MISP Use Case Report

BoB 8th Kang Seong Min

Contents

I. What is MISP?

MISP is a threat intelligence platform for sharing, storing and correlating Indicators of Compromise of targeted attacks, threat intelligence, financial fraud information, vulnerability information or even counter-terrorism information. Discover how MISP is used today in multiple organisations. Not only to store, share, collaborate on cyber security indicators, malware analysis, but also to use the IoCs and information to detect and prevent attacks, frauds or threats against ICT infrastructures, organisations or people.

II. Create events

You can create an event based on a report. I found report about attack could related to North korea APT Group. I append this information to my MISP Instance. I could add this event by clicking the "Add Event" option or Using API.

figure 1. Report related to North kore apt group

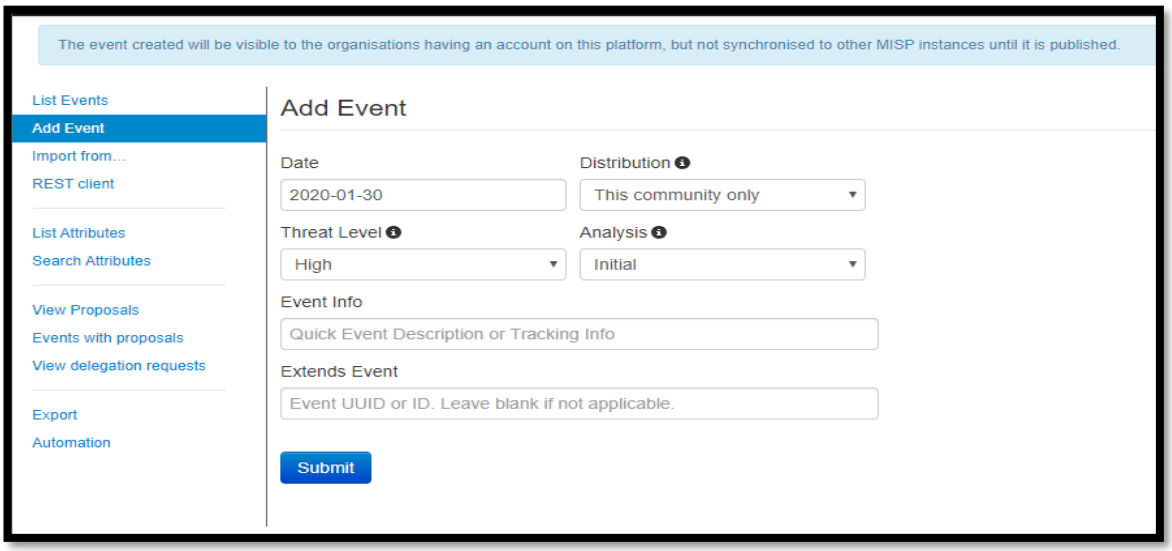


figure 2. Add Event option

```
C:₩Users₩SeongMin Kang₩Desktop₩MISP_API>python create_events.py -h
usage: create_events.py [-h] [-d DISTRIB] [-i INFO] [-a ANALYSIS] [-t THREAT]

Create an event on MISP.

optional arguments:
  -h, --help            show this help message and exit
  -d DISTRIB, --distrib DISTRIB
                        The distribution setting used for the attributes and
                        for the newly created event, if relevant. [0-3].
  -i INFO, --info INFO  Used to populate the event info field if no event ID
                        supplied.
  -a ANALYSIS, --analysis ANALYSIS
                        The analysis level of the newly created event, if
                        applicable. [0-2]
  -t THREAT, --threat THREAT
                        The threat level ID of the newly created event, if
                        applicable. [1-4]
```

figure 3. Create events by using API

Using these option, I could add Date, Distribution(choose sharing range), Threat Level, Analysis, Event Info. Additionally could make tags by using "Add tag" or API. By using tags, we could know the events is related to what efficiently.

figure 4. Add Tag options

```
C:₩PyMISP-master₩PyMISP-master₩examples>python addtag.py -h
usage: addtag.py [-h] -e EVENT [-a ATTRIBUTE] -t TAG [-m]

Get an event from a MISP instance.

optional arguments:
  -h, --help            show this help message and exit
  -e EVENT, --event EVENT
                        Event ID to get.
  -a ATTRIBUTE, --attribute ATTRIBUTE
                        Attribute ID to modify. A little dirty for now,
                        argument need to be included in event
  -t TAG, --tag TAG     Tag ID.
  -m, --modify_attribute
                        If set, the tag will be add to the attribute,
                        otherwise to the event.
```

figure 5. Add Tag by using API

And we could add attributes(IoC, reports link...etc) to events by using "Add Attribute" or API. Or could append using Free text tool by clicking

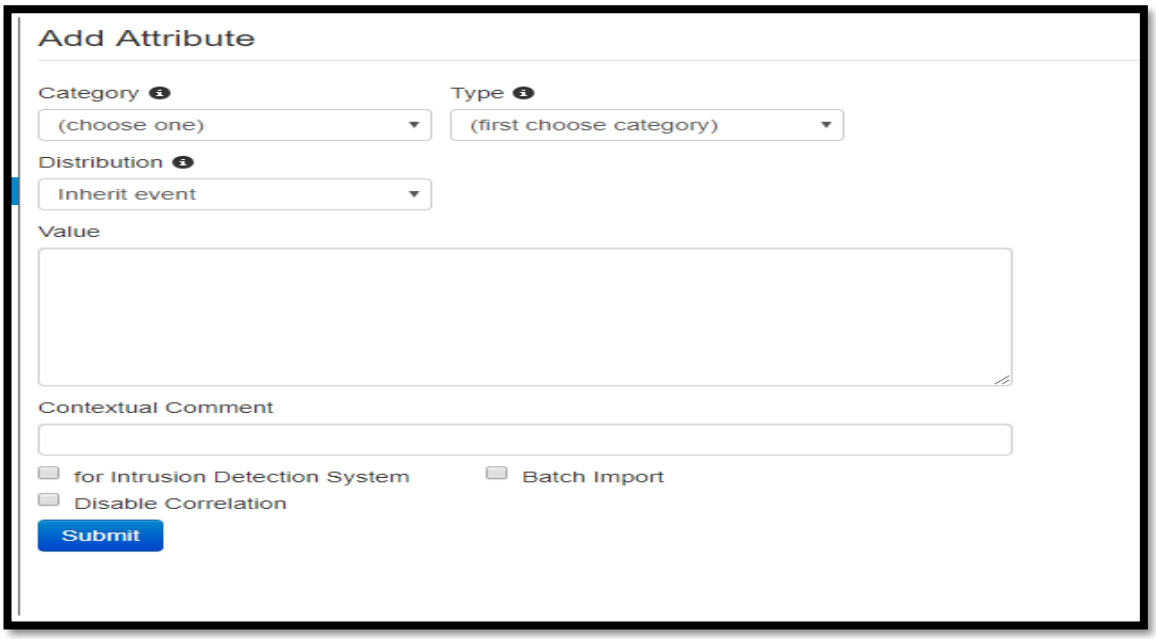


figure 6. Add Attribute options

```
C:₩Users₩SeongMin Kang₩Desktop₩MISP_API>python add_named_attribute.py -h
usage: add_named_attribute.py [-h] [-e EVENT] [-t TYPE] [-v VALUE]

Add an attribute to an event

optional arguments:
  -h, --help            show this help message and exit
  -e EVENT, --event EVENT
                        The id, uuid or json of the event to update.
  -t TYPE, --type TYPE  The type of the added attribute
  -v VALUE, --value VALUE
                        The value of the attribute
```

figure 7. Add Attribute by using API

You also could add attribute by using Freetxt Import tool. It allings attributes about their characteristics.

figure 8. Free text tool button

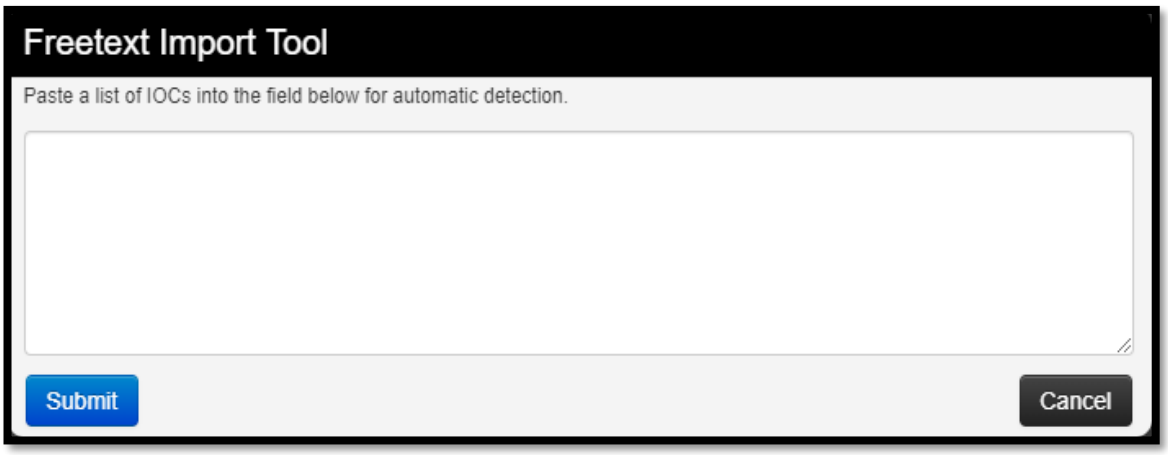
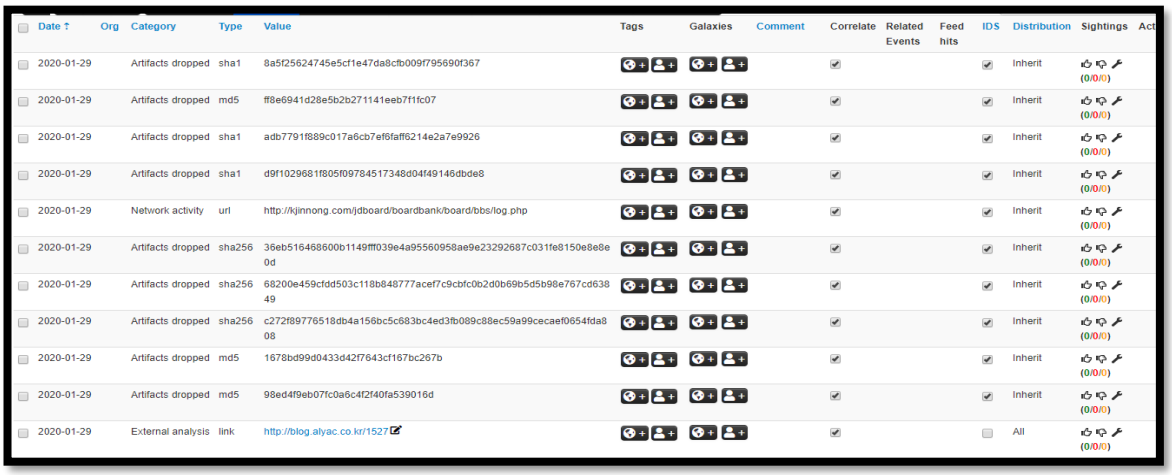


figure 9. Freextext Import Tool



Date	Org	Category	Type	Value	Tags	Galaxies	Comment	Correlate	Related Events	Feed hits	IDS	Distribution	Sightings	Act
2020-01-29		Artifacts dropped	sha1	8a5f25624745e5cf1e47da8cfb009f795690f367				✓			✓	Inherit	(0/0/0)	
2020-01-29		Artifacts dropped	md5	ff8e6941d28e5b2b271141eeb7f1fc07				✓			✓	Inherit	(0/0/0)	
2020-01-29		Artifacts dropped	sha1	adb7791f889c017a6cb7ef6faff6214e2a7e9926				✓			✓	Inherit	(0/0/0)	
2020-01-29		Artifacts dropped	sha1	d9f1029681f805f09784517348d04f49146dbde8				✓			✓	Inherit	(0/0/0)	
2020-01-29		Network activity	url	http://kjinnong.com/jdboard/boardbank/board/bbs/log.php				✓			✓	Inherit	(0/0/0)	
2020-01-29		Artifacts dropped	sha256	36eb516468600b1149fff039e4a95560958ae9e23292687c031fe8150e8e8e0d				✓			✓	Inherit	(0/0/0)	
2020-01-29		Artifacts dropped	sha256	68200e459cfdd503c118b848777acef7c9cbfc0b2d0b69b5d5b98e767cd63849				✓			✓	Inherit	(0/0/0)	
2020-01-29		Artifacts dropped	sha256	c272f89776518db4a156bc5c683bc4ed3fb089c88ec59a99cecaef0654fda808				✓			✓	Inherit	(0/0/0)	
2020-01-29		Artifacts dropped	md5	1678bd99d0433d42f7643cf167bc267b				✓			✓	Inherit	(0/0/0)	
2020-01-29		Artifacts dropped	md5	98ed4f9eb07fc0a6c4f2f40fa539016d				✓			✓	Inherit	(0/0/0)	
2020-01-29		External analysis	link	http://blog.alyac.co.kr/1527				✓				All	(0/0/0)	

figure 10. Attributes added by Freetext import tool

III. Using API

PyMISP is a Python library to access MISP platforms via their REST API. You can Install PyMISP by either pip or by getting the last version from the Github repository[1]. Before using this, you should know your baseurl and Auth key. You can find your Auth key in https://<base url>/users/view/me.

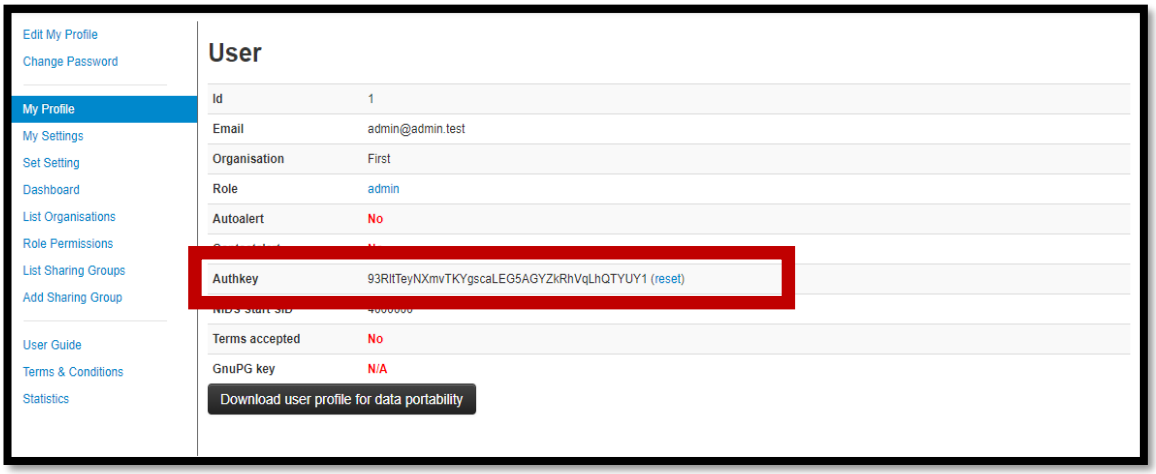


figure 11. Auth key

I made API with regex and using Curl.

[1] https://github.com/MISP/PyMISP

```
def check_data_character(data):
    md5 = re.findall(r"([a-fA-F\d]{32})", data)
    sha1 = re.findall(r"([a-fA-F\d]{40})", data)
    sha256 = re.findall(r"([a-fA-F\d]{64})", data)
    IP = re.findall(r"\b(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\b", data)
    Email = re.findall(r"\b([a-z][_a-z0-9-.]+@[a-z0-9-]+\.[a-z]+)\b", data)
    CVE = re.findall(r"\b(CVE\-[0-9]{4}\-[0-9]{4,6})\b", data)
    Filename = re.findall(r"\b([A-Za-z0-9-_\.]+\.(exe|dll|bat|sys|htm|html|js|jar|jpg|png|vb|scr|pif|chm|zip|rar|cab|pdf|doc|docx|ppt|pptx|xls|xlsx|swf|gif))\b", data)
```

Figure 12. regex to find IoCs

The command about curl could be found in MISP API document[2]. Using this code. I uploaded bulk of Iocs to my MISP instance.

Figure 13. uploaded Iocs

Also you could add tags to events or each of IoCs. Tag is important for distinguish the Attacker group. You can check this option in "Event Actions"

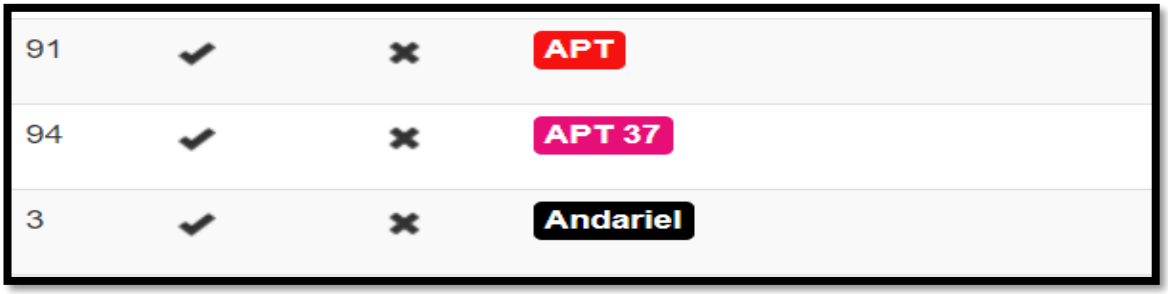


Figure 14. Tags

If there exist same IoCs with other events, MISP create Corelation Graph

[2] https://www.circl.lu/doc/misp/automation/

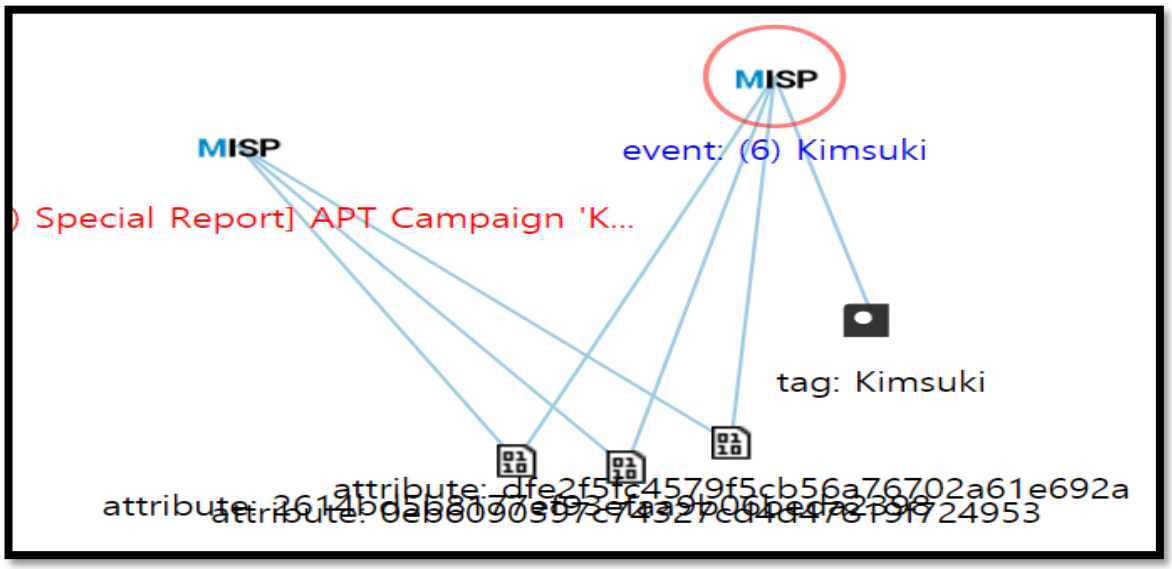


Figure 15. Correlation Graph

IV. MISP-Cloud

You can install MISP in EC2 instance by selecting MISP-Cloud

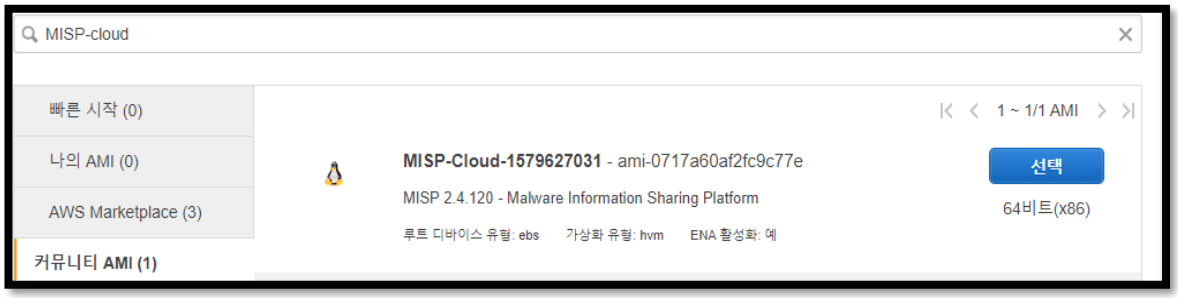


figure 12. MISP-Cloud

Start by selecting **"Community AMIs"** and search for **MISP-Cloud**. The builds are always created with "MISP" and the creation date. Chose "Select" after finding the MISP image.

figure 13. Choose instance type

The images are built to run on a t2.micro instance, which falls under the "Free Tier" option of AWS. You're free to select another instance type. You can accept the defaults and proceed until **Step 6** where you'll get to configure the firewall rules *(security groups)*:

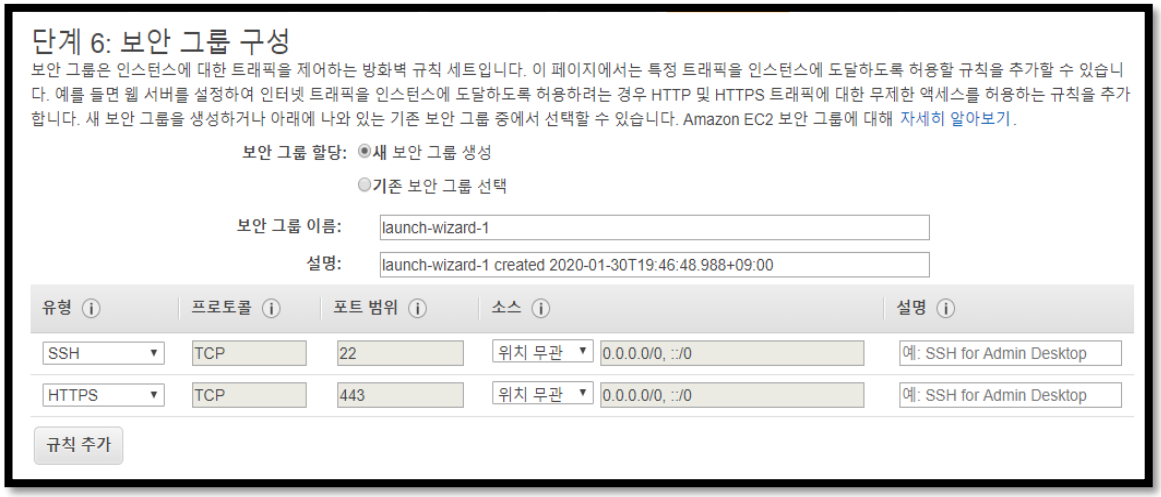


figure 14. Security settings

MISP-Cloud requires at least 443 *(HTTPS)*. You can always choose **My IP** to restrict the source to your IP address. After that, you'll be able to launch your instance. Before doing that, however, you need to handle SSH access *(even if you don't plan on using it, AWS requires this step to be completed)*:

V. Sharing/synchronization

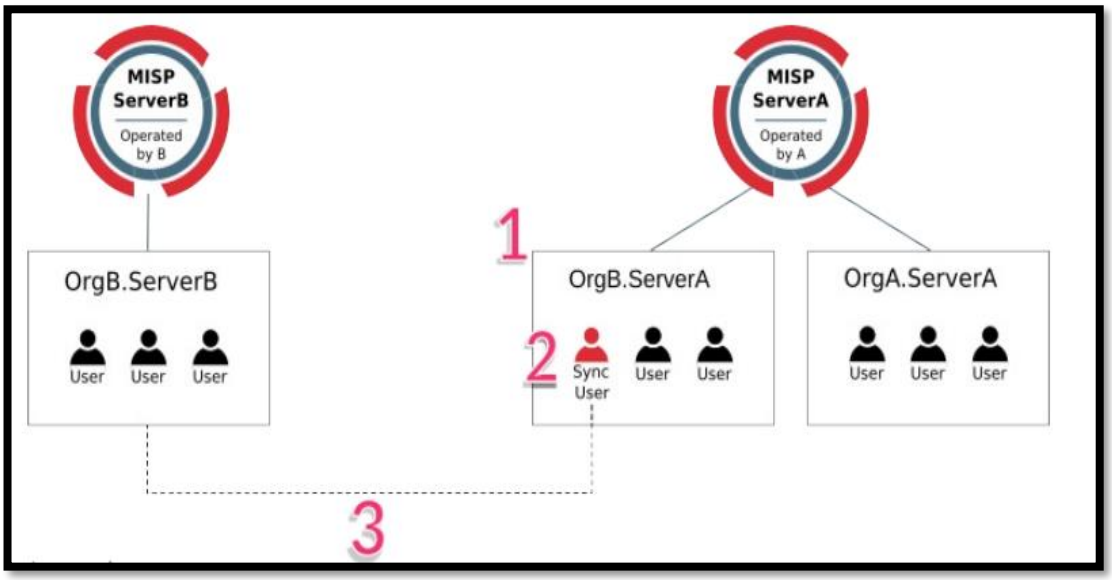


figure 15. Synchronization diagram

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.

Step 1:

I try this server B(https://3.87.219.193), server A(52.90.175.205)

I append Org_KSM.Server_KSM to server B

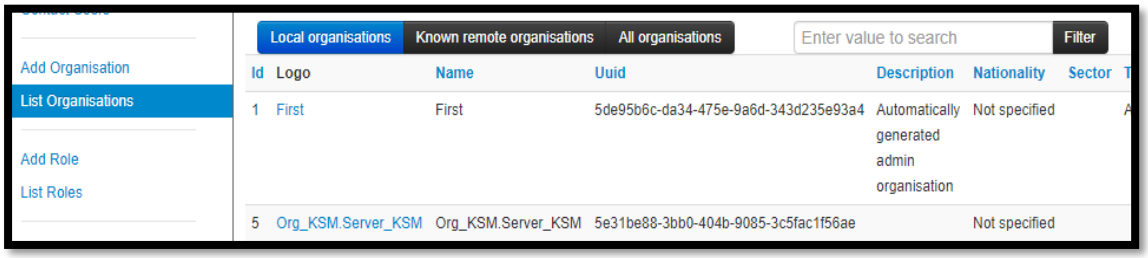


figure 16. Add orgB.ServerB

And append this organization to server A as local organization

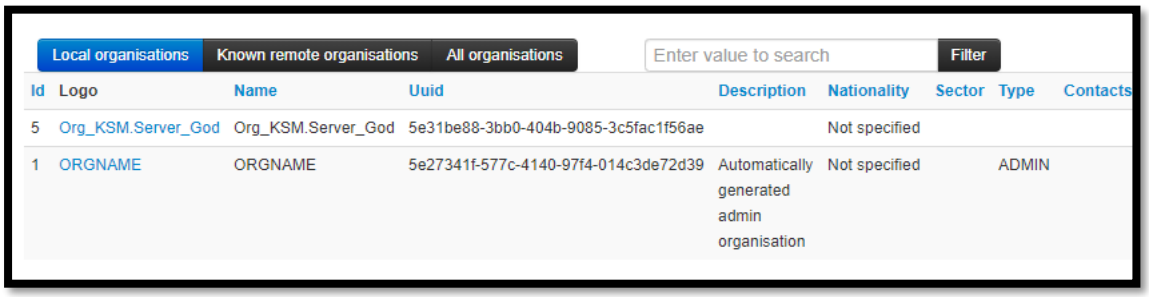


figure 17. Add OrgB.Server A

Step 2:

I Add sync user in Org_KSM.Server_God organization

figure 18: Add sync user in OrgB.Server A

Step 3:

Add Server by using "New Servers" option(Sync Actions -> List Servers -> New Servers) I fill Auth key with Sync user in Org_KSM.Server_GOd

figure 19. Add Server by fill auth key with sync users' auth key

VI. Use case

I made two events related to apt 37, Kimsuki. Two groups are related to North Korea. I uploaded total 267 IoCs using automate API. MISP show correlation IoC in events. For Example, I uploaded ioc related to the report about "Continued targeting of crypto-currencies in South Korea". I could find 6 attribute was related. By using this graph. We

could know the attack in report was realted to APT 37.

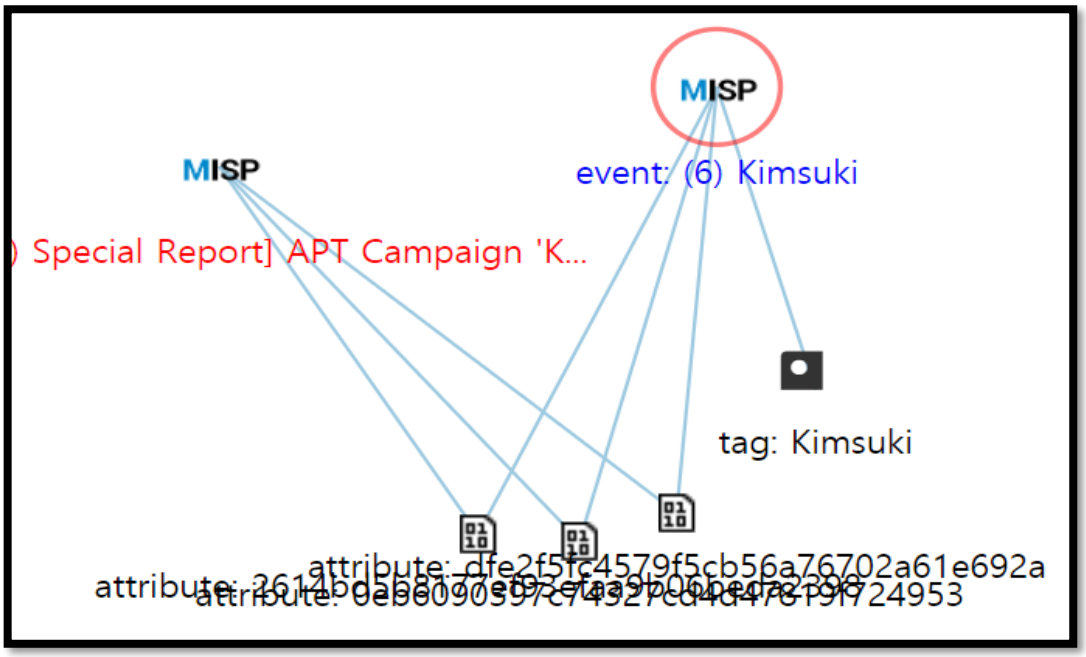


figure 20. Correlation view graph

I shared IoCs with ill-hyeon, Se-young but MISP didn't make the correlation graph. It means the APT groups are act independently. And I add the feeds to my events. I find correlations with APT 37 and "Korea the Cross Hairs".

Figure 21. Correlation Graph with feeds

Ⅶ. Problems

1. Redirect strange IP

It direct strange IP sometimes. My MISP instance ip is 3.87.219.193 but, sometimes the site go to 3.87.222.81. I solved the problem by changing contents of "/var/www/MISP/app/Config/config.php" file. The base url was set 3.87.222.81, I changed it to 3.87.219.193. The problem solved.

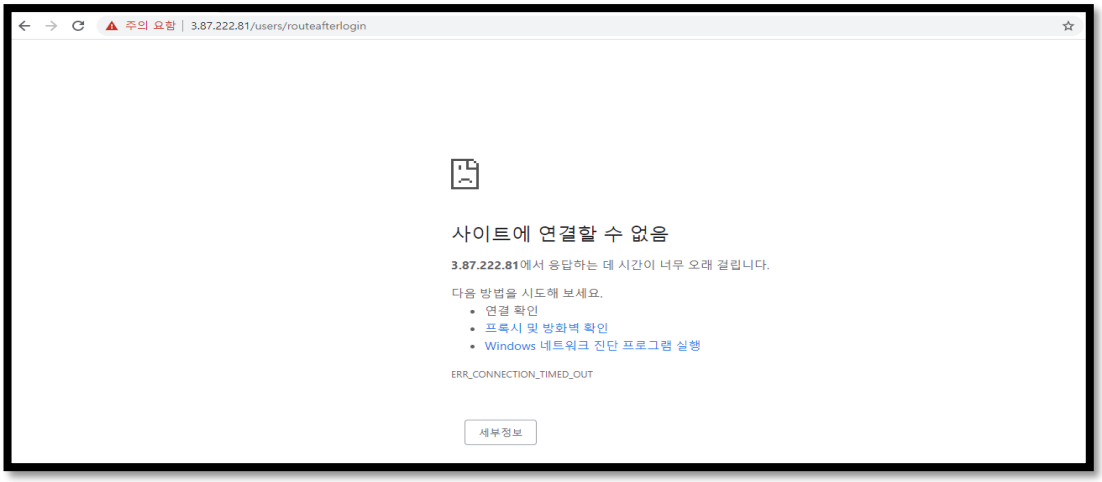


figure 20. Problems with redirecting wrong ip address

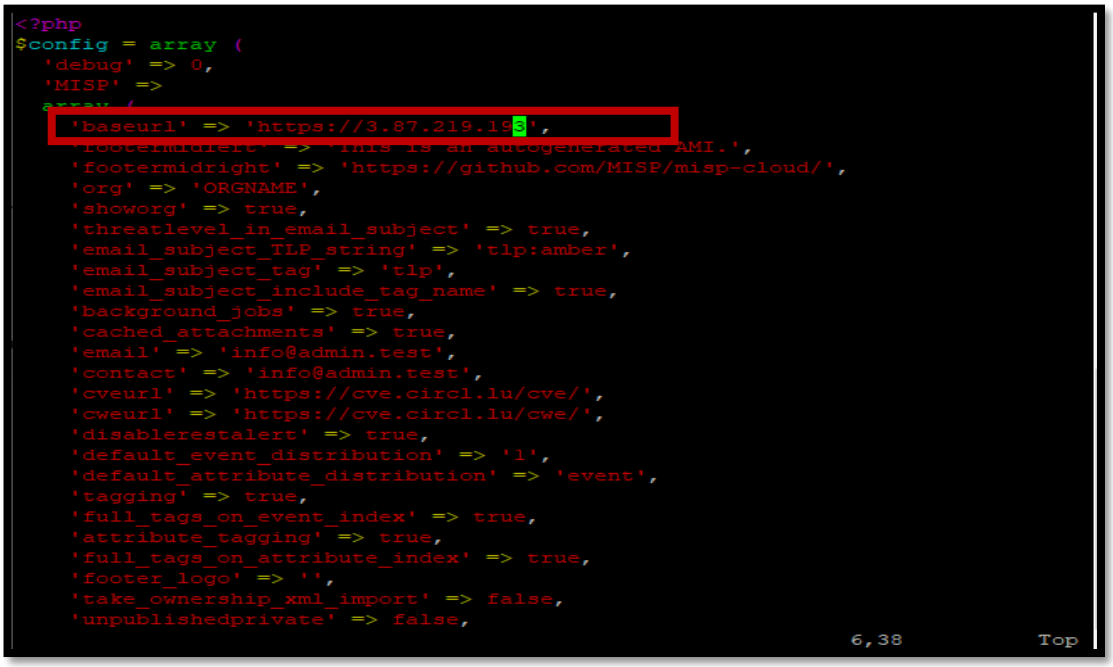


figure 21. Checking base url

2. Synchronizing Errors(server unreachable)

I try to synchronize with another MISP instance by following step of MISP user guide. But failed. Server unreachable error occurred. SSL certification error occurred. To solve this problem, It is important to check "self Singed" box. I sovled connetion problem and check the Other student's instance.

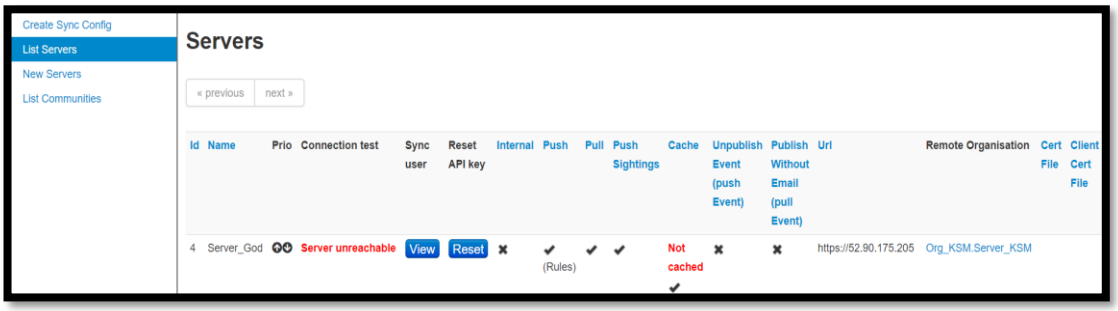


Figure 22. Server unreachable error

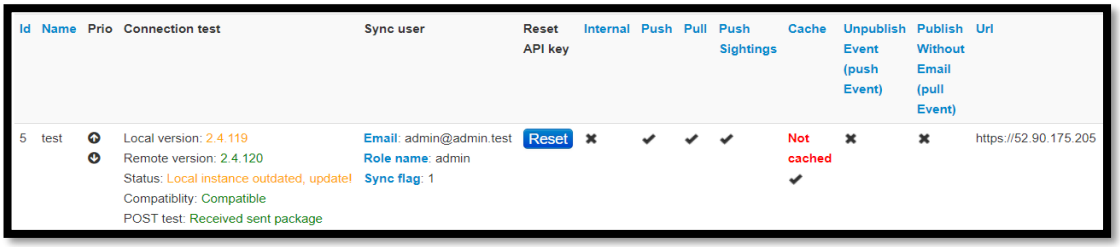


Figure 23. Check Server Connection

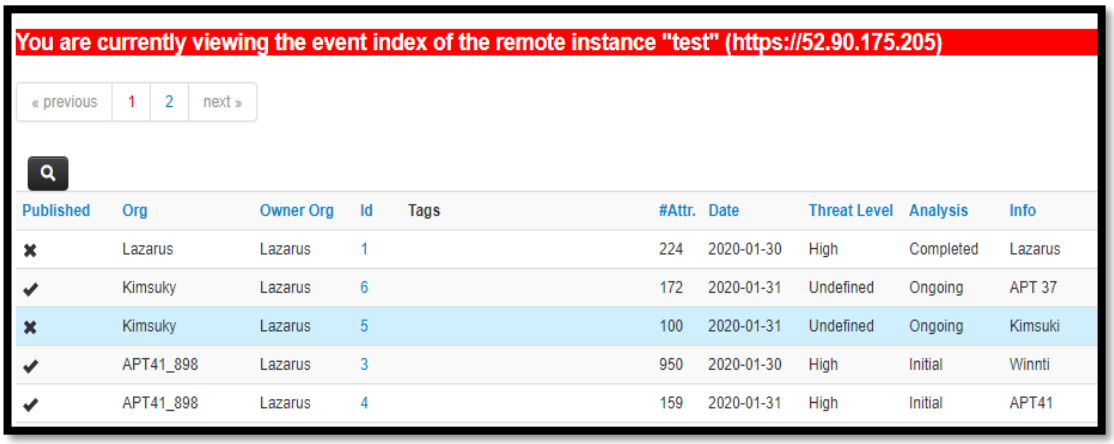


Figure 24. Check Other instance's Events

You can add Other instance's events to Sharing Group in Global actions -> Add Sharing Group

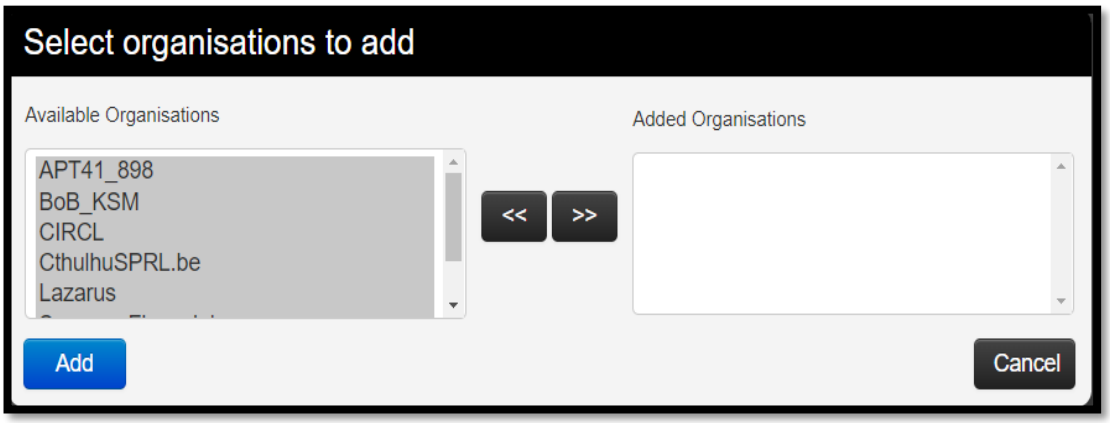


Figure 25. Select Sharing groups

3. API misp_verificert error

To use PyMISP, should set information of url, Authkey, verificert. First, I set the verificer "True" because it related to security problems, and user guide set "ture" but the error occurred. I solve this problem by change setting "True" to "False."

```
[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate (_ssl.c:1076)'))
```

Figure 26. PyMISP SSL Certificate error

```
keys.py > ...
1   misp_url = "https://3.87.219.193/"
2   misp_key = '93RltTeyNXmvTKYgscaLEG5AGYZkRhVqLhQTYUY1'
3   misp_verifycert = False
```

Figure 27. Change verifycert option to "False"

Ⅷ. Else

1. Feed

Feeds are remote or local resources containg indicatiors that can be automatically imported in MISP at regular intervals. You can check or add Feeds in "Sync Actions -> Feeds".

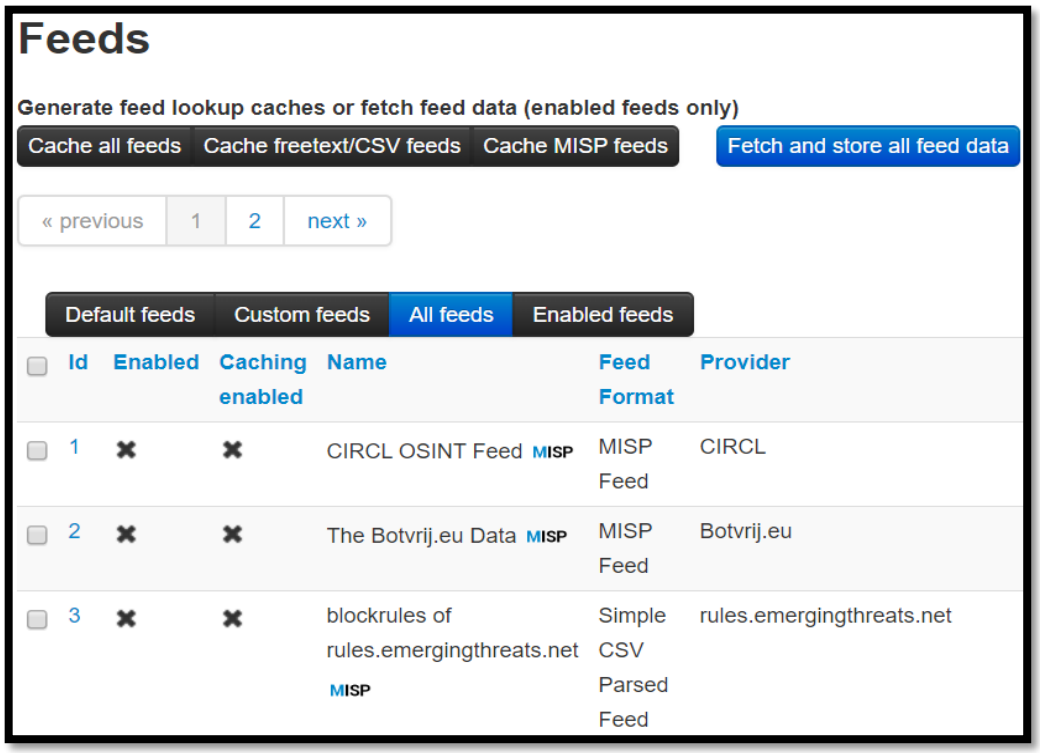


Figure 28. Feeds

2. MISP Communities

MISP is an open source software and it is also a large community of MISP users creating, maintaining and operating communities of users or organizations sharing information about threats or cyber security indicators worldwide. The MISP project doesn't maintain an exhaustive list of all communities relying on MISP especially that some communities use MISP internally or privately.

Figure 29. Misp community