MISP Dashboard

MISP DASHBOARD

INFO@CIRCLLU AUGUST 3, 2022



MISP DASHBOARD

REAL-TIME OVERVIEW OF THREAT INTELLIGENCE FROM

CIRCL / TEAM MISP PROJECT

INFO@CIRCL.LU

AUGUST 3, 2022



MISP ZEROMQ

MISP ZEROMQ

MISP includes a flexible publish-subscribe model to allow real-time integration of the MISP activities:

- Event publication
- Attribute creation or removal
- Sighting
- User login

 \rightarrow Operates at global level in MISP

MISP Dashboard

MISP ZeroMQ

MISP ZeroMQ

MISP ZeroMQ

MISP ZeroMQ

All Contracts of finishing publish culturation model to allow extension or more allow an integration of the MEP activation.

I write the contraction or more all a significance.

I were publication or more all a significance.

I were publication or more all a significance.

I were publication or more allowed as a global level in MEP.

MISP ZEROMQ

MISP ZeroMQ functionality can be used for various model of integration or to extend MISP functionalities:

- Real-time search of indicators into a SIEM¹
- Dashboard activities
- Logging mechanisms
- Continuous indexing
- Custom software or scripting

MISP Dashboard

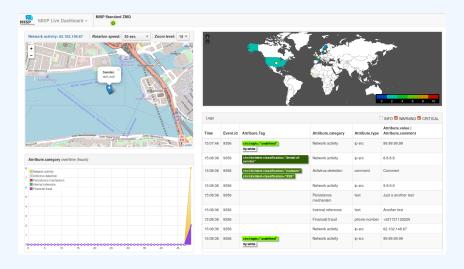
MISP ZeroMQ functionality can be used for surrous received of integration or to extend (APP inctionalities:

Mischard desiries:
Mischard desiries:
Continue indexing
Control surfaces of the Control of Integration (Integration Integration Int

¹Security Information & Event Management

MISP-DASHBOARD: AN INTRODUCTION

MISP-DASHBOARD - REALTIME ACTIVITIES AND THREAT INTELLIGENCE



MISP Dashboard

MISP-Dashboard: An introduction

-MISP-Dashboard - Realtime activities and threat intelligence



MISP-DASHBOARD - FEATURES





- Subscribe to multiple **ZMQ** MISP instances
- Provides historical geolocalised information
- Present an experimental **Gamification of the platform**
- Shows when and how MISP is used
- Provides real time information showing current threats and activity

MISP Dashboard

-MISP-Dashboard: An introduction

-MISP-Dashboard - Features



MISP-DASHBOARD: ARCHITECTURE AND DEVELOPMENT

SETTING UP THE DASHBOARD

- 1. Be sure to have a running redis server: e.g.
 - ► redis-server -p 6250
- 2. Update your configuration in config.cfg
- 3. Activate your virtualenv:
 - ▶ . ./DASHENV/bin/activate
- 4. Listen to the MISP feed by starting the zmq_subscriber:
 - ► ./zmq_subscriber.py
- 5. Start the dispatcher to process received messages:
 - ► ./zmg dispatcher.py
- 6. Start the Flask server:
 - ► ./server.py
- 7. Access the interface at http://localhost:8001/

MISP Dashboard

-MISP-Dashboard: Architecture and development

__Setting up the dashboard

SETTING OF THE BASHBOARD

to have a running redis server: e.g.

Update your configuration in config.

► . ./DASHENV/bin/activate

./zmg_subscriber.py

F Start the dispatcher to prover received morrantee

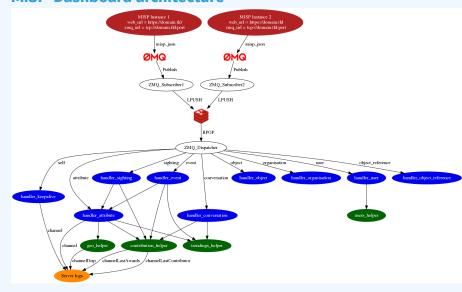
Start the dispatcher to process received m
 ./zmq_dispatcher.py

Start the Flask server:
 ▶ ./server.ov

./server.py

8

MISP-Dashboard architecture



14

MISP Dashboard

MISP-Dashboard: Architecture and development

MISP-Dashboard architecture

WRITING YOUR HANDLER

```
1 # Register your handler
2 dico_action = {
                                        handler_dispatcher,
          "misp_json":
          "misp_json_event":
                                        handler_event,
          "misp json self":
                                        handler keepalive,
          "misp json attribute":
                                        handler attribute,
          "misp json object":
                                        handler object,
          "misp json sighting":
                                        YOUR CUSTOM SIGHTINGS HANDLER,
          "misp json organisation":
                                        handler log,
          "misp json user":
                                        handler user,
          "misp json conversation":
                                        handler conversation,
          "misp json object reference": handler log,
```

MISP Dashboard

MISP-Dashboard: Architecture and development

-Writing your handler

Magicine route studies

Magicine pour studies

```
1 # Implement your handler
3 # e.g. user handler
4 def handler user(zmg name, jsondata):
      # json action performed by the user
     action = jsondata['action']
      # user json data
      json user = jsondata['User']
      # organisation json data
      json org = jsondata['Organisation']
      # organisation name
      org = json_org['name']
      # only consider user login
     if action == 'login':
          timestamp = time.time()
          # users_helper is a class to interact with the DB
         users_helper.add_user_login(timestamp, org)
```

MISP Dashboard MISP-Dashboard: Architecture and development

```
Indicated year harder

**Commission of the commission of the commi
```

RECENT CHANGES IN THE MISP-DASHBOARD

- MISP authentication can now be used in the misp-dashboard
- Improved TLS/SSL support in the default misp-dashboard
- Self-test tool to debug and test ZMQ connectivity

MISP Dashboard

MISP-Dashboard: Architecture and development

Recent changes in the misp-dashboard

FUTURE DEVELOPMENT

Optimizing contribution scoring and model to encourage sharing and contributions enrichment



Increasing geolocation coverage



Global filtering capabilities

- Geolocation: Showing wanted attribute or only on specific region
 - Trendings: Showing only specified taxonomies



Tighter integration with MISP

- Present in MISP by default
- ACL enabled version

MISP Dashboard

MISP-Dashboard: Architecture and development

Future development

Optimizing contribution scoring and model to secondary barling and contributions enrichment of the contribution coverage of increasing geolocation coverage of closh fiftering capabilities - desciration flowing susted attribute or only on specific - Treadings chosing only specified tenomies - Trigets integration with MED - Province is MED by dataset.

CONCLUSION

MISP-Dashboard can provides realtime information to support security teams, CSIRTs or SOC showing current threats and activity by providing:

- Historical geolocalised information
- Geospatial information from specific regions
- The most active events, categories, tags, attributes, ...

It also propose a prototype of gamification of the platform providing incentive to share and contribute to the community

MISP Dashboard

MISP-Dashboard: Architecture and development

Conclusion

ON

MISP-Dashboard can provides realtime information to suppor security teams, CSIRTs or SOC showing current threats and

Historical geolocalised information

Geospatial information from specific regions

The most active events, categories, tags, attributes,
the least active events and formities of the obtains

The most active events are formities of the obtain.

also propose a prototype of gamification of the platform roviding incentive to share and contribute to the commun