how to counter it	• Attach and download files and malware samples from events• Search for hashes/IPs/domains/URLs from malware events, or add malware samples hashes to an event• Analyse observables and malware collected during an incident (e.g. domain name, IP addresses etc.) by checking whether observables are IoCs or false positives using 'correlation graph' and 'expansion modules'.• Enrich malware events by querying data sources external to MISP using modules• Perform dynamic malware analysis correlations• Submit events with malware samples to analysis tools (e.g VirusTotal, VMRay) for further analysis, and then extend MISP with malware analysis results
As a lead threat intelligence analyst, I want to convert threat data into actionable threat intelligence so that I can improve security posture.	• Import data from external sources• Add feeds• Contextualise events and attributes using tags, taxonomies and galaxies
As a threat analyst, I want to exchange threat information with third parties so that we can gain shared situational awareness	• Setup different models of distribution on MISP instance• Sync events and attributes between instances• Use filtering functionalities to meet an organisation's sharing policy• Share information, pentest information, malware samples, vulnerabilities internally and externally
As a threat analyst, I want to monitor threats and access live data so that I can manage threats before they cause major damage	• Import lists of indicators and check if the IOCs are present in feeds.• Monitor statistics and sightings using widgets• Show live data and stats from one or more MISP instances via the Dashboard• Process information in real-time when it's updated, created, or published by instances by integrating with ZMQ• Use sightings to notify an instance about activities related to an indicator
As a threat analyst, I want to aggregate and compare indicators from various sources so that I can connect the dots between various threats	• Join communities and subscribe to the feeds• Add events and assign events to specific feeds• Correlate indicators using MISP's automated correlation engine• Use the overlap feed analysis available in MISP• Link events and attributes using the correlation graph• Analyse and gain more information on attributes using modules• Link events with malware, threat actors etc using galaxies (e.g ATT&CK)

User story	Example workflow
As a threat analyst, I want to have a structured database of threat data that I can use to perform lookups/queries when investigating new threats	• Store information in a structured format using STIX• Import unstructured reports using the free-text import tool• Use MISP as a centralized hub for security and fraud threat intel. Centralize threat intel by aggregating indicators from OSINT and commercial feeds• Remove false positives and duplicates• Score indicators based on Sightings and other metrics• Import/integrate feeds or threat intelligence from third parties• Generate, select, exchange, and collect intelligence using feeds• Select and import events• Look for correlations between events using the correlation graph• Build filtered subsets of the data repository for feed creation.• Preview and correlate feed data directly for evaluation
As a threat analyst, I want to contextualize and enrich raw threat data so that I can produce actionable intelligence	• Understand attacker TTPs by using taxonomies to link events• Categorize risks and incidents using galaxies and taxonomies• Quickly classify information using tags collections• Contextualise sightings with information on the source• Enrich IDSes export with tags to fit your NIDS deployment• Decay attributes and score indicators using sightings (reported by IDSes)• Describe and visualise complex scenarios using MISP's richer data structure• Allow advanced combination of attributes using MISP objects
As a threat analyst, I want to investigate threats so that I can protect computer systems from attacks	• Find relevant data for investigations from MISP communities. Preview new MISP events and alerts from multiple sources such as email reports, CTI providers, and SIEMs• Query a MISP instance for events that include a given IOC. Browse through other MISP events, attributes, objects, tags, and galaxies• Create events, add IoCs (attributes), and contextualise (using tags)• Pivot an event into its attributes, objects, tags, galaxies, and/or related Events• Explore further details from Galaxies and related Events• Categorize available related information within the ATT&CK framework.• Query tools (e.g Cytomic Orion API) to check if certain MISP indicators have been observed, and the import sighting details to add them to MISP events• Prioritize threats using Sightings collected from users, scripts and IDSes.• Decay/expire indicators using sightings reported by users, scripts and IDSes• Launch lookups from MISP against SIEMs as part of an investigation• Correlate network forensic flows from several tools
As a SOC team, we want to ingest, analyse, store and make connections between threat data so as to discover potential threats	• See connections between events using the correlations graph• Import CVEs and vulnerabilities (e.g from MetaSploit) and contextualise them• Contextualise CVEs using events gotten from articles/reports• Convert CVE information into a feed• Pull shared CVE feeds• Combine collected data with your MISP data set for correlation•

User story	Example workflow
	Share correlated info to the team using the export function or API search• View current threats and activity, historical, geolocalized information using MISP Dashboard
As a junior SOC analyst, I want to enrich alerts so that I can "punch above my weight" and make connections that would have otherwise required more experience	• Create events, add/import observables• Use Cortex and its analyzers to gain insight• Leverage tags, sightings, and previously-seen observables to feed your threat intelligence• Export IOCs to MISP instances after investigations are complete• Integrate MISP with Maltego to generate visualisations of data• Integrate MISP with Elastic to access threat data without the complexities of the MISP interface.• Push attributes from MISP to Elastic and have a representation with graphs, an alternative to using MISP Dashboard.• Create taxonomies using the taxonomy editor.• Contextualise data using taxonomies, clusters and galaxies
As a SOC analyst, I want to customize risk feeds to ignore or downgrade alerts that do not match organization/ industry-specific criteria, so that I can focus on relevant alerts	• Filter incidents based on taxonomies (e.g the veris country taxonomy to indicate countries affected by an incident)• Normalise external input and feeds in MISP (e.g. feed importer).• Compare feeds before import to find similarities and false positives.• Evaluate the quality of the information before importing it (warning-list lookups at feed evaluation)
As a SOC analyst, I want to share real-time information pertaining to new or existing cases/ observables to team members so that we can collaborate on investigations simultaneously	• Control threat sharing using 'distribution settings': sharing group, community-only, connected communities, all communities.• Share sensitive and confidential events using the 'sharing group' functionality• Measure the impact of an incident using taxonomies based on NISD/OESs impact criteria• Export and share sightings in ATT&CK sightings format to give insights on TTPs and frequency of usage
As a SOC analyst, I want to rule out false positives so that I can focus on significant threats	• Weed out false positives using warning lists• Crowd source data validation from community• Filter indicators based on specific criteria• Receive information on false positives using collaborative tools (proposals, sightings)
As a threat analyst, I want to remove false positives, filter and prioritize alerts so that I can focus on what really matters to my organization	• Evaluate the quality and freshness of indicators using decaying models• Enforce warninglists to exclude events with certain attributes• Enable warninglists to alert for certain issues• Classify information (add/remove tags) based on their score or visibility via sightings• Use tags to set events or attributes for further processing by external tools (e.g. VirusTotal auto-expansion using enrichment modules)• Notify an instance about activities related to an indicator via Sighting• Limit NIDS exports and improve rules using Sightings• Filter indicators based on specific criteria• Filter out relevant data when feeding protective tools

User story	Example workflow
As a security analyst, I want to unravel the inner workings of a malicious file, phishing email or domain so that I can prevent attacks	• Integrate MISP with a Security Incident Response Platform (e.g TheHive)• Import indicators from MISP into the SIRP for further analysis
As a security analyst, I want to create blacklists/ whitelists (e.g of domains) so that I can protect customers from malicious activity	• Import threat data into MISP from synced servers and label using taxonomies• Enable warning lists, and exclude attributes that exist on the warning lists• Create lists with preferred attributes and export the list in an easy accessible format as CSV
As a security analyst, I need a real-time overview of threat information so that I can quickly glance at important metrics	• Integrate ZMQ to access a dashboard showing live data and stats• Monitor ongoing trends based on interests using the EventStream widget• Monitor activity in real-time on MISP dashboard by subscribing to ZMQ feeds• View immediate contributions made by organisations from MISP's live dashboard• Find threats within your constituency using MISP Geolocalisation Dashboard• Get geospatial threat information from specific regions using the Geolocalisation Dashboard
As a security analyst, I want to automate repetitive tasks related to data normalization, importation, aggregation and enrichment so that I can have more time to put into threat analysis efforts	• Automate tasks using PyMISP• Use PyMISP for Scripted processing of events and attributes
As a security analyst, I want to collaborate with other analysts within and out of my organization's sector so that we can support one another	• Build or join communities to exchange specific data structures• Share real-time analysis of an incident• Propose modifications to someone else's analysis using Proposals
As a security analyst, I want to triage and prioritize alerts so as to avoid alert fatigue	• Evaluate the quality and freshness of indicators using decaying models• Weed out false positives using warning lists• Enable warning lists to alert for critical issues• Filter indicators based on specific criteria• Score indicators based on user sightings, including negative sightings and expiration sightings.• Classify information (add/ remove tags) based on their score or visibility via sightings
As an incident responder, I want to get an up-to-date picture of the threat landscape so that I can prepare for threats in advance	• Describe the impact of threat using taxonomies (e.g using the veris timeline taxonomy to indicate the duration of the incident)• Classify data to gain insight into the threat landscape.• Classify data so IDSes can alert on a rule• Integrate ZMQ to have a dashboard showing live data and statistics.• Integrate ZMQ to process information in real-time when it's updated, created, or gathered in MISP.
As an incident responder, I want to identify and respond to incidents so that I can reduce the impact and severity of an attack	• Report false or true positives using the sighting mechanism, based on an incident investigation• Decay indicators to guarantee the quality of the indicators
As an incident responder, I want to receive early warnings and alerts about threats/incidents so that I can retaliate before they cause any harm	• Receive correlated threat intel from sharing groups and communities• Monitor MISP feeds for alerts• Preview new events and alerts from multiple sources• Automate import/export of IoCs to/

User story	Example workflow
	from protective or detection tools like IDSes and IPSes• Dispatch notifications when certain events are created or modified using the alert feature• Create filter rules based on personalised uses. Restrict alert messaged by tags, publishing organisation or other metrics
As an incident responder, I want to store information identified during an incident investigation so that I can perform lookups/queries against the historical database during future incidents	• Use a MISP instance as a database of events representing incidents. Store incident response data internally in a structured manner on MISP• Represent indicators using attributes. Attributes such as network indicators (e.g. IP address) or system indicators (e.g. a string in memory)• Combine OSINT and your own intelligence• Create events made up of indicators (attributes) and then leverage these as a threat data feed• Modify events representing incidents to enable monitoring over time• Add object types to describe incidents• Monitor indicators for relevancy using Sightings• Ensure information quality and freshness by expiring indicators depending on their personalised objectives• Pull events from indicator lists to perform lookups against SIEMs• Use indicators to check logs and verify if you're affected by a threat• Correlate indicators with actual incidents to get more information• Integrate MISP with IR tools (e.g TheHive) to (1) analyse observables during an incident, (2) import and (3) export events from MISP to TheHive and vice-versa• Perform large-scale bulk data/traffic analysis and correlation against your MISP database using SightingsDB
As an incident responder, I want to export and feed data between security tools so that I can enhance their functionalities	• Export data from MISP to feed protective/detective tools and early warning systems. Export formats support IDSes / IPSes (e.g. Suricata, Bro, Snort), SIEMs (eg CEF), Host scanners (e.g. OpenIOC, STIX, CSV, yara), analysis tools (e.g. Maltego), DNS policies (e.g. RPZ)• Feed MISP using automatic tools (e.g. Sandbox Analysis, low-value information needing correlation, Analyst workbench)• Pull events from feeds or indicator lists to perform lookups against SIEMs• Subscribe to ZMQ pub-sub to get published events for use in lookup processes• Match attributes against SIEMs using the lookup expansion module• Import activities from a SIEM (e.g. Splunk lookup validation or false-positive feedback), NIDS or honeypot devices• Post Sightings from IDSes, IPSes, SIEMs back to MISP• Use sightings to improve NIDS' rule-sets• Generate IDS and NIDS rules automatically or manually using IoCs• Feed data to honeypots to generate blocklists and DNS RPZ zones• Consume correlated results in SIEMs using the API• Search indicators in real-time into a SIEM using MISP ZMQ• Submit large sets of IoCs from MISP into SIEMs using PyMISP• Import indicators

User story	Example workflow
	into MISP from other tools (SIEMs, IDSes) and be notified when those indicators appear again
As a CSIRT, we want to exchange and discuss information related to incidents and associated risks so that we can collaboratively respond to incidents	• Build communities to exchange specific data structures• Discuss non-event related topics in Forums• Add comments to events (which may represent an incident)• Contact a reporter (e.g. another CSIRT) via email (encrypted, anonymously or not) to discuss commercially-sensitive information related to an incident
As a CSIRT, we want to interact with threat data in various ways during the threat investigation and incident response process	• View events, indicators and feeds• Search and filter the data set• Classify, contextualize and correlate data• Download the viewed data in various formats• Interact with MISP data using other tools in the MISP ecosystem (e.g. PyMISP, the built-in REST client, and MISP modules)
As a CSIRT, we want to coordinate with team members and other organisations so that we can avoid duplication of work	• Create and manage sharing groups between sectors• Join existing communities or sharing groups• Create and exchange events and indicators• Propose changes to existing analysis or reports• Enhance an analysis with additional information using Extended Events• Report sightings as false-positive or true-positive (e.g. a partner/ analyst has seen a similar indicator)• Contribute to threat intel feeds and analyse overlapping data
As a CSIRT, we want to share incident information and discuss risks with other team members so that we can collaboratively perform incident analysis	• Create, modify, delete and exchange events and indicators• Modify distribution settings to exchange individual incidents and ensure confidentiality• Use taxonomies and galaxies to classify data before exchange (e.g Indicate the confidentiality of incidents using the NATO classification, indicate the risk of an incident using the threat-level taxonomy)• Edit, visualize, and share reports using Event Report• Incorporate reports from information sources using the Event Report module• Share indicators derived during incident response• Correlate and enrich data derived during incidents• Coordinate with affected parties during incident response using MISP's collaborative tools (proposals, sightings, emails)
As a fraud analyst, I want to investigate financial threats so that I can help financial institutions and consumers prevent financial fraud	• Join communities and receive shared IOCs• Subscribe to feeds and get IOCs in an easily accessible format• Access lists and public feeds of malicious domains (e.g phishing sites) and threats• Use indicators to check logs and verify if you're affected by a threat• Gather information related to a phishing site and create events• Integrate MISP with Maltego to visualise the full ATT&CK framework
As a fraud analyst, I want to blend updated threat intel with anti-fraud tools so that I can prevent fraud in real-time	• Feed data from MISP to fraud prevention tools• Report sightings to MISP from fraud prevention tools

User story	Example workflow
As a fraud analyst, I want to collaborate with analysts from other institutions so that we can gain shared situational awareness	• Implement a MISP instance, and join relevant communities• Publish fraud perpetrators for others to see• Exchange events containing fraud information (e.g a bank account number)• Use shared fraud data to feed firewalls and blocklists• Warn of false positives by alerting for invalid financial indicators• Give more credibility to indicators by reacting to event attributes (Sightings)• Get feedback from the community on the quality of indicators (Sightings)
As a customs and border control agent, I want to facilitate the flow of legal immigration and goods while preventing the illegal trafficking of people and contraband so that I can ensure homeland security	• Create or join sharing groups and communities• Share information (e.g travel documents / biometric information) between border control agencies using MISP• Categorize data using predefined types such PNR (passenger name records)• Share information / involve experts for the identification of smuggled goods• Perform anonymised lookups against exported data sets information (e.g. offline border control check)
As a law enforcement officer, I want to investigate digital crimes and threats so that I can apprehend criminals	• Access information sharing communities• Get indicators and actionable information from CSIRTs/CERTs networks or researchers• Exchange information with other officers via sharing communities• Exchange and store incident information on MISP, enabling the system to act as a forensic tool over time
As a law enforcement officer, I want to collect and verify evidence of digital crimes so that I can bootstrap my DFIR cases	• Collect indicators from shared events• Propose changes to existing analysis or reports• Enhance existing events with additional pieces of evidence using Extended Events• Exchange analysis and reports of digital forensic evidence• Correlate indicators corresponding to forensic pieces of evidence• Import Mactime timelines to describe forensic activities on an analysed file system• Describe forensic analysis cases using objects templates• Create, modify and visualise the timeline of events• Share analysis and reports of digital forensic evidence• Report sightings such as false-positive or true-positive (e.g. a partner/analyst has seen a similar indicator)
As a cybersecurity consultant, I want to provide structured threat intelligence to cross-sector partners with diverse requirements so that I can secure their infrastructure	• Implement an instance and join relevant communities• Integrate MISP with an organisation's existing solutions using the API• Exchange events containing indicators• Setup distribution levels to ensure confidentiality during threat sharing• Sync between untrusted and trusted networks using Feed support• Notify the community about activities related to an indicator using Sightings• Score indicators based on user sightings, including negative sightings and expiration sightings• Propose updates to an event owner or indicate a sighting• Share attacker techniques via integration with ATT&CK• Set an attribute for detection tools using the IDS flag

User story	Example workflow
As a cybersecurity specialist, I want to anonymously publish threat intel so that I can protect the identity of people who don't want to be associated with the information	• Pseudo-anonymously publish data using Event Delegation
As a cybersecurity specialist, I want to investigate threats so that I can remediate and prevent cyber attacks	• Query an instance for events that include a given IOC• Explore more details from Galaxies and related events• Categorize related information within the MITRE ATT&CK framework
As a security analyst, I want to access threat data so that I can use it to support my research	• Contextualise indicators (attributes) using categories, taxonomies and galaxies• Reinforce an analysis using correlation features (e.g. do other analysts have the same hypothesis?)• Confirm a specific aspect using correlation features (e.g. are the sinkhole IP addresses used for one campaign?)• Verify if a threat is new or unknown in your community using correlation features
As a security analyst, I want to access updated threat data so that I can build protection in real time	• Monitor feeds for recent indicators• Monitor activity in real-time on MISP dashboard by subscribing to ZMQ feeds• Process information in real-time when it's updated, created or gathered using ZMQ
As a risk analyst, I want to identify and predict risks to my organization so that I can improve the organization's security posture and situational awareness	• Use a MISP instance as a database of events representing threats• Classify risks using taxonomies and galaxies• Generate statistics from your MISP instance to deduce from incidents the current operational status, risk posture, and threats to the cyber environment• Monitor trends and adversary TTPs using MISP-dashboard and built-in statistics
As a risk analyst, I want to present risk data to stakeholders in various formats (depending on their technical ability), so that I can justify the need for risk-mitigating strategies	• Show trends within the sector/geographical region using MISP dashboard and built-in statistics• Turn MISP data into explorable graphs or timelines representing their activity or events• Export data from MISP in various formats• Share reports along with actionable data using Events Report
As a disinformation researcher, I want to identify indicators associated with a specific operation or campaign so that I can help track and mitigate threats	• Monitor MISP feeds for indicators• Find relationships between indicators using correlation
As a disinformation researcher and journalist, I want to investigate information campaigns so that I can report whether there is or isn't disinformation or misinformation	• Compare external feeds information with already-available information• Analyze the connections between incident objects• Map data with DISARM (embedded in MISP) to understand threat actor capabilities• Generate events that can be shared directly, via email or MISP• Add object types (e.g for common social media platforms), relationship types (to make the graphs that users can traverse in MISP richer) and taxonomies (e.g DFRLab's Dichotomies of Disinformation, and a NATO-led tactical variant) to describe indicators and events• Generate and share information operations data in MISP JSON or STIX format for

User story	Example workflow
	easy sharing• Classify events with DISARM techniques using the DISARM matrix• Describe attack patterns using DISARM• Track disinformation techniques using the DISARM galaxy• Integrate MISP with TheHive for case tracking• Describe additional disinformation cases using object templates
As a disinformation researcher, I want to connect with other researchers and responders so that we can collaboratively verify if an article/video/image contains disinformation and verify that a source (publisher, domain, etc) doesn't distribute disinformation	• Join a disinformation community• Notify the community about activities related to an indicator• Score indicators based on users sighting• Corroborate a finding using correlation features (e.g. is this the same campaign?)
As a disinformation researcher, I want to collaborate with other researchers and responders so that we can collectively stop disinformation campaigns	• Browse and Join disinformation communities (e.g CogSec Collab MISP)• Contextualise data using tags, taxonomies and galaxies• Describe information campaigns indicators and events using taxonomies (e.g DFRLab Dichotomies of Disinformation)• Find relationships between indicators using correlation• Describe misinformation tactics/techniques using the DISARM framework (galaxy, formerly AMITT)• Include relevant techniques found in a report or sighting in misinformation event data using the DISARM matrix
As a data scientist, I want to automate tasks related to data collection, curation, analysis, and visualization so that I can reduce security analysts' workloads	• Collect, add, update, search events/attributes/tags using PyMISP• Study malware samples using PyMISP• Write scripts to import (from other tools such as VirusTotal) additional attributes or IOC data (such as hashes) to build up knowledge on an event• Automatically handle indicators in third-party tools using PyMISP• Integrate MISP with existing infrastructure using PyMISP• Automate the dissemination of threat intelligence and threat data using the API• Generate exports to be ingested into other platforms• Create a range of filtered subsets of the dataset for various protective measures• Write scripts to disable the IDS flag based on the number of false-positive reported sightings, in order to prevent using false-positive indicators for detection or correlation actions• Generate data statistics and send reports via email, attached as CSV files using the API• Feed processed data into IDSes and 3rd party visualization using PyMISP• Build custom widgets to visualise/track data via the Dashboard• Extend MISP with Python scripts using MISP modules• Auto-discover new modules with their features using the API
As a data scientist, I want to collect and analyze data from various sources so that I can prioritize and predict risk	• Aggregate indicators and sightings of all attributes/objects, useful for detecting particular security events or threats• Use PyMISP for Scripted processing of events and attributes• Collect data from open data portals using the API• Publish open data and create data sets• Investigate file

User story	Example workflow
	hashes, malicious website URLs, IP Addresses and domain names using shared indicators• Aggregate data sets for security research and threat analysis• Analyse and select threat feeds for incorporation into other tools to hunt known indicators• Indicate if an attribute should be used for detection or correlation actions using the IDS flag• Download data in various formats for ingestion in other tools, and for training ML models
As a MISP site administrator, I want to automate repetitive instance housekeeping and enforce policy so that data quality stays high without manual effort	• Build Workflows that react to instance events (publish, attribute add, before save) by running a chain of logic and action blocks• Use blocking Workflows to stop a publish that is missing a required TLP/PAP tag, and non-blocking Workflows to notify, enrich or tag automatically• Share and reuse automation across instances using Workflow Blueprints• Tune indicator lifetime centrally with Decaying Models and keep noise down with warninglists and noticelists
As a MISP site administrator, I want to run a trustworthy, well-connected instance so that partners can rely on the data we publish	• Sign published events with cryptographic keys so downstream instances can verify their integrity and origin (Protected Events)• Manage synchronisation with other MISP servers and expose or consume feeds and TAXII 2.x servers• Broker organisation, sharing-group and sync trust at scale with Cerebrate• Keep the platform healthy by watching background workers and jobs, scheduled tasks, diagnostics and the audit log
As an organisation administrator, I want to run our day-to-day sharing operations so that our community always has current, well-scoped intelligence	• Manage the organisation's users, roles and API keys, and curate the feeds we ingest• Define reusable distribution scopes with Sharing Group Blueprints instead of rebuilding sharing groups by hand• Control what leaves the organisation with distribution levels, publish/alert filters and delegation• Monitor sighting and correlation pipelines and keep dashboards current for the team
As a threat analyst, I want to record my assessment of shared data without altering it so that my judgement travels with the intelligence	• Attach Analyst Notes (free-text, markdown) to any event, attribute, object or galaxy cluster• Express confidence with Opinions (0–100) and connect related elements — even across events and organisations — using typed Relationships• Keep this analyst layer distinct from the underlying data (it has its own ownership and distribution, and syncs across the federation)
As a threat analyst, I want to package related intelligence together so that a whole investigation or campaign can be shared and synced as one unit	• Group related Events and Galaxy Clusters into a Collection• Share and synchronise the collection as a single object so recipients get the full picture, not scattered fragments• Curate and version the collection as the investigation evolves

54. User Personas

These personas are fictional, but they are based on concrete representations of the people who use MISP. They help us keep in mind who we are building for, what their needs are, and what MISP should help them accomplish. These personas are based on OSINT about current MISP users, including Gitter chats, GitHub issues, LinkedIn, and other cybersecurity sources.

54.1 Primary personas

Farrah and Adam represent the users that are the most important to us.

54.1.1 Farrah

The Threat Hunter

Farrah works in threat intelligence for a security service provider in Malaysia that offers a range of cybersecurity solutions. He leads a threat intelligence team made up of experienced intelligence analysts who are former military/government employees and contractors.

Farrah uses MISP to analyze malware, gather information about specific adversary groups, and discover emerging threats. He also uses MISP for data normalization (consolidating data across different source formats), de-duplication (removal of duplicate information), and enrichment (removal of false positives, scoring of indicators, and the addition of context).

"To address threats effectively, you must maintain a team focused on monitoring, generating, and triaging alerts."

54.1.1.1 Role

Lead Threat Intelligence Analyst

54.1.1.2 His primary goals are to:

- Hunt down threats, analyze malware, manage vulnerabilities and prevent attacks against ICT infrastructures, organizations, or people.
- Improve security posture through the aggregation, correlation, and analysis of threat data from multiple sources
- Investigate and understand adversarial capabilities, infrastructure & TTPs.
- Turn threat data from various sources into actionable threat intelligence.

54.1.1.3 He uses MISP to:

- Dispatch notifications containing IoCs to various parties via mail_to_misp.
- Monitor feeds for indicators and correlate attributes and analyze malware (check ransom notes, look for any indicators, check the origin, etc).
- Store attack info in a structured format and allow for automated use of the database for various purposes using the API.
- Prioritize indicators using sighting reports and purge false positives using warning lists.
- Classify and contextualize data using taxonomies and galaxies, and keep track of the advancement of an analysis using tags.
- View and visualize events and activities using MISP-dashboard or Maltego.
- Automatically import, aggregate, compare, contextualize, query, and cross-reference data using PyMISP
- Import, export and enrich data using MISP modules
- Aggregate, curate, and validate indicators from various feeds, then feed the data into detection and analysis tools like NIDs, IDS, and SIEMs
- Query vulnerability scan results in MISP, automatically create/classify events on matching results, then create blocklists by excluding attributes that exist on warning lists.

- Collaborate with others in a sharing community using Proposals (send and receive feedback), Extended Events (add additional information to other's analytics), and Event Reports (supply resolution steps and recommendations).
- Share and receive reports of a specific threat, false positives, or post-mortem analysis of an incident from sharing groups.
- Automate repetitive enrichment, tagging and notification with Workflows, and reuse that automation across instances via Blueprints.

54.1.1.4 His objectives are to:

- Join relevant sharing communities, produce and publish indicators and share information across sectors to avoid hybrid threats.
- Use IoCs from feeds to identify vulnerabilities, compromised assets, data leaks and to verify the results of a malware scan.
- Triage threat intel, prioritize vulnerabilities, and customize risk feeds to ignore or downgrade irrelevant alerts in order to avoid alert fatigue.
- Generate and share alerts to provide critical information to internal teams and external peers.
- Share information about relevant vulnerabilities, coordinate with security vendors to get notifications, and community sharing of pentest results with other analysts.
- Research the evolution of high-risk malware families, validate malware signatures and domain reputations.
- Use indicators to query security logs/systems and databases, identify compromised systems, and add/modify signatures (used by firewalls, intrusion detection systems, etc), and block or alert on activity matching the indicators.
- Correlate shared indicators from feeds with those captured by other security and network tools to produce intelligence placed in the context of wider threat landscapes.
- Integrate with existing security solutions so as to centralize security in one place. Leverage integrations to alert on information leaks (AIL 2), hunt down threats (OpenDXL), share attacker's techniques (ATT&CK), query and prioritize indicators (Trellix EDR), speed up investigations (Cytomic Orion), and enhance the power of threat data (Carbon Black).

54.1.2 Adam

The Remediator

Adam is part of the Computer Security Incident Response Team (CSIRT) at a Belgian cybersecurity consulting firm. His responsibilities involve incident response, incident coordination, threat intelligence, and vulnerability management. He monitors potential threats, investigates attacks, and works with other security personnel to reduce the impact and severity of an attack.

Adam uses MISP to monitor incidents, provide early warnings/alerts about incidents, respond to incidents and provide incident analysis and situational awareness.

"A breach alone is not a disaster, but mishandling it is. The goal is to handle the situation in a way that limits damage and reduces recovery and time costs"

54.1.2.1 Role

Incident Response

54.1.2.2 His primary goals are to:

- Uncover the effects of attacks, determine how to clean up its impact, and inform a response to an existing incident to mitigate its extent or impacts.
- Develop and maintain strong processes for the most common incidents and threats, and create actionable results and remediation plans for internal stakeholders to proactively improve the security posture and maturity.
- Accelerate incident investigations, management, and prioritization by looking for information on the who/what/why/when/how of an incident.
- Determine the scope of incidents and limit the potential damage.

54.1.2.3 He uses MISP to:

- Store incidents as a database of events, describe incidents through event classification (using taxonomies and galaxies) and use the API to deduce from all incidents the current operational status, risk posture, and threats to the cyber environment.
- Join sharing groups and communities to share incident information with others and discuss information related to risks associated with incidents via Forums, comments to events, and contact a reporter.
- Analyze observables/malware collected during an incident, determining whether they are IoCs or false positives using the correlation graph and expansion modules.
- Alert and send emails when events are created in the system or major changes occur in the events, serving as part of an early warning system.
- Pull events via the API or export IOCs in formats for easy ingestion into other tools (such as SIEMs and IDS) and carrying out investigations by launching lookups against databases.
- Collaborate and get feedback from team members and affected parties during incident response using Proposals.
- Dismiss false positives (using warning lists) and enable alert prioritization.
- Aggregate and compare information from internal and external feeds to identify genuine threats.
- Perform large-scale data/traffic analysis and correlation through lookups against SightingDB.
- Share, receive, store and forward incidents and information identified during an incident investigation, enabling the MISP system to act as a forensic tool over time.
- Correlate and reference network forensic flows from different tools or network equipment
- Speed up incident response via integration with TheHive.
- Capture the team's assessment alongside shared data using Analyst Data — Notes, Opinions (confidence) and typed Relationships — and bundle an incident's events and clusters into a Collection for handover.

54.1.2.4 His objectives are to:

- Share information and get critical alerts and relevant actionable information in the event of a crisis situation.
- Support forensic analysts and collaborate with law enforcement.
- Improve incident response functionality and increase coverage and detection through integrations with tools like SIEMs.
- Use threat data to validate alarms/events and decide which to escalate to the rest of the incident response team for remediation.
- Aggregate information from various sources and correlating in order to understand how this data fits together in the broad threat landscape.
- Get insights (e.g using data feeds) into attacks, thereby helping incident response teams understand the nature, intent, and timing of specific attacks.
- Prioritize incidents based on risk and impact to the organization and filter out false alerts.

54.2 Secondary personas

Tina, Henry, Jacob, Jay, Sarah, Priya, and Marco represent other users that are also important to us.

54.2.1 Tina

The Fraud Catcher

Tina works as a fraud analyst at a National bank in Canada. She is responsible for investigating any forgery or theft within customers' accounts and transactions on behalf of the bank.

Tina uses MISP to find and share financial indicators in order to detect financial frauds.

"Fighting fraud with threat intelligence is all about alerting"

54.2.1.1 Role

Fraud analyst

54.2.1.2 Her primary goals are to:

- Identify and trace fraudulent activity.

- Create models for analyzing and determining financial fraud in order to protect consumers and stakeholders.
- Assess and analyze the attack surface, conduct threat modeling, and deliver actionable intelligence with a focus on current and emerging cyber-attacks against financial sector assets.

54.2.1.3 She uses MISP to:

- Map legacy and internal systems/models using MISP objects.
- Find IoCs, malware, vulnerabilities, financial threat, fraud information and share data between other banks and financial institutions using sharing groups.
- Create, modify and visualize the timeline of events, use MISP Dashboard to provide real-time information showing current threats and activity.
- Minimize false positives during the fraud vetting process using warning lists and sightings.
- Detect fraud using threat intel such as real-time notifications for stolen credit cards and phishing URLs from MISP feeds.
- Prevent fraud by integrating MISP with a network of crawlers, honeypots, and other techniques that can cross-reference indicators against feeds, enabling the bank to intercept cards before they are sold on the black market and therefore reduce this risk of fraud.
- Monitor feeds for specific indicators (e.g. email header content, attachments, embedded URLs) related to phishing and fraud attacks.
- Block wire transfers to money mule accounts by integrating MISP warning lists and sightings with blocking systems.
- Aggregate sightings of attributes/objects so as to detect particular security events or threats.

54.2.1.4 Her objectives are to:

- Investigate financial indicators and handle false positives in order to detect and alert for certain potentially invalid data points.
- Aggregate, correlate, and analyze financial indicators from multiple feeds to discover any fraudulent activity.
- Blend threat intel from MISP with anti-fraud tools to identify and prevent fraud in real-time.
- Use threat intel to produce awareness reports informing the institution of threats in the financial sector, and then develop proactive defense strategies against fraud activity.
- Engage with sharing communities that allow individual enterprises to receive and share data so they can protect themselves before they are compromised.

54.2.2 Henry

The Enforcer

Henry is a law enforcement officer living in Florida, USA. He works with the Digital Forensics and Incident Response (DFIR) team. He is responsible for investigating digital security incidents, identifying digital assets targeted during attacks, and documenting all findings.

He uses MISP to support or bootstrap his DFIR cases.

"I worry about what I don't know, not what I know"

54.2.2.1 Role

Law Enforcement Officer

54.2.2.2 His primary goals are to:

- Find, gather and analyze digital evidence for criminal investigations.
- Carry out data breach and malware investigations.

54.2.2.3 He uses MISP to:

- Propose changes to existing analyses or reports.
- Correlate (1-to-1 value matches, fuzzy hashing, CIDR block matching) evidence against external/local attributes.
- Correlate and reference network forensic flows from different tools or network equipment using the community-id feature.
- Export data in various formats to feed into and lookup in other security tools.

- Join sharing groups and collaborate with other investigators.
- Receive, gather, analyze and share intelligence on digital crimes.
- Report digital evidence (in STIX) in a structured way for forensic use.
- Collect evidence for forensic analysis from feeds, using shared indicators to carry out cybercriminal behavior investigations, attribution, and identifying the link to organized crime activities.
- Exchange, store and forward incidents/information identified during an incident investigation, enabling the MISP system to act as a forensic tool over time.

54.2.2.4 His objectives are to:

- Share indicators, analyses, and reports of forensic evidence among other law enforcement officers within and out of his team.
- Collaborate with CSIRT/CERTs and security researchers in the investigation of cyberattacks.
- Correlate data identified in a recent incident with data from previous investigations or external feeds.
- Bridge their use-cases with MISP's information-sharing mechanism.

54.2.3 Jacob

The Veteran

Jacob is a cybersecurity consultant for organizations looking to secure their infrastructure. He has founded a cybersecurity agency that provides threat intel and security consulting services to small and medium-sized businesses.

Jacob uses MISP to investigate threats and find IOCs. He works with many clients and typically wants to integrate MISP into existing client solutions.

"There's a difference between threat data and threat intelligence"

54.2.3.1 Role

Cyber Security consultant

54.2.3.2 His primary goals are to:

- Produce intelligence that will be embedded into organizational workflows and would serve decision-makers.
- Scope and implement custom security solutions across a variety of client software, architectures, and tools.
- Detect, contain, and remediate cybersecurity incidents, manually or programmatically.

54.2.3.3 He uses MISP to:

- Create, collaborate, automate and share threat intel using flexible sharing groups, automatic correlation, free-text import helper, event distribution, and proposals.
- Allow users to notify a MISP instance about activities (gotten from SIEMs, NIDS, honeypot devices, etc) related to an indicator using sightings.
- Monitor feeds delivered through a REST API and correlate IOCs with firewall and other logs to identify potential threats in the organization.
- Push/pull events between local and client MISP instances in order to exchange intel internally and externally.
- Import, export, and enrich data using modules, automate such tasks using PyMISP.
- Create sub-communities and MISP object templates to allow rapid sharing of information using specific data models with existing communities.
- Validate data and flag false positives using warning lists and sightings.
- View live data/statistics and process information in real-time through integration with ZMQ to access MISP-dashboard.
- Pseudo-anonymously publish data using the MISP delegation system.
- Contextualize shared information within MISP instances and communities (using taxonomies and tags), and attach more complex structures to data (using MITRE ATT&CK and other galaxies)
- Broker trust with client and partner instances using Cerebrate and reusable Sharing Group Blueprints, and sign the events he publishes with cryptographic keys so clients can verify their authenticity (Protected Events).

54.2.3.4 His objectives are to:

- Run a Cyber Threat Intel platform using MISP integrated with existing client solutions (such as Active Directory, Splunk ES, ThreatConnect, Recorded Future, and Crowdstrike).
- Gather unstructured data from various sources and connect the dots to provide context on IoCs and TTPs of threat actors.
- Identify incoming threats, triage and prioritize alerts as they emerge.
- Feed SIEMs from MISP and feed MISP from other sources (SIEMs included).
- Share incidents and IOCs for detection (checking if IoCs are present in client infrastructure), blocking (using attributes to block, sinkhole, or divert the traffic), and intelligence (gathering information about campaigns and attacks) purposes.

54.2.4 Jay

The Inquisitor

Jay is a risk analyst for a large technology company in the USA. He is responsible for identifying and predicting risks, as well as forecasting the cost of certain attacks to the organization.

Jay uses MISP data to learn about the broad threat landscape and analyze the likelihood of certain risks, so as to gain situational awareness.

"The more certain you can be about the probability of a specific exploit impacting your environment, the easier it is to manage risk"

54.2.4.1 Role

Risk analyst

54.2.4.2 His primary goals are to:

- Improve the organization's security posture, situational awareness, and resilience.
- Forecast evolving threats before they materialize, provide detailed insights into which vulnerabilities pose the greatest risk, and plan accordingly to avoid them.
- Assess business and technical risks, identify the right strategies and technologies to mitigate the risks, communicate the nature of the risks to top management and justify investments in defensive measures.

54.2.4.3 He uses MISP to:

- Monitor trends and adversary TTPs within the company's sector/geographical region, share and track information emerging on a particular topic from the MISP dashboard in order to gain situational awareness.
- Monitor IoCs from various technical feeds and add additional context to internal sources of data using the automatic correlation engine.
- Access risk scores using correlation and sightings.
- Present data using different export formats, event reports, and the MISP dashboard timeline.

54.2.4.4 His objectives are to:

- Use shared indicators to perform a risk assessment, identify key information/assets and illustrate the intent/capability of actors to target these assets through impact assessments.
- Score threats according to the organization's specific needs and prepare processes in advance based on threat data gathered from feeds
- Present data to stakeholders in various formats – articles, timelines, graphs, raw data – depending on their technical knowledge.
- Gain shared situational awareness through information sharing and collaboration with other experts in the same sector.

54.2.5 Sarah

The Fact Checker

Sarah is a disinformation researcher and journalist working for a large American newspaper. She works with security researchers around the world to investigate cybercrimes and report disinformation. In the past, she has written about national security and geopolitics. She is used to making decisions on what should or shouldn't be published or shared.

Sarah uses MISP to collaborate with security researchers and investigate disinformation as it happens.
"Decisions as to what is or isn't published or shared go far beyond what is technically interesting"

54.2.5.1 Role

Disinformation researcher and journalist

54.2.5.2 Her primary goals are to:

- Conduct research and write intelligence reports about up-and-coming emerging threats and recent breaches.
- Investigate and report disinformation as it happens.
- Convert technical data into articles and reports that non-technical people can understand.

54.2.5.3 She uses MISP to:

- Write/read event reports, create misinformation events using relevant techniques found in a report or sighting.
- Join sharing groups and communities (e.g Cogsec Collab) that connect misinformation researchers and responders, share incident data with organizations focusing on response and counter-campaigns.
- Integrate with the DISARM Framework (as a galaxy, formerly AM!TT) in order to describe misinformation tactics/techniques, break an incident into techniques that can be analyzed/countered, and check for disinformation through mapping.
- Monitor feeds, investigate disinformation using shared indicators in feeds, generate structured intelligence using the automated correlation engine, and decide if there are any falsehoods in data.
- Enrich threat data by adding object types, new relationship types (to make the graphs that users can traverse in MISP richer), and taxonomies to cover things like types of threat actors.
- Classify events, indicators, and threats using taxonomies (such as the Admiralty Scale taxonomy), which ranks the reliability of a source and the credibility of the information.

54.2.5.4 Her objectives are to:

- Distill essential information from a large piece of data, making it clear to the reader what really matters.
- Integrate MISP with TheHive for enhanced disinformation investigation and reporting.
- Verify that an article (or image, video, etc) doesn't contain disinformation and verify that a source (publisher, domain, etc) doesn't distribute disinformation.
- Extend MISP for disinformation, adding object types for incidents and narratives, and using DISARM (formerly AMITT) for attack patterns.

54.2.6 Priya

The Platform Administrator

Priya is a MISP site administrator at a national CERT. She keeps the instance that her analysts and partner organisations depend on healthy, current, and trustworthy — provisioning access, wiring up sharing, and automating the tedious parts so the team can focus on the intelligence itself. She rarely creates events herself; her product is a reliable platform.

"My users should never have to think about the plumbing — it should just work, and it should be trustworthy by default."

54.2.6.1 Role

MISP Site Administrator

54.2.6.2 Her primary goals are to:

- Run a secure, available, and up-to-date instance that analysts and partners can rely on.
- Make the right data reach the right people automatically, while keeping sensitive data contained.
- Reduce manual, error-prone housekeeping so the analysts' time goes into analysis.

54.2.6.3 She uses MISP to:

- Manage users, organisations, roles and API keys, and audit activity through the logs and user login profiles.

- Automate housekeeping and policy enforcement with Workflows (blocking checks such as "refuse to publish without a TLP tag", plus non-blocking enrichment, tagging and notifications), and share that automation as Workflow Blueprints.
- Connect the instance to the wider community: synchronisation servers, feeds, TAXII 2.x servers, and organisation/sharing-group trust brokered through Cerebrate.
- Guarantee the integrity and origin of what the instance publishes by signing events with cryptographic keys (Protected Events).
- Tune data quality centrally with Decaying Models, warninglists and noticelists.
- Keep the platform healthy by monitoring background workers and jobs, scheduled tasks, diagnostics, and the update process.

54.2.6.4 Her objectives are to:

- Onboard new users and partner organisations quickly and safely.
- Keep the instance patched and its reference data (galaxies, taxonomies, warninglists, noticelists) current.
- Encode sharing and quality policy once, as reusable automation, rather than relying on every analyst to remember it.
- Give partners cryptographic assurance that intelligence published by the instance is authentic and unmodified.

54.2.7 Marco

The Operator

Marco is an organisation administrator at a sectoral ISAC. He does not run the MISP server itself, but he runs his organisation's day-to-day presence on it: which feeds come in, which partners they exchange with, and what the organisation publishes to the community. He is the person who keeps the intelligence flowing and correctly scoped every day.

"Good sharing is boring sharing — the right data, to the right people, on time, every day."

54.2.7.1 Role

Organisation Administrator / Operations

54.2.7.2 His primary goals are to:

- Keep a steady, current flow of relevant intelligence into and out of the organisation.
- Make sure everything the organisation shares is scoped to exactly the right audience.
- Keep his team's users productive and their data clean.

54.2.7.3 He uses MISP to:

- Manage the organisation's users, roles and API keys, and curate the feeds the organisation ingests and evaluates before import.
- Control distribution precisely using distribution levels, sharing groups and reusable Sharing Group Blueprints, and pseudo-anonymise sensitive publications through delegation.
- Bundle related intelligence into Collections so a whole investigation or campaign can be shared and synchronised as one unit.
- Keep detection value high by managing sightings, decaying models and warninglists, and by publishing and alerting on a controlled schedule.
- Watch the organisation's dashboards and background jobs to confirm feeds, correlations and sync are running as expected.

54.2.7.4 His objectives are to:

- Onboard his organisation's analysts and keep their access and API keys in order.
- Maintain trusted exchange relationships with partner organisations and communities.
- Ensure nothing over- or under-shares by encoding the organisation's distribution policy into sharing groups and blueprints.
- Provide clean, well-contextualised, and timely data to the community the organisation belongs to.

54.3 Other personas

Malcolm represents users that we care about but aren't so important to us.

54.3.1 Malcolm

The Data Expert

Malcolm is a data scientist for a telecom operator in the USA. He assists the Security Operations Center with tasks related to anomaly detection, exploratory data analysis, data visualization, modeling, and optimization of security solutions.

Malcolm uses data from MISP alongside natural language processing, predictive modeling, and other data science techniques to assess, prioritize, and even predict risk. He can process threat data to help with alert prioritization and data-driven decision making.

"It is a mistake to theorize before one has data. Insensibly, one begins to twist facts to suit theories, instead of theories to suit facts"

54.3.1.1 Role

Data Scientist

54.3.1.2 His primary goals are to:

- Develop tools to help businesses detect threats so they can develop solid plans of action and better protect themselves.
- Make predictions, perform data analysis, and detect patterns in data.
- Support the threat analysis team with the development of new and innovative ways of extracting insight from large sets of structured and unstructured data.
- Translate complex data into relevant insights and visualize information.

54.3.1.3 He uses MISP to:

- Collect IoCs and sift through data from feeds to derive useful insights and connect dots between actors from various sources.
- Join sharing groups to collaborate with threat analysts and reduce analysts' workload by taking on many tasks related to data collection and correlation.
- Automatically aggregate, parse, de-duplicate, and manage indicators using the API.
- Visualize events in real-time by setting up MISP-dashboard.
- Export large threat data sets that can be used to train ML models

54.3.1.4 His objectives are to:

- Combine data from MISP and other security sources to find patterns/relationships in data, develop models, assess, prioritize, and predict risk using data science and AI techniques.
- Produce informative visualizations and knowledge graphs based on large data sets.
- Eliminate manual tasks by writing scripts that automatically aggregate, parse, de-duplicate, and manage indicators in MISP.
- Classify risks in order to save analysts time sorting through false positives and deciding what to prioritize.

Appendices

A. Appendices

B. Appendix A: External Authentication

B.0.0.1 The external authentication mechanism described

The external authentication allows a user or an external tool to authenticate with MISP using an arbitrary value passed along in a custom header. This authentication method overrides the regular authentication mechanisms and is customisable by a site-admin.

It is possible to create a mixed mode MISP setup where certain users can go through the normal authentication mechanism and other users are required to use the external authentication method.

B.0.0.2 Setting up the external authentication mechanism

To change the authentication settings, navigate to Administration - Server settings - Plugin settings

The settings associated with the external authentication can be found by pressing the CustomAuth button as depicted below:

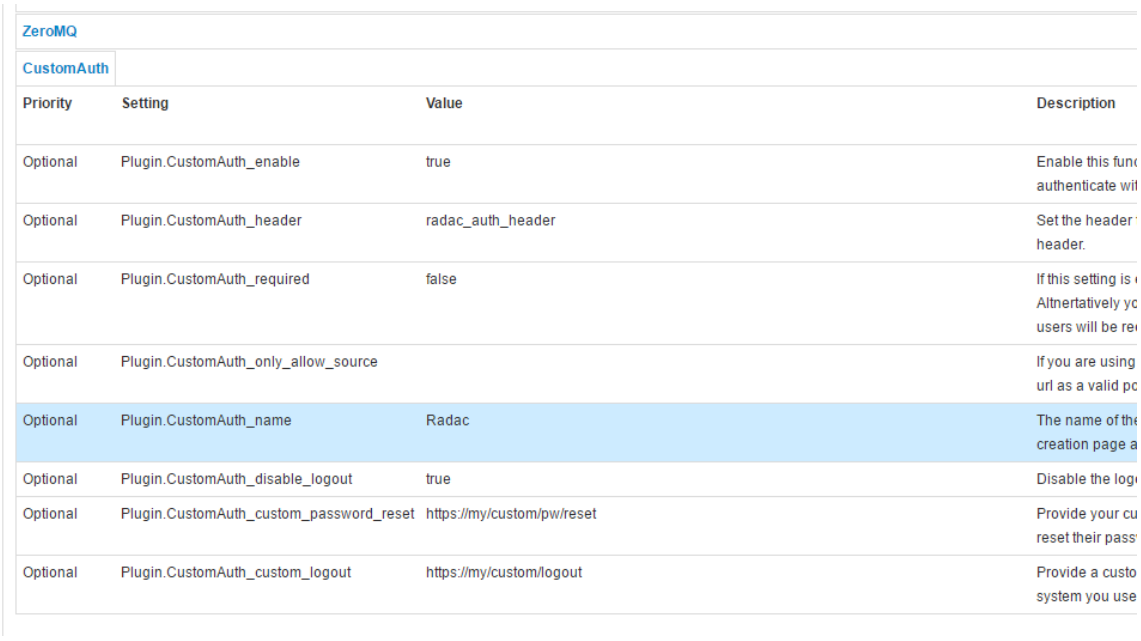


Figure B.1: The server settings relevant to the Custom Authentication system.

To change a setting simply double click on the value to edit the field. Use the guidance provided by the setting tool to configure the external authentication. The accessible settings are as follows:

- **enable**: Enable or disable external authentication (off by default)
- **header**: The header which MISP will use to identify users
- **required**: Enabling this setting will force all users to use the external authentication. Leave this disabled allows administrators to assign external authentication or regular authentication users.
- **only_allow_source**: Setting a url / IP address here will only allow requests that originated from the given address
- **name**: The name to be used for the authentication mechanism. This is reflected in the user creation / edit views, the logs and the error messages on failed logins.
- **disable_logout**: Disable the default logout button. Using an external authentication mechanism that authenticates via the header with each requests makes the logout button obsolete.

- **custom_password_reset**: If your authentication system has a url that a user can access to reset his/her password, please specify the full url for it here. This will then be reused in the UI.
- **custom_password_logout**: If your authentication system has a url that a user can access to logout, please specify the full url for it here. This will then be reused in the UI.

B.0.0.3 User management

Using a new setting, user self management can be disabled for all users that are not administrators via the MISP.disableUserSelfManagement setting, found in the MISP settings tab. Enabling this setting removes the ability of users to change their user settings and reset their authentication keys. All other functionality remains unchanged.

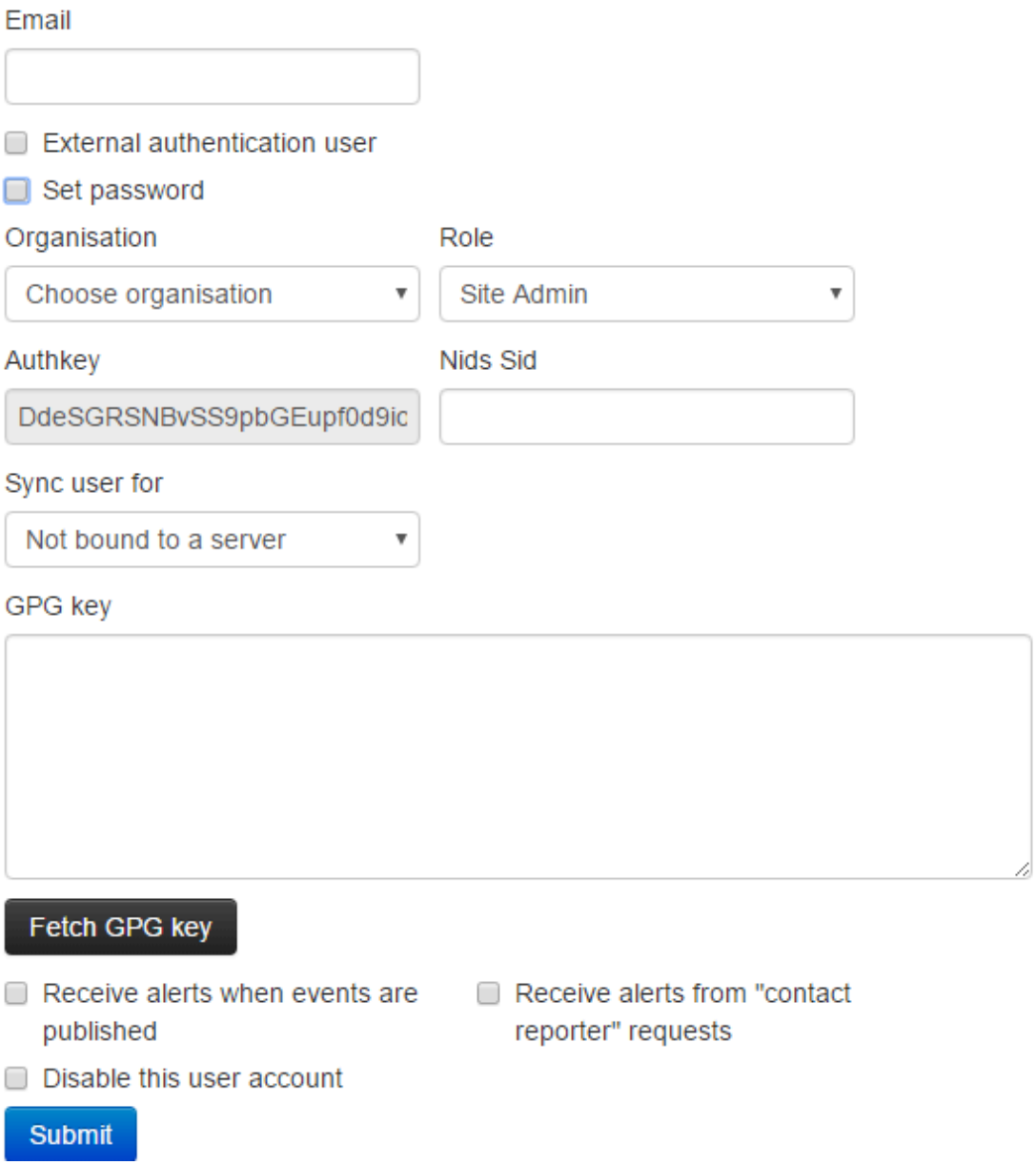


Figure B.2: Adding a user with the external authentication enabled.

To create an external authenticated user, simply tick the External authentication user checkbox, after which an external auth key field will appear. This will be used to identify the users via the passed along header.

B.0.0.4 Logging

For a description of the logging facilities provided by this plugin, please refer to the “Logging of failed authentication attempts” section of the Administration section.

C. Appendix B: ACL descriptors

C.0.0.1 Querying the ACL system

MISP allows site admins to query the ACL system for various types of data. This can be interesting when tuning for example WAF access to MISP. All applicable queries can be requested via /servers/queryACL

C.0.0.2 Getting a list of URLs accessible to a role

```
https://<misp url>/servers/queryACL/printRoleAccess/<role id>
```

The above URL will return a JSON with all accessible URLs for the given role ID. If no Role ID is provided, a JSON containing all roles and their access lists will be returned.

Example:


```
{
    "3": {
        "name": "User",
        "urls": [
            "/*/restSearch",
            "/attributes/add",
            "/attributes/add_attachment",
            "/attributes/add_threatconnect",
            "/attributes/addTag",
            "/attributes/attributeReplace",
            "/attributes/attributeStatistics",
            "/attributes/bro",
            "/attributes/delete",
            "/attributes/deleteSelected",
            "/attributes/describeTypes",
            "/attributes/download",
            "/attributes/downloadAttachment",
            "/attributes/downloadSample",
            "/attributes/edit",
            "/attributes/editField",
            "/attributes/editSelected",
            "/attributes/exportSearch",
            "/attributes/fetchEditForm",
            "/attributes/fetchViewValue",
            "/attributes/getMassEditForm",
            "/attributes/hoverEnrichment",
            "/attributes/index",
            "/attributes/removeTag",
            "/attributes/restore",
            "/attributes/restSearch",
            "/attributes/returnAttributes",
            "/attributes/rpz",
            "/attributes/search",
            "/attributes/searchAlternate",
            "/attributes/toggleCorrelation",
            "/attributes/text",
            "/attributes/toggleToIDS",
            "/attributes/updateAttributeValues",
            "/attributes/view",
            "/attributes/viewPicture",
            "/authKeys/add",
```

```
            "/authKeys/delete",
            "/authKeys/edit",
            "/authKeys/index",
            "/authKeys/view",
            "/auth_keys/add",
            "/auth_keys/delete",
            "/auth_keys/edit",
            "/auth_keys/index",
            "/auth_keys/view",
            "/dashboards/getForm",
            "/dashboards/index",
            "/dashboards/updateSettings",
            "/dashboards/getEmptyWidget",
            "/dashboards/renderWidget",
            "/dashboards/listTemplates",
            "/dashboards/saveTemplate",
            "/dashboards/export",
            "/dashboards/import",
            "/dashboards/deleteTemplate",
            "/decayingModel/export",
            "/decayingModel/import",
            "/decayingModel/view",
            "/decayingModel/index",
            "/decayingModel/add",
            "/decayingModel/edit",
            "/decayingModel/delete",
            "/decayingModel/enable",
            "/decayingModel/disable",
            "/decayingModel/decayingTool",
            "/decayingModel/getAllDecayingModels",
            "/decayingModel/decayingToolBasescore",
            "/decayingModel/decayingToolSimulation",
            "/decayingModel/decayingToolRestSearch",
            "/decayingModel/decayingToolComputeSimulation",
            "/decaying_model/export",
            "/decaying_model/import",
            "/decaying_model/view",
            "/decaying_model/index",
            "/decaying_model/add",
            "/decaying_model/edit",
            "/decaying_model/delete",
            "/decaying_model/enable",
            "/decaying_model/disable",
            "/decaying_model/decayingTool",
            "/decaying_model/getAllDecayingModels",
            "/decaying_model/decayingToolBasescore",
            "/decaying_model/decayingToolSimulation",
            "/decaying_model/decayingToolRestSearch",
            "/decaying_model/decayingToolComputeSimulation",
            "/decayingModelMapping/viewAssociatedTypes",
            "/decayingModelMapping/linkAttributeTypeToModel",
            "/decaying_model_mapping/viewAssociatedTypes",
            "/decaying_model_mapping/linkAttributeTypeToModel",
            "/eventBlocklists/add",
            "/eventBlocklists/delete",
            "/eventBlocklists/edit",
            "/eventBlocklists/index",
            "/eventBlocklists/massDelete",
            "/event_blocklists/add",
            "/event_blocklists/delete",
```

```
            "/event_blocklists/edit",
            "/event_blocklists/index",
            "/event_blocklists/massDelete",
            "/eventReports/add",
            "/eventReports/view",
            "/eventReports/viewSummary",
            "/eventReports/edit",
            "/eventReports/delete",
            "/eventReports/reportFromEvent",
            "/eventReports/restore",
            "/eventReports/index",
            "/eventReports/getProxyMISPElements",
            "/eventReports/extractAllFromReport",
            "/eventReports/extractFromReport",
            "/eventReports/replaceSuggestionInReport",
            "/eventReports/importReportFromUrl",
            "/event_reports/add",
            "/event_reports/view",
            "/event_reports/viewSummary",
            "/event_reports/edit",
            "/event_reports/delete",
            "/event_reports/reportFromEvent",
            "/event_reports/restore",
            "/event_reports/index",
            "/event_reports/getProxyMISPElements",
            "/event_reports/extractAllFromReport",
            "/event_reports/extractFromReport",
            "/event_reports/replaceSuggestionInReport",
            "/event_reports/importReportFromUrl",
            "/events/add",
            "/events/addIOC",
            "/events/addTag",
            "/events/add_misp_export",
            "/events/automation",
            "/events/checkLocks",
            "/events/checkPublishedStatus",
            "/events/contact",
            "/events/csv",
            "/events/delegation_index",
            "/events/delete",
            "/events/deleteNode",
            "/events/downloadExport",
            "/events/downloadOpenIOCEvent",
            "/events/edit",
            "/events/enrichEvent",
            "/events/export",
            "/events/exportChoice",
            "/events/exportModule",
            "/events/filterEventIndex",
            "/events/freeTextImport",
            "/events/getEditStrategy",
            "/events/getEventInfoById",
            "/events/getEventGraphReferences",
            "/events/getEventGraphTags",
            "/events/getEventGraphGeneric",
            "/events/getEventTimeline",
            "/events/genDistributionGraph",
            "/events/getDistributionGraph",
            "/events/getReferenceData",
            "/events/getReferences",
```

```
            "/events/getObjectTemplate",
            "/events/handleModuleResults",
            "/events/hids",
            "/events/index",
            "/events/importChoice",
            "/events/importModule",
            "/events/merge",
            "/events/nids",
            "/events/proposalEventIndex",
            "/events/publishSightings",
            "/events/queryEnrichment",
            "/events/removePivot",
            "/events/removeTag",
            "/events/restSearch",
            "/events/runTaxonomyExclusivityCheck",
            "/events/saveFreeText",
            "/events/stix",
            "/events/stix2",
            "/events/toggleCorrelation",
            "/events/unpublish",
            "/events/updateGraph",
            "/events/upload_analysis_file",
            "/events/upload_sample",
            "/events/upload_stix",
            "/events/view",
            "/events/viewClusterRelations",
            "/events/viewEventAttributes",
            "/events/viewGraph",
            "/events/viewGalaxyMatrix",
            "/events/xml",
            "/favouriteTags/toggle",
            "/favouriteTags/getToggleField",
            "/favourite_tags/toggle",
            "/favourite_tags/getToggleField",
            "/feeds/compareFeeds",
            "/feeds/feedCoverage",
            "/feeds/index",
            "/feeds/previewEvent",
            "/feeds/previewIndex",
            "/feeds/searchCaches",
            "/feeds/view",
            "/galaxies/attachCluster",
            "/galaxies/attachMultipleClusters",
            "/galaxies/export",
            "/galaxies/forkTree",
            "/galaxies/index",
            "/galaxies/relationsGraph",
            "/galaxies/selectGalaxy",
            "/galaxies/selectGalaxyNamespace",
            "/galaxies/selectCluster",
            "/galaxies/showGalaxies",
            "/galaxies/view",
            "/galaxies/viewGraph",
            "/galaxyClusters/attachToEvent",
            "/galaxyClusters/detach",
            "/galaxyClusters/index",
            "/galaxyClusters/restSearch",
            "/galaxyClusters/view",
            "/galaxyClusters/viewGalaxyMatrix",
            "/galaxyClusters/viewRelations",
```

```
            "/galaxyClusters/viewRelationTree",
            "/galaxy_clusters/attachToEvent",
            "/galaxy_clusters/detach",
            "/galaxy_clusters/index",
            "/galaxy_clusters/restSearch",
            "/galaxy_clusters/view",
            "/galaxy_clusters/viewGalaxyMatrix",
            "/galaxy_clusters/viewRelations",
            "/galaxy_clusters/viewRelationTree",
            "/galaxyClusterRelations/index",
            "/galaxyClusterRelations/view",
            "/galaxy_cluster_relations/index",
            "/galaxy_cluster_relations/view",
            "/galaxyElements/index",
            "/galaxy_elements/index",
            "/jobs/cache",
            "/jobs/getGenerateCorrelationProgress",
            "/jobs/getProgress",
            "/logs/event_index",
            "/logs/returnDates",
            "/modules/index",
            "/modules/queryEnrichment",
            "/news/index",
            "/noticelists/index",
            "/noticelists/view",
            "/objects/add",
            "/objects/addValueField",
            "/objects/delete",
            "/objects/edit",
            "/objects/get_row",
            "/objects/editField",
            "/objects/fetchEditForm",
            "/objects/fetchViewValue",
            "/objects/quickAddAttributeForm",
            "/objects/quickFetchTemplateWithValidObjectAttributes",
            "/objects/restSearch",
            "/objects/proposeObjectsFromAttributes",
            "/objects/groupAttributesIntoObject",
            "/objects/revise_object",
            "/objects/view",
            "/objectReferences/add",
            "/objectReferences/delete",
            "/objectReferences/view",
            "/object_references/add",
            "/object_references/delete",
            "/object_references/view",
            "/objectTemplates/objectChoice",
            "/objectTemplates/objectMetaChoice",
            "/objectTemplates/view",
            "/objectTemplates/viewElements",
            "/objectTemplates/index",
            "/object_templates/objectChoice",
            "/object_templates/objectMetaChoice",
            "/object_templates/view",
            "/object_templates/viewElements",
            "/object_templates/index",
            "/objectTemplateElements/viewElements",
            "/object_template_elements/viewElements",
            "/organisations/fetchSGOrgRow",
            "/organisations/index",
```

```
            "/organisations/view",
            "/pages/display",
            "/posts/add",
            "/posts/delete",
            "/posts/edit",
            "/regexp/index",
            "/restClientHistory/delete",
            "/restClientHistory/index",
            "/rest_client_history/delete",
            "/rest_client_history/index",
            "/roles/index",
            "/roles/view",
            "/servers/getApiInfo",
            "/servers/getPyMISPVersion",
            "/servers/getVersion",
            "/servers/idTranslator",
            "/servers/postTest",
            "/servers/rest",
            "/shadowAttributes/accept",
            "/shadowAttributes/acceptSelected",
            "/shadowAttributes/add",
            "/shadowAttributes/add_attachment",
            "/shadowAttributes/delete",
            "/shadowAttributes/discard",
            "/shadowAttributes/discardSelected",
            "/shadowAttributes/download",
            "/shadowAttributes/edit",
            "/shadowAttributes/index",
            "/shadowAttributes/view",
            "/shadowAttributes/viewPicture",
            "/shadow_attributes/accept",
            "/shadow_attributes/acceptSelected",
            "/shadow_attributes/add",
            "/shadow_attributes/add_attachment",
            "/shadow_attributes/delete",
            "/shadow_attributes/discard",
            "/shadow_attributes/discardSelected",
            "/shadow_attributes/download",
            "/shadow_attributes/edit",
            "/shadow_attributes/index",
            "/shadow_attributes/view",
            "/shadow_attributes/viewPicture",
            "/sharingGroups/index",
            "/sharingGroups/view",
            "/sharing_groups/index",
            "/sharing_groups/view",
            "/sightings/add",
            "/sightings/restSearch",
            "/sightings/advanced",
            "/sightings/delete",
            "/sightings/index",
            "/sightings/listSightings",
            "/sightings/quickDelete",
            "/sightings/viewSightings",
            "/sightings/bulkSaveSightings",
            "/sightings/quickAdd",
            "/tagCollections/index",
            "/tagCollections/view",
            "/tag_collections/index",
            "/tag_collections/view",
```

```
            "/tags/attachTagToObject",
            "/tags/index",
            "/tags/removeTagFromObject",
            "/tags/search",
            "/tags/selectTag",
            "/tags/selectTaxonomy",
            "/tags/showEventTag",
            "/tags/showAttributeTag",
            "/tags/showTagControllerTag",
            "/tags/tagStatistics",
            "/tags/view",
            "/tags/viewGraph",
            "/tags/viewTag",
            "/taxonomies/index",
            "/taxonomies/taxonomyMassConfirmation",
            "/taxonomies/taxonomyMassHide",
            "/taxonomies/taxonomyMassUnhide",
            "/taxonomies/view",
            "/taxonomies/unhideTag",
            "/taxonomies/hideTag",
            "/templateElements/index",
            "/template_elements/index",
            "/templates/deleteTemporaryFile",
            "/templates/index",
            "/templates/populateEventFromTemplate",
            "/templates/submitEventPopulation",
            "/templates/templateChoices",
            "/templates/uploadFile",
            "/templates/view",
            "/threads/index",
            "/threads/view",
            "/threads/viewEvent",
            "/users/attributehistogram",
            "/users/change_pw",
            "/users/checkIfLoggedIn",
            "/users/dashboard",
            "/users/downloadTerms",
            "/users/edit",
            "/users/email_otp",
            "/users/searchGpgKey",
            "/users/fetchGpgKey",
            "/users/histogram",
            "/users/login",
            "/users/logout",
            "/users/register",
            "/users/resetauthkey",
            "/users/request_API",
            "/users/routeafterlogin",
            "/users/statistics",
            "/users/tagStatisticsGraph",
            "/users/terms",
            "/users/updateLoginTime",
            "/users/view",
            "/users/getGpgPublicKey",
            "/userSettings/index",
            "/userSettings/view",
            "/userSettings/setSetting",
            "/userSettings/getSetting",
            "/userSettings/delete",
            "/userSettings/setHomePage",
```

```
            "/user_settings/index",
            "/user_settings/view",
            "/user_settings/setSetting",
            "/user_settings/getSetting",
            "/user_settings/delete",
            "/user_settings/setHomePage",
            "/warninglists/checkValue",
            "/warninglists/index",
            "/warninglists/view",
            "/allowedlists/index",
            "/eventGraph/view",
            "/eventGraph/add",
            "/eventGraph/delete",
            "/event_graph/view",
            "/event_graph/add",
            "/event_graph/delete"
        ]
    }
}
```

C.0.0.3 Getting a list of all accessible controllers and actions in MISP

```
https://<misp url>/servers/queryACL/printAllFunctionNames
```

This URL will return a JSON with all controller and all mapped functions within them.

C.0.0.4 Viewing a list of yet unmapped functions

```
https://<misp url>/servers/queryACL/findMissingFunctionNames
```

Functions that have not been tied into the new ACL yet show up here. These functions will (until added to the ACL) only be accessible to site admins.

D. Appendix C: Official MISP developments

This section lists the projects that can be found on the main MISP GitHub page e know of but not officially support and rely on their respective maintainers to keep up to date to the MISP 2.5 developments.

Project	Description	Status
misp-objects	Definition, description and relationship types of MISP objects	Core to MISP, frequently updated and tested
Best Practices in ThreatIntel	Best practices in threat intelligence	Book available here: https://www.misp-project.org/best-practices-in-threat-intelligence.html

E. Appendix D: Third-party development

This section lists some projects we know of but not officially support and rely on their respective maintainers to keep up to date to the MISP 2.5 developments.

Project	Description	Status
MISP-STIX-ESM	Exports MISP events to STIX and ingest into McAfee ESM	Not tested by MISP core team
Docker MISP	Automated Docker MISP container	Not tested by MISP core team
misp42splunk	A Splunk app to use MISP in background and combine with TheHive	Not tested by MISP core team
getmispioc	getiocmisp is a Splunk custom search command that helps to extract IOCs from a MISP instance.	Not tested by MISP core team
OTX MISP	Imports Alienvault OTX pulses to a MISP instance	Not tested by MISP core team
BTG	BTG's purpose is to make fast and efficient search on IOC	Not tested by MISP core team
MISP OSINT Collection	Collection of best practices to add OSINT into MISP and/or MISP communities	Not tested by MISP core team
IBM XFE module	Various IBM X-Force Exchange modules	Not tested by MISP core team
MISP dockerized	MISP dockerized is a project designed to provide an easy-to-use and easy-to-install'out of the box' MISP instance that includes everything you need to run MISP with minimal host-side requirements.	Not tested by MISP core team
MISP dockerized modules	MISP-modules for MISP dockerized	Not tested by MISP core team
FireMISP	FireEye Alert json files to MISP Malware information sharing plattform (Alpha)	Not tested by MISP core team
MISP Chrome Plugin	MISP Chrome plugin for adding and looking up indicators	Not tested by MISP core team
PySight2MISP	PySight2MISP is a project that can be run to be used as glue between iSight intel API and MISP API	Not tested by MISP core team

Project	Description	Status
tie2misp	Import DCSO TIE IOCs as MISP events	Not tested by MISP core team
security onion MISP	Grab NIDS rules and Bro Intel generated from a MISP instance and use them in Security Onion	Not tested by MISP core team
virustream	A script to track malware IOCs with OSINT on Twitter.	Not tested by MISP core team
LAC CSV Import	Register MISP events based on information described in files such as CSV and TSV.	Not tested by MISP core team
The Hive	TheHive: a Scalable, Open Source and Free Security Incident Response Platform	Strong links between core team members, tested and known working
puppet-misp	This module installs and configures MISP - puppet forge site	Not tested by MISP core team
Ansible MISP	Ansible playbook to install Malware Information Sharing Platform (MISP)	**unmaintained**
ansible MISP	ansible role to setup MISP	Not tested by MISP core team
OpenDXL ATD MISP	Automated threat intelligence collection with McAfee ATD, OpenDXL and MISP	Not tested by MISP core team
IMAP Proxy	Modular IMAP proxy (including PyCIRCLeanMail and MISP forward modules)	Not tested by MISP core team
AutoMISP	automate your MISP installs - This shell script is designed to automatically install MISP and the misp-modules extension on either Ubuntu 16.04, or 18.04.	Not tested by MISP core team
Palo Alto Networks report_to_misp	Parse a report and import the events into MISP	Not tested by MISP core team
Palo Alto Networks minemeld-misp	MineMeld nodes for MISP	Not tested by MISP core team
golang-misp	Golang Library to interact with your MISP instance	Not tested by MISP core team
go-misp	Golang MISP API Client	Not tested by MISP core team
MISP MAR	Integration between MISP platform and McAfee Active Response	Not tested by MISP core team
MISP IoC Validator	Validate IOC from MISP ; Export results and iocs to SIEM and sensors using syslog and CEF format	Not tested by MISP core team

Project	Description	Status
vt2misp	Script to fetch data from virustotal and add it to a specific event as an object	Not tested by MISP core team
Threat Pinch Lookup	Documentation and Sharing Repository for ThreatPinch Lookup Chrome & Firefox Extension	Not tested by MISP core team
dovehawk	Dovehawk is a Bro module that automatically imports MISP indicators and reports Sightings	Not tested by MISP core team
yara-exporter	Exporting MISP event attributes to yara rules usable with Thor apt scanner	Not tested by MISP core team
volatility-misp	Volatility plugin to interface with MISP	Not tested by MISP core team
misp2bro	Python script that gets IOC from MISP and converts it into BRO intel files.	Not tested by MISP core team
TA-misp	Splunk integration with MISP	Not tested by MISP core team
Tinycheck	TinyCheck integration with MISP	Not tested by MISP core team
MISP QRadar	The Project can used to integrate QRadar with MISP Threat Sharing Platform	Not tested by MISP core team
pymisp-suricata_search	Multi-threaded suricata search module for MISP	Not tested by MISP core team
MISP-ThreatExchange	Script to interface MISP with Facebook ThreatExchange	Not tested by MISP core team
aptc	Automated Payload Test Controller	Not tested by MISP core team
aptmap	A map displaying threat actors from the misp-galaxy	Not tested by MISP core team
mispy	Another MISP module for Python	Not tested by MISP core team
MispSharp	C# Library for MISP	Not tested by MISP core team
misp_btc	get BTC addresses from MISP and fetch BTC transactions	Tested by MISP core team
Privacy Aware Sharing of IoCs in MISP	Master Thesis including MISP data.	Master thesis
sam-bot	Bot to create MISP events from data in Slack	Not tested by MISP core team
Polarity.io Connector	"Polarity is the memory augmentation platform that makes your team smarter"	Not tested by MISP core team

F. Appendix E: Other Threat Intel Ressources

A brief list of online ressources that around #ThreatIntel

- A curated list of awesome malware analysis tools and resources. Inspired by awesome-python and awesome-php.
- An authoritative list of awesome devsecops tools with the help from community experiments and contributions.DEV.SEC.OPS
- Advance Python IoC extractor

G. Appendix F: LDAP Authentication

MISP can authenticate users against an LDAP server through the shipped **LdapAuth** plugin (`app/Plugin/LdapAuth`). When enabled, a user logs in on the normal MISP login form with their LDAP credentials; if the account does not yet exist in MISP it is created on first login. The authoritative option reference is the plugin's own README; the summary below covers the essentials.

G.0.0.1 Prerequisites

Install the PHP LDAP extension:

```
# RHEL / CentOS / Fedora
sudo dnf install php-ldap
# Debian / Ubuntu
sudo apt install php-ldap
```

G.0.0.2 Configuration

LDAP is configured entirely in `app/Config/config.php`, in **two** parts. Both are required — the second is the step most commonly missed, and without it LDAP is simply never used.

1. Add the `LdapAuth` settings block with your server details and a reader account used to search the directory:

```
'LdapAuth' => [
    'ldapServer'         => 'ldap://ldap.example.com',
    'ldapDn'             => 'dc=example,dc=com',
    'ldapReaderUser'     => 'cn=reader,dc=example,dc=com',
    'ldapReaderPassword' => 'password',
    'ldapSearchFilter'   => '(objectClass=inetOrgPerson)',
    'ldapSearchAttribute'=> 'mail',
    'ldapEmailField'     => ['mail'],
    'ldapDefaultOrg'     => 1,
    'ldapDefaultRoleId'  => 3,
    'updateUser'         => true,
],
```

2. Enable the plugin by adding it to the `Security.auth` array — this is what actually turns LDAP on:

```
'Security' => [
    // ... your existing Security settings ...
    'auth' => [
        0 => 'LdapAuth.Ldap',
    ],
],
```

> ! Important
>
> Configuring the `LdapAuth` block alone does **not** enable LDAP. You must also add `'LdapAuth.Ldap'` to `Security.auth`. If you skip this step, LDAP login silently does nothing.

G.0.0.3 Key settings

The plugin exposes many options; the most important are:

Setting	Purpose	Default
`ldapServer`	LDAP server URI (`ldap://host:port` or `ldaps://…`)	—
`ldapDn`	Search-base DN containing your users	—
`ldapReaderUser` / `ldapReaderPassword`	Bind account used to search the directory	—
`ldapSearchFilter`	Filter used to locate the user entry	—
`ldapSearchAttribute`	Attribute matched against the login identifier	`mail`
`ldapEmailField`	Attribute(s) holding the user's e-mail	`['mail']`
`ldapDefaultOrg`	Organisation ID assigned to newly created LDAP users	—
`ldapDefaultRoleId`	Default role ID, or a group→role mapping (see below)	`3`
`updateUser`	Refresh a user's e-mail and role from LDAP on each login	`true`
`mixedAuth`	Allow both LDAP and local (MISP password) authentication	`true`
`starttls`	Use StartTLS for a secure connection	`false`
`ldapNetworkTimeout`	Connection timeout in seconds (`-1` = none)	`-1`
`ldapProtocol`	LDAP protocol version	`3`

G.0.0.4 Mapping LDAP groups to MISP roles

`ldapDefaultRoleId` can be a single role ID, or an array mapping an LDAP group to a MISP role ID, so group membership drives the assigned role:

```
'ldapDefaultRoleId' => [
    'misp_admin'     => 1,
    'misp_orgadmin'  => 2,
    'misp_user'      => 3,
    'misp_publisher' => 4,
    'misp_syncuser'  => 5,
    'misp_readonly'  => 6,
],
```

For **Microsoft Active Directory**, set `ldapUseMemberOf => true` to resolve group membership from the user's `memberOf` attribute; when you do, the keys in the mapping above must be the full group DN (for example `CN=MISP Admins,OU=Groups,DC=example,DC=com`) rather than the short group name.

G.0.0.5 TLS / LDAPS

For encrypted connections use an `ldaps://` URI or enable `starttls`. A set of `ldapTls*` options (`ldapTlsRequireCert`, `ldapTlsCustomCaCert`, `ldapTlsCrlCheck`, `ldapTlsProtocolMin`) map to the corresponding PHP `LDAP_OPT_X_TLS_*` constants for cases where you need a custom CA or to relax certificate checking during testing — see the plugin README for the full list.

G.0.0.6 Diagnostics

The plugin ships a standalone test script that reads your `LdapAuth` configuration and walks through a bind and a user search, which is the fastest way to debug a connection:

```
php app/Plugin/LdapAuth/Controller/Component/Auth/TestLdapAuth.php
```

You can also inspect `app/tmp/logs/error.log` for the raw error responses from the LDAP server. A good troubleshooting order is: set `debug => true`, start with `starttls => false`, and if you hit `LDAP user search failed: Operations error`, set `ldapAllowReferrals => 0`.

G.0.0.7 Migrating existing users to LDAP

LDAP and MISP users are paired by e-mail address. When you enable LDAP and an LDAP user logs in, an existing MISP user with the same e-mail address is found and linked to the LDAP identity, so existing accounts are preserved.

G.0.0.8 Caveats

- A user disabled in LDAP (or removed from the required group) is not disabled in MISP automatically — the block takes effect the next time they try to log in.
- With `mixedAuth` enabled (the default) both LDAP and local MISP authentication are accepted; set it to `false` to force LDAP only.
- The `Security.require_password_confirmation` setting does not apply to LDAP authentication.

G.0.0.9 Other external authentication plugins

`LdapAuth` is one of several external authentication plugins shipped with MISP. Each is enabled the same way — configured in `app/Config/config.php` and added to `Security.auth` — and each has its own README under `app/Plugin/`:

- **OidcAuth** — OpenID Connect (e.g. Keycloak, generic OIDC identity providers).
- **AadAuth** — Microsoft Entra ID / Azure Active Directory.
- **ShibbAuth** — Apache Shibboleth / SAML via an Apache module.
- **CertAuth** — client-certificate (X.509) authentication.
- **LinOTPAuth** — LinOTP one-time-password authentication (do not combine with MISP's built-in TOTP).

Choose at most one primary identity source for a given deployment, and consult the relevant plugin README for its exact option list.

H. Appendix G: SimpleBackgroundJobs Migration guide

Since **MISP 2.5**, background jobs are handled by **SimpleBackgroundJobs**, a simpler system that no longer relies on the unmaintained CakeResque library. SimpleBackgroundJobs is now the default (and only) background-job backend — CakeResque has been removed.

The backend requires Supervisor and some extra PHP packages. On a fresh 2.5 install these are set up for you by the installer scripts (the required packages are already declared in MISP's `composer.json`); the steps below are mainly relevant when migrating an older instance.

This guide is intended for Ubuntu/Debian systems

H.1 Install requirements

Run on your MISP instance the following commands.

1. Install **Supervisord**: `sudo apt install supervisor -y`
2. Install required PHP packages: ``` cd /var/www/MISP/app sudo -u www-data php composer.phar require –with-all-dependencies supervisorphp/supervisor:^4.0
   guzzlehttp/guzzle
   php-http/message
   lstrojny/fxmlrpc
   ```
3. Add the following settings at the bottom of the **Supervisord** conf file, usually located in:

   `/etc/supervisor/supervisord.conf` `[inet_http_server]` `port=127.0.0.1:9001 username=supervisor password=PWD_CHANGE_ME`
4. Use the following configuration as a template for the services, usually located in:

   `/etc/supervisor/conf.d/misp-workers.conf` ``` [group:misp-workers] programs=default,email,cache,prio,update

   [program:default] directory=/var/www/MISP command=/var/www/MISP/app/Console/cake start_worker default process_name=%(program_name)s_%(process_num)02d numprocs=5 autostart=true autorestart=true redirect_stderr=false stderr_logfile=/var/www/MISP/app/tmp/logs/misp-workers-errors.log stdout_logfile=/var/www/MISP/app/tmp/logs/misp-workers.log directory=/var/www/MISP user=www-data

   [program:prio] directory=/var/www/MISP command=/var/www/MISP/app/Console/cake start_worker prio process_name=%(program_name)s_%(process_num)02d numprocs=5 autostart=true autorestart=true redirect_stderr=false stderr_logfile=/var/www/MISP/app/tmp/logs/misp-workers-errors.log stdout_logfile=/var/www/MISP/app/tmp/logs/misp-workers.log directory=/var/www/MISP user=www-data

   [program:email] directory=/var/www/MISP command=/var/www/MISP/app/Console/cake start_worker email process_name=%(program_name)s_%(process_num)02d numprocs=5 autostart=true autorestart=true redirect_stderr=false stderr_logfile=/var/www/MISP/app/tmp/logs/misp-workers-errors.log stdout_logfile=/var/www/MISP/app/tmp/logs/misp-workers.log directory=/var/www/MISP user=www-data

   [program:update] directory=/var/www/MISP command=/var/www/MISP/app/Console/cake start_worker update process_name=%(program_name)s_%(process_num)02d numprocs=1 autostart=true autorestart=true redirect_stderr=false stderr_logfile=/var/www/MISP/app/tmp/logs/misp-workers-errors.log stdout_logfile=/var/www/MISP/app/tmp/logs/misp-workers.log directory=/var/www/MISP user=www-data
   ```

[program:cache] directory=/var/www/MISP command=/var/www/MISP/app/Console/cake start_worker cache process_name=%(program_name)s_%(process_num)02d numprocs=5 autostart=true autorestart=true redirect_stderr=false stderr_logfile=/var/www/MISP/app/tmp/logs/misp-workers-errors.log stdout_logfile=/var/www/MISP/app/tmp/logs/misp-workers.log user=www-data ```

5. Restart **Supervisord** to load the changes: `sudo service supervisor restart`
6. Check **Supervisord** workers are running: `$ sudo supervisorctl status misp-workers:cache_00 RUNNING pid 1673228, uptime 1:37:54 misp-workers:cache_01 RUNNING pid 1673225, uptime 1:37:54 misp-workers:cache_02 RUNNING pid 1673375, uptime 1:37:53 misp-workers:cache_03 RUNNING pid 1673398, uptime 1:37:52 misp-workers:cache_04 RUNNING pid 1673303, uptime 1:37:53 misp-workers:default_00 RUNNING pid 1673222, uptime 1:37:54 misp-workers:default_01 RUNNING pid 1673385, uptime 1:37:52 misp-workers:default_02 RUNNING pid 1673391, uptime 1:37:52 misp-workers:default_03 RUNNING pid 1673223, uptime 1:37:54 misp-workers:default_04 RUNNING pid 1673393, uptime 1:37:52 misp-workers:email_00 RUNNING pid 1673394, uptime 1:37:52 misp-workers:email_01 RUNNING pid 1673312, uptime 1:37:53 misp-workers:email_02 RUNNING pid 1673224, uptime 1:37:54 misp-workers:email_03 RUNNING pid 1673227, uptime 1:37:54 misp-workers:email_04 RUNNING pid 1673333, uptime 1:37:53 misp-workers:prio_00 RUNNING pid 1673279, uptime 1:37:54 misp-workers:prio_01 RUNNING pid 1673304, uptime 1:37:53 misp-workers:prio_02 RUNNING pid 1673305, uptime 1:37:53 misp-workers:prio_03 RUNNING pid 1673232, uptime 1:37:54 misp-workers:prio_04 RUNNING pid 1673319, uptime 1:37:53 misp-workers:update_00 RUNNING pid 1673327, uptime 1:37:53`

H.2 MISP Config

1. Go to your **MISP** instances `Server Settings & Maintenance` page, and then to the new **SimpleBackgroundJobs** tab.
2. Update the `SimpleBackgroundJobs.supervisor_password` with the password you set in the *Install requirements* section 3.
3. Verify Redis and other settings are correct and then set `SimpleBackgroundJobs.enabled` to `true`.
4. Use **MISP** normally and visit **Administration -> Jobs** (`/jobs/index` on your instance) to check Jobs are running correctly. If there are any issues check the logs:
   - /var/www/MISP/app/tmp/logs/misp-workers-errors.log
   - /var/www/MISP/app/tmp/logs/misp-workers.log
5. Once the new workers are functioning as expected, you can remove the previous workers service: `bash $ sudo systemctl stop --now misp-workers $ sudo systemctl disable --now misp-workers`

H.2.1 Notes

Scheduled tasks (TasksController) are not supported with the new backend, however this feature is going to be deprecated, it is recommended to use cron jobs instead.

I. Glossary

This glossary is intended as a quick reference for clarifying threat-sharing and threat-intelligence terminology. Be careful when adding terms to the glossary. Adding a generic term such as *MISP* will prevent more specific terms such as *MISP noticelist* from being added. As a general rule, please use the singular form for each term. If you use any CC BY-SA licensed content, or any other material that is subject to licensing, make sure to add the appropriate attribution at the end of the entry.

I.1 API

MISP makes extensive use of its RESTful API (Application programming interface) both internally and provides an external API for automation, synchronisation or any other tasks requiring a machine to machine interface. In general terms, it is a set of clearly defined methods of communication between various software components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer. An API may be for a web-based system, operating system, database system, computer hardware or software library. The de-facto standard for talking to MISP via an API is PyMISP. Partial source "API" - CCBYSA.

I.2 RESTful

Representational state transfer (REST) or RESTful web services are a way of providing interoperability between computer systems on the Internet. REST-compliant Web services allow requesting systems to access and manipulate textual representations of Web resources using a uniform and predefined set of stateless operations. Other forms of Web services exist which expose their own arbitrary sets of operations such as WSDL and SOAP. Source "REST" - CCBYSA.

I.3 Sharing groups

Sharing groups in MISP are a more granular way to create re-usable distribution lists for events/attributes that allow users to include organisations from their own instance (local organisations) as well as organisations from directly, or indirectly connected instances (external organisations). More about creating and managing sharing groups

I.4 Site admin

As an admin (not to be confused with Org Admin), you can set up new accounts for users, edit user profiles, delete them, or just have a look at all the viewers' profiles. Site admins have access to every administrator feature for all the data located on the system including global features such as the creation and modification of user roles and instance links. You will also see all other organizations connected to or set up in the instance. The site admin can be considered as a super-user of a MISP instance.

I.5 Indicators

Indicators contain a pattern that can be used to detect suspicious or malicious cyber activity.

I.6 IOC

Indicator of compromise (IOC or IoC) is an artefact observed on a network or in an operating system or information channel that could reference an intrusion or a reference to a technique used by an attacker. IoCs are a subset of indicators.

I.7 NIDS

Network Intrusion Detection System eg. Snort, Suricata

I.8 MISP

MISP is a recursive acronym for the **MISP Intelligence Sharing Platform**. The project was originally named the *Malware Information Sharing Platform*, but as its scope grew well beyond malware to cover every kind of threat intelligence, the name was redefined recursively — in the tradition of GNU ("GNU's Not Unix"). It is commonly known simply as MISP.

I.9 MISP Attribute

Attributes in MISP can be network indicators (e.g. IP address), system indicators (e.g. a string in memory) or even bank account details. ◦ A type (e.g. MD5, url) is how an attribute is described. ◦ An attribute is always in a category (e.g. Payload delivery) which puts it in a context. • A category is what describes an attribute. ◦ An IDS flag on an attribute allows to determine if an attribute can be automated (such as being exported as an IDS ruleset or used for detection). If the IDS flag is not present, the attribute can be useful for contextualisation only.

I.10 Observable

Some other SIEMs or formats (STIX) use the term observable. This is the same as an attribute in MISP-speak. Usually an observable is a MISP attribute without the IDS flag set.

I.11 MISP Event

MISP events are encapsulations for contextually related information represented as attribute and object.

I.12 MISP Extended Events

MISP can extend an event (since version 2.4.90). This allows users to build full blown events that extend an existing event, giving way to a combined event view that includes a sum total of the event along with all extending events. More

I.13 MISP feeds

MISP includes a set of public OSINT feeds in its default configuration. The feeds can be used as a source of correlations for all of your events and attributes without the need to import them directly into your system. The MISP feed system allows for fast correlation but also a for quick comparisons of the feeds against one another. To get started with MISP we advise to enable the CIRCL OSINT feed within your MISP instance. This feed is generated with the PyMISP feed-generator. More

I.14 MISP format

MISP formats are described in specification document based on the current implementation of MISP core and PyMISP. These specifications are available for other developers willing to develop their own tools or software supporting the MISP format.

I.15 MISP Galaxy Cluster

MISP galaxy is a simple method to express a large object called cluster that can be attached to MISP events or attributes. A cluster can be composed of one or more elements. Elements are expressed as key-values. There are default vocabularies available in MISP galaxy but those can be overwritten, replaced or updated as you wish. Existing clusters and vocabularies can be used as-is or as a template. MISP distribution can be applied to each cluster to permit a limited or broader distribution scheme. The following document is generated from the machine-readable JSON describing the MISP galaxy. There is a Python module

available to work with Galaxy Cluster in a Pythonic way called PyMISPGalaxies. MISP galaxy GitHub Repo More

I.16 MISP modules

MISP modules are autonomous modules that can be used for expansion and other services in MISP. The modules are written in Python 3 following a simple API interface. The objective is to ease the extensions of MISP functionalities without modifying core components. The API is available via a simple REST API which is independent from MISP installation or configuration. MISP modules GitHub Repo

I.17 MISP Instance

A MISP instance is an installation of the MISP software and the connected database. All the data visible to the users is stored locally in the database and data that is shareable (based on the distribution settings) can be synchronised with other instances via the Sync actions. The instance that you are reading this manual on will be referred to as "this instance" or "your instance". The instances that your instance synchronises with will be referred to as "remote instances".

I.18 MISP Objects

MISP objects are used in MISP (starting from version 2.4.80) system and can be used by other information sharing tool. MISP objects are in addition to MISP attributes to allow advanced combinations of attributes. The creation of these objects and their associated attributes are based on real cyber security use-cases and existing practices in information sharing. The objects are just shared like any other attributes in MISP even if the other MISP instances don't have the template of the object. The following document is generated from the machine-readable JSON describing the MISP objects. MISP objects GitHub Repo More

I.19 MISP GnuPG Key

or GnuPG instance key is the GnuPG (Gnu Privacy Guard) key used by the MISP instance and which is only used to sign notification. The GnuPG key used in the MISP instance must **not** be used anywhere else and should not be valuable.

I.20 MISP Sightings

Basically, sighting is a system allowing people to react on attributes on an event. It was originally designed to provide an easy method for user to tell when they see a given attribute, giving it more credibility.

I.21 MISP Taxonomies

Taxonomy is the practice and science of classification. The word is also used as a count noun: a taxonomy, or taxonomic scheme, is a particular classification. The word finds its roots in the Greek language τάξις, taxis (meaning 'order', 'arrangement') and νόμος, nomos ('law' or 'science'). Taxonomies that can be used in MISP (2.5) and other information sharing tools, expressed in Machine Tags (Triple Tags). A machine tag is composed of a namespace (MUST), a predicate (MUST) and an (OPTIONAL) value. Machine tags are often called triple tag due to their format. For more details on taxonomies and classification see the taxonomies catalogue. Partial source "Taxonomy_(general)" - CCBYSA. There is a Python module available to work with Taxonomies in a Pythonic way called PyTaxonomies. MISP taxonomies GitHub Repo

I.22 Local tags

You can create tags locally if you are a member of the given MISP instance's host organisation, enabling "in-place" tagging for synchronisation and export filtering. MISP events are not modified while using the local tags and are in turn always stripped before being synchronised with other MISP instances and sharing communities. Local tags allow users to avoid violating the ownership model of MISP, but still

be able to tag any event or attribute for further dissemination and data contextualisation. Local tagging works for tags, tag collections, galaxies and matrix-like galaxies such as ATT&CK.

I.23 MISP warninglists

MISP warninglists are lists of well-known indicators that can be associated to potential false positives, errors or mistakes. There is a Python module available to work with warninglists in a Pythonic way called PyMISPWarningLists. MISP warninglists GitHub Repo

I.24 MISP noticelist

Notice lists to inform MISP users of the legal, privacy, policy or even technical implications of using specific attributes, categories or objects. MISP noticelist GitHub Repo

I.25 Org Admin

Organisation admins (Org Admin) are restricted to executing site-admin actions exclusively within their own organisation's users only. They can administer users, events and logs of their own respective organisations.

I.26 OSINT

Open-source intelligence (OSINT) is data collected from publicly available sources to be used in an intelligence context.[1] In the intelligence community, the term "open" refers to overt, publicly available sources (as opposed to covert or clandestine sources). It is not related to open-source software or public intelligence. OSINT under one name or another has been around for hundreds of years. With the advent of instant communications and rapid information transfer, a great deal of actionable and predictive intelligence can now be obtained from public, unclassified sources. Source "Open-source intelligence" - CCBYSA.

I.27 Pivot path

The (branching) path taken by a user from event to event while following correlation links. This is represented by the branching graph in the event view.

I.28 Pivoting

The act of navigating from event to event through correlation links.

I.29 Proposals

Each event can only be directly edited by users of the original creator organisation (and site admins). However, if another organisation would like to amend an event with extra information on an event, or if they'd like to correct a mistake in an attribute, they can create a Proposal. These proposals could then be accepted by the original creator organisation. These proposals can be pulled to another server, allowing users on connected instances to propose changes which then could be accepted by the original creators on another instance (and subsequently pushed back).

I.30 Publishing

When an event is first created by a user, it is visible to everyone on the instance based on the access rights ("Your organisation only" events will not be visible to users of other organisations), but they will not be synchronised and they won't be exportable. For this, a user with publishing permission of the organisation that created the event has to publish the event. The system will then inform all the users of the instance that are subscribing to e-mail notifications and who have access to view the published event via an e-mail.

I.31 Pull

Pulling is the process of using the configured sync user on a remote instance to REST GET all of the accessible data (based on the distribution rights) to your instance and store it.

I.32 Push

Pushing is the process of using a configured instance link to send an event or all accessible events (limited by the distribution rights) through the REST interface to a remote instance.

I.33 Roles

Roles are the central place where you can define ACL roles for your local users on your instance. You can add new Roles depending on your use case. The following permissions can be given depending on which overall *Permission* the role inherits.

- Permission -> Read Only, Manage Own Events, Manage Organisation Events, Manage and Publish Organisation Events
- Site Admin, Org Admin
- Sync Actions (and Internal / Authoritative Sync Actions)
- Audit Actions, Auth key access, Regex Actions
- Tagger, Tag Editor, Template Editor, Sharing Group Editor, Delegations Access
- Sighting Creator, Object Template Editor, Galaxy Editor, Decaying Model Editor
- ZMQ publisher, Kafka publisher, Warninglist Editor, View Feed Correlations
- Analyst Data Creator, Skip OTP Reqs, Server Signing

See the Roles section for the full list of permissions with descriptions.

I.34 Scheduled Tasks

Certain common tasks can be scheduled for a later execution or for regular recurring executions. These tasks currently include caching all of the export formats, pulling from all eligible instances and pushing to all eligible instances.

I.35 Standard MISP Install

Any MISP instance install that is strongly aligned with our official install guides. This is mostly to make sure you have a similar folder structure, /var/www/MISP for an Ubuntu Server Install. It will also be easier to debug any Web Server issues or other system related problems.

I.36 Sync User

A user of a role that grants sync permissions, these users (and their authentication keys) are used to serve as the points of connection between instances. Events pushed to an instance are pushed to a sync user, who then creates the events on the remote instance. Events pulled are added by the sync user that is used to connect the remote instance to your instance. As an administrator, keep in mind that a sync user needs auth key and publish permissions, has to have undergone the mandatory password change and has to have accepted the Terms of Use in order for the sync to work. Please make sure that all of these steps are taken before attempting to push or pull.

I.37 Synchronisation

What we call synchronisation is an exchange of data between two (or more) MISP instances through our pull and push mechanisms.

I.38 Tagging

Users with tagging rights can assigned various dynamically created tags to events, allowing an arbitrary link between events to be created. It is possible to filter events based on these tags and they can also be used to filter events for the automation.

I.39 Templating

Users with templating rights can create easy to fill forms that help with the event creation process.

I.40 Analyst Data

Collaborative annotations attached to almost any MISP element (events, attributes, objects, galaxy clusters, event reports). There are three kinds: **Notes** (free-text commentary), **Opinions** (a confidence/agreement score with an optional comment) and **Relationships** (typed links between elements). Analyst data is shared and synchronised alongside the data it annotates.

I.41 Event Report

A rich-text (Markdown) report attached to an event, used to narrate the intelligence around the indicators. Event reports support indicator extraction, import from a URL, and PDF export.

I.42 Decaying Model

A configurable model that scores how relevant an indicator remains over time, so stale indicators can be automatically aged out instead of being trusted indefinitely.

I.43 Workflow

An event-driven automation graph that runs when something happens in MISP (for example when an event is published), used to block, enrich or otherwise act on data. Workflows are built from triggers and modules and can be shared as blueprints.

I.44 Cerebrate

A companion platform for managing organisations and sharing communities; a MISP instance can synchronise organisation and sharing-group definitions from a connected Cerebrate node.

I.45 Collection

A named bundle of MISP elements (such as events and galaxy clusters) that can be managed and synchronised as a unit.

I.46 Community

A community is composed of the local organisations on a MISP server and the remote organisations connected by the sync users.