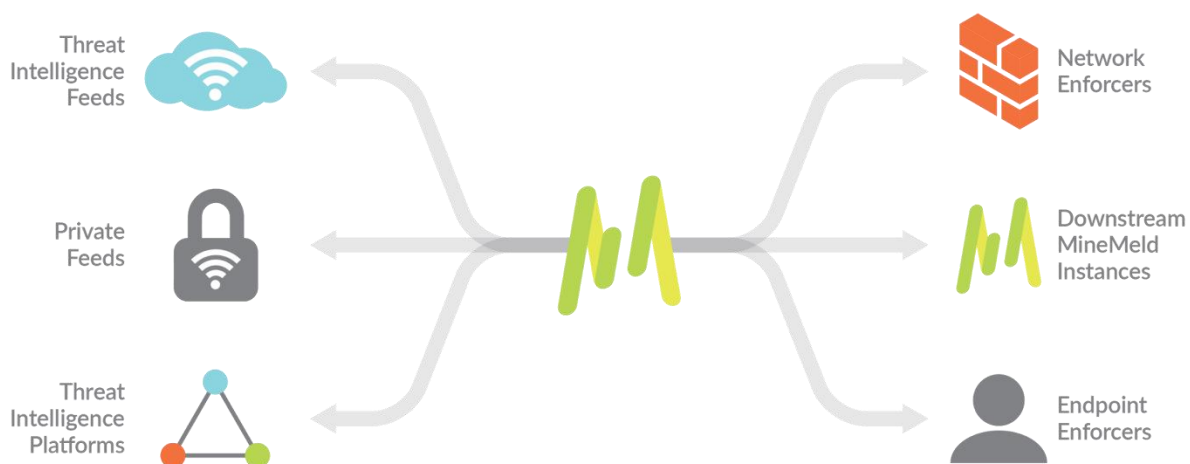


Minemeld User Guide

Minemeld Overview	2
Minemeld Installation Options	4
Virtual Machine Setup Recommendations (minemeld-ansible)	5
Installing Minemeld via minemeld-ansible	8
Minemeld Web Interface Components	9
Changing Admin Credentials	10
Dashboard	11
Nodes	13
Node Details	14
Node Logs	18
Whitelist Miners, and Adding Whitelist Entries	18
Config	21
Prototype Collection	25
Reviewing Prototype Details	27
Cloning Prototypes into Nodes	28
Creating New Prototypes	31
Enabling expert mode	34
Logs	35
System	35
Creating Backups	36
Restoring backups	37
Third-Party Extensions	38
Logging Out	41
Conclusions	42

Minemeld Overview

The Minemeld (<https://www.paloaltonetworks.com/products/secure-the-network/subscriptions/minemeld>) project is an open-source threat intelligence platform that exists to acquire threat intelligence feeds from public and private sources, perform correlation and aggregation across multiple feeds and blacklists, and output deduplicated threat intelligence data. This deduplicated data can then be consumed by endpoint and/or network security devices for enforcement as necessary.

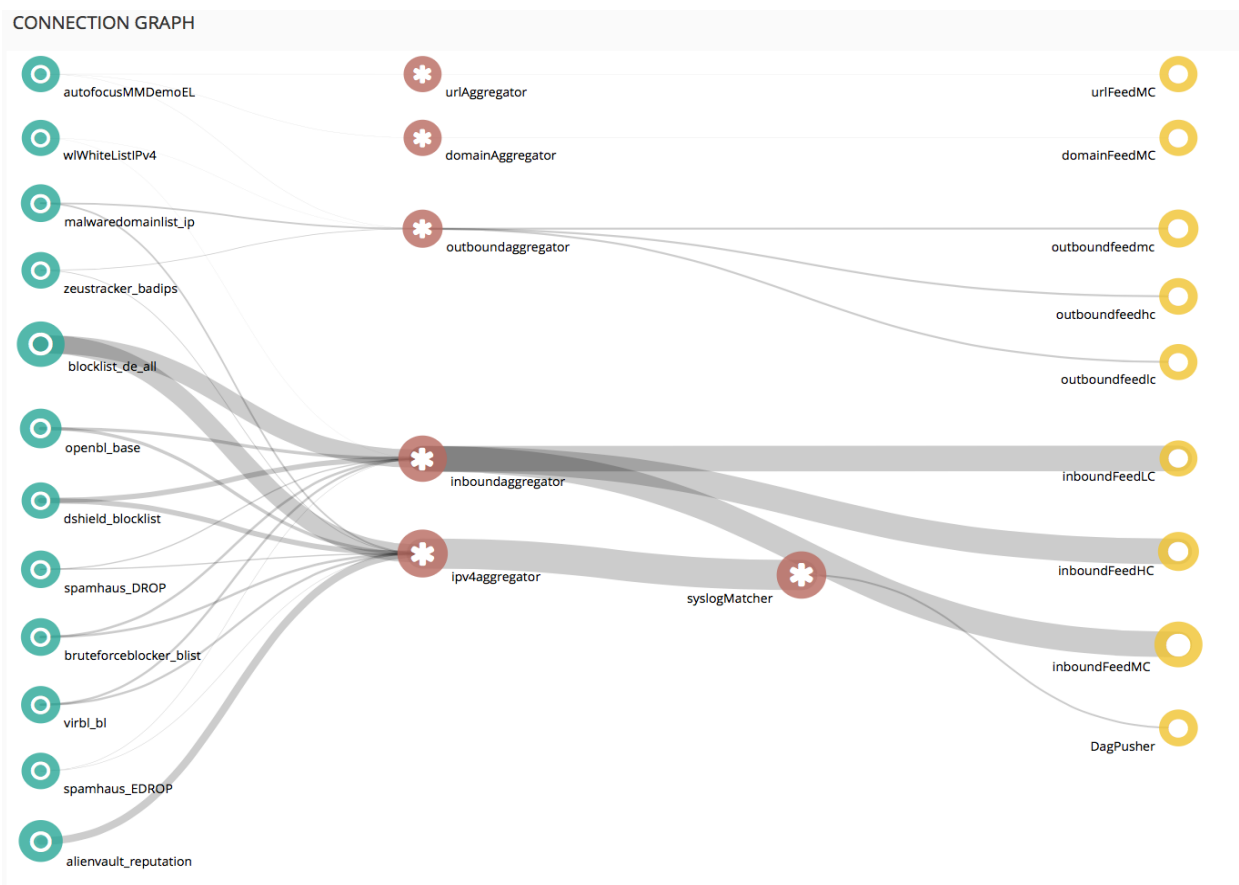


Minemeld operates on a three-step process. First, is the process of consuming logs from a given intelligence source. In minemeld terminology, a process that collects threat intelligence data from a disparate source for consolidation and correlation by the minemeld platform and/or downstream devices is called a miner.

By default, minemeld has several pre-built miners that will collect threat intelligence data from a variety of sources. If a miner doesn't exist in minemeld by default to harvest a particular type of data, or data from a particular source there is a good chance that someone in the open-source community may have developed a miner, and that it can be acquired on [github.com](https://github.com/PaloAltoNetworks/minemeld-misp) as third party extension. An example of this would be the MISP mining module (<https://github.com/PaloAltoNetworks/minemeld-misp>).

The second step of the process is selecting a processor to analyze the mined data for a specific data type, aggregating that data type from miners, then deduplicating it. The processor is where users can choose the type of data or IOC that to aggregate and/or deduplicate. Each processor can handle MULTIPLE miners, and a single miner can send its data to multiple processors for different data types. that is the point of the aggregators -- to collect threat intel data from multiple sources and deduplicate data of the specific type specified, such as URLs, file hashes, domains, ipv4 addresses, etc.

The third step of the process involves configuring outputs. Processors send the aggregated and deduplicated data of a particular type to an output for consumption by downstream products and/or analysts. There are a variety of data types that can be utilized to output data for consumption. In addition to configuring output formats, it is here where data can be output specified on the confidence in the data (e.g. High, Medium and/or Low confidence data), as well as by the classification of the data source(s) by the Traffic Light Protocol (TLP). This graphic may provide additional understanding in how the miners, processors and outputs all fit together.



The miners are denoted in blue, processors are denoted in red, and outputs are denoted in yellow. Notice how there are lines from the miners to the aggregators to the output feeds to indicate where data is coming from, and where it is going. Each miner has a label to indicate where it is collecting its data from, while each processor is labeled with the type of IOC or data it

analyzes from a miner, and last but not least, each output is labeled with the type of IOC data it provides, while most also have the notation "HC, LC, and MC" -- which stand for High Confidence, Low Confidence, and Medium Confidence indicators, respectively.

To make a long story short, minemeld is threat feed aggregator. Think of it as RSS, except for threat intel. Data is gathered from public and/or private threat intel lists, processed by the type of IOC they are (e.g. IP address, hash, etc.), correlated by confidence level tags assign to the miner, IOCs deduplicated if they appear multiple times on different threat feeds, then the aggregated, cleaned data for consumption by security devices, analysts, etc. in a variety of output formats. Minemeld is simply middleware for collecting a bunch of disparate threat intel feeds and cleaning them up for consumption. Consumption for what purpose is up to the user and/or organization.

Minemeld Installation Options

The recommended minemeld installation method is utilizing Ansible to automate the install process. Navigate to the minemeld-ansible github (<https://github.com/PaloAltoNetworks/minemeld-ansible>) and following the instructions for available for various Linux distributions. This guide will walk users through install minemeld via minemeld-ansible on Ubuntu 16.04.3 LTS.

Virtual Machine Setup Recommendations (minemeld-ansible)

When it comes to sizing physical and/or virtual machines for minemeld, the amount of resources the system will require depends highly on the number of miners/processors/outputs that will be running on the instance. However, per discussion on the Palo Alto community discussion boards for minemeld, the minimum requirements are 1GB of RAM, 1 Virtual CPU, and at least 8GB of disk space (<https://live.paloaltonetworks.com/t5/MineMeld-Discussions/Recommended-MineMeld-System-Requirements/td-p/140542>).

On VMware ESXi , it is recommended to create an Ubuntu 16.04.3 VM with at least 1vCPU, 2GB of RAM, and 80GB of disk space.

▼ Hardware Configuration		
▶ CPU	1 vCPUs	
▶ Memory	2 GB	
▶ Hard disk 1	80 GB	
▶ Network adapter 1	Bridged (Connected)	
▶ Video card	4 MB	
▶ CD/DVD drive 1	ISO [Rogue 1] Linux_and_BSD_ISOs/ubuntu-16.04.3-server-amd64.iso	 Select disc image
▶ Others	Additional Hardware	

Install Ubuntu 16.04.3 with most of the settings, create a default user for administering the system (note: the username for these instructions is trobinson), select guided - use entire disk for the partitioning scheme, configure the system to install automatic updates, and install OpenSSH server for remote access and administration.

[!] Software selection

At the moment, only the core of the system is installed. To tune the system to your needs, you can choose to install one or more of the following predefined collections of software.

Choose software to install:

- ☐ Manual package selection
- ☐ DNS server
- ☐ LAMP server
- ☐ Mail server
- ☐ PostgreSQL database
- ☐ Samba file server
- ☒ standard system utilities
- ☐ Virtual Machine host
- ☒ OpenSSH server

<Continue>

<Tab> moves; <Space> selects; <Enter> activates buttons

On first login, sudo to root (`sudo su -`), and run `apt-get update; apt-get -y dist-upgrade; init 6` to fully update the system, then reboot it.

```
werk minemeld
Using username "trobinson".
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-87-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

100 packages can be updated.
51 updates are security updates.

Last login: Wed Jan 17 12:13:41 2018
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

trobinson@minemeld:~$ sudo su -
[sudo] password for trobinson:
root@minemeld:~# apt-get update; apt-get -y dist-upgrade; init 6
```

After rebooting the system, create a VM snapshot prior to installing minemeld to serve as an emergency fallback.

Take snapshot for minemeld

Name	minemeld - preinstallation baseline
Description	pre-minemeld installation baseline

☐ Snapshot the virtual machine's memory.

☐ Quiesce guest file system (needs VMware tools installed).

Take snapshot Cancel

Installing Minemeld via minemeld-ansible

Refer to <https://github.com/PaloAltoNetworks/minemeld-ansible> for install instructions on Ubuntu 16.04 (or the Linux distro chosen).

Warning: Pay close attention to the “MineMeld version” section of the minemeld-ansible github page. By default, the ansible install method will install the latest, unstable version from git. It is recommended to install the latest stable (master) version of minemeld for use in any sort of production capacity. To do this, either modify the local.yml file included with minemeld-ansible per the instructions on the minemeld-ansible github, or modify the ansible-playbook command used to install minemeld to:

```
ansible-playbook -K -e "minemeld_version=master" -i 127.0.0.1, local.yml
```

In my experience, editing the local.yml file caused more problems than helped, so using the modified ansible-playbook command above is preferred.

Note: Currently, as of 1/17/2018, there is a problem with the ansible-playbook in which a required node.js component used with minemeld isn’t installed and results in failing to install minemeld properly. After downloading minemeld-ansible from github, change directories to minemeld-ansible/roles/minemeld/tasks, and modify the webui.yml file, and add the following lines:

```
- name: npm install lodash
  command: npm i --save lodash._reinterpolate
  chdir="{{webui_repo_directory}}"
  environment:
    NODE_VIRTUAL_ENV: "{{www_venv_directory}}"
    PATH:
      "{{www_venv_directory}}/lib/node_modules/.bin:{{www_venv_directory}}/bin:{{webui_repo_directory}}/node_modules/.bin:{{ ansible_env.PATH }}"
    NODE_PATH: "{{www_venv_directory}}/lib/node_modules"
    NPM_CONFIG_PREFIX: "{{www_venv_directory}}"
    npm_config_prefix: "{{www_venv_directory}}"
```

These lines above MUST be added immediately prior to the line:

```
- name: gulp build
```

For more information, check out issue 31 from the minemeld-ansible github issues collection: <https://github.com/PaloAltoNetworks/minemeld-ansible/issues/31>


```

Cloning into 'minemeld-ansible'...
remote: Counting objects: 850, done.
remote: Total 850 (delta 0), reused 0 (delta 0), pack-reused 850
Receiving objects: 100% (850/850), 101.04 KiB | 0 bytes/s, done.
Resolving deltas: 100% (348/348), done.
Checking connectivity... done.
SUDO password:

PLAY [minemeld playbook] *****

TASK [Gathering Facts] *****
ok: [127.0.0.1]

TASK [infrastructure : debug] *****
ok: [127.0.0.1] => {
    "msg": "Loading vars for Ubuntu 16.04"
}

TASK [infrastructure : include_vars] *****
ok: [127.0.0.1] => (item=/home/trobinson/minemeld-ansible/roles/infrastructure/vars/Ubuntu-16.04.yml)

TASK [infrastructure : create minemeld group] *****
changed: [127.0.0.1]

TASK [infrastructure : include task based on distribution] *****
skipping: [127.0.0.1]

TASK [infrastructure : install packages] *****

```

If the installation process is successful there should be no errors, and a reminder to add the Ubuntu user to the minemeld group (if running minemeld-ansible on Ubuntu). Please note that the usermod command will require sudo in order to perform its task.

```

TASK [minemeld : debug] *****
ok: [127.0.0.1] => {
    "msg": "Remember to add your user to the minemeld group"
}

RUNNING HANDLER [minemeld : restart collectd] *****
changed: [127.0.0.1]

RUNNING HANDLER [minemeld : restart nginx] *****
changed: [127.0.0.1]

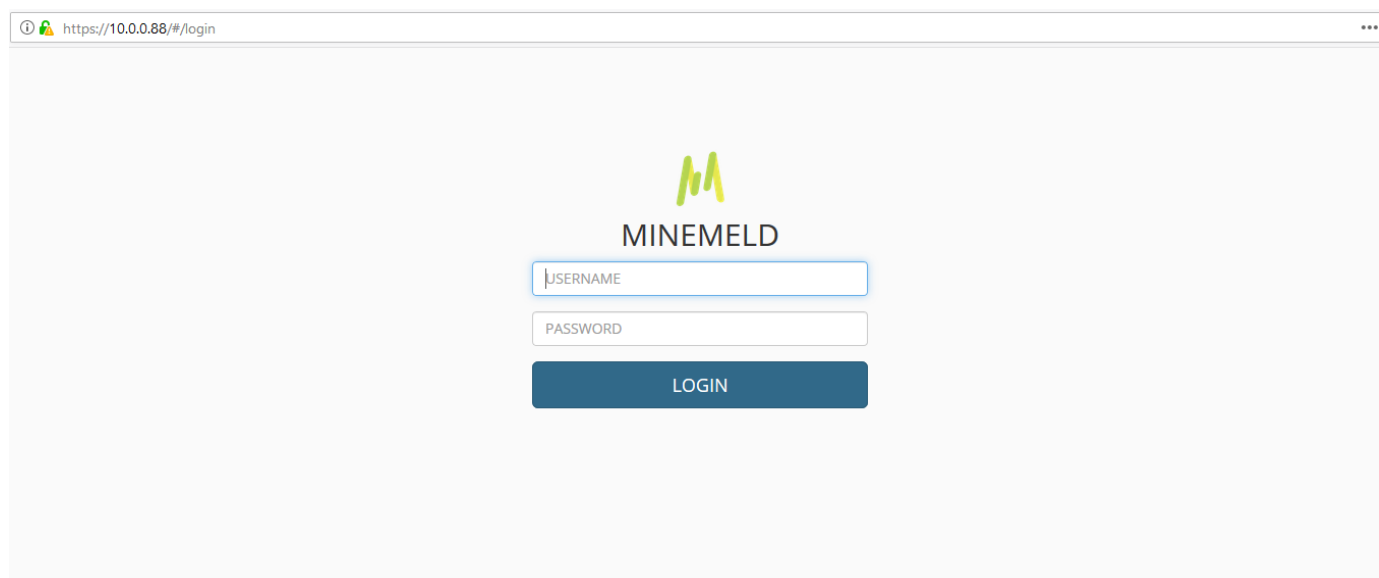
PLAY RECAP *****
127.0.0.1                : ok=76   changed=67   unreachable=0   failed=0

trobinson@minemeld:~/minemeld-ansible$ sudo usermod -a -G minemeld trobinson

```

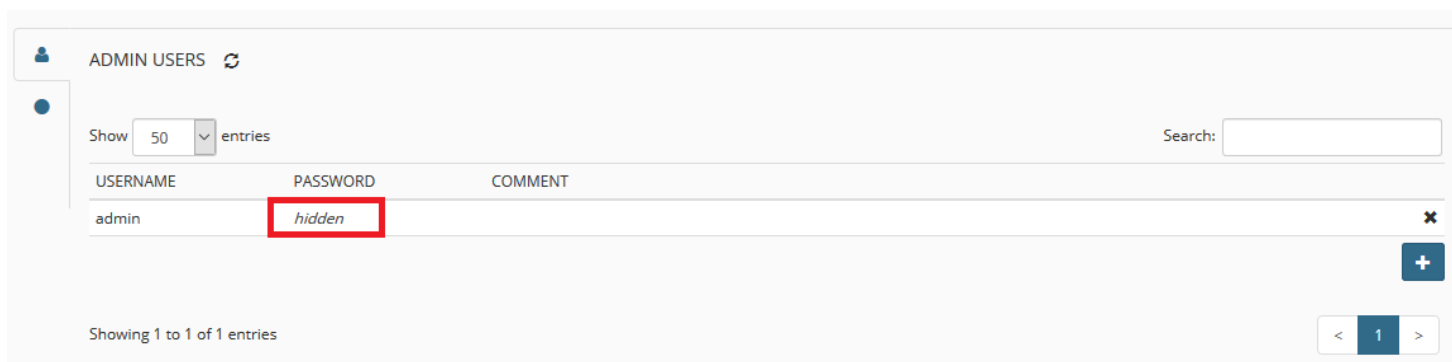
Minemeld Web Interface Components

Once the installation steps have been completed, point a web browser to the IP address of the minemeld VM. For example, the IP address of the VM in the illustration below is 10.0.0.88.



Changing Admin Credentials

When greeted with the minemeld login page, take a moment to take another VM snapshot before continuing, in order to have a snapshot of the minemeld application in its default state. Afterwards, log in to continue. The default credentials are **admin / minemeld**. Best practice dictates that these default credentials be changed immediately. This can be done by clicking the admin user icon in the upper right corner, then clicking on the hidden password field next to the admin user on the admin users page. Enter a new password, then click OK.



USERNAME	PASSWORD	COMMENT
admin	hidden	

SET PASSWORD

USERNAME

admin

PASSWORD

●●●●●●●●●●●●●●●●

REPEAT PASSWORD

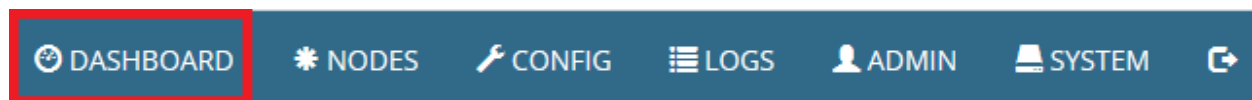
●●●●●●●●●●●●●●●●|

OK

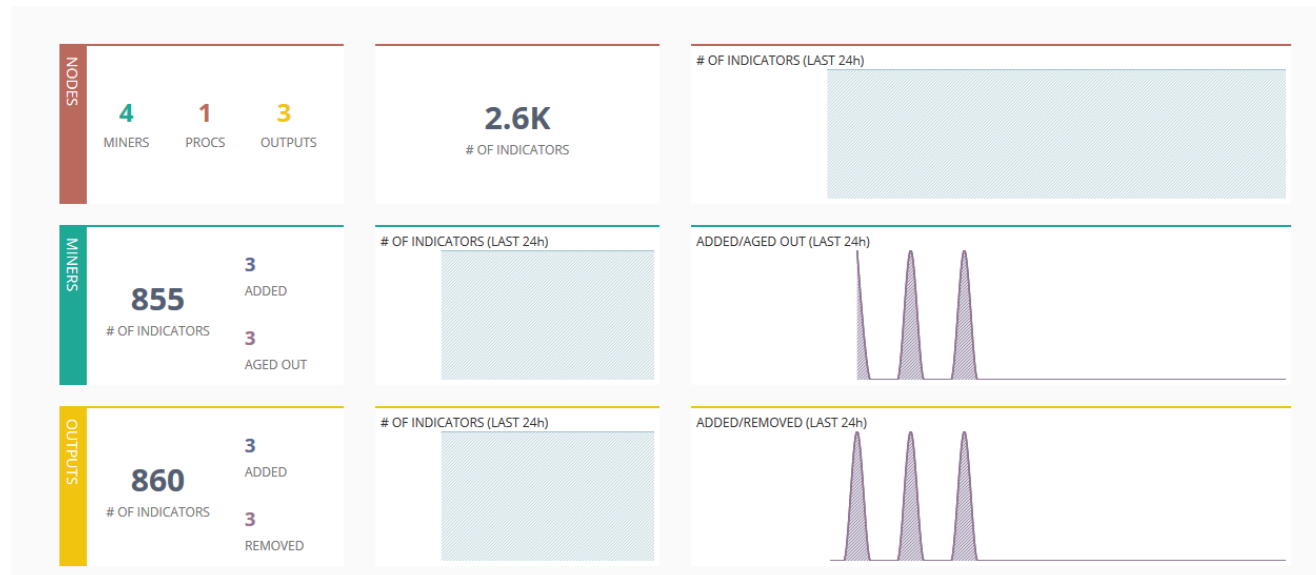
CANCEL

Dashboard

Select the dashboard option in the upper right navigation bar.



This is the dashboard, the landing page for the minemeld web interface upon login.



The dashboard screen is divided into three sections, labeled **NODES**, **MINERS**, and **OUTPUTS**. The purpose of this page is to provide an at a glance view of how many miners, processors, and outputs are currently defined and active, as well as a count of processed indicators.

The **NODES** section is further divided into a count of how many miners, processors and outputs are currently active and processing indicators. Additionally, the nodes section defines how many indicators have been processed by all active nodes -- Miners, Processors, and Output nodes. Note that the same indicators can and will be counted multiple times -- We will be discussing this shortly. The final section in the nodes portion of the dashboard displays a graph that shows how many indicators have been processed in the last 24 hours.

The **MINERS** section shows the number of mined indicators for a 24 hour period, included new and/or “aged out” indicators -- indicators that have been removed from a mined resource, or new entries that have recently been added. Not unlike the **NODES** section, there is a graph that indicates how many indicators have been processed in the last 24 hours, and a graph showing the added and/or aged out indicators in the last 24 hours.

The **OUTPUTS** section provides a count of processed indicators, including a count of added/removed indicators, and just like the **MINERS** section, there is a section that details the number of output indicators in the last 24 hours, as well as the number of added/removed indicators in the same time period.

Nodes

Select the nodes option in the upper right navigation menu.



This will display the nodes page.

ADD INDICATOR

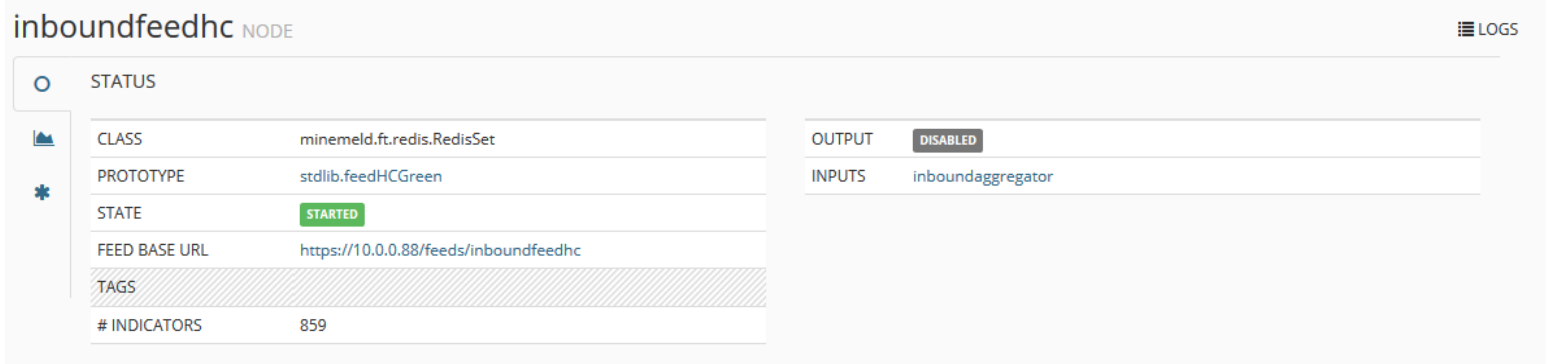
NAME	TYPE	STATE	INDICATORS	ADD/REM/AO	UPDATES	WITHDRAWALS
dshield_blocklist	MINER	STARTED	20	ADDED: 3 AGED OUT: 3	RX: 0 PROCESSED: 0 TX: 354	RX: 0 PROCESSED: 0 TX: 3
spamhaus_DROP	MINER	STARTED	784	ADDED: 0 REMOVED: 0	RX: 0 PROCESSED: 0 TX: 0	RX: 0 PROCESSED: 0 TX: 0
spamhaus_EDROP	MINER	STARTED	50	ADDED: 0 AGED OUT: 1	RX: 0 PROCESSED: 0 TX: 0	RX: 0 PROCESSED: 0 TX: 1
wlWhiteListIPv4	MINER	STARTED	0	ADDED: 0 REMOVED: 0	RX: 0 PROCESSED: 0 TX: 0	RX: 0 PROCESSED: 0 TX: 0
inboundfeedhc	OUTPUT	STARTED	859	ADDED: 3 REMOVED: 4	RX: 354 PROCESSED: 354 TX: 0	RX: 4 PROCESSED: 4 TX: 0
inboundfeedlc	OUTPUT	STARTED	0	ADDED: 0 REMOVED: 0	RX: 354 PROCESSED: 0 TX: 0	RX: 4 PROCESSED: 358 TX: 0
inboundfeedmc	OUTPUT	STARTED	0	ADDED: 0 REMOVED: 0	RX: 354 PROCESSED: 0 TX: 0	RX: 4 PROCESSED: 358 TX: 0
inboundaggregator	PROCESSOR	STARTED	854	ADDED: 3 REMOVED: 4	RX: 354 PROCESSED: 354 TX: 354	RX: 4 PROCESSED: 4 TX: 4

Showing 1 to 8 of 8 entries

This page will display a list of all started and enabled nodes on the minemeld instance, how many indicators have been processed by a particular node, as well as what type of node it is (e.g. miner, processor, or output).

Node Details

Clicking on a node will direct users to a new page with tabs that provide status, statistics , and a connection graph to show where the node gets its data from, and which other nodes the selected node sends its data to. First, lets go over the status tab:



The screenshot shows the 'inboundfeedhc' node details page. The 'STATUS' tab is selected. The page displays the following information:

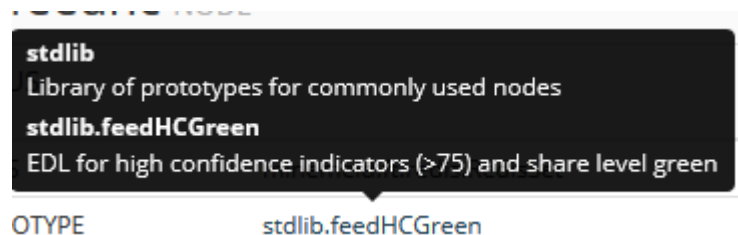
Field	Value
CLASS	minemeld.ft.redis.RedisSet
PROTOTYPE	stdlib.feedHCGreen
STATE	STARTED
FEED BASE URL	https://10.0.0.88/feeds/inboundfeedhc
TAGS	
# INDICATORS	859

On the right side, there are two sections: 'OUTPUT' with a 'DISABLED' button, and 'INPUTS' with the value 'inboundaggregator'.

The status tab features a set of fields, that provides more information about the node that has been selected. “Inboundfeedhc”, has the following fields:

Class - Defines what kind of processing is applied to indicators. In this case “minemeld.ft.redis.RedisSet” is collecting these indicators from a Redis database

Prototype - this is what defines how the node operators and what sort of data it operates on. In our case, the “inboundfeedhc” prototype is “stdlib.feedHCGreen”. Hovering over this text provides a description of the prototype



The screenshot shows a tooltip for the 'stdlib.feedHCGreen' prototype. The tooltip contains the following text:

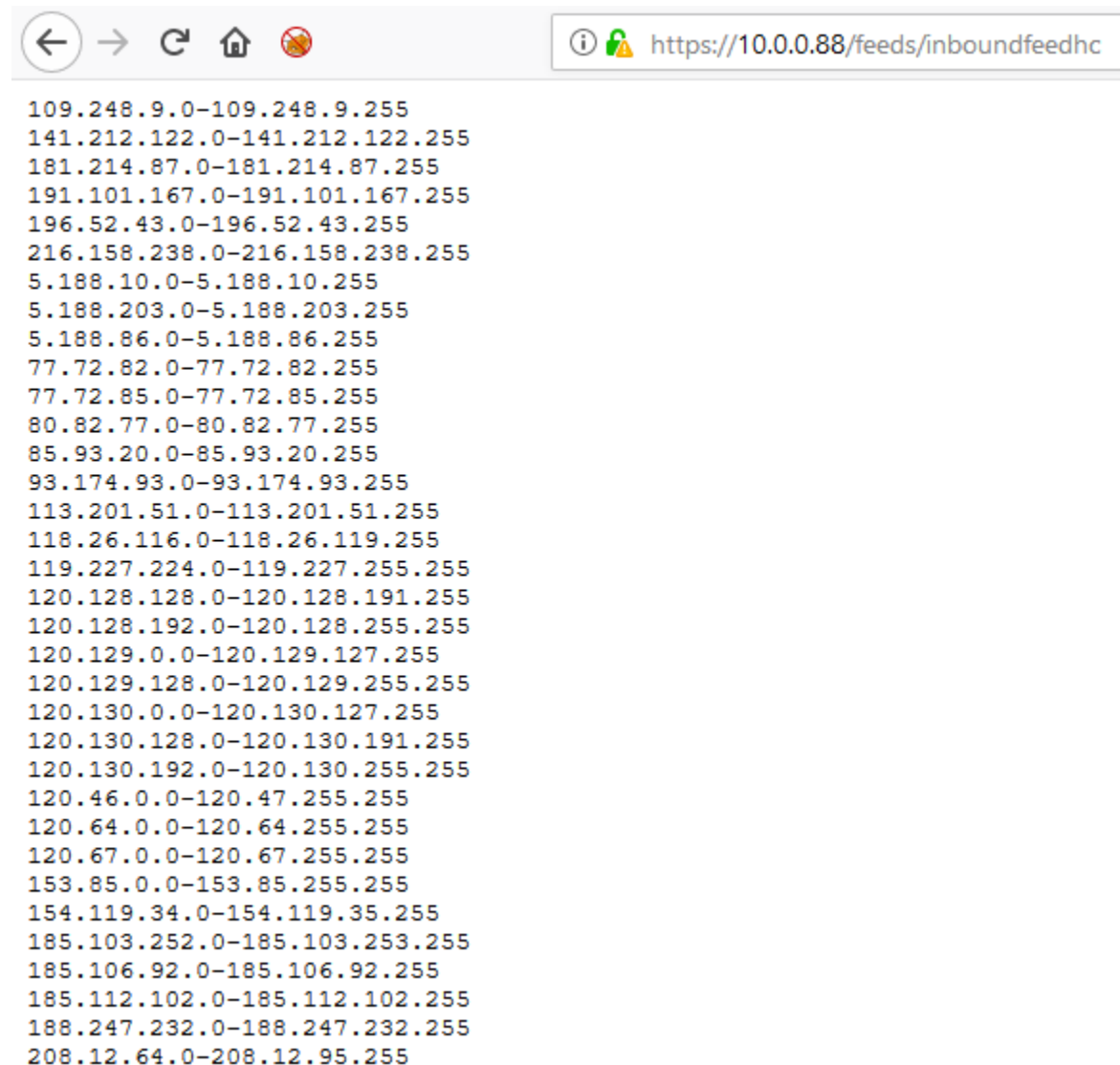
- stdlib**
- Library of prototypes for commonly used nodes
- stdlib.feedHCGreen**
- EDL for high confidence indicators (>75) and share level green

Below the tooltip, the text 'OTYPE' is visible, followed by the link 'stdlib.feedHCGreen'.

Since inboundfeedhc is an output node, this is an output style prototype. The description states that this is a feed for high confidence indicators (indicators/feeds can be assigned confidence ratings from 0-100, with a share level of green (indicators/feeds can also be provided share levels that correspond to the Traffic Light Protocol (TLP), Green indicating that the indicators can be shared freely

State - defines whether or not the node is running

Feed base URL - Since this node is an output feed, and its providing plaintext output, this a url that can be accessed by downstream devices to access the output data. By clicking on the link in the Feed base URL field, the user is provided a plain text feed of IP addresses:



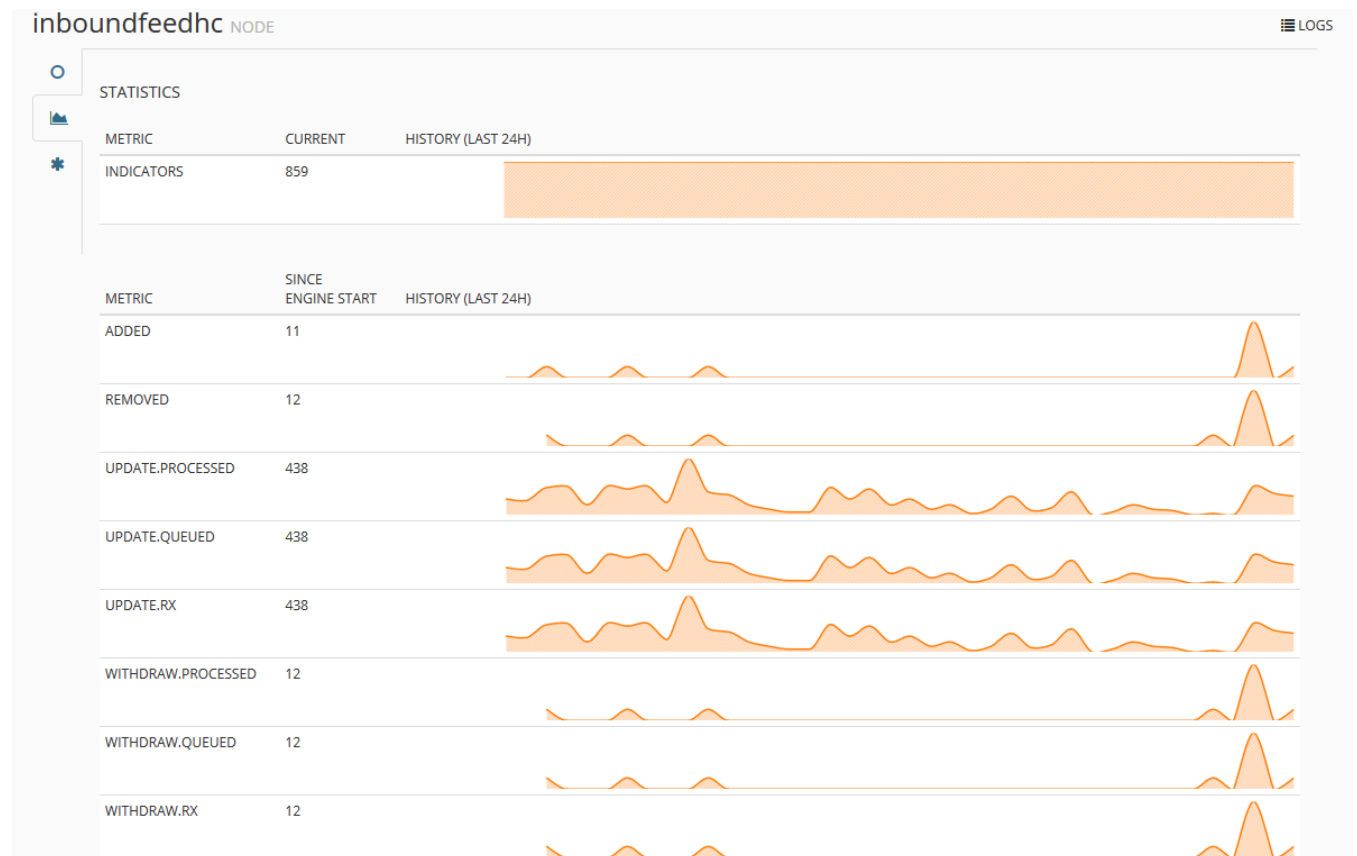
Tags - Metadata tags assigned to the node being inspected

Indicators - number of indicators the node has processed

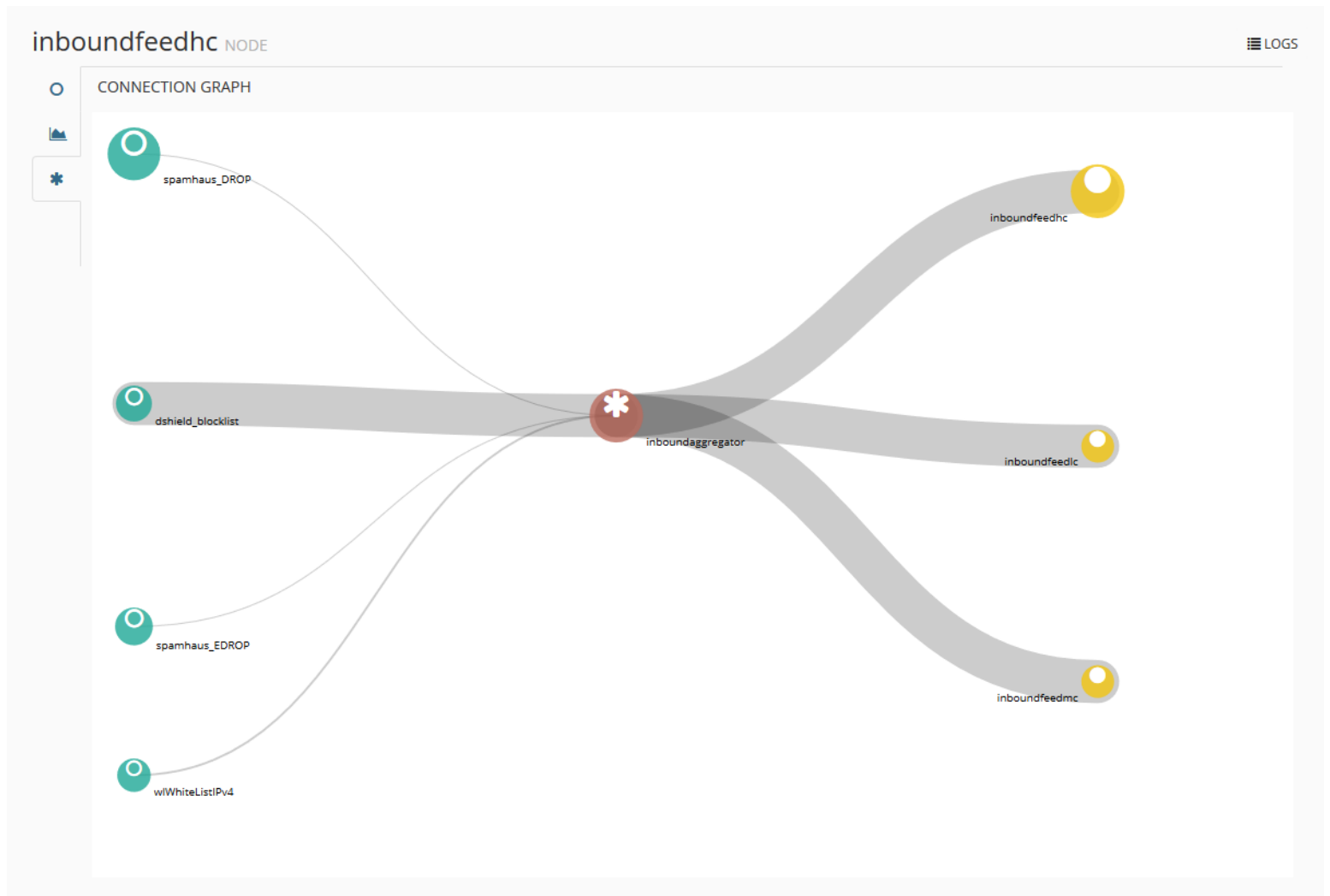
Output - Indicates if this node is outputting its data to another node. Since this node is an output node, the output setting is set to disabled. While, a miner that is sending its data to a processor node, or a processor sending its data to an output node would be seen as enabled

Inputs - defines what other nodes the current node is receiving input from. Note that this means that miners will typically have “none” listed for their inputs, due to the fact that they do not mine their data from another minemeld node

The other tabs include the statistics tab that provides a host of information about the activity the current node is responsible for




And the connection graph tab that shows the nodes and all of their various connections (e.g. what miners connect to what processors, and what processors connect to what outputs, etc.)



Node Logs

In the upper right section of the node details page, that there is a section labeled logs.

 LOGS

clicking on this icon, this will display system logs related to the current node.

source:inboundfeedhc					
LOGS					
Scroll up for latest entries. Or click here					
18/1/2018 12:15:57 -0500	inboundfeedhc	ACCEPT_WITHDRAW	191.96.249.0-191.96.249.255	direction: inbound dshield_name: Digital Energy Technologies Chile SpA, confidence: 100 share_level: green sour...	
18/1/2018 12:15:57 -0500	inboundfeedhc	RECD_WITHDRAW	191.96.249.0-191.96.249.255	direction: inbound dshield_name: Digital Energy Technologies Chile SpA, confidence: 100 share_level: green sourc...	
18/1/2018 12:15:57 -0500	inboundfeedhc	ACCEPT_WITHDRAW	66.118.142.0-66.118.142.255	direction: inbound dshield_name: SAGO NET - Sago Networks, LLC, confidence: 100 share_level: green sources: [...	
18/1/2018 12:15:57 -0500	inboundfeedhc	RECD_WITHDRAW	66.118.142.0-66.118.142.255	direction: inbound dshield_name: SAGO NET - Sago Networks, LLC, confidence: 100 share_level: green sources: [...	
18/1/2018 12:15:57 -0500	inboundfeedhc	ACCEPT_UPDATE	125.212.217.0-125.212.217.255	direction: inbound dshield_name: VIETEL-AS-AP Viettel Corporation, confidence: 100 share_level: green sources: [...	
18/1/2018 12:15:57 -0500	inboundfeedhc	RECD_UPDATE	125.212.217.0-125.212.217.255	direction: inbound dshield_name: VIETEL-AS-AP Viettel Corporation, confidence: 100 share_level: green sources: [...	
18/1/2018 12:15:57 -0500	inboundfeedhc	ACCEPT_UPDATE	185.107.83.0-185.107.83.255	direction: inbound dshield_name: NFORCE, confidence: 100 share_level: green sources: ["dshield.block"] dshield_co...	
18/1/2018 12:15:57 -0500	inboundfeedhc	RECD_UPDATE	185.107.83.0-185.107.83.255	direction: inbound dshield_name: NFORCE, confidence: 100 share_level: green sources: ["dshield.block"] dshield_co...	
18/1/2018 12:15:57 -0500	inboundfeedhc	ACCEPT_UPDATE	71.6.146.0-71.6.146.255	direction: inbound dshield_name: CARINET - CariNet, Inc., confidence: 100 share_level: green sources: ["dshield.block"] ds...	
18/1/2018 12:15:57 -0500	inboundfeedhc	RECD_UPDATE	71.6.146.0-71.6.146.255	direction: inbound dshield_name: CARINET - CariNet, Inc., confidence: 100 share_level: green sources: ["dshield.block"] ds...	
18/1/2018 12:15:57 -0500	inboundfeedhc	ACCEPT_UPDATE	5.188.11.0-5.188.11.255	direction: inbound dshield_name: PIN-AS, confidence: 100 share_level: green sources: ["dshield.block"] dshield_country:...	

These logs can be used to debug different nodes and verify that nodes are sending and receiving data properly.

Whitelist Miners, and Adding Whitelist Entries

Miners starting with “wl” indicate that they are indicator whitelists. Indicators associated with a whitelist miner are indicators that users do not want to see show up in any outputs the miner is associated with. Whitelist miners work almost the same as standard miners, in that they must be associated to a processor, and that processor must be associated to an output to ensure that whitelisted items are removed from a given output. The primary difference is that whitelist miners typically have the prototype, “minemeld.ft.local.*”, meaning that users are expected to manually add indicators to these miners.

Back on the nodes page, the “add indicator” option is available on the upper right of the page.

 ADD INDICATOR

This link can be used to manually specify indicators to add, as well as which miner to add it to.

ADD INDICATOR

INDICATOR

TYPE

SHARE LEVEL

COMMENT

MINERS

Review indicators manually added to whitelist miners by clicking on the whitelist node (by default, the only whitelist node available in minemeld is the “wlWhiteListIPv4” node), and clicking on the “indicators” tab. The blue “+” sign in the lower right area of this tab can also be used to add additional indicators to the whitelist miner as well.

wlWhiteListIPv4

NODE

LOGS

INDICATORS

50 entries

Search:

INDICATOR	DIRECTION	SHARE LEVEL	COMMENT	
255.255.255.255			broadcast address	✕
224.0.0.0/4			multicast	✕
127.0.0.1			loopback is something I don't want to see in my ipv4 blacklists.	✕

Showing 1 to 3 of 3 entries

< 1 >

+

Config

Navigate to the config option in the navigation menu.



COMMIT

REVERT

LOAD

IMPORT

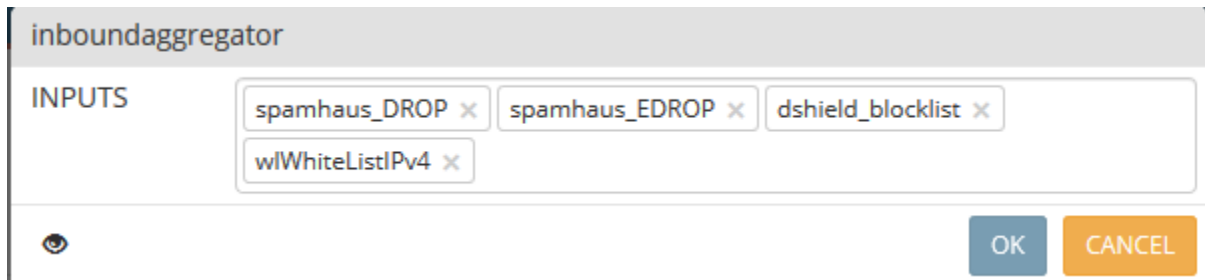
EXPORT

Search:

NAME	TYPE	PROTOTYPE	INPUTS	
dshield_blocklist	MINER	dshield.block	None	✕
spamhaus_DROP	MINER	spamhaus.DROP	None	✕
spamhaus_EDROP	MINER	spamhaus.EDROP	None	✕
wlWhiteListIPv4	MINER	stdlib.listIPv4Generic	None	✕
inboundfeedhc	OUTPUT	stdlib.feedHCGreen	inboundaggregator	✕
inboundfeedlc	OUTPUT	stdlib.feedLCGreen	inboundaggregator	✕
inboundfeedmc	OUTPUT	stdlib.feedMCGreen	inboundaggregator	✕
inboundaggregator	PROCESSOR	stdlib.aggregatorIPv4Inbound	spamhaus_DROP spamhaus_EDROP dshield_blocklist wlWhiteListIPv4	✕

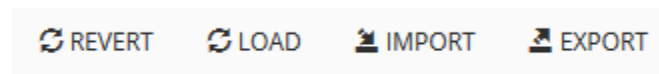
The config page defines all active processors, miners and output nodes on the minemeld instance, what prototypes or data they handle, and what inputs (if any) they accept data from.

Additionally, for nodes that accept input from other nodes (e.g. processors and/or outputs), users can select the inputs column for that node to modify and/or add additional nodes to process input from. For example, if input column from the inboundagggregator row were clicked, users could choose to add additional miners to process indicators from, or remove existing miners.



The screenshot shows a configuration window titled "inboundagggregator". Inside, there is a section labeled "INPUTS" which contains a list of input sources: "spamhaus_DROP", "spamhaus_EDROP", "dshield_blocklist", and "wlWhiteListIPv4". Each item has a small "x" icon to its right, indicating it can be removed. At the bottom right of the window are two buttons: "OK" (blue) and "CANCEL" (orange). A small eye icon is visible at the bottom left of the window.

The node config page also features several options in the upper right of the screen:



Revert: If users chose to add additional nodes and did not commit those changes via the commit button, this reverts the node config to the last running version.

Load: Reloads the current running config.

Export/Import:

The export function allows users to export the node configuration for the minemeld instance by copying the configuration file to the system clipboard, and allowing the data to be saved it to a text file.

EXPORT CANDIDATE CONFIGURATION

CONFIG

1 nodes:

2 spamhaus_EDROP:

3 output: true

4 prototype: spamhaus.EDROP

5 dshield_blocklist:

6 output: true

7 prototype: dshield.block

8 inboundaggregator:

9 inputs:

10 - spamhaus_DROP

11 - spamhaus_EDROP

12 - dshield_blocklist

13 - wlWhitelistIPv4

14 output: true

15 prototype: stdlib.aggregatorIPv4Inbound

16 inboundfeedhc:

17 inputs:

18 - inboundaggregator

19 output: false

20 prototype: stdlib.feedHCGreen

21 spamhaus_DROP:

22 output: true

23 prototype: spamhaus.DROP

24 wlWhitelistIPv4:

25 inputs: []

COPY TO CLIPBOARD

OK

The import function on the other hand, allows users to paste in a backed up configuration file to be applied to the minemeld instance. This configuration can be Appended to the current configuration, or Replace the current configuration entirely.

IMPORT CANDIDATE CONFIGURATION

CONFIG

1

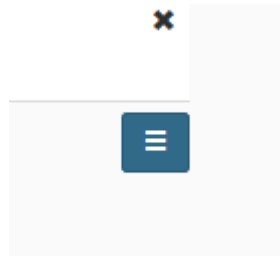
APPEND







REPLACE

CANCEL

Prototype Collection


The icon in the lower right corner of the screen is used to access the list of available prototypes:



PROTOTYPES			
Show <input type="text" value="50"/> entries	Search: <input type="text"/>		
NAME	TYPE	INDICATORS	DESCRIPTION
 ETOpen.blockIPs <small>GREGORY ROEHL (PALOALTONETWORKS.COM)</small>	MINER	IPv4	ETOpen Emerging Threats Open rulesets ETOpen.blockIPs Raw IPs for the firewall block lists. These come from Spam nets identified by Spamhaus (www.spamhaus.org), Top Attackers listed by DShield (www.dshield.org), Abuse.ch. TAGS ConfidenceMedium ShareLevelGreen OSINT
 ETOpen.compromisedIPs <small>VICTOR BARAHONA (UAM.ES)</small>	MINER	IPv4	ETOpen Emerging Threats Open rulesets ETOpen.compromisedIPs This ruleset is compiled from a number of sources. It's contents are hosts that are known to be compromised by bots, phishing sites, etc, or known to be spewing hostile traffic. These are not your everyday infected and sending a bit of spam hosts, these are significantly infected and hostile hosts. TAGS ConfidenceMedium ShareLevelGreen OSINT
 alienvault.reputation <small>MINEMELD CORE TEAM</small>	MINER	IPv4	alienvault Open Source AlienVault Reputation Data. alienvault.reputation this just catches everything TAGS OSINT ShareLevelGreen
 anomali.opticAPI <small>MINEMELD CORE TEAM</small>	MINER	domain URL IPv4 IPv6	EXPERIMENTAL anomali Anomali delivers earlier detection and identification of adversaries in your organizations network by making it possible to correlate tens of millions of threat indicators against your real time network activity logs and up to a year or more of forensic log data. Anomali's approach enables detection at every point along the kill chain, making it possible to mitigate threats before material damage to your organization has occurred. anomali.opticAPI Miner for Anomali Optic API. You need a valid Optic API Key to use this Miner. TAGS ConfidenceHigh ConfidenceLow ConfidenceMedium ShareLevelRed
 auscert.1day_combo <small>SIMON COGGINS</small>	MINER	URL	auscert AusCERT is a leading Cyber Emergency Response Team (CERT) in Australia and the Asia/Pacific region. auscert.1day_combo 1 day combo TAGS ConfidenceHigh ShareLevelRed
 auscert.1day_dumpsites <small>MINEMELD CORE TEAM</small>	MINER	URL	auscert AusCERT is a leading Cyber Emergency Response Team (CERT) in Australia and the Asia/Pacific region. auscert.1day_dumpsites Do not use ! TAGS Deprecated ShareLevelRed

The prototypes page shows what the prototype is (miner, processor, or output format), the type of data the prototype collects or processes (such as ipv4 addresses), details about the prototype

itself (such as, the Emerging Threats Open blocklist IP addresses), as as any pre-defined metadata tags (e.g. ShareLevelGreen, ConfidenceHigh, OSINT, etc.)


ETOpen.blockIPs
GREGORY ROEHL (PALOALTONETWORKS.COM)

MINER

IPv4











ETOpen Emerging Threats Open rulesets
ETOpen.blockIPs Raw IPs for the firewall block lists. These come from Spam nets identified by Spamhaus (www.spamhaus.org), Top Attackers listed by DShield (www.dshield.org), Abuse.ch.
 TAGS

ConfidenceMedium

ShareLevelGreen

OSINT

The vast majority of prototypes are going to be miners used for collecting indicators from disparate locations for aggregation (such as the ETOpen.blockIPs list above). But there are also several output and processor prototypes.

 stdlib.aggregatorSHA1 <small>MINEMELD CORE TEAM</small>	PROCESSOR	sha1	stdlib Library of prototypes for commonly used nodes stdlib.aggregatorSHA1 Aggregator for SHA1 indicators. Inputs with names starting with "wl" will be interpreted as whitelists.
 stdlib.aggregatorSHA256 <small>MINEMELD CORE TEAM</small>	PROCESSOR	sha256	stdlib Library of prototypes for commonly used nodes stdlib.aggregatorSHA256 Aggregator for SHA256 indicators. Inputs with names starting with "wl" will be interpreted as whitelists.
 stdlib.aggregatorSSDEEP <small>MINEMELD CORE TEAM</small>	PROCESSOR	ssdeep	stdlib Library of prototypes for commonly used nodes stdlib.aggregatorSSDEEP Aggregator for ssdeep indicators. Inputs with names starting with "wl" will be interpreted as whitelists.
 stdlib.aggregatorURL <small>MINEMELD CORE TEAM</small>	PROCESSOR	URL	stdlib Library of prototypes for commonly used nodes stdlib.aggregatorURL Aggregator for URL indicators. Inputs with names starting with "wl" will be interpreted as whitelists.
 stdlib.aggregatorUserAgentFragment <small>MINEMELD CORE TEAM</small>	PROCESSOR	user-agent.fragment	stdlib Library of prototypes for commonly used nodes stdlib.aggregatorUserAgentFragment Aggregator for user-agent.fragment indicators. Inputs with names starting with "wl" will be interpreted as whitelists.
 stdlib.aggregatorWindowsRegistryValue <small>MINEMELD CORE TEAM</small>	PROCESSOR	windows-registry-value	stdlib Library of prototypes for commonly used nodes stdlib.aggregatorWindowsRegistryValue Aggregator for windows-registry-value indicators. Inputs with names starting with "wl" will be interpreted as whitelists.
 stdlib.dagPusher <small>MINEMELD CORE TEAM</small>	OUTPUT	IPv4 IPv6	stdlib Library of prototypes for commonly used nodes stdlib.dagPusher Push IP unicast indicators to PAN-OS devices via DAG.
 stdlib.feedGreenWithValue <small>MINEMELD CORE TEAM</small>	OUTPUT	any	stdlib Library of prototypes for commonly used nodes stdlib.feedGreenWithValue EDL for indicators with share level green, with value TAGS <div>ShareLevelGreen</div>
 stdlib.feedHCGreen <small>MINEMELD CORE TEAM</small>	OUTPUT	any	stdlib Library of prototypes for commonly used nodes stdlib.feedHCGreen EDL for high confidence indicators (>75) and share level green TAGS <div>ConfidenceHigh</div> <div>ShareLevelGreen</div>
 stdlib.feedHCGreenWithValue <small>MINEMELD CORE TEAM</small>	OUTPUT	any	stdlib Library of prototypes for commonly used nodes stdlib.feedHCGreenWithValue EDL for high confidence indicators (>75) and share level green, with value TAGS <div>ConfidenceHigh</div> <div>ShareLevelGreen</div>

Reviewing Prototype Details

By clicking on a prototype, users are directed to a page that provides more details about the prototype being reviewed. Remember to be aware that Miners are for collecting data from external indicators, or local whitelists, processors aggregate and deduplicate data from miners of the same indicator type, and that outputs are output formats that take the aggregated and deduplicated data from processors. For the purposes of this exercise, we will look at the ETOpen.blockIPs prototype.

ETOpen.blockIPs PROTOTYPE

CLONE NEW

MINER **STABLE**

ABOUT ETOpen

Emerging Threats Open rulesets
For more details: <http://doc.emergingthreats.net>

ABOUT ETOpen.blockIPs

Raw IPs for the firewall block lists. These come from Spam nets identified by Spamhaus (www.spamhaus.org), Top Attackers listed by DShield (www.dshield.org), Abuse.ch.

AUTHOR

Gregory Roehl (paloaltonetworks.com)

CLASS

minemeld.ft.http.HttpFT

INDICATOR TYPES

IPv4

TAGS

ConfidenceMedium **ShareLevelGreen** **OSINT**

CONFIG

attributes	confidence: 50 share_level: green type: IPv4
ignore_regex	^#
source_name	ET.block_ips
url	http://rules.emergingthreats.net/fwrules/emerging-Block-IPs.txt

The prototype page features a whole host of information about the the prototype selected. Each section provides information about the purpose of the prototype, what data it gathers, what class it uses to perform its task, the indicator type the prototype applies to, any metadata tags for the data it gathers, and finally, there is a config section. The config section varies greatly from prototype to prototype and influences how that prototype operates. For instance, in the image above, the config has multiple entries:

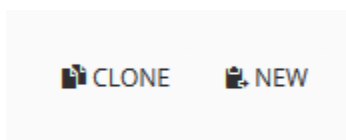
Attributes - Governs metadata and/or data type associated with the prototype. For instance, we see the attributes confidence:50, share_level: green and type: IPv4 that seem to directly correlate to the Indicator Types, and Tags section.

Ignore_regex - this appears to be a configuration that states if this regular expression is matched (a hash mark at the beginning of a line) to ignore that line entirely. This ignore regex is designed to ignore comment lines in the retrieved data.

Source_name - the name of the data source this prototype gathers indicators from

Url - this prototype is a miner that collects data from a threat intelligence source. The URL field of the config section denotes what URL and website the data is sourced from.

On the page for the current prototype, in the upper right corner, there are two buttons labeled “Clone”, and “New”.



Cloning Prototypes into Nodes

If the “CLONE” option is selected, the Add Node page appears, to add the reviewed prototype as a new node.

A screenshot of a form titled 'ADD NODE'. It contains three input fields: 'NAME' with the value 'blockIPs-1516373854595', 'PROTOTYPE' with a dropdown menu showing 'ETOpen.blockIPs', and 'INPUTS' with a placeholder text 'Select input nodes...'. At the bottom right, there are two buttons: 'OK' (blue) and 'CANCEL' (orange).

The clone feature replicates the prototype into a node with no modification whatsoever. Let’s review the fields available on the Add Node page:






Name - This defines the name of the node being created based off the prototype just reviewed. It is advised to enter a descriptive name for the new node. In the case of this miner,

ETOpenBlockIPs-IPv4 would be a sufficiently descriptive name to describe the purpose of this node.

Prototype - Here users can confirm the prototype to create the node from. Prototypes from other nodes can also be selected, if desired.

Inputs - If the prototype being reviewed is either a processor or output, users can specify the miner(s) or processor(s) the node accepts data from here. Since the the ETOpen.blockIPs is a miner node, the input field is disabled.

By clicking the OK button, the new node is added to the node list on the config page. Note that users will have to click the commit button in the upper left corner of the commit page before any new nodes are considered active.

 COMMIT		 REVERT  LOAD  IMPORT  EXPORT	
		Search: <input type="text"/>	
NAME	TYPE	PROTOTYPE	INPUTS
dshield_blocklist	MINER	dshield.block	None ✕
ETOpenBlockIPs-IPv4	MINER	ETOpen.blockIPs	None ✕
spamhaus_DROP	MINER	spamhaus.DROP	None ✕
spamhaus_EDROP	MINER	spamhaus.EDROP	None ✕
wlWhiteListIPv4	MINER	stdlib.listIPv4Generic	None ✕
inboundfeedhc	OUTPUT	stdlib.feedHCGreen	inboundaggregator ✕
inboundfeedlc	OUTPUT	stdlib.feedLCGreen	inboundaggregator ✕
inboundfeedmc	OUTPUT	stdlib.feedMCGreen	inboundaggregator ✕
inboundaggregator	PROCESSOR	stdlib.agggregatorIPv4Inbound	spamhaus_DROP spamhaus_EDROP dshield_blocklist wlWhiteListIPv4 ✕

Note: New miners MUST have a corresponding processor AND output before they will output data that can be consumed by downstream devices and/or analysts. For instance, the ETOpenBlockIPs-IPv4 node should be added to the inputs column of the inboundaggregator node. This is what it should look like when set up correctly:

inboundaggregator	PROCESSOR	stdlib.aggregatorIPv4Inbound	spamhaus_DROP spamhaus_EDROP dshield_blocklist w/WhiteListIPv4 ETOpenBlockIPs-IPv4	X
-------------------	-----------	------------------------------	--	---

Do not forget to commit new nodes and changes to existing nodes on the node config page.

Creating New Prototypes

By clicking on the “New” button on the prototype details page instead, users can opt to create a new prototype based off of the old prototype. All of the fields on the prototype details page for the prototype selected are available to be edited.

NEW LOCAL PROTOTYPE

NAME	<input type="text" value="ETOpen_blockIPs-1516378417188"/>
NODE TYPE	<input type="text" value="miner"/>
DEVEL STATUS	<input type="text" value="STABLE"/>
DESCRIPTION	<div>Raw IPs for the firewall block lists. These come from Spam nets identified by Spamhaus (www.spamhaus.org), Top Attackers listed by DShield (www.dshield.org), Abuse.ch.</div>
CLASS	<input type="text" value="minemeld.ft.http.HttpFT"/>
INDICATOR TYPES	<div>IPv4</div>
TAGS	<div>ConfidenceMedium ShareLevelGreen OSINT</div>
CONFIG	<div><pre>1 attributes: 2 confidence: 50 3 share_level: green 4 type: IPv4 5 ignore_regex: ^# 6 source_name: ET.block_ips 7 url: http://rules.emergingthreats.net/fwrules/emerging-Block-IPs.txt 8</pre></div>

Lets say for example, the user wanted to create a prototype based of the Cisco TALOS IP blacklist (<https://talosintelligence.com/documents/ip-blacklist>), since a prototype does not exist for this blacklist. This document is nearly identically formatted the same way the Emerging Threats Block IP list is formatted, so one could reconfigure the ETOpen.BlockIPs prototype to collect data from this blacklist instead, like so:

NEW LOCAL PROTOTYPE

NAME	TALOS_ipfilter
NODE TYPE	miner
DEVEL STATUS	STABLE
DESCRIPTION	Raw IPs for the firewall block lists. These come Cisco TALOS (talosintelligence.com)
CLASS	minemeld.ft.http.HttpFT
INDICATOR TYPES	IPv4
TAGS	ConfidenceMedium ShareLevelGreen OSINT
CONFIG	<pre>1 attributes: 2 confidence: 50 3 share_level: green 4 type: IPv4 5 ignore_regex: ^# 6 source_name: TALOS_ipfilter.blf 7 url: https://talosintelligence.com/documents/ip-blacklist 8</pre>


OK CANCEL

In the screen capture above, I edited the name and description field to denote that this new prototype collects data from Cisco's TALOS information security research team. In the config section itself, I changed the source_name to denote that the ipfilter.blk file from talosintelligence.com is being mined, and changed the url configuration option to point to the ip-blacklist file, provided by the TALOS team. By clicking OK the new prototype gets added to the prototype collection list, which can be found by using the search input box in the upper right corner.

PROTOTYPES

Show 50 entries

Search: talos


NAME	TYPE	INDICATORS	DESCRIPTION
 minemeldlocal.TALOS_ipfilter	MINER	IPv4	<p>minemeldlocal Local prototype library managed via MineMeld WebUI</p> <p>minemeldlocal.TALOS_ipfilter Raw IPs for the firewall block lists. These come Cisco TALOS (talosintelligence.com)</p> <p>TAGS</p> <p>ConfidenceMedium ShareLevelGreen OSINT</p>

Showing 1 to 1 of 1 entries (filtered from 199 total entries)

< 1 >

By clicking on the new prototype, the prototype details page is displayed, where more can be learned about this new prototype, it can be cloned into a node, a new prototype can be built based off this prototype, or the new prototype can be deleted via the remove option. Note that only third party prototypes can be removed.

minemeldlocal.TALOS_ipfilter PROTOTYPE

 CLONE  NEW  REMOVE

MINER STABLE

ABOUT minemeldlocal

Local prototype library managed via MineMeld WebUI

ABOUT minemeldlocal.TALOS_ipfilter

Raw IPs for the firewall block lists. These come Cisco TALOS (talosintelligence.com)

CLASS

minemeld.ft.http.HttpFT

INDICATOR TYPES

IPv4

TAGS

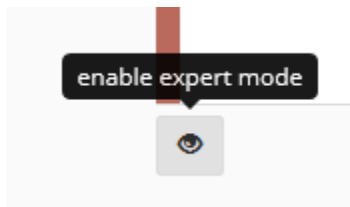
ConfidenceMedium ShareLevelGreen OSINT

CONFIG

attributes	<p>confidence: 50</p> <p>share_level: green</p> <p>type: IPv4</p>
ignore_regex	^#
source_name	TALOS_ipfilter.blk
url	https://talosintelligence.com/documents/ip-blacklist




Enabling expert mode

Back on the node configuration page, in the left corner, there is an icon in the shape of an eye. Click on this icon toggles expert mode.



Expert mode allows users view whether or not nodes have output enabled (and also allows the toggling of the output status from Disabled to Enabled, and vice versa, by clicking on the status for a particular node), and adds a blue “+” icon in the lower right corner, that when clicked, directs users to the add nodes page.

NAME	TYPE	PROTOTYPE	INPUTS	OUTPUT	
dshield_blocklist	MINER	dshield.block	None	ENABLED	✕
ETOpenBlockIPs-IPv4	MINER	ETOpen.blockIPs	None	ENABLED	✕
spamhaus_DROP	MINER	spamhaus.DROP	None	ENABLED	✕
spamhaus_EDROP	MINER	spamhaus.EDROP	None	ENABLED	✕
wlWhiteListIPv4	MINER	stdlib.listIPv4Generic	None	ENABLED	✕
inboundfeedhc	OUTPUT	stdlib.feedHCGreen	inboundaggregator	DISABLED	✕
inboundfeedlc	OUTPUT	stdlib.feedLCGreen	inboundaggregator	DISABLED	✕
inboundfeedmc	OUTPUT	stdlib.feedMCGreen	inboundaggregator	DISABLED	✕
inboundaggregator	PROCESSOR	stdlib.aggregatorIPv4Inbound	spamhaus_DROP spamhaus_EDROP dshield_blocklist wlWhiteListIPv4	ENABLED	✕



inboundaggregator

OUTPUT ENABLED

OK CANCEL

Logs

Click on the Logs option in the navigation menu.



LOGS

Scroll up for latest entries. Or click [here](#)

19/1/2018 12:51:21 -0500	inboundfeedmc	DROP_UPDATE	104.236.178.0-104.236.178.255	direction: inbound dshield_name: DIGITALOCEAN-ASN - Digital Ocean, Inc., confidence: 100 share_level: green sour...
19/1/2018 12:51:21 -0500	inboundfeedmc	RECD_UPDATE	104.236.178.0-104.236.178.255	direction: inbound dshield_name: DIGITALOCEAN-ASN - Digital Ocean, Inc., confidence: 100 share_level: green sou...
19/1/2018 12:51:21 -0500	inboundfeedmc	DROP_UPDATE	93.174.93.0-93.174.93.255	direction: inbound dshield_name: QUASINETWORKS, confidence: 100 share_level: green sources: ["dshield.block"] dshi...
19/1/2018 12:51:21 -0500	inboundfeedmc	RECD_UPDATE	93.174.93.0-93.174.93.255	direction: inbound dshield_name: QUASINETWORKS, confidence: 100 share_level: green sources: ["dshield.block"] dshi...

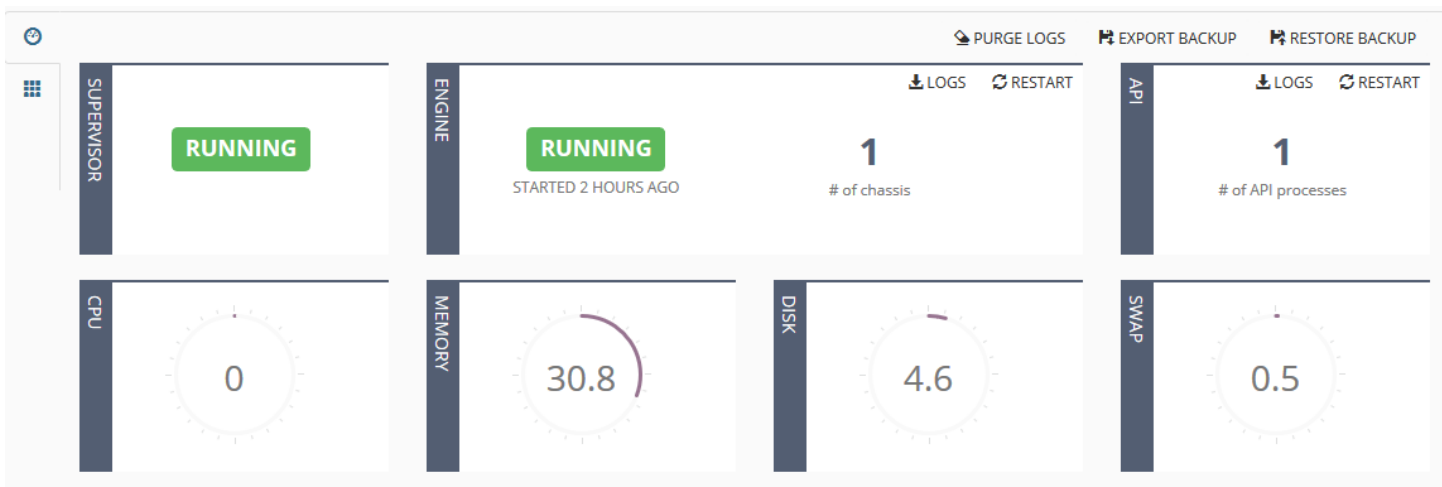
Simply specify a node, or a term to search for in the search bar, and minemeld will attempt to find logs related to the search term.

System

Click on the System option in the navigation menu.



The system page is divided into two tabs. The first tab is a dashboard that provides information on how the system is performing, and also provides numerous options related to system backups, and logs for the various subsystems on the minemeld instance.



The dashboard tab provides information about the status of the Supervisor and Engine core services, the minemeld API, as well as performance statistics for the system's CPU, Memory, Disk, and Swap usage.

In the upper right corner of the Engine and API sections of the dashboard are options to download logs related to those subsystems, and/or restart system services related to those subsystems. In the upper right portion of the dashboard tab itself, users are provided with options to purge all of the system logs on the minemeld instance, as well as options to create and/or restore backups.

Creating Backups

When the Export Backup option is selected, the interface immediately prompts the user to enter a password for the backup being created, to ensure that the data is protected at rest.

The screenshot shows a 'GENERATE BACKUP' dialog box. It contains two input fields for passwords, each with a blue highlight and a blue border. The first field is labeled 'BACKUP PASSWORD' and the second is labeled 'REPEAT PASSWORD'. At the bottom right, there are two buttons: 'OK' (blue) and 'CANCEL' (orange).

When the user clicks OK, the system schedules a backup of the system to be performed. When the backup is complete, the user is provided with a link to download the backup for external storage.

DOWNLOAD BACKUP

Backup ready. [Click here to download.](#)

DONE

Restoring backups

When the Restore Backup option is selected, users are prompted to browse to a backup file on their local system, and provide the password for the backup in order for it to be decrypted. Browse to the backup file, enter the backup's password, then click upload.

RESTORE LOCAL BACKUP

BACKUP FILE

Browse...

072e2a7b-c6bd-4658-85f9-f2084a95fd51

PASSWORD

●●●●●●

UPLOAD

CANCEL

Users are prompted to select what data they wish to restore -- node configuration data, and/or locally created prototypes. Click next to each option to check or uncheck the data to restore, then click the Restore button to start the backup restoration process.

RESTORE LOCAL BACKUP

Select backup content to restore

CONFIGURATION

☒

LOCAL PROTOTYPES

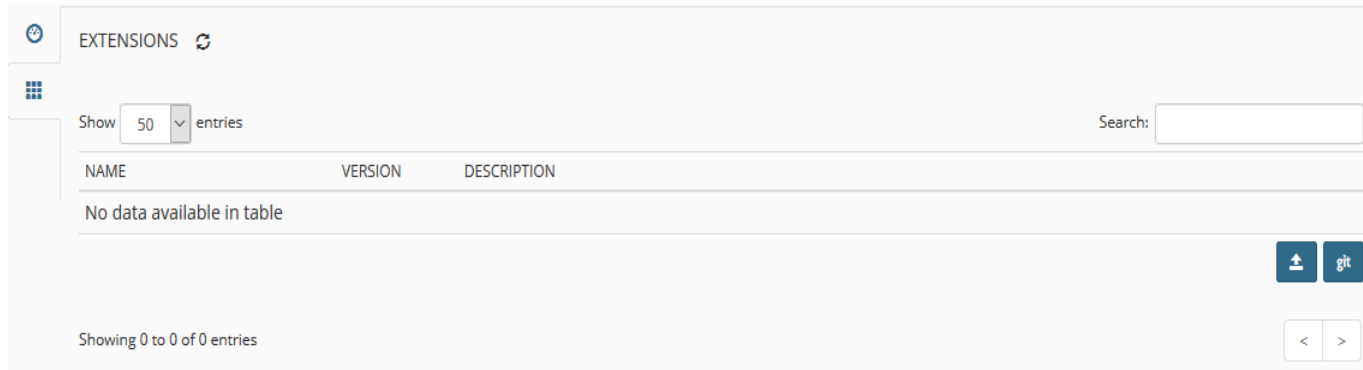
☒

RESTORE

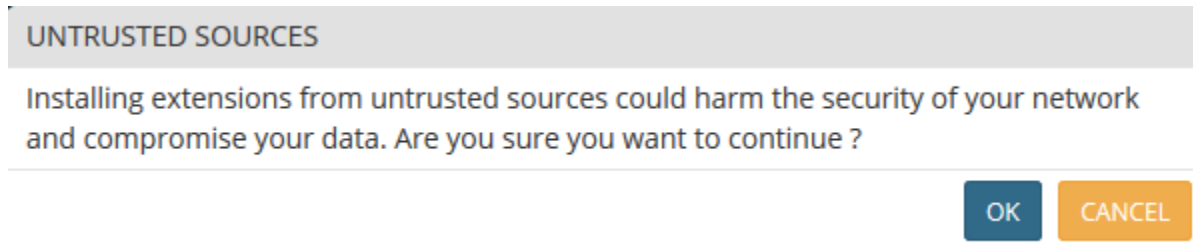
CANCEL

Third-Party Extensions

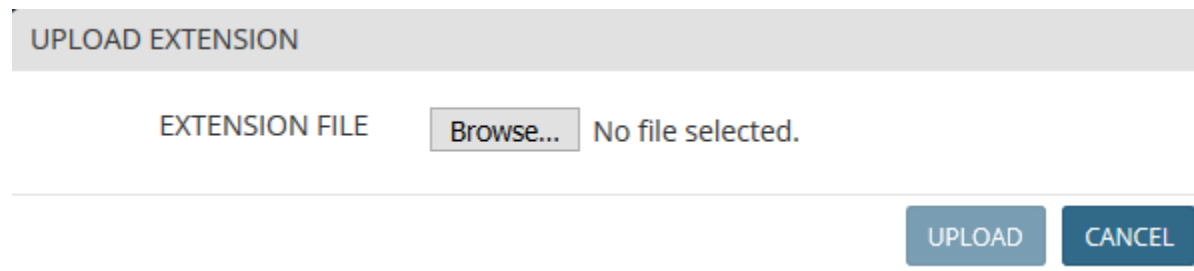
Minemeld allows for third party extensions to be installed via the Extensions tab of the System page. On this tab, there are two blue and white icons in the lower right corner. One that is an upward pointing arrow, and the second labeled “git”.



The first icon allows users to upload a minemeld extension from their local computer, while the git option allows users to download and install minemeld extensions from github.com. Regardless of the method chosen to use for installing minemeld extensions, clicking either button causes to issue a warning about risks of installing third party code and extensions. Click OK to accept the risks and continue. Users will be required to click OK anytime they wish to upload a third party extension.



When the upward facing arrow icon is selected, the Upload Extension dialog pops up. This dialog simply asks the user to browse to the location of the minemeld extension on their computer, select it, and click upload to continue.



Selecting the icon labeled “git”, causes the “Install Extension from Git” dialog to pop up. Users will be required to input the full github repository URL (ending with “.git”, as indicated) in order to install the third party extension. For demonstration purposes, we will be installing the MISP extension for minemeld, available at <https://github.com/PaloAltoNetworks/minemeld-misp>. Enter the URL of the git file into the repository URL field, then click the retrieve button next to the version field.

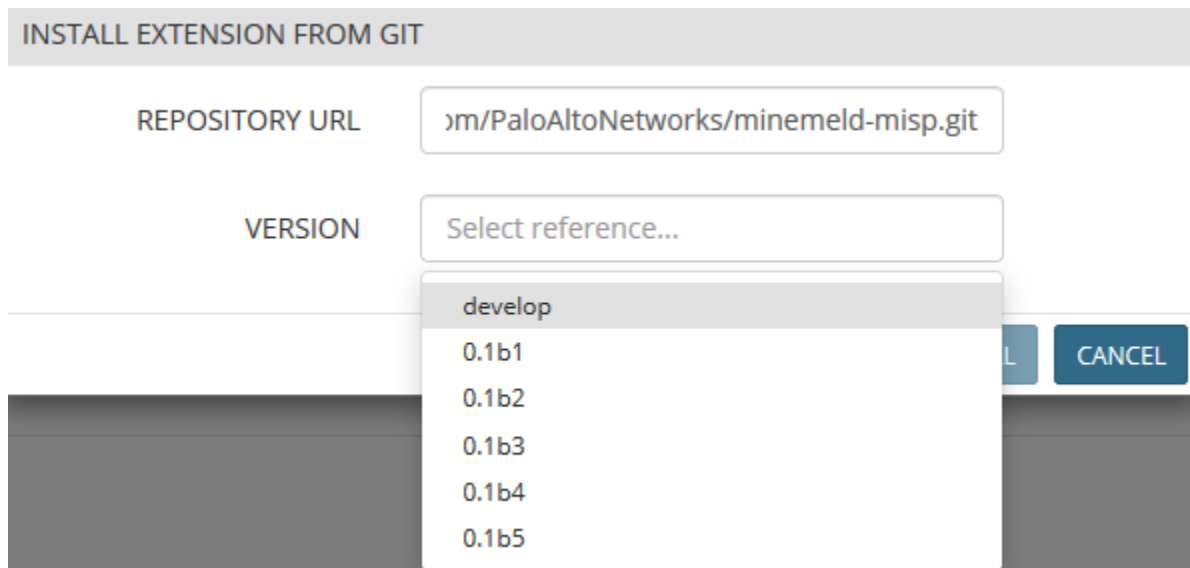


INSTALL EXTENSION FROM GIT

REPOSITORY URL

VERSION

The version field updates and shows versions of the code repository available for download via a drop-down menu. It is best practice to select the latest version available. The version drop-down lists all entries from oldest at the top of the drop-down to newest at the bottom of the drop-down. We will be choosing version 0.1b5. After selecting the version of the extension to download, click the install button to continue.






INSTALL EXTENSION FROM GIT

REPOSITORY URL

VERSION

- develop
- 0.1b1
- 0.1b2
- 0.1b3
- 0.1b4
- 0.1b5

This uploads the extension to the minemeld instance (provided a working internet connection), and the extension is added to the extensions list, along with two icons on the right side of the extension, a small “x” and a small checkbox icon.

EXTENSIONS 		
<div> <div>Show</div> <div>50</div> <div>entries</div> </div> <div>Search: <input type="text"/></div>		
NAME	VERSION	DESCRIPTION
minemeld-misp PALO ALTO NETWORKS	0.1b5	<div>MineMeld extension for MISP</div> <div>PATH</div> <div>/opt/minemeld/local/library/2255fbc7-a77d-4f8c-997f-d22a74689a28</div> <div>TAGS</div> <div>git nodes prototypes webui</div> <div>   </div>

Clicking the “x” icon uninstalls the extension from the minemeld instance, while clicking the checkbox enables the extension for use. In both cases the user will be asked to confirm deleting OR enabling the extension.


ACTIVATE EXTENSION

Are you sure you want to activate extension minemeld-misp v0.1b5 ?

OK

CANCEL

When OK is selected to enable the new extension, the striped lines through the extension disappear, and the two buttons on the right side of the extension are replaced with a single small square icon. The square icon can be used to deactivate the extension so that it may be uninstalled at a later date.



NAME	VERSION	DESCRIPTION
minemeld-misp PALO ALTO NETWORKS	0.1b5	<div>MineMeld extension for MISP</div> <div>PATH</div> <div>/opt/minemeld/local/library/2255fbc7-a77d-4f8c-997f-d22a74689a28</div> <div>TAGS</div> <div>git nodes prototypes webui</div> <div></div>

Note: The MISP extension creates a miner for collecting MISP events. Per the github page for the MISP minemeld extension, it recommends restarting the minemeld API service, then logging out of the minemeld instance, then back in. I was unable to use mm-supervisorctl, but using the system page’s restart option for the API subsystem, followed by logging out and back into the minemeld instance worked, allowing the Prototype collection to update, displaying the new MISP prototype.

PROTOTYPES

Show 50 entries

Search: misp

NAME	TYPE	INDICATORS	DESCRIPTION
 misp.anyEvent <small>PALO ALTO NETWORKS TBD</small>	MINER	any	<div><div> EXPERIMENTAL</div><div>misp MineMeld nodes for MISP Threat Intelligence Platform. misp.anyEvent Miner for MISP</div><div>TAGS<div>extensionmisp</div></div></div>

Logging Out

Clicking the navigation option farthest to the right will log the user from the minemeld instance.



Conclusions

Minemeld is a platform designed to collect and aggregate threat intelligence sources both public and private. Minemeld allows aggregation of a variety of threat intelligence indicators, aggregate them, deduplicate them, apply confidence and TLP tags to the collected data, and output the aggregated and deduplicated data for consumption by other security products, and/or analysts and incident responders looking for a quick yes or no to determine if a given collected indicator should be considered malicious.

It is extremely important to understand that the basic operation of minemeld relies on the existence of miner nodes to collect the threat intelligence and indicators from disparate external sources, processor nodes for processing a single type of indicator (e.g. file hashes, registry keys, file hashes, mutexes, URLs, domains, etc.), and output nodes for output the processed data into an output format suitable for consumption in most environments. **Processors require input from miners, and outputs require input from processors to produce indicators for consumption.**

Nodes for mining, processing, and outputting data are created from prototypes that define what types of data can be mined (and in the case of a miner, from where), processed, and what formats the data can be output to.

Minemeld simply serves as a middleware platform for preprocessing external data, and optimizing the data for consumption. Minemeld is not a threat intelligence platform OR reporting tool, and has no effective ways for reviewing the data put into the system. The onus of vetting the data is placed upon the users, through the use of “confidence level” metadata tagging.