



## SECURITY OPERATIONS FUNDAMENTALS V2

### Lab 7: Threat Intelligence

Document Version: **2022-12-23**

Copyright © 2022 Network Development Group, Inc.  
[www.netdevgroup.com](http://www.netdevgroup.com)

NETLAB+ is a registered trademark of Network Development Group, Inc.

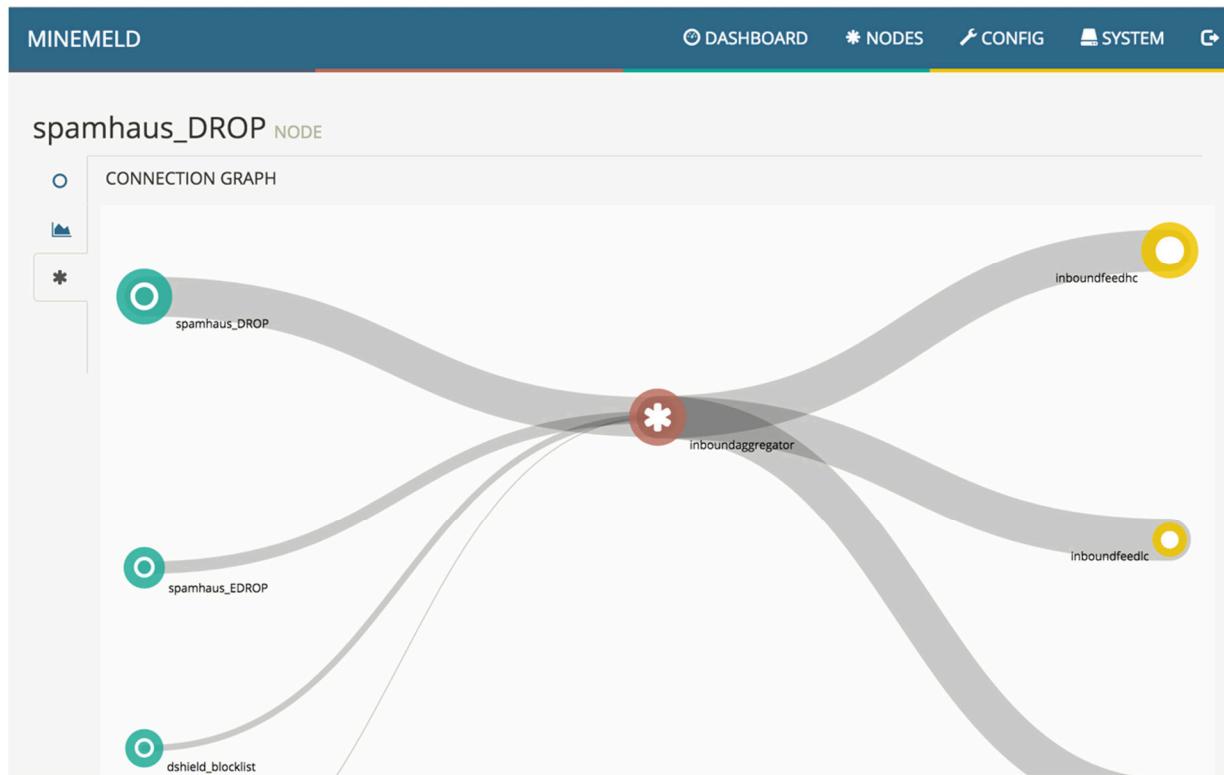
Palo Alto Networks and the Palo Alto Networks logo are trademarks or registered trademarks of Palo Alto Networks, Inc.

## Contents

|   |    |
|---|----|
| Introduction .....  | 3  |
| Objective .....   | 3  |
| Lab Topology.....   | 4  |
| Lab Settings.....   | 5  |
| 1      Threat Intelligence.....   | 6  |
| 1.0    Load Lab Configuration .....   | 6  |
| 1.1    Create a Docker Volume on the Client for a MineMeld Container.....   | 11 |
| 1.2    Launch a MineMeld Container using Docker Compose .....   | 12 |
| 1.3    Access the MineMeld Web UI to View the Default Configurations .....  | 13 |
| 1.4    Configure an External Dynamic List (EDL) on the Firewall Appliance Using a<br>MineMeld Output Feed.....                  | 17 |
| 1.5    Configure Custom MineMeld Miner, Processor and Output Nodes, and<br>Configure an EDL to use the Custom Output Node ..... | 24 |

## Introduction

In this lab, you will analyze data from the Palo Alto Networks Firewall. The data will be coming from the logs on the Palo Alto Networks Firewall. To effectively utilize the information, you will become familiar with a variety of logs and how to search the logs.

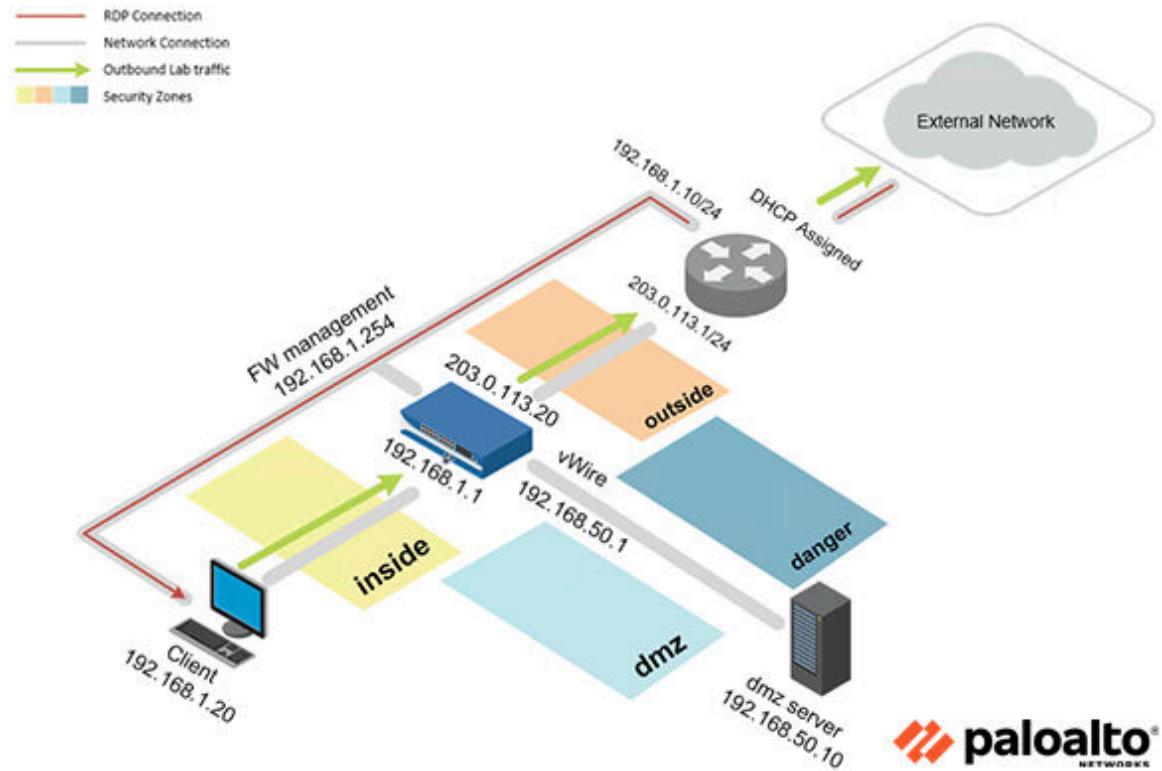


## Objective

In this lab, you will perform the following tasks:

- Create Docker volumes to store MineMeld container data on client
- Deploy a MineMeld container image using Docker Compose
- Log on to the MineMeld Web UI and observe default configurations
- On the firewall appliance configure an External Dynamic List (EDL) to use MineMeld's default threat intelligence IP block list feeds
- Create a custom MineMeld intelligence feed and use it to configure another EDL block list on the firewall appliance

# Lab Topology



## Lab Settings

The information in the table below will be needed in order to complete the lab. The task sections below provide details on the use of this information.

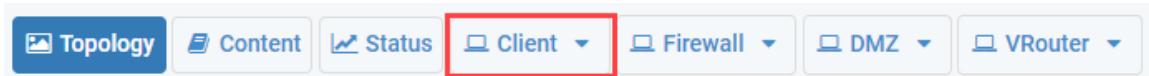
| Virtual Machine | IP Address    | Account<br>(if needed) | Password<br>(if needed) |
|-----------------|---------------|------------------------|-------------------------|
| Client          | 192.168.1.20  | lab-user               | Pal0Alt0!               |
| DMZ             | 192.168.50.10 | root                   | Pal0Alt0!               |
| Firewall        | 192.168.1.254 | admin                  | Pal0Alt0!               |

## 1 Threat Intelligence

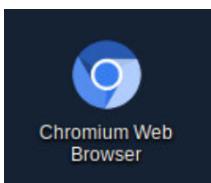
### 1.0 Load Lab Configuration

In this section, you will load the Firewall configuration file.

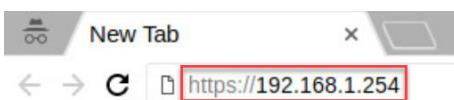
1. Click on the **Client** tab to access the client PC.



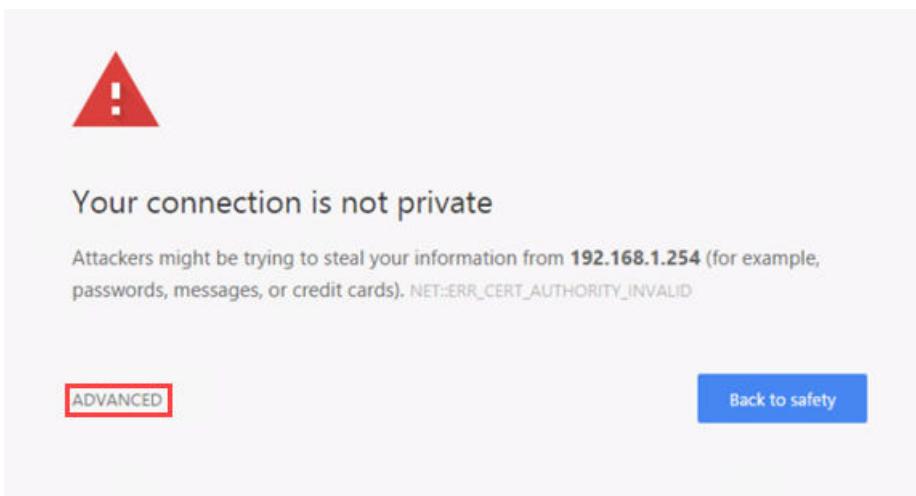
2. Log in to the client PC with the username `lab-user` and password `PaloAlt0!`.
3. Double-click the **Chromium** icon located on the desktop.



4. In the *Chromium* address field, type `https://192.168.1.254` and press **Enter**.

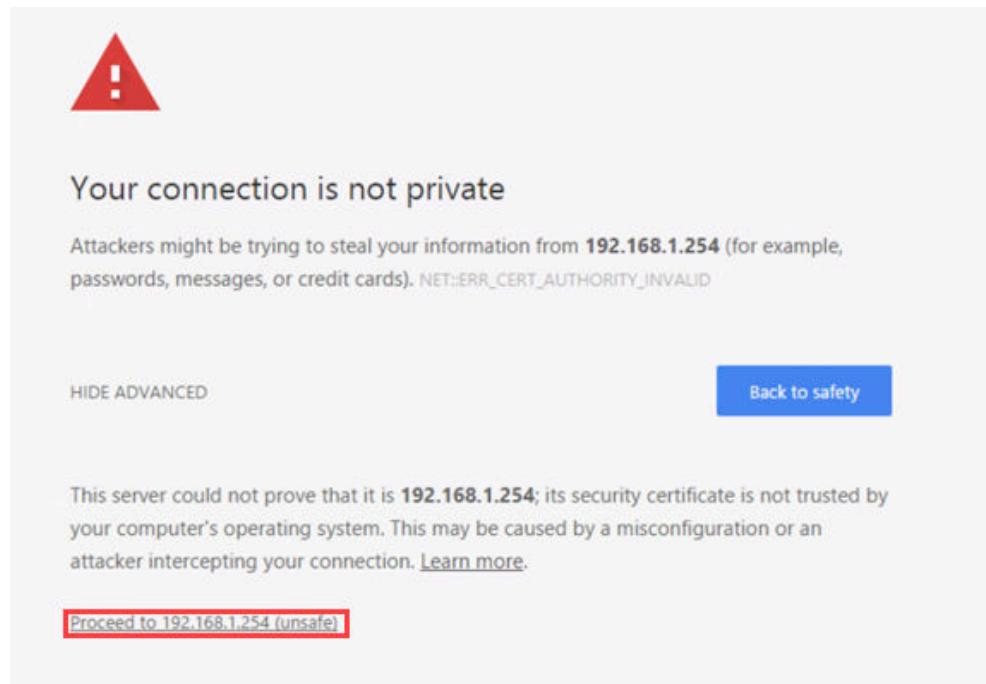


5. You will see a “*Your connection is not private*” message. Click on the **ADVANCED** link.



If you encounter the “*Unable to connect*” or “*502 Bad Gateway*” message while attempting to connect to the IP specified above, please wait an additional 1-3 minutes for the Firewall to fully initialize. Refresh the page to continue.

6. Click on **Proceed to 192.168.1.254 (unsafe)**.



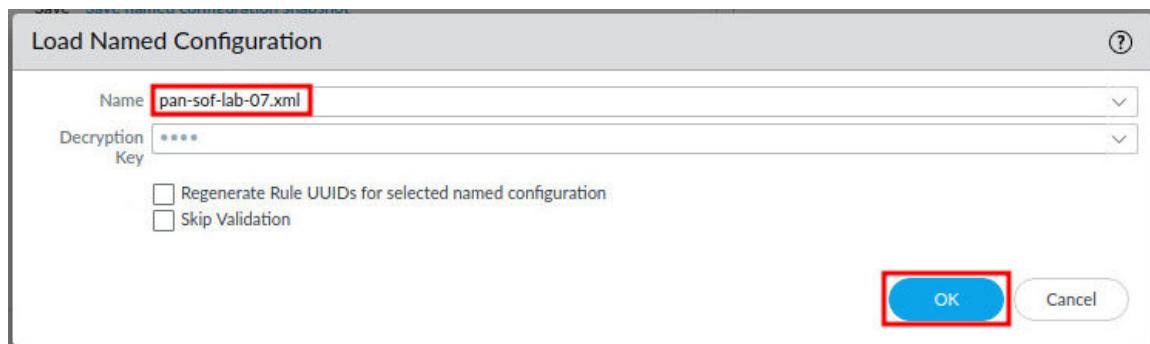
7. Log in to the Firewall web interface as username admin, password PaloAlt0!.



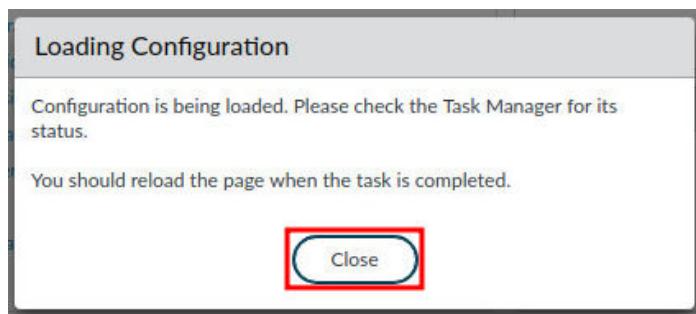
8. In the web interface, navigate to **Device > Setup > Operations** and click on **Load named configuration snapshot** in the *Configuration Management* section.

The screenshot shows the PA-VM web interface. The top navigation bar includes links for DASHBOARD, ACC, MONITOR, POLICIES, OBJECTS, NETWORK, and DEVICE. The DEVICE link is highlighted with a red box. The left sidebar has a 'Setup' section with various configuration items like High Availability, Config Audit, and Certificate Management. The main content area is titled 'Configuration Management' and contains several options under 'Operations': Revert, Save, Load, Export, Import, and Device Operations. The 'Load named configuration snapshot' option is specifically highlighted with a red box.

9. In the *Load Named Configuration* window, select **pan-sof-lab-07.xml** from the **Name** dropdown list and click **OK**.



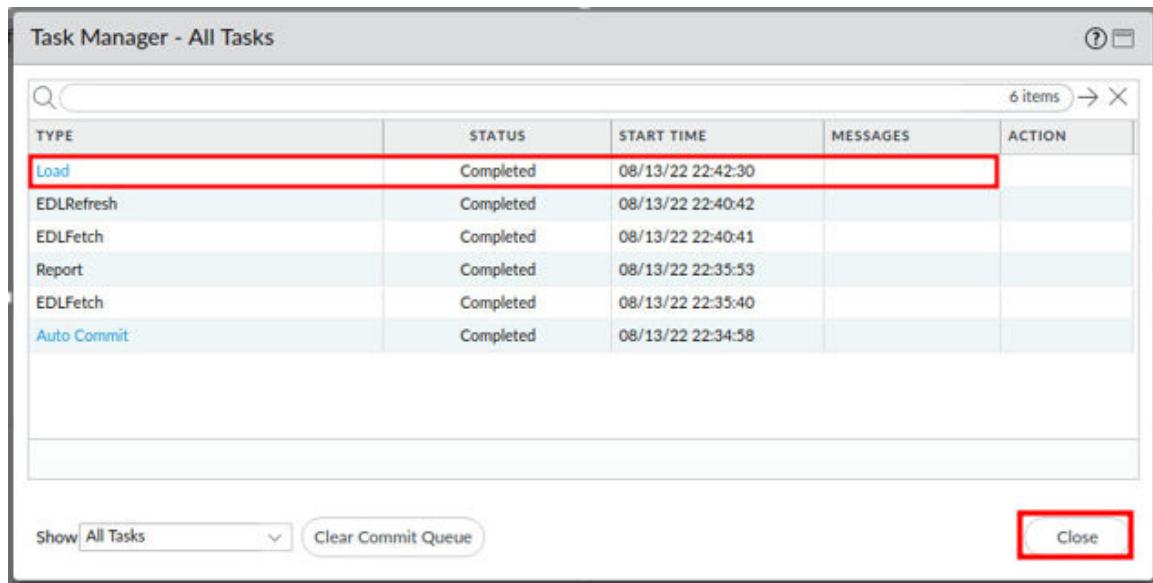
10. In the *Loading Configuration* window, a message will say *Configuration is being loaded. Please check the Task Manager for its status. You should reload the page when the task is completed.* Click **Close** to continue.



11. Click the **Tasks** icon located at the bottom-right of the web interface.

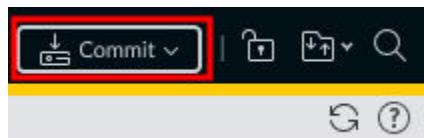


12. In the *Task Manager – All Tasks* window, verify that the *Load* type has successfully completed. Click **Close**.

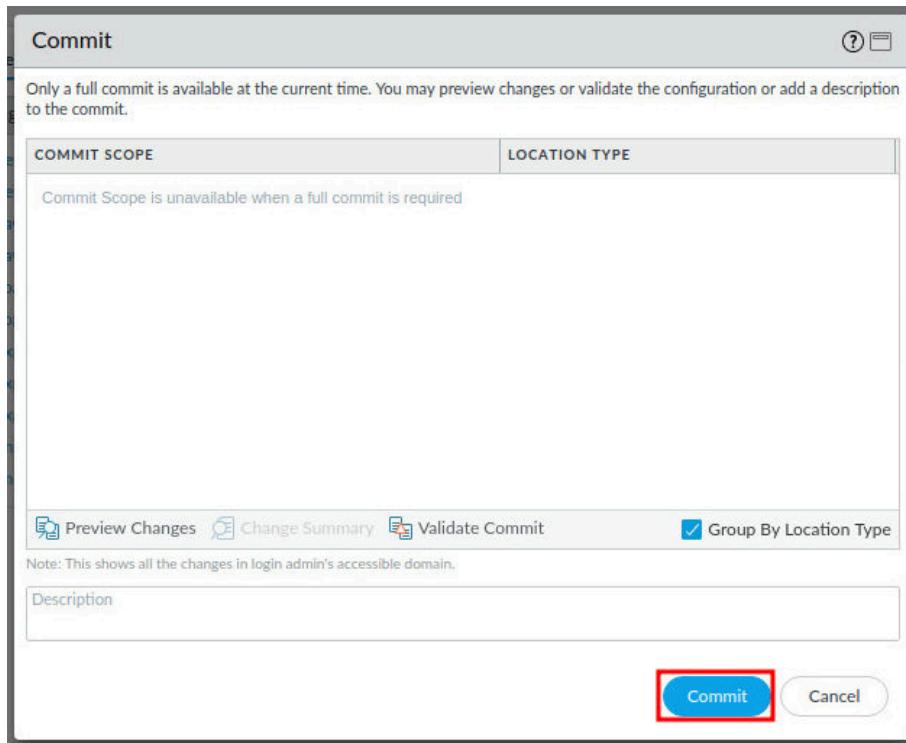


| TYPE        | STATUS    | START TIME        | MESSAGES | ACTION |
|-------------|-----------|-------------------|----------|--------|
| Load        | Completed | 08/13/22 22:42:30 |          |        |
| EDLRefresh  | Completed | 08/13/22 22:40:42 |          |        |
| EDLFetch    | Completed | 08/13/22 22:40:41 |          |        |
| Report      | Completed | 08/13/22 22:35:53 |          |        |
| EDLFetch    | Completed | 08/13/22 22:35:40 |          |        |
| Auto Commit | Completed | 08/13/22 22:34:58 |          |        |

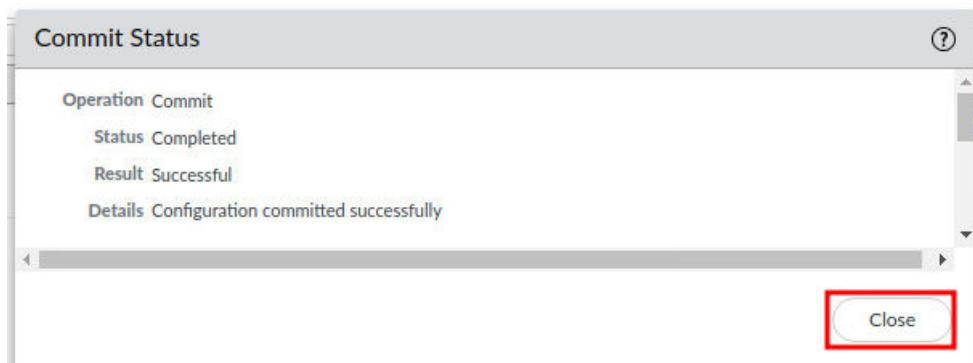
13. Click the **Commit** link located at the top-right of the web interface.



14. In the **Commit** window, click **Commit** to proceed with committing the changes.



15. When the commit operation successfully completes, click **Close** to continue.



The commit process takes changes made to the Firewall and copies them to the running configuration, which will activate all configuration changes since the last commit.

## 1.1 Create a Docker Volume on the Client for a MineMeld Container

In this section, you will create a Docker volume for a MineMeld container.



When you create a Docker volume, a directory is created in the host's `/var/lib/docker/volumes` directory for containers to use. You can then map the Docker volume on a host to a container's directory in either a Docker run command or by using a `docker-compose.yml` file. The container's data can then persist if the container stops running.

1. On the student desktop, open an *Xfce Terminal* window by clicking on the **Terminal** icon.



2. Create a Docker volume called **minemeld-logs** by typing the command below.

```
C:\home\lab-user> docker volume create minemeld-logs
```

```
C:\home\lab-user> docker volume create minemeld-logs  
minemeld-logs  
C:\home\lab-user>
```

3. Create a Docker volume called **minemeld-local** by typing the command below.

```
C:\home\lab-user> docker volume create minemeld-local
```

```
C:\home\lab-user> docker volume create minemeld-local  
minemeld-local  
C:\home\lab-user>
```

4. View the Docker volumes created by typing the command below. When prompted for the password, type Pal0Alt0! and press **Enter**.

```
C:\home\lab-user> sudo ls /var/lib/docker/volumes
```

```
C:\home\lab-user> sudo ls /var/lib/docker/volumes  
[sudo] password for lab-user:  
1b9309e5c3e59fddd22e912dbc0341259502be07ead235e6483cc84a4ccb9786  
4a8fbfb165f0829b3ee57e62395a23d7b82c5ba057f80a39ea07adc23e523d37  
64c5fc596b31fc5fc79873a4f65faf3c348a2f75559698dc87a5b1clafe4c  
8e2f5259adc3124524e4a1e953760cbaed0deec5df4c735b9b7b7c446bce367  
a8696cf7a3f105cd64afe198b9b52859dbaf6d0ad0fe911a21f76ec72060a02  
a9cc1d3876bc2a40789f724d2acf753408755eb083d016958f1b88f669850139  
metadata.db  
minemeld-local  
minemeld-logs  
C:\home\lab-user>
```

5. Leave the *terminal* window open and continue to the next task.

## 1.2 Launch a MineMeld Container using Docker Compose

In this section, you will create a MineMeld container and view the **docker-compose.yml** file using *vi editor*.

1. Navigate to the **minemeld** directory by typing the command below.

```
C:\home\lab-user> cd minemeld
```

```
C:\home\lab-user> cd minemeld
C:\home\lab-user\minemeld>
```

2. Observe the contents of the **docker-compose.yml** file by entering the command below.

```
[root@pod-dmz ~]# vi docker-compose.yml
```

```
C:\home\lab-user\minemeld> vi docker-compose.yml
```

3. Examine the contents of the **docker-compose.yml** file. Lastly, type :q and press **Enter** to exit the *vi editor* and return to the *terminal* shell.

```
File Edit View Terminal Tabs Help
version: '3.0'
services:
  minemeld:
    container_name: minemeld
    tmpfs: /run
    dns: 1.1.1.1
    volumes:
      - 'minemeld-local:/opt/minemeld/local'
      - 'minemeld-logs:/opt/minemeld/log'
    ports:
      - '443:443'
      - '80:80'
    image: paloaltonetworks/minemeld
    volumes:
      - minemeld-local
      - minemeld-logs
```

The terminal window shows the **docker-compose.yml** file with annotations explaining various configuration parameters:

- Container name: minemeld**
- Deletes files in volumes after container stops**
- Sets the preferred dns server for the container**
- Maps docker volumes to container directories**
- Maps host ports 443 & 80 to container ports 443 & 80**
- Image used to create the container**
- Declares the docker volumes you created**

4. Launch the MineMeld container by typing the command below.

```
[root@pod-dmz ~]# docker-compose up -d
```

```
C:\home\lab-user\minemeld> docker-compose up -d
Creating network "minemeld_default" with the default driver
Creating minemeld ... done
C:\home\lab-user\minemeld>
```

5. View the **minemeld-logs** volume by typing the command below. If prompted for the password, type Pal0Alt0! and press **Enter**.

```
[root@pod-dmz ~]# sudo docker logs minemeld
```

```
C:\home\lab-user\minemeld>sudo docker logs minemeld
[sudo] password for lab-user: [REDACTED]
*** Running /etc/rc.local...
*** Booting runit daemon...
*** Runit started as PID 7
minemeld: checking if dependencies are running...
run: redis: (pid 20) 0s
run: collectd: (pid 21) 0s
Copying constraints
Starting redis-server...
Regenerating CA bundle
2021-01-04T04:23:48 (34)cacert_merge.main INFO: config: {'cafile': ['/opt/minemeld/local/certs/site/'], 'dst': '/opt/minemeld/local/certs/bundle.crt', 'config': '/opt/minemeld/local/certs/cacert-merge-config.yml', 'no_merge_certifi': False}
0
Starting minemeld...
/opt/minemeld/engine/0.9.70.post1/local/lib/python2.7/site-packages/supervisor/options.py:383: PkgResourcesDeprecationWarning: Parameters to load are deprecated. Call .resolve and .require separately.
    return pkg_resources.EntryPoint.parse("x="+spec).load(False)
```

6. Minimize the *terminal* window by clicking the **Minimize** icon in the top-right of the *terminal* window and continue to the next task.



### 1.3 Access the MineMeld Web UI to View the Default Configurations

In this section, you will access MineMeld to aggregate threat intelligence feeds across public, private and commercial intelligence sources that include government and commercial organizations.

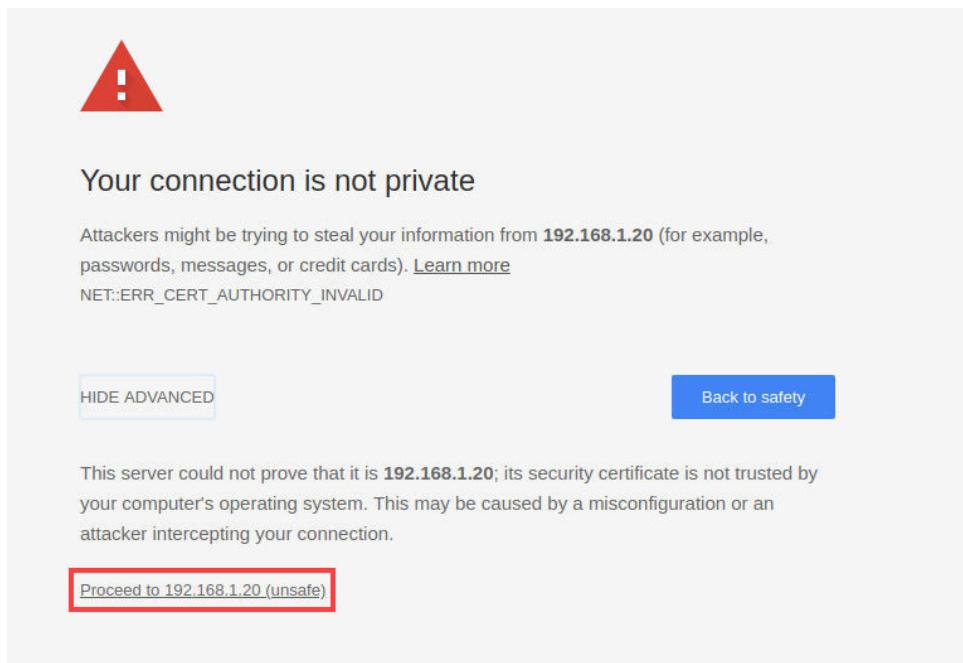
1. In the *Chromium* web browser, click on the **New tab** button in the upper-left.



2. Enter **https://192.168.1.20** in the address bar and press **Enter**.



3. In the *Your connection is not private* window, advance through the security warning and click **Proceed to 192.168.1.20 (unsafe)**.



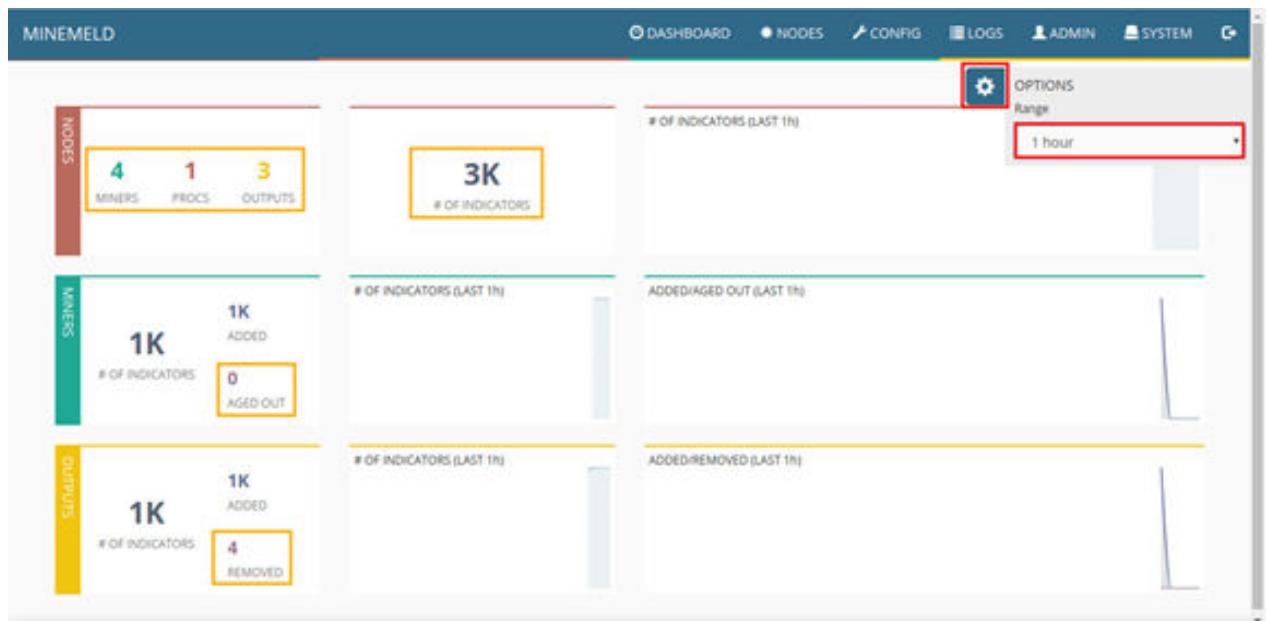
4. Log on to the *MineMeld Web UI* by typing admin for the username and minemeld for the password. Click **Login**.



5. In the *MineMeld Web UI*'s dashboard, click the settings gear icon on the far-right and change the range to **1 hour**. In the **Nodes** widget, note you have 4 Miners, 1 Processor and 3 Outputs.

**Please Note**

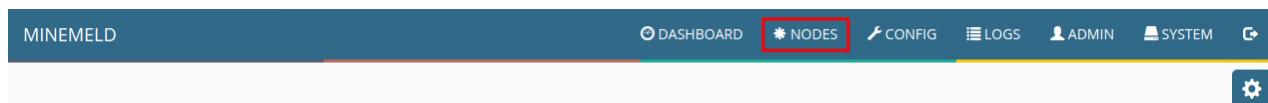
The **Miners** collect and serve the intelligence feed indicators containing block lists to the **Processors**. The **Processors** deduplicate and process the intelligence feeds for the **Outputs**. The **Outputs** can send the intelligence indicators to downstream MineMeld instances, to firewall/IPS devices, and/or to endpoint protection software.



**Please Note**

Note that there are 3K block list indicators and the intelligence feed indicators are constantly being updated. The number of indicators may vary.

6. In the *MineMeld UI*, navigate to the *Nodes* webpage by clicking on the **NODES** link.



7. On the *nodes* webpage, select the **spamhaus\_DROP** Miner.

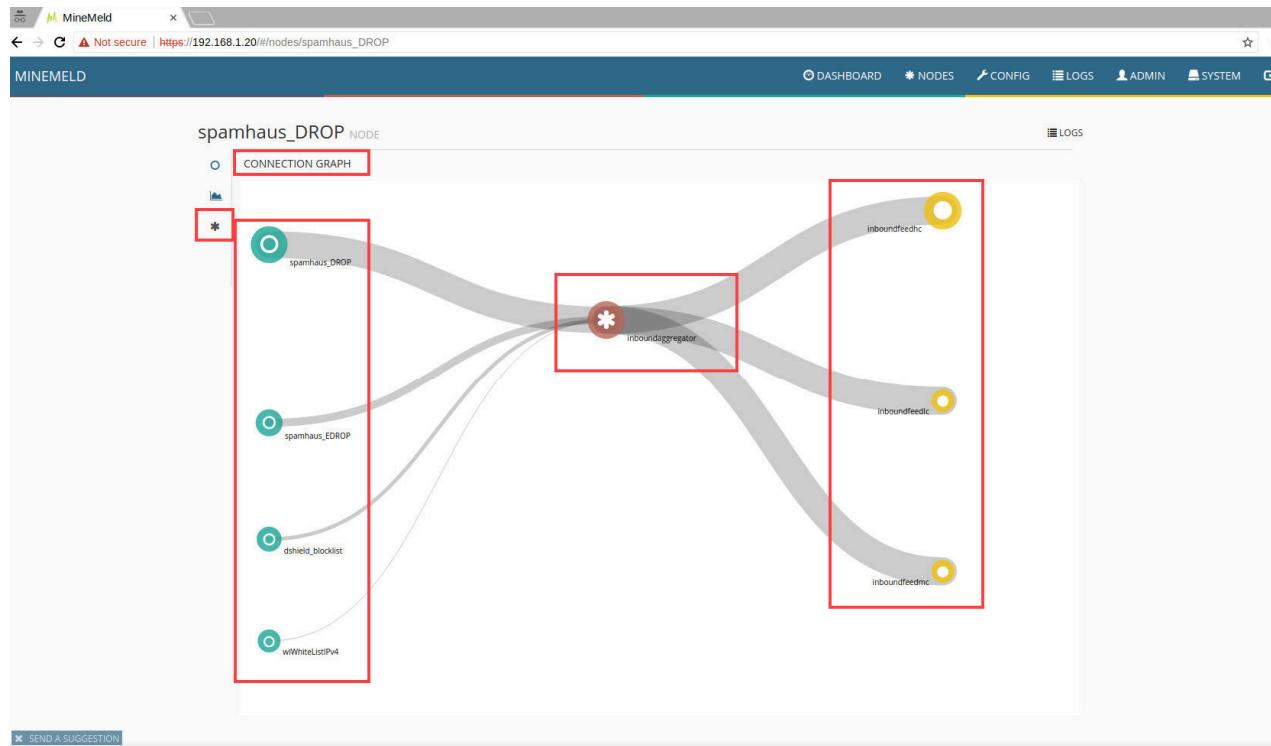
| MINEMELD          |        |         |            |                          |                                  |                                | DASHBOARD     | NODES | CONFIG | LOGS | ADMIN | SYSTEM |  |
|-------------------|--------|---------|------------|--------------------------|----------------------------------|--------------------------------|---------------|-------|--------|------|-------|--------|--|
|                   |        |         |            |                          |                                  |                                | ADD INDICATOR |       |        |      |       |        |  |
| Show              | All    | entries |            |                          |                                  |                                | Search:       |       |        |      |       |        |  |
| ▲ NAME            | ▲ TYPE | STATE   | INDICATORS | ADD/REM/AO               | UPDATES                          | WITHDRAWALS                    |               |       |        |      |       |        |  |
| dshield_blocklist | MINER  | STARTED | 20         | ADDED: 21<br>AGED OUT: 1 | RX: 0<br>PROCESSED: 0<br>TX: 40  | RX: 0<br>PROCESSED: 0<br>TX: 1 |               |       |        |      |       |        |  |
| spamhaus_DROP     | MINER  | STARTED | 898        | ADDED: 898<br>REMOVED: 0 | RX: 0<br>PROCESSED: 0<br>TX: 898 | RX: 0<br>PROCESSED: 0<br>TX: 0 |               |       |        |      |       |        |  |
| spamhaus_EDROP    | MINER  | STARTED | 69         | ADDED: 69<br>REMOVED: 0  | RX: 0<br>PROCESSED: 0<br>TX: 69  | RX: 0<br>PROCESSED: 0<br>TX: 0 |               |       |        |      |       |        |  |
| wlWhiteListIPv4   | MINER  | STARTED | 0          | ADDED: 0<br>REMOVED: 0   | RX: 0<br>PROCESSED: 0<br>TX: 0   | RX: 0<br>PROCESSED: 0<br>TX: 0 |               |       |        |      |       |        |  |

8. On the **spamhaus\_DROP** node, select the **graph** icon.

spamhaus\_DROP NODE

|  |                               |
|--|-------------------------------|
|  | STATUS                        |
|  | CLASS minemeld.ft.http.HttpFT |
|  | PROTOTYPE spamhaus.DROP       |
|  | STATE STARTED                 |

9. In the *CONNECTION GRAPH*, you should see 4 Miners (which collect the block list indicators) connect to the **inboundaggregator** Processor. This Processor feeds the processed indicators to 3 Output nodes: **inboundfeedhc** (hc=high confidence), **inboundfeedlc** (lc=low confidence), and **inboundfeedmc** (mc = medium confidence).

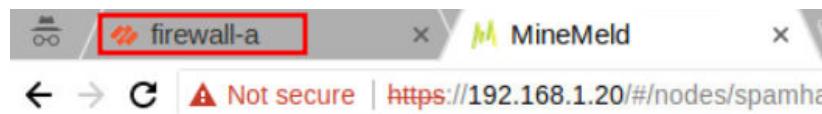


10. Leave the *Chromium* web browser open and continue to the next task.

#### 1.4 Configure an External Dynamic List (EDL) on the Firewall Appliance Using a MineMeld Output Feed

In this section, you will configure an External Dynamic List (EDL) on the Firewall to use the *inboundfeedhc* Output node feed and then use the EDL in a Security Policy rule to block incoming traffic.

1. Click on the **firewall-a** tab in the upper-left to return to the firewall web interface.



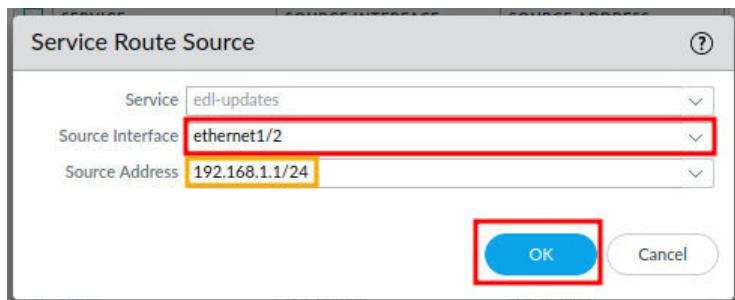
2. Navigate to **Device > Setup > Services** and select **Service Route Configuration**.

The screenshot shows the PA-VM management interface. The top navigation bar includes links for DASHBOARD, ACC, MONITOR, POLICIES, OBJECTS, NETWORK, and DEVICE. The DEVICE link is highlighted with a red box. Below the navigation is a secondary menu with links like Management, Operations, Services (which is also highlighted with a red box), Interfaces, Telemetry, Content-ID, WildFire, Session, and Help. On the left, a sidebar titled "Setup" contains various configuration options, with "Service Route Configuration" highlighted with a red box under the "Services Features" section.

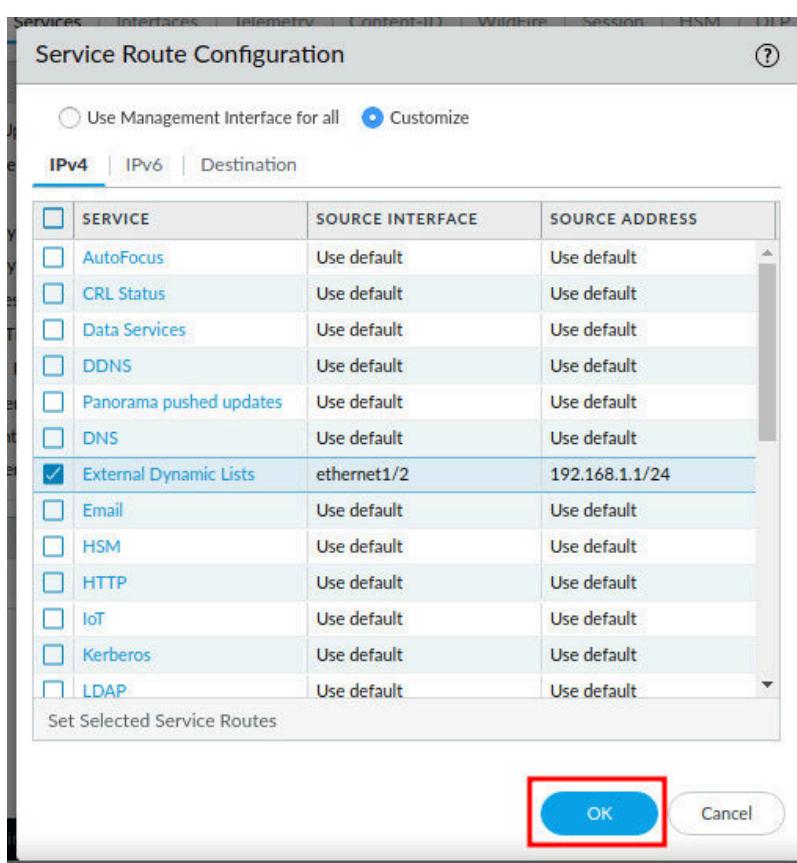
3. In the *Service Route Configuration* dialog box, select **Customize**. Select and click **External Dynamic Lists**.

The screenshot shows the "Service Route Configuration" dialog box. At the top, there is a radio button for "Use Management Interface for all" and a checked radio button for "Customize", which is highlighted with a red box. Below this are tabs for "IPv4" (which is selected and highlighted with a blue border), "IPv6", and "Destination". The main area is a table with columns: SERVICE, SOURCE INTERFACE, and SOURCE ADDRESS. The table lists several services: AutoFocus, CRL Status, Data Services, DDNS, Panorama pushed updates, DNS, External Dynamic Lists (which is highlighted with a red box), and Email. All entries show "Use default" for both source interface and address.

4. In the *Service Route Source* dialog box, select **ethernet1/2** for the *Source Interface* and verify **192.168.1.1/24** for the *Source Address*. Click **OK**.



5. In the *Service Route Configuration* window, click **OK**.



6. Navigate to **Objects > External Dynamic Lists** and select **Add**.

The screenshot shows the PA-VM interface with the 'OBJECTS' tab selected. In the left sidebar, under 'External Dynamic Lists', the 'Add' button is highlighted with a red box. The main pane displays a table of predefined dynamic lists:

| NAME  | LOCATION   | DESCRIPTION  | SOURCE                    |
|---|------------|--|---------------------------|
| Palo Alto Networks - Tor exit IP addresses        | Predefined | IP addresses supplied by multiple providers and validated with Palo Alto Networks threat intelligence data as active Tor exit nodes. Traffic from Tor exit nodes can serve a legitimate purpose, however, is disproportionately associated with malicious activity, especially in enterprise environments. | Palo Alto Ne addresses    |
| Palo Alto Networks - Bulletproof IP addresses     | Predefined | IP addresses that are provided by bulletproof hosting providers. Because bulletproof hosting providers place few, if any, restrictions on content, attackers can use these services to host and distribute malicious, illegal, and unethical material.   | Palo Alto Ne IP addresses |
| Palo Alto Networks - High risk IP addresses       | Predefined | IP addresses that have recently been featured in threat activity advisories distributed by high-trust organizations. However, Palo Alto Networks does not have direct evidence of maliciousness for these IP addresses.  | Palo Alto Ne addresses    |
| Palo Alto Networks - Known malicious IP addresses | Predefined | IP addresses that are currently used almost exclusively by malicious actors for malware distribution, command-and-control, and for launching various attacks.  | Palo Alto Ne malicious IP |

7. In the *External Dynamic Lists* window, type **MineMeld high confidence list** in the **Name** field, and enter **<https://192.168.1.20/feeds/inboundfeedhc>** for the **Source**. Click **OK**.

The dialog box is titled 'External Dynamic Lists'. It has two tabs: 'Create List' (selected) and 'List Entries And Exceptions'. The 'Create List' tab contains the following fields:

- Name:** MineMeld high confidence list (highlighted with a red box)
- Type:** IP List (dropdown menu)
- Description:** (empty text area)
- Source:** https://192.168.1.20/feeds/inboundfeedhc (highlighted with a red box)
- Server Authentication:** Certificate Profile: None
- Check for updates:** Every five minutes

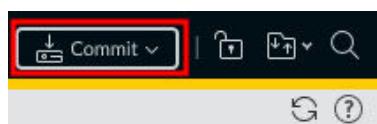
At the bottom right are 'OK' and 'Cancel' buttons, with 'OK' highlighted with a red box.

8. Navigate to **Policies > Security** and select the **outside-inside** policy.

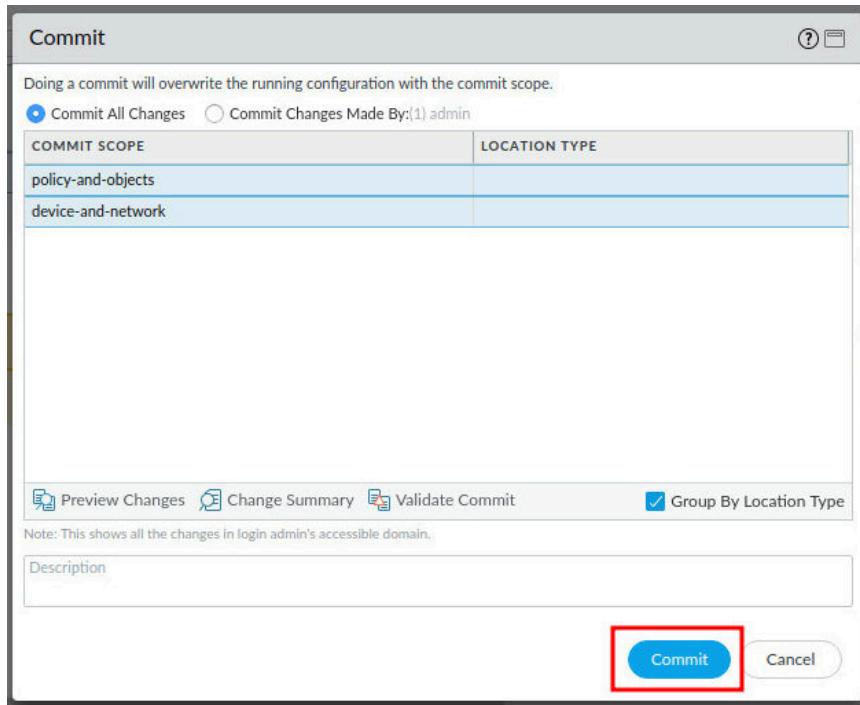
| NAME                  | TAGS     | TYPE      | Source  |         |      |
|-----------------------|----------|-----------|---------|---------|------|
|                       |          |           | ZONE    | ADDRESS | USER |
| 1 outside-inside      | internal | universal | outside | any     | any  |
| 2 internal-inside-dmz | internal | universal | inside  | any     | any  |
| 3 egress-outside      | egress   | universal | dmz     | any     | any  |
| 4 intrazone-default   | none     | intrazone | any     | any     | any  |
| 5 interzone-default   | none     | interzone | any     | any     | any  |

9. In the **Security Policy Rule** window, select the **Source** tab. In the **Source Address** box, click **Add**. Select the **MineMeld high confidence list** from the dropdown menu. Click **OK** to close the **Security Policy Rule** window.

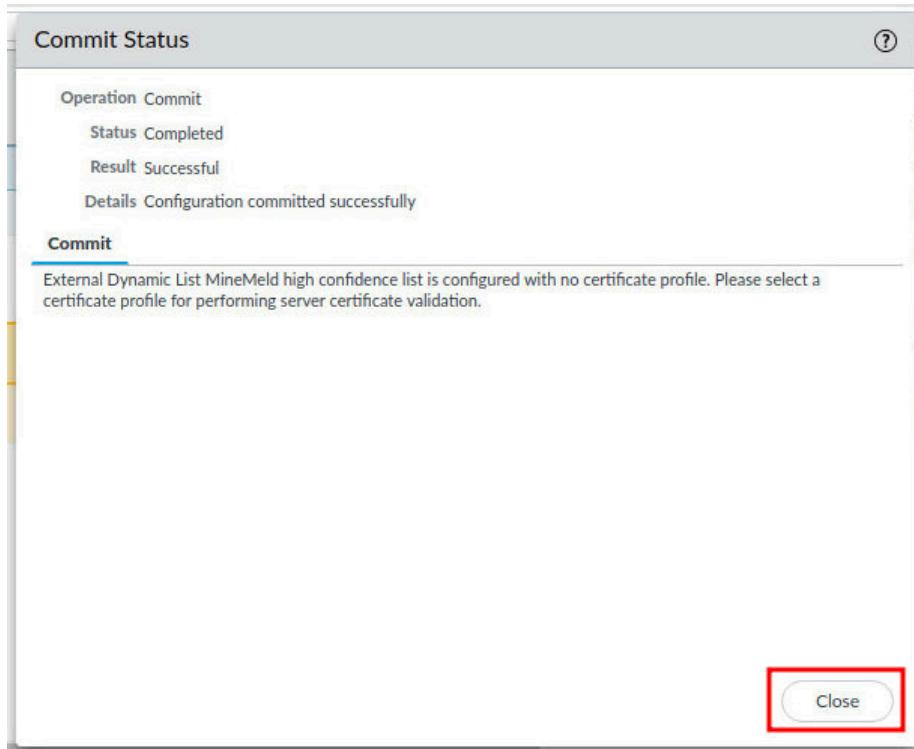
10. Click the **Commit** link located at the top-right of the web interface.



11. In the *Commit* window, click **Commit**.



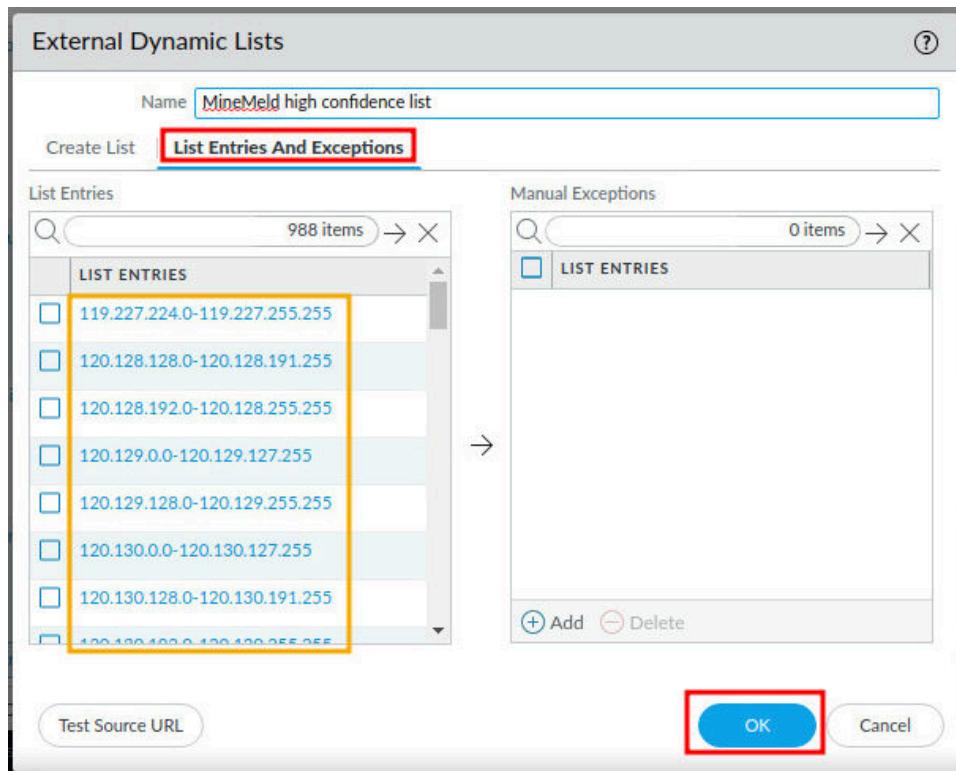
12. Once the commit finishes, click **Close**.



13. Navigate to **Objects > External Dynamic Lists** and click the **MineMeld high confidence list**.

| <input type="checkbox"/> | NAME  | LOCATION   | DESCRIPTION  | SOURCE  |
|--------------------------|---|------------|--|---|
| <input type="checkbox"/> | Palo Alto Networks - Tor exit IP addresses        | Predefined | IP addresses supplied by multiple providers and validated with Palo Alto Networks threat intelligence data as active Tor exit nodes. Traffic from Tor exit nodes can serve a legitimate purpose, however, is disproportionately associated with malicious activity, especially in enterprise environments. | Palo Alto Net addresses                           |
| <input type="checkbox"/> | Palo Alto Networks - Bulletproof IP addresses     | Predefined | IP addresses that are provided by bulletproof hosting providers. Because bulletproof hosting providers place few, if any, restrictions on content, attackers can use these services to host and distribute malicious, illegal, and unethical material.   | Palo Alto Net IP addresses                        |
| <input type="checkbox"/> | Palo Alto Networks - High risk IP addresses       | Predefined | IP addresses that have recently been featured in threat activity advisories distributed by high-trust organizations. However, Palo Alto Networks does not have direct evidence of maliciousness for these IP addresses.  | Palo Alto Net addresses                           |
| <input type="checkbox"/> | Palo Alto Networks - Known malicious IP addresses | Predefined | IP addresses that are currently used almost exclusively by malicious actors for malware distribution, command-and-control, and for launching various attacks.  | Palo Alto Net malicious IP addresses              |
| <input type="checkbox"/> | <b>MineMeld high confidence list</b>              |            |  | <a href="https://192.1.1.1">https://192.1.1.1</a> |

14. In the *External Dynamic Lists* window, select **List Entries and Exceptions** and observe the *IP block list indicators* that MineMeld is feeding the Palo Alto Networks Firewall. Click **OK**.

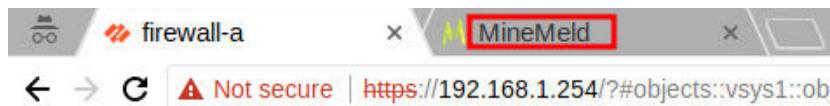


15. Leave *Chromium* open and continue to the next task.

## 1.5 Configure Custom MineMeld Miner, Processor and Output Nodes, and Configure an EDL to use the Custom Output Node

In this section, you will configure custom MineMeld Miner, Processor and Output nodes. You will also configure an EDL to use the custom Output node.

1. Change focus back to MineMeld by clicking on the **MineMeld** tab.



2. In the *MineMeld Web UI*, browse to the **CONFIG** tab. On the *config* webpage, click the **eye** icon to *enable expert mode* in the lower-left corner and then click the **+** icon to *add node* that appears at the far-right bottom.

| NAME              | TYPE      | PROTOTYPE                    | INPUTS  | OUTPUT   |
|-------------------|-----------|------------------------------|---|----------|
| dshield_blocklist | MINER     | dshield.block                | 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<br>spamhaus_EDROP<br>dshield.blocklist<br>wlWhiteListIPv4 | ENABLED  |

3. On the *ADD NODE* webpage, enter `bad-ip-miner` for the *NAME*. Select `blocklist_de.all` for the *PROTOTYPE*. Leave all other defaults and click **OK**.

**ADD NODE**

|   |   |
|---|---|
| NAME  | <input type="text" value="bad-ip-miner"/>     |
| PROTOTYPE   | <input type="text" value="blocklist_de.all"/> |
| INPUTS  | Select input nodes...                         |
| <input type="button" value="OK"/> <input type="button" value="CANCEL"/> |   |

4. On the **CONFIG** tab, on the *config* webpage, click the **eye** icon to *enable expert mode* in the lower-left corner and then click the **+** icon to *add node* that appears at the far-right bottom.

The screenshot shows the MINEMELD interface with the 'CONFIG' tab selected. The main area displays a table of nodes with columns: NAME, TYPE, PROTOTYPE, INPUTS, and OUTPUT. A red box highlights the eye icon in the bottom-left corner of the table header. Another red box highlights the '+' icon in the bottom-right corner of the table.

| NAME              | TYPE      | PROTOTYPE                    | INPUTS  | OUTPUT   |
|-------------------|-----------|------------------------------|---|----------|
| bad-ip-miner      | MINER     | blocklist_de.all             | None  | ENABLED  |
| dshield_blocklist | MINER     | dshield.block                | 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.feedHGreen            | inboundaggregator   | DISABLED |
| inboundfeedlc     | OUTPUT    | stdlib.feedLGreen            | inboundaggregator   | DISABLED |
| inboundfeedmc     | OUTPUT    | stdlib.feedMGreen            | inboundaggregator   | DISABLED |
| inboundaggregator | PROCESSOR | stdlib.aggregatorIPv4Inbound | spamhaus_DROP<br>spamhaus_EDROP<br>dshield_blocklist<br>wlWhiteListIPv4 | ENABLED  |

5. On the *ADD NODE* webpage, enter **bad-ip-processor** for the **NAME**. Select **stdlib.aggregatorIPv4Generic** for the **PROTOTYPE**. For the **INPUTS**, select **bad-ip-miner** and click **OK**.

The screenshot shows the 'ADD NODE' form. The fields are: NAME (bad-ip-processor), PROTOTYPE (stdlib.aggregatorIPv4Generic), and INPUTS (bad-ip-miner). The 'OK' button is highlighted with a red box. The 'Inputs' field has a red box around it.

6. On the **CONFIG** tab, on the *config* webpage, click the **eye** icon to *enable expert mode* in the lower-left corner and then click the **+** icon to *add node* that appears at the far-right bottom.

| NAME              | TYPE      | PROTOTYPE                    | INPUTS  | OUTPUT   |
|-------------------|-----------|------------------------------|---|----------|
| bad-ip-miner      | MINER     | blocklist_de.all             | None  | ENABLED  |
| dshield_blocklist | MINER     | dshield.block                | 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 |
| bad-ip-processor  | PROCESSOR | stdlib.aggregatorIPv4Generic | bad-ip-miner  | ENABLED  |
| inboundaggregator | PROCESSOR | stdlib.aggregatorIPv4Inbound | spamhaus_DROP<br>spamhaus_EDROP<br>dshield_blocklist<br>wlWhiteListIPv4 | ENABLED  |

7. On the **ADD NODE** webpage, enter **sof-bad-ip-output** for the **NAME**. Select **stdlib.feedMCGreenWithValue** for the **PROTOTYPE**. For the **INPUTS**, select **bad-ip-processor** and click **OK**.

|   |                             |
|---|-----------------------------|
| NAME  | sof-bad-ip-output           |
| PROTOTYPE   | stdlib.feedMCGreenWithValue |
| INPUTS  | bad-ip-processor            |
| <input type="button" value="OK"/> <input type="button" value="CANCEL"/> |                             |

8. On the *config* webpage, verify that **bad-ip-miner**, **bad-ip-processor** and **sof-bad-ip-output** are showing. Click **COMMIT** to save your changes.

**COMMIT**

REVERT LOAD IMPORT EXPORT

Search:

| NAME              | TYPE      | PROTOTYPE                    | INPUTS  | OUTPUT |
|-------------------|-----------|------------------------------|---|--------|
| bad-ip-miner      | MINER     | blocklist_de.all             | None  |        |
| 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   |        |
| sof-bad-ip-output | OUTPUT    | stdlib.feedMCGreenWithValue  | bad-ip-processor  |        |
| bad-ip-processor  | PROCESSOR | stdlib.aggregatorIPv4Generic | bad-ip-miner  |        |
| inboundaggregator | PROCESSOR | stdlib.aggregatorIPv4Inbound | spamhaus_DROP<br>spamhaus_EDROP<br>dshield_blocklist<br>wlWhiteListIPv4 |        |



You may need to wait for a minute to ensure that the STOP/START sequence has completed.

9. On the *config* webpage, click the eye icon. For **sof-bad-ip-output** click **DISABLED** to the far right.

**COMMIT**

REVERT LOAD IMPORT

Search:

| NAME              | TYPE      | PROTOTYPE                    | INPUTS  | OUTPUT   |
|-------------------|-----------|------------------------------|---|----------|
| bad-ip-miner      | MINER     | blocklist_de.all             | None  | ENABLED  |
| dshield_blocklist | MINER     | dshield.block                | 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 |
| sof-bad-ip-output | OUTPUT    | stdlib.feedMCGreenWithValue  | bad-ip-processor  | DISABLED |
| bad-ip-processor  | PROCESSOR | stdlib.aggregatorIPv4Generic | bad-ip-miner  | ENABLED  |
| inboundaggregator | PROCESSOR | stdlib.aggregatorIPv4Inbound | spamhaus_DROP<br>spamhaus_EDROP<br>dshield_blocklist<br>wlWhiteListIPv4 | ENABLED  |

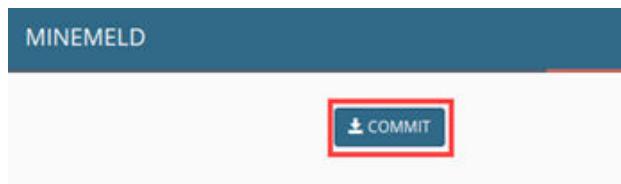
10. In the *sof-bad-ip-output* window dialog box, click **DISABLED**.



11. In the *sof-bad-ip-output* window dialog box, the output should now be **ENABLED**. Click **OK**.



12. On the *config* webpage, click **COMMIT** to save your changes.



13. On the *MineMeld* webpage, click the **NODES** tab and select **sof-bad-ip-output**.

| MINEMELD          |        |         |            |                        |                                |                                |  |
|-------------------|--------|---------|------------|------------------------|--------------------------------|--------------------------------|--|
|                   |        |         |            |                        |                                | DASHBOARD                      |  |
|                   |        |         |            | NODES                  |                                |                                |  |
| Show              | All    | entries |            |                        |                                | Search:                        |  |
| ▲ NAME            | ▲ TYPE | STATE   | INDICATORS | ADD/REM/AO             | UPDATES                        | WITHDRAWALS                    |  |
| bad-ip-miner      | MINER  | STARTED | 32588      | ADDED: 0<br>REMOVED: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 |  |
| dshield_blocklist | MINER  | STARTED | 20         | ADDED: 0<br>REMOVED: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 |  |
| spamhaus_DROP     | MINER  | STARTED | 971        | ADDED: 0<br>REMOVED: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 |  |
| spamhaus_EDROP    | MINER  | STARTED | 82         | ADDED: 0<br>REMOVED: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 |  |
| wlWhiteListIPv4   | MINER  | STARTED | 0          | ADDED: 0<br>REMOVED: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 |  |
| inboundfeedhc     | OUTPUT | STARTED | 1078       | ADDED: 0<br>REMOVED: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 |  |
| inboundfeedlc     | OUTPUT | STARTED | 0          | ADDED: 0<br>REMOVED: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 |  |
| inboundfeedmc     | OUTPUT | STARTED | 0          | ADDED: 0<br>REMOVED: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 |  |
| sof-bad-ip-output | OUTPUT | STARTED | 32515      | ADDED: 0<br>REMOVED: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 | RX: 0<br>PROCESSED: 0<br>TX: 0 |  |

14. In the *sof-bad-ip-output* window, right-click **FEED BASE URL** and click **Copy link address**. You will use the URL to create another External Dynamic List on your Palo Alto Networks Firewall.

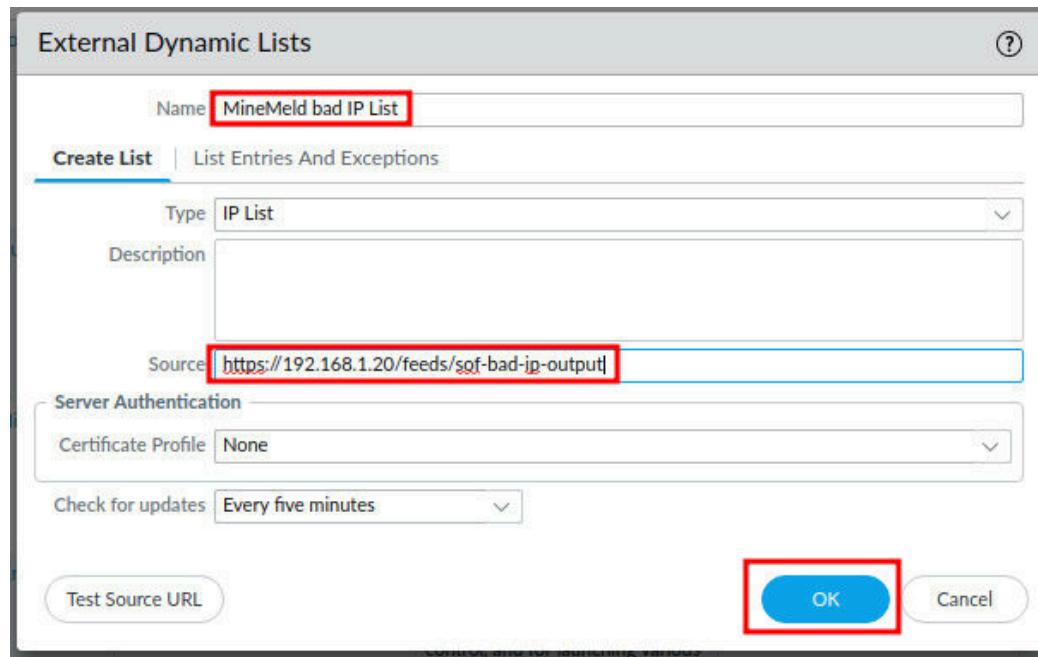
**sof-bad-ip-output** NODE

|               |   |  |                  |
|---------------|---|--|------------------|
| STATUS        |   |  |                  |
| CLASS         | minemeld.ft.redis.RedisSet  | OUTPUT   | DISABLED         |
| PROTOTYPE     | stdlib.feedMCGreenWithValue   | INPUTS   | bad-ip-processor |
| STATE         | STARTED   |  |                  |
| FEED BASE URL | <a href="https://192.168.1.20/feeds/sof-bad-ip-output">https://192.168.1.20/feeds/sof-bad-ip-output</a> | <a href="#">Open link in new tab</a><br><a href="#">Open link in new window</a><br><a href="#">Open link in incognito window</a><br><br><a href="#">Save link as...</a><br><div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px;">Copy link address</div><br><a href="#">Inspect</a> <a href="#">Ctrl+Shift+I</a> |                  |
| TAGS          |   |  |                  |
| # INDICATORS  | 32515   |  |                  |

15. Change focus back to the **firewall-a** tab in the *Chromium* web browser. Navigate to **Objects > External Dynamic Lists** and click **Add**.

| NAME  | LOCATION   | DESCRIPTION  | SOURCE  |
|---|------------|--|---|
| Palo Alto Networks - Tor exit IP addresses                        | Predefined | IP addresses supplied by multiple providers and validated with Palo Alto Networks threat intelligence data as active Tor exit nodes. Traffic from Tor exit nodes can serve a legitimate purpose, however, is disproportionately associated with malicious activity, especially in enterprise environments. | Palo Alto Networks  |
| Palo Alto Networks - Bulletproof IP addresses                     | Predefined | IP addresses that are provided by bulletproof hosting providers. Because bulletproof hosting providers place few, if any, restrictions on content, attackers can use these services to host and distribute malicious, illegal, and unethical material.   | Palo Alto Networks  |
| Palo Alto Networks - High risk IP addresses                       | Predefined | IP addresses that have recently been featured in threat activity advisories distributed by high-trust organizations. However, Palo Alto Networks does not have direct evidence of maliciousness for these IP addresses.  | Palo Alto Networks  |
| Palo Alto Networks - Known malicious IP addresses                 | Predefined | IP addresses that are currently used almost exclusively by malicious actors for malware distribution, command-and-control, and for launching various attacks.  | Palo Alto Networks  |
| <input checked="" type="checkbox"/> MineMeld high confidence list |            |  | https://192.168.1.254/?#objects::vsys1::objects/dynamic-block-lists |

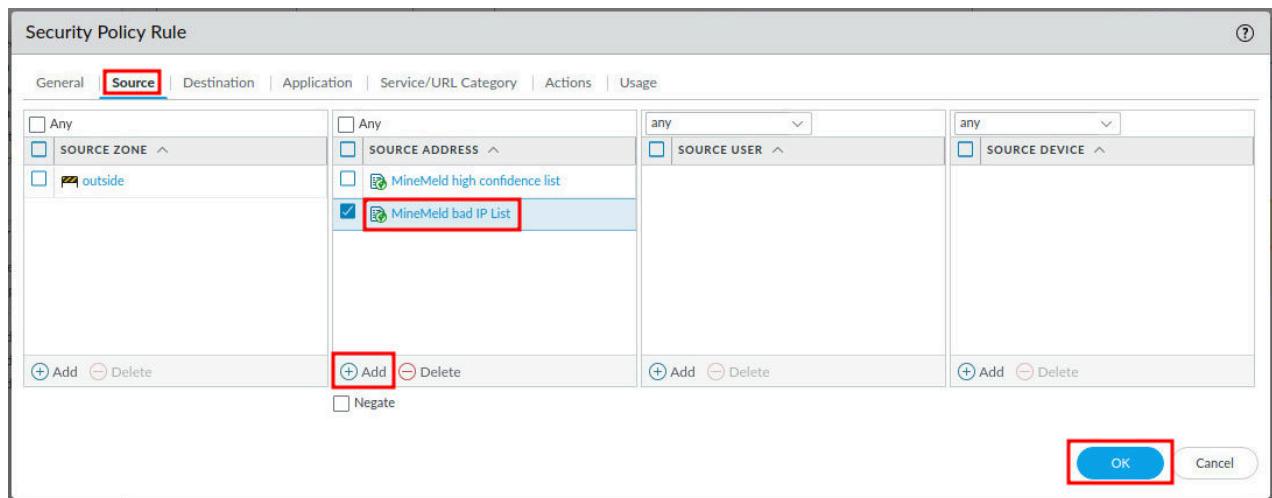
16. In the *External Dynamic Lists* dialog box, enter MineMeld bad IP List for the **Name**. Paste the *MineMeld output node URL* you copied from step 14 for the **Source**. Click **OK**.



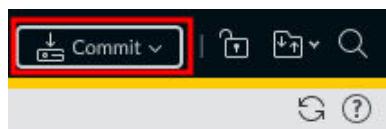
17. Navigate to **Policies > Security** and click the **outside-inside** security policy to open it.

| NAME                  | TAGS     | TYPE      | Source  |         |      |
|-----------------------|----------|-----------|---------|---------|------|
|                       |          |           | ZONE    | ADDRESS | USER |
| 1 outside-inside      | internal | universal | outside | any     | any  |
| 2 internal-inside-dmz | internal | universal | inside  | any     | any  |
| 3 egress-outside      | egress   | universal | dmz     | any     | any  |
| 4 intrazone-default   | none     | intrazone | any     | any     | any  |
| 5 interzone-default   | none     | interzone | any     | any     | any  |

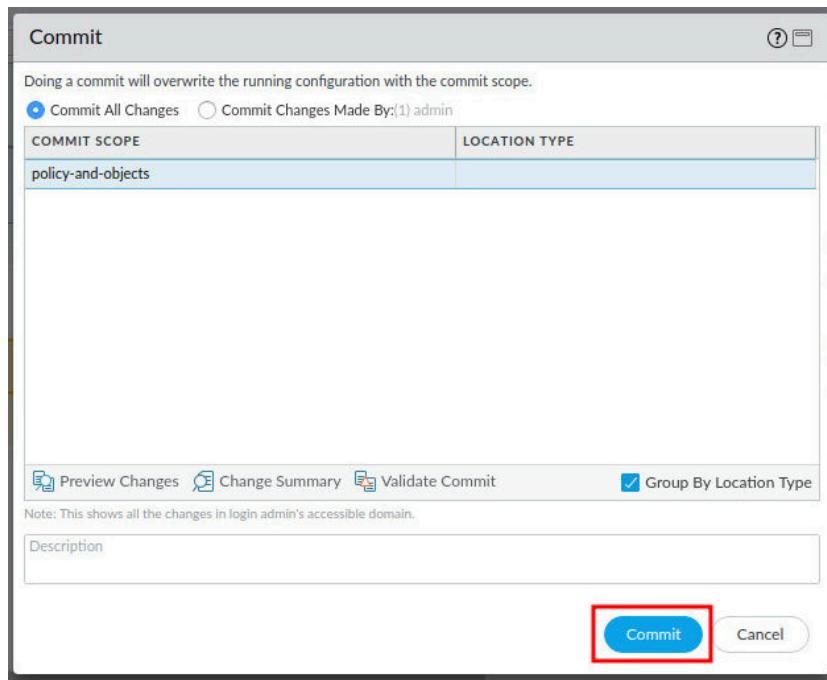
18. In the *Security Policy Rule* window, select the **Source** tab. In the *Source Address* window, click **Add** and select the **MineMeld bad IP List** EDL. Click **OK**.



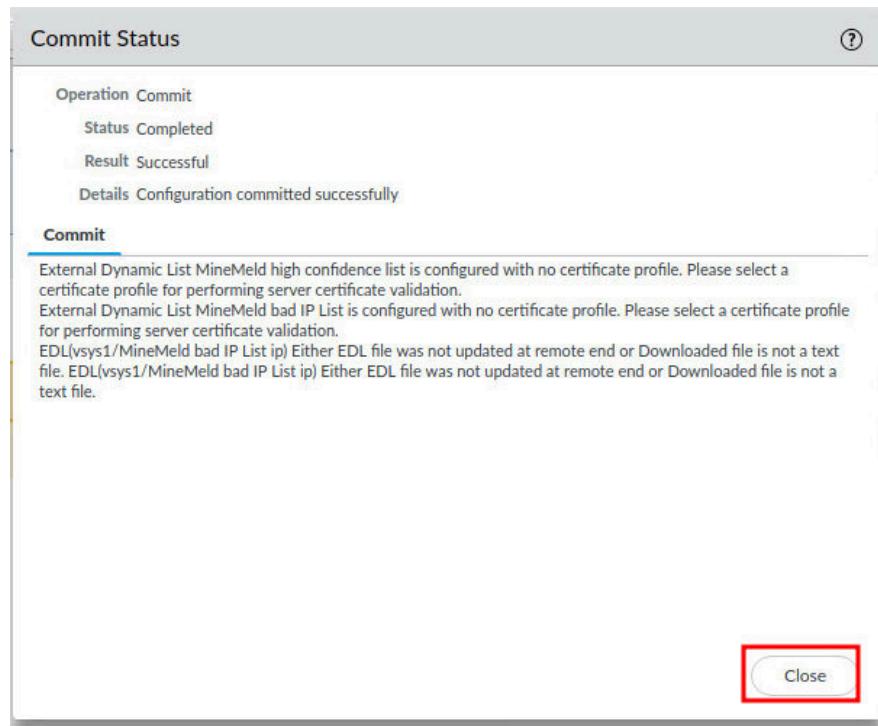
19. Click **Commit** on the Palo Alto Networks Firewall.



20. In the *Commit* window, click **Commit**.



21. In the *Commit Status* window, click **Close**. You may ignore the warnings.



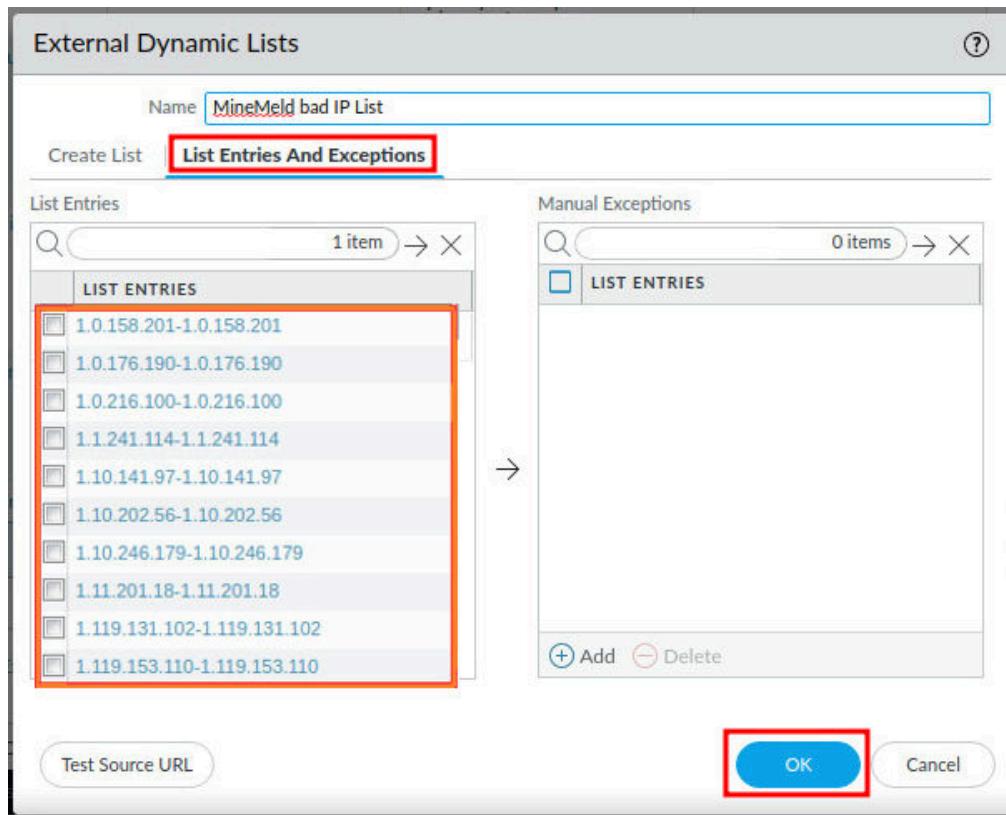
22. Navigate to **Objects > External Dynamic Lists**. Open the **MineMeld bad IP List**.

The screenshot shows the PA-VM interface with the following details:

- Header:** DASHBOARD, ACC, MONITOR, POLICIES, **OBJECTS** (highlighted), NETWORK, DEVICE.
- Left Sidebar:** Addresses, Address Groups, Regions, Dynamic User Groups, Applications, Application Groups, Application Filters, Services, Service Groups, Tags, Devices, GlobalProtect (with HIP Objects and HIP Profiles), **External Dynamic Lists** (highlighted with a red box), Custom Objects (with Data Patterns, Spyware, Vulnerability, URL Category).
- Table:** Displays external dynamic lists with the following columns: NAME, LOCATION, DESCRIPTION, and SOURCE.
- Data:**

| NAME  | LOCATION   | DESCRIPTION  | SOURCE                        |
|---|------------|--|-------------------------------|
| Palo Alto Networks - Bulletproof IP addresses     | Predefined | IP addresses that are provided by bulletproof hosting providers. Because bulletproof hosting providers place few, if any, restrictions on content, attackers can use these services to host and distribute malicious, illegal, and unethical material. | Palo Alto Net IP addresses    |
| Palo Alto Networks - High risk IP addresses       | Predefined | IP addresses that have recently been featured in threat activity advisories distributed by high-trust organizations. However, Palo Alto Networks does not have direct evidence of maliciousness for these IP addresses.                                | Palo Alto Net addresses       |
| Palo Alto Networks - Known malicious IP addresses | Predefined | IP addresses that are currently used almost exclusively by malicious actors for malware distribution, command-and-control, and for launching various attacks.  | Palo Alto Net malicious IP ad |
| MineMeld high confidence list                     |            |  | https://192.16                |
| MineMeld bad IP List                              |            |  | https://192.16 bad-ip-output  |

23. In the *External Dynamic Lists* window, select **List Entries And Exceptions**. View the *IP Address block list indicators* that MineMeld is now feeding the Palo Alto Networks Firewall.

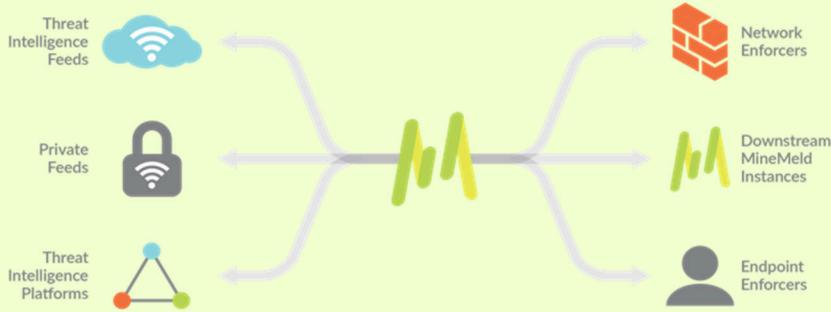


Resources like this are dependent on the sources of information they gather intelligence from. In this section (see step 3), the data source is **blocklist.de**. If that website is unavailable, there will be no data to aggregate, and the EDL (External Dynamic List) created above will not populate with IP address data.

Aggregators like *MineMeld* are especially helpful when there are multiple updating sources for data, as the loss of data from a single source would not eliminate all data for the EDL.



*MineMeld* simplifies the collection and correlation of intelligence across commercial threat intelligence feeds, open-source intelligence (OSINT) providers, threat intelligence platforms, ISACs, CERTs and other *MineMeld* users.



24. The lab is now complete; you may end your reservation.