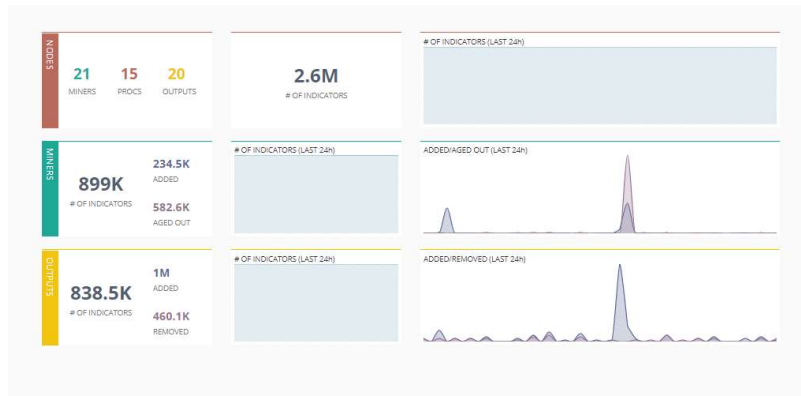


How to Integrate Minemeld with FortiSIEM

What is minemeld?

Minemeld is an IOC Aggregator, its used for EDL (External Dynamic List), it will collect IOC(s) from OTX, FortiGuard, join together and reduce duplicate entry's, then EDL will distribute to SIEM, Firewall, etc. the output can be in STIXX/TAXI, CSV, etc..



What is a Miner?

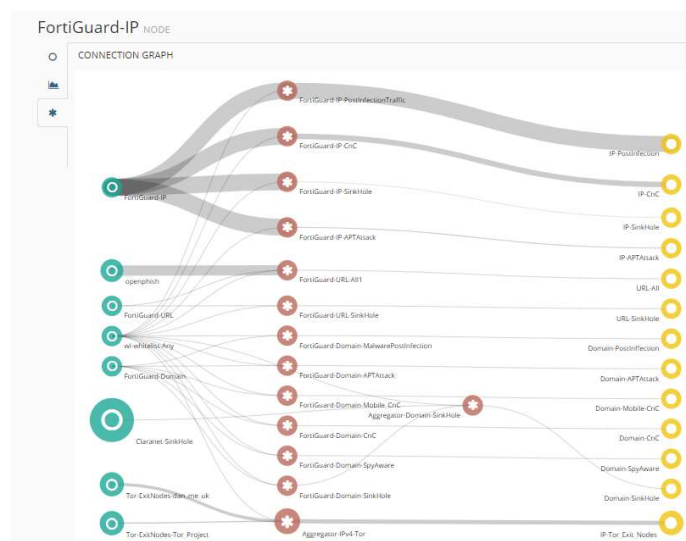
Miner is the process that will collect IOC(s)

What is a Node?

Node will process the Miner IOC(s), put a confidence and correlate.

What is an Output?

The output will generate a “Feed”, in CSV, STIXX/TAXI, the feed can be Public or Private.



How to Push FortiGuard IOC, for separate the threats?

We will Push IOC(s) from the Fortiguard, for then separate the indicator for Threat Type, like APT Attack, Malware Post Infection.

Extract IOC(s) form PostgreSQL

To extract the IOC(s) from FortiSIEM, make an SSH Session to the Super and run this Command, and then save in CSV in `/var/www/html/xxxxx.csv`

Fortiguard IP

```
psql -U phoenix -d phoenixdb -c "Copy (Select malware_type, high_ip,
description, last_seen from ph_malware_ip WHERE group_id = '500620') To
STDOUT With CSV HEADER DELIMITER ',';" >
/var/www/html/fortiguard_malware_ip.csv
```

Fortiguard Domains

```
psql -U phoenix -d phoenixdb -c "Copy (Select malware_type, domain_name,
description, last_seen from ph_malware_site WHERE group_id = '500611') To
STDOUT With CSV HEADER DELIMITER ',';" >>
/var/www/html/fortiguard_domain.csv
```

Fortiguard URL

```
psql -U phoenix -d phoenixdb -c "Copy (Select malware_type, url,
description, last_seen from ph_malware_url WHERE group_id = '500626') To
STDOUT With CSV HEADER DELIMITER ',';" >> /var/www/html/fortiguard_url.csv
```

Special Notes:

- Before Push the Incidents, make a query to the Database to collect `"group_id"` from your instance. Use the bellow command:

```
psql -U phoenix -d phoenixdb
Select * from from ph_malware_ip;
```

Check for the group_id, of na IOC from FortiGuard from GUI (Resources)

Don't Forget to add the commands to Crontab.

Create a Miner to consume the FortiGuard IOC

We go to config in Menu -> Clone a Prototype (in this case AlienVault OTX),

NAME	TYPE	INDICATORS	DESCRIPTION
alienvault.reputation MINEMELD CORE TEAM	MINER	IPV4	alienvault: Open Source AlienVault Reputation Data. alienvault.reputation this just catches everything TAGS: OSINT, ShareLevelGreen
minemeldlocal.AlienVaultOTX:	MINER	any	EXPERIMENTAL minemeldlocal: Local prototype library managed via MineMeld WebUI minemeldlocal.AlienVaultOTX: public TAXII feed from Abuse_ch TAGS: OSINT, ShareLevelGreen, ConfidenceHigh, ConfidenceMedium, ConfidenceLow
minemeldlocal.alienvault_reputation	MINER	IPV4	minemeldlocal: Local prototype library managed via MineMeld WebUI minemeldlocal.alienvault_reputation this just catches everything TAGS: OSINT, ShareLevelGreen

We will use this Prototype because the Miner uses the *CLASS minemeld.ft.csv.CSVFT*

alienvault.reputation PROTOTYPE

MINER STABLE

ABOUT alienvault

Open Source AlienVault Reputation Data.
For more details: <http://labs.alienvault.com/labs/index.php/projects/open-source-ip-reputation-portal/download-ip-reputation-database/>

ABOUT alienvault.reputation

this just catches everything

AUTHOR

MineMeld Core Team

CLASS

minemeld.ft.csv.CSVFT

INDICATOR TYPES

IPV4

TAGS

OSINT, ShareLevelGreen

CONFIG

attributes	confidence: 80 share_level: green type: IPV4
delimiter	#
fieldnames	indicator alienvault_reliability alienvault_risk alienvault_type
interval	3600
source_name	alienvault.reputation
url	http://reputation.alienvault.com/reputation.data

Then click on New, because we need to modify some settings, like the *FieldNames, URL, Source and the Delimiter*.

Change the config to this one:

```
age_out:
  default: null
  interval: 3600
  sudden_death: true
attributes:
  confidence: 100
  share_level: green
  type: IPv4
delimiter: ','
fieldnames:
- malware_type
- indicator
- malware_description
interval: 3600
source_name: FortiGuard
url: https://contoso.com/fortiguard_malware_ip.csv
```

*** Caution with the Age Out, put the value as your flavour.**

Change the Name, of Prototype for [FortiGuard-IP](#)

NEW LOCAL PROTOTYPE

NAME: FortiGuard-IP

NODE TYPE: miner

DEVEL STATUS: STABLE

DESCRIPTION: this just catches everything

CLASS: minemeld.ft.csv.CSVFT

INDICATOR TYPES: IPv4, URL, domain

TAGS: OSINT, ShareLevelGreen

CONFIG

```
1 * age_out:
2   default: null
3   interval: 3600
4   sudden_death: true
5 * attributes:
6   confidence: 100
7   share_level: green
8   type: IPv4
9 delimiter: ','
10 fieldnames:
11 - malware_type
12 - indicator
13 - malware_description
14 interval: 3600
15 source_name: FortiGuard
16 url: https://[REDACTED]/fortiguard_malware_ip.csv
17
```

OK CANCEL

Click ok to create the Prototype, then search for **FortiGuard-IP** we need to clone this, to create a miner:

PROTOTYPES

Show 50 entries Search: fortiguard-ip

NAME	TYPE	INDICATORS	DESCRIPTION
minemeldlocal.FortiGuard-IP	MINER	IPv4, URL, domain	minemeldlocal Local prototype library managed via MineMeld WebUI minemeldlocal.FortiGuard-IP this just catches everything

Showing 1 to 1 of 1 entries (filtered from 309 total entries)

Note that every time that you customize any prototype, will have the suffix of **minemeldlocal**

minemeldlocal.FortiGuard-IP PROTOTYPE

MINER STABLE

ABOUT minemeldlocal

Local prototype library managed via MineMeld WebUI

ABOUT minemeldlocal.FortiGuard-IP

this just catches everything

CLASS

minemeld.ft.csv.CSVFT

INDICATOR TYPES

IPv4, URL, domain

TAGS

OSINT, ShareLevelGreen

CONFIG

age_out	default: null interval: 3600 sudden_death: true
attributes	confidence: 100 share_level: green type: IPv4
delimiter	,
fieldnames	malware_type indicator malware_description
interval	3600
source_name	FortiGuard
url	https://[redacted]fortiguard_malware_ip.csv

After clone, **commit** the setting:

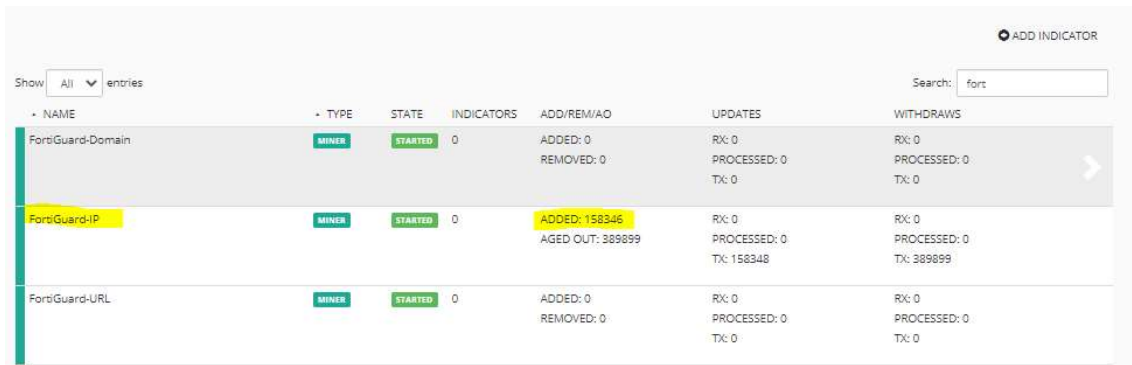
COMMIT

REVERT LOAD IMPORT EXPORT

Search: fortiguard-IP

NAME	TYPE	PROTOTYPE	INPUTS
FortiGuard-Domain	MINER	minemeldlocal.FortiGuard-Domain	None
FortiGuard-IP	MINER	minemeldlocal.FortiGuard-IP	None
FortiGuard-URL	MINER	minemeldlocal.FortiGuard-URL	None

Then wait for the minemeld restart, and then you can see the new miner working on the [nodes](#) section.



The screenshot shows a web interface with a table of miner nodes. At the top right, there is a button labeled "ADD INDICATOR". Below it, there is a search bar with the text "fort". The table has columns for NAME, TYPE, STATE, INDICATORS, ADD/REM/AO, UPDATES, and WITHDRAWS. The 'FortiGuard-IP' row is highlighted in yellow.

NAME	TYPE	STATE	INDICATORS	ADD/REM/AO	UPDATES	WITHDRAWS
FortiGuard-Domain	MINER	STARTED	0	ADDED: 0 REMOVED: 0	RX: 0 PROCESSED: 0 TX: 0	RX: 0 PROCESSED: 0 TX: 0
FortiGuard-IP	MINER	STARTED	0	ADDED: 158346 AGED OUT: 389899	RX: 0 PROCESSED: 0 TX: 158348	RX: 0 PROCESSED: 0 TX: 389899
FortiGuard-URL	MINER	STARTED	0	ADDED: 0 REMOVED: 0	RX: 0 PROCESSED: 0 TX: 0	RX: 0 PROCESSED: 0 TX: 0

Reference:

<https://live.paloaltonetworks.com/t5/minemeld-articles/using-minemeld-to-create-a-custom-miner/ta-p/227694>

Create a Processor to FortiGuard IOC

Now we gone create a processor for splitting the IOC in threat types, we go to the Prototypes, and then search for a **Processor of IPv4**, in this case I will use the **stdlib.aggregatorIPv4Generic** Processor.

The screenshot shows the configuration page for the `stdlib.aggregatorIPv4Generic` prototype. It includes sections for 'ABOUT stdlib', 'ABOUT stdlib.aggregatorIPv4Generic', 'AUTHOR', 'CLASS', 'INDICATOR TYPES', 'TAGS', and 'CONFIG'. The 'CONFIG' section contains a table of infilters and a whitelist_prefixes entry.

NAME	CONDITIONS	ACTIONS
accept withdraws	<code>__method == 'withdraw'</code>	accept
accept IPv4	<code>type == 'IPv4'</code>	accept
drop all		drop

whitelist_prefixes: wl

Create a New Prototype, and change the bellow values:

```
infilters:
- actions:
  - accept
  conditions:
  - __method == 'withdraw'
  name: accept withdraws
- actions:
  - accept
  conditions:
  - type == 'domain'
  - malware_type == 'Malware/APTAttack'
  name: accept domain
- actions:
  - drop
  name: drop all
whitelist_prefixes:
- wl
```

The screenshot shows the 'NEW LOCAL PROTOTYPE' dialog box. The fields are filled with the following values:

- NAME: FortiGuard-APT-Domain
- NODE TYPE: processor
- DEVEL STATUS: STABLE
- DESCRIPTION: Aggregator for domain indicators. Inputs with names starting with "wl" will be interpreted as whitelists.
- CLASS: minimeid.ft.op.AggregateFT
- INDICATOR TYPES: domain
- TAGS: Add tags...
- CONFIG: The same configuration code as shown in the previous block.

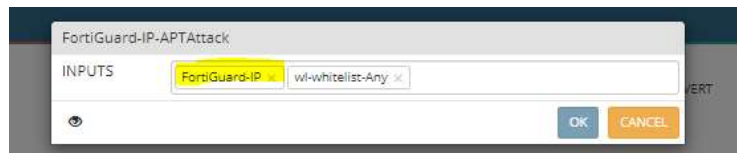
Put the name `Malware-APTAttack_aggregator_IPv4`, then save the Prototype, and clone, Prototype and input the miner:

NAME	TYPE	PROTOTYPE	INPUTS
IP-APTAttack	OUTPUT	stdlib.feedHCGreenWithValue	FortiGuard-IP-APTAttack
IP-CnC	OUTPUT	stdlib.feedHCGreenWithValue	FortiGuard-IP-CnC
IP-PostInfection	OUTPUT	stdlib.feedHCGreenWithValue	FortiGuard-IP-PostInfectionTraffic
IP-SinkHole	OUTPUT	stdlib.feedHCGreenWithValue	FortiGuard-IP-SinkHole
FortiGuard-IP-APTAttack	PROCESSOR	minemeldlocal.Malware-APTAttack_aggregator_IPv4-1	FortiGuard-IP-wl-whitelist-Any
FortiGuard-IP-CnC	PROCESSOR	minemeldlocal.Malware-CnC_aggregator_IPv4	FortiGuard-IP-wl-whitelist-Any

Click on Input and select the:

NAME	TYPE	PROTOTYPE	INPUTS
IP-APTAttack	OUTPUT	stdlib.feedHCGreenWithValue	FortiGuard-IP-APTAttack
IP-CnC	OUTPUT	stdlib.feedHCGreenWithValue	FortiGuard-IP-CnC
IP-PostInfection	OUTPUT	stdlib.feedHCGreenWithValue	FortiGuard-IP-PostInfectionTraffic
IP-SinkHole	OUTPUT	stdlib.feedHCGreenWithValue	FortiGuard-IP-SinkHole
FortiGuard-IP-APTAttack	PROCESSOR	minemeldlocal.Malware-APTAttack_aggregator_IPv4-1	FortiGuard-IP-wl-whitelist-Any
FortiGuard-IP-CnC	PROCESSOR	minemeldlocal.Malware-CnC_aggregator_IPv4	FortiGuard-IP-wl-whitelist-Any

Then Input will be the Miner `FortiGuard-IP`



Then `commit` the changes, and you will see the connection on the Nodes.

PROPERTY	VALUE
CLASS	minemeld.ft.ipop.AggregatorIPv4FT
PROTOTYPE	minemeldlocal.Malware-APTAttack_aggregator_IPv4-1
STATE	STARTED
# INDICATORS	0
OUTPUT	ENABLED
INPUTS	FortiGuard-IP wl-whitelist-Any

Create a Output to FortiGuard IOC

Now we gone create an Output Feed for consuming the feed FortiSIEM, in CSV mode.

Go to the Prototypes, and search for `stdlib.feedGreenWithValue`, note that the Output is in TLP:Green, click on [Clone](#) and change the Name to `IP-APTAttack`, and select the Input the Processor `FortiGuard-IP-APTAttack`

NAME	TYPE	PROTOTYPE	INPUTS
IP-APTAttack	OUTPUT	stdlib.feedHCGreenWithValue	FortiGuard-IP-APTAttack
IP-CnC	OUTPUT	stdlib.feedHCGreenWithValue	FortiGuard-IP-CnC
IP-PostInfection	OUTPUT	stdlib.feedHCGreenWithValue	FortiGuard-IP-PostInfectionTraffic
IP-SinkHole	OUTPUT	stdlib.feedHCGreenWithValue	FortiGuard-IP-SinkHole
FortiGuard-IP-APTAttack	PROCESSOR	minemeldlocal.Malware-APTAttack_aggregator_IPv4-1	FortiGuard-IP-wl-whitelist-Any
FortiGuard-IP-CnC	PROCESSOR	minemeldlocal.Malware-CnC_aggregator_IPv4	FortiGuard-IP-wl-whitelist-Any
FortiGuard-IP-PostInfectionTraffic	PROCESSOR	minemeldlocal.Malware-PostInfectionTraffic_aggregator_IPv4	FortiGuard-IP-wl-whitelist-Any
FortiGuard-IP-SinkHole	PROCESSOR	minemeldlocal.Malware-Sinkhole_aggregator_IPv4	FortiGuard-IP-wl-whitelist-Any

Commit the Changes, and the go to Nodes, search for IP-APTAttack, on the Feed Base URL

IP-APTAttack NODE

STATUS

CLASS: minemeld.ft.redis.RedisSet

PROTOTYPE: stdlib.feedHCGreenWithValue

STATE: **STARTED**

FEED BASE URL: https://[redacted]/feeds/IP-APTAttack

TAGS

INDICATORS: 0

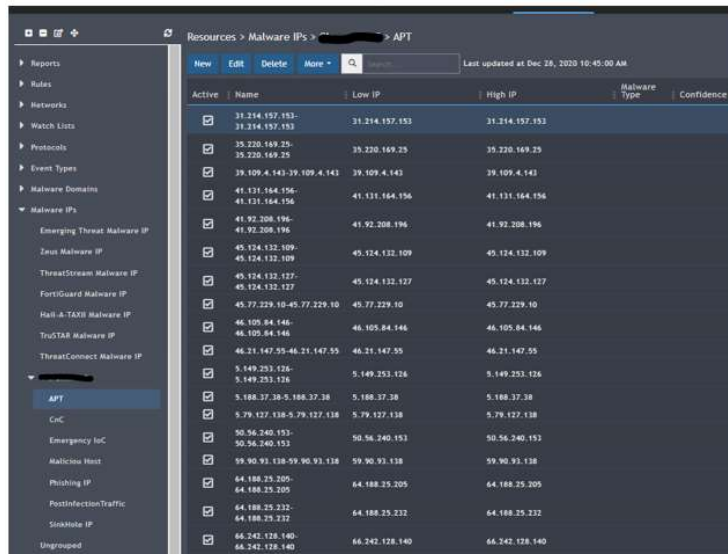
OUTPUT: **DISABLED**

INPUTS: FortiGuard-IP-APTAttack

The list:

```
← → ↻ 🌐 https://contoso.com/feeds/IP-APTAttack  
117.199.10.14-117.199.10.14  
117.2.139.117-117.2.139.117  
117.206.4.236-117.206.4.236  
117.215.4.29-117.215.4.29  
117.216.67.123-117.216.67.123  
117.239.241.2-117.239.241.2  
117.242.253.163-117.242.253.163  
117.248.60.13-117.248.60.13  
118.168.232.142-118.168.232.142  
118.200.151.113-118.200.151.113  
118.31.48.220-118.31.48.220  
118.69.70.109-118.69.70.109
```

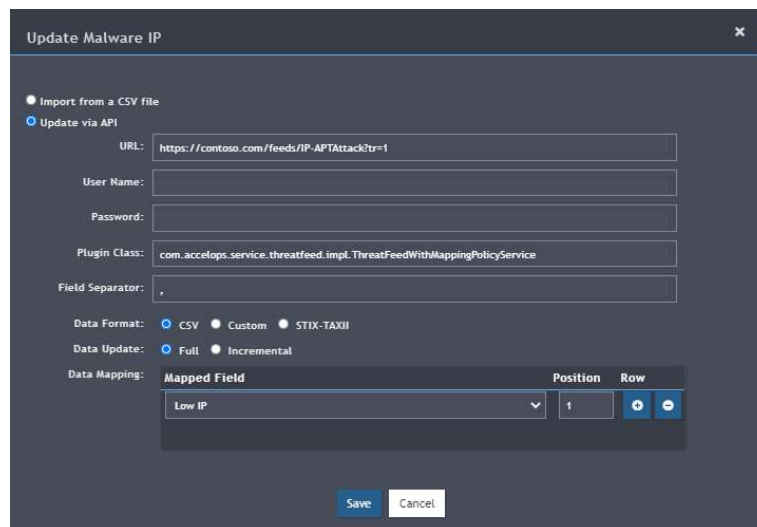
Clone the link **Feed Base URL** <https://contoso.com/feeds/IP-APTAttack>, go to the FortiSIEM, resources.create a New List.



The screenshot shows the FortiSIEM interface for a list of Malware IPs. The left sidebar contains a navigation menu with categories like Reports, Rules, Networks, Watch Lists, Protocols, Event Types, Malware Domains, and Malware IPs. The 'Malware IPs' section is expanded, showing various sub-categories such as Emerging Threat Malware IP, Zeus Malware IP, ThreatStream Malware IP, FortiGuard Malware IP, Hail-A-TAXII Malware IP, TruSTAR Malware IP, ThreatConnect Malware IP, and the selected 'APT' category. The main panel displays a table of IP ranges with columns for Active status, Name, Low IP, High IP, Malware Type, and Confidence. The table contains 15 rows of IP ranges, each with a checkbox in the Active column.

Active	Name	Low IP	High IP	Malware Type	Confidence
<input checked="" type="checkbox"/>		31.214.197.153-31.214.197.153	31.214.197.153		
<input checked="" type="checkbox"/>		35.220.169.25-35.220.169.25	35.220.169.25		
<input checked="" type="checkbox"/>		39.109.4.143-39.109.4.143	39.109.4.143		
<input checked="" type="checkbox"/>		41.131.164.156-41.131.164.156	41.131.164.156		
<input checked="" type="checkbox"/>		41.92.208.196-41.92.208.196	41.92.208.196		
<input checked="" type="checkbox"/>		45.124.132.109-45.124.132.109	45.124.132.109		
<input checked="" type="checkbox"/>		45.124.132.127-45.124.132.127	45.124.132.127		
<input checked="" type="checkbox"/>		45.77.229.10-45.77.229.10	45.77.229.10		
<input checked="" type="checkbox"/>		46.105.84.146-46.105.84.146	46.105.84.146		
<input checked="" type="checkbox"/>		46.21.147.55-46.21.147.55	46.21.147.55		
<input checked="" type="checkbox"/>		5.149.253.126-5.149.253.126	5.149.253.126		
<input checked="" type="checkbox"/>		5.188.37.38-5.188.37.38	5.188.37.38		
<input checked="" type="checkbox"/>		5.79.127.138-5.79.127.138	5.79.127.138		
<input checked="" type="checkbox"/>		50.56.240.153-50.56.240.153	50.56.240.153		
<input checked="" type="checkbox"/>		59.90.93.138-59.90.93.138	59.90.93.138		
<input checked="" type="checkbox"/>		64.188.25.205-64.188.25.205	64.188.25.205		
<input checked="" type="checkbox"/>		64.188.25.232-64.188.25.232	64.188.25.232		
<input checked="" type="checkbox"/>		66.242.128.140-66.242.128.140	66.242.128.140		

Then Update the Feed as your flavour:



The screenshot shows the 'Update Malware IP' dialog box in FortiSIEM. It has two radio buttons: 'Import from a CSV file' (unselected) and 'Update via API' (selected). The 'Update via API' section contains the following fields: URL (https://contoso.com/feeds/IP-APTAttack?tr=1), User Name (empty), Password (empty), Plugin Class (com.accelops.service.threatfeed.impl.ThreatFeedWithMappingPolicyService), and Field Separator (empty). Below these are three radio buttons for 'Data Format': CSV (selected), Custom, and STIX-TAXII. There are also two radio buttons for 'Data Update': Full (selected) and Incremental. At the bottom, there is a 'Data Mapping' table with columns 'Mapped Field', 'Position', and 'Row'. The table has one row with 'Low IP' in the 'Mapped Field' column, '1' in the 'Position' column, and a '+' icon in the 'Row' column. At the bottom of the dialog are 'Save' and 'Cancel' buttons.

Special Notes: As you can see we are passing in the URI the `?tr=1`, this is because some time the Lists of the IP is 8.8.8.8-8.8.8.8 and with this option we are telling the minemeld to generate in CSV the IP in 8.8.8.8.

References:

<https://live.paloaltonetworks.com/t5/minemeld-articles/parameters-for-the-output-feeds/ta-p/146170>

Prototype Examples:

I put here some Prototypes examples, for Tor Browser, DNS Shields (For DNS Firewall):

Name of Prototype: stdlib.listIPv4Generic ([For Emergency IOC](#))

Class: minemeld.ft.local.YamlIPv4FT

```
age_out:
  default: null
  interval: 67
  sudden_death: true
attributes:
  confidence: 100
  share_level: red
interval: 3600
```

Here I have created an Emergency Feed, that will add IOC, in case of an Emergency (Ransomware), Malware, or for Threat Hunting that we have done from FortiSIEM.

Name of Prototype: TorExitNodes-dan_me_uk

Class: minemeld.ft.http.HttpFT

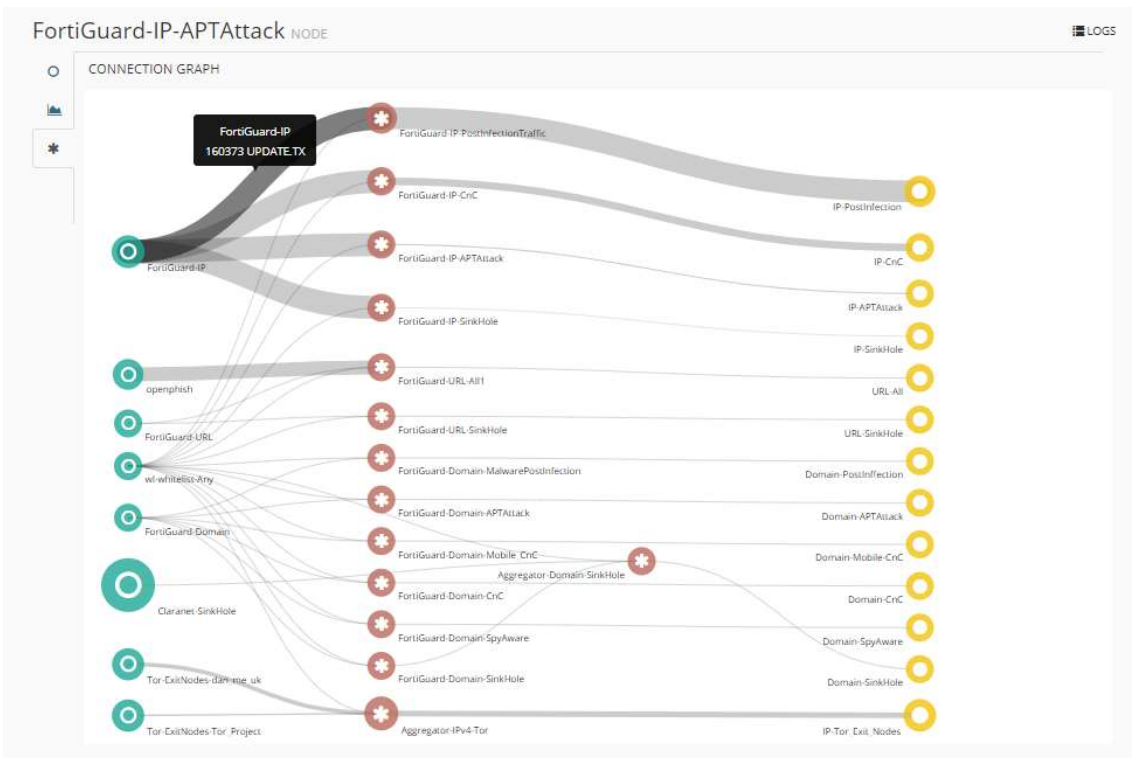
```
age_out:
  default: null
  interval: 3600
  sudden_death: true
attributes:
  confidence: 100
  direction: inbound
  share_level: green
  type: IPv4
indicator:
  regex: ^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$
source_name: itcertpa.IP
url: https://www.dan.me.uk/torlist/?exit
```

Name of Prototype:.DNS-Shield-Porn

Class: minemeld.ft.http.HttpFT

```
age_out:  
  default: null  
  interval: 3600  
  sudden_death: true  
attributes:  
  confidence: 100  
  direction: inbound  
  share_level: green  
  type: domain  
indicator:  
  regex: ^.*  
source_name: DNS.Shield.Porn  
url: https://contoso.com/FamilyShield/BL/porn/domains
```

At the End of the Day, my Prototypes are:



Possible Integrations

Office 365 (Send IOC to O365 using graphAPI)

<http://live.paloaltonetworks.com/t5/MineMeld-Articles/Send-IOCs-to-Microsoft-Graph-API-With-MineMeld/ta-p/258540>

Azure Sentinel (Send IOC to AZSentinel)

<https://medium.com/@antonio.formato/azure-sentinel-minemeld-bring-your-own-threat-intelligence-feeds-7e2f622d6c66>