SECURITY OPERATIONS FUNDAMENTALS V2

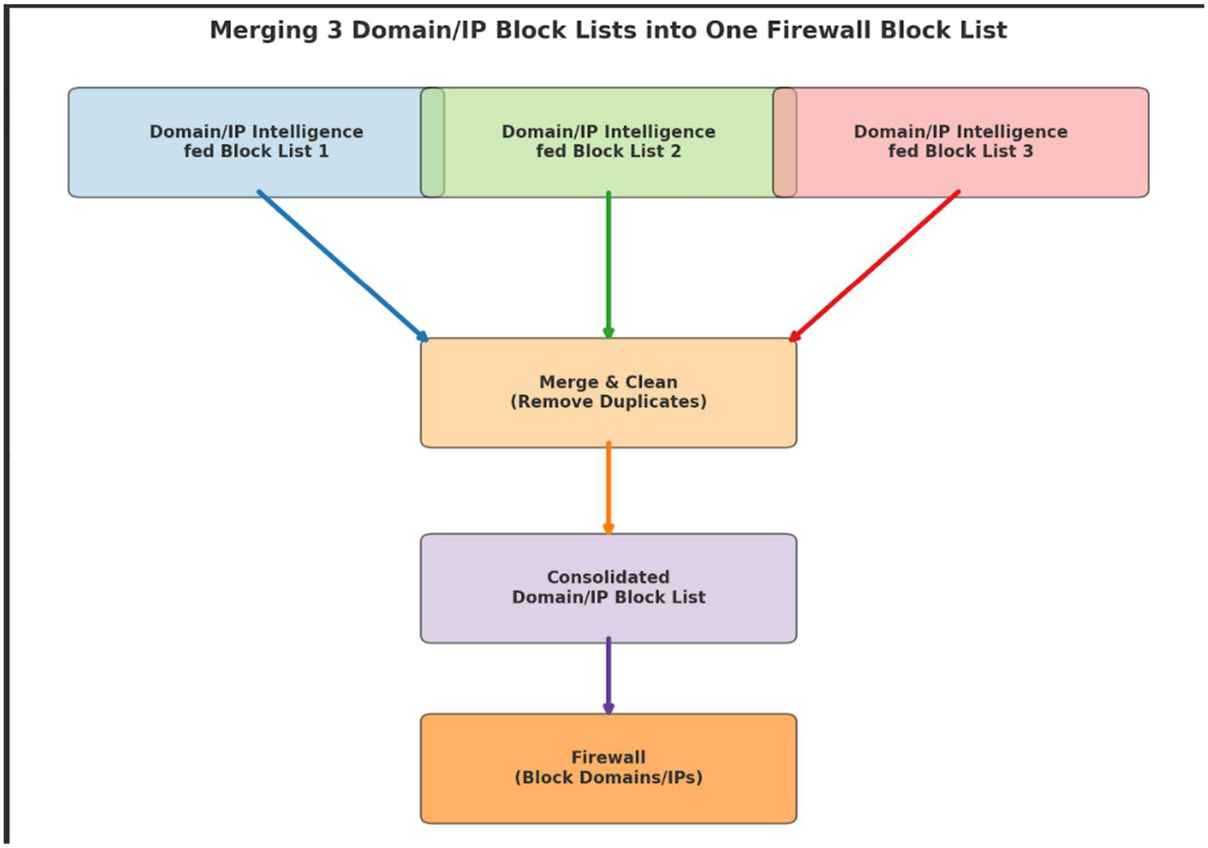
Lab 7: Threat Intelligence

Document Version: 2025-04-18

Contents

Introduction

In this lab, you will use python scripts to create a domain block list and an IP address blocklist. The scripts will use intelligence feeds from 3 different sources and combine them into one domain blocklist and one IP address blocklist. The scripts will also create localhost websites to post the blocklists on. You will then configure your firewall to use the blocklists to block traffic sourced from the entries in the blocklists.

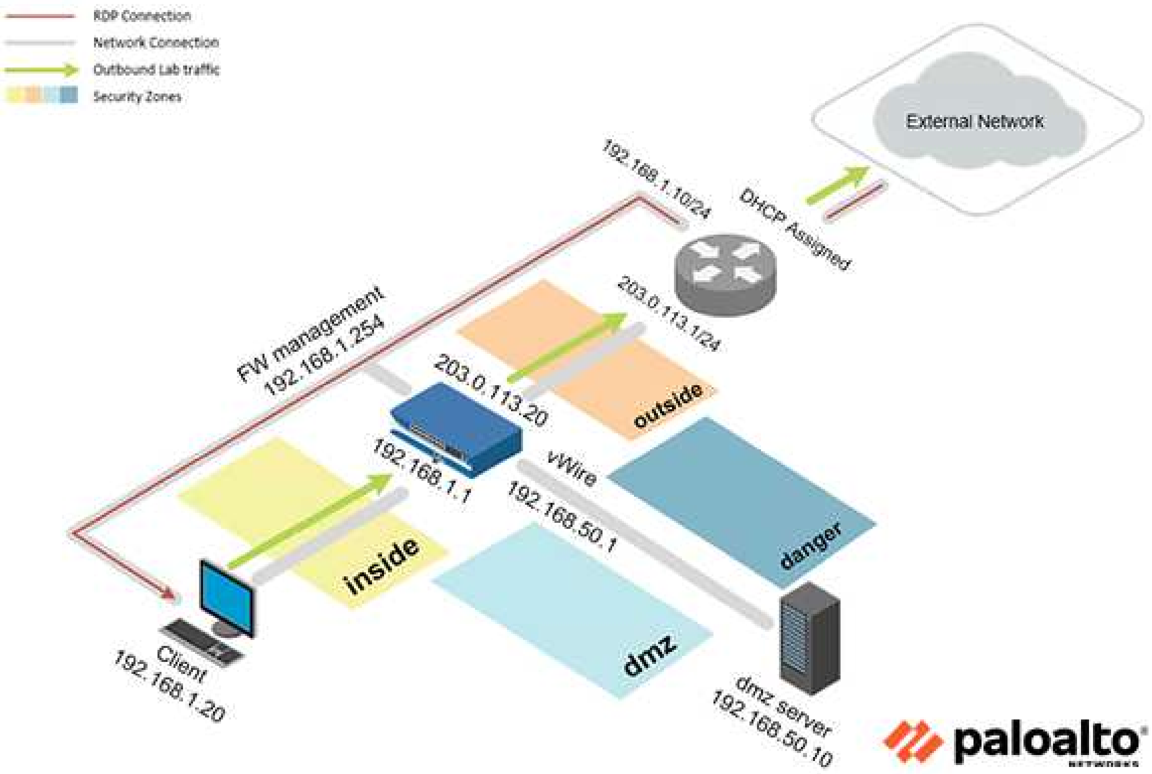

Merging 3 Domain/IP Block Lists into One Firewall Block List
Domain/IP Intelligence fed Block List 1
Domain/IP Intelligence fed Block List 2
Domain/IP Intelligence fed Block List 3
Merge & Clean (Remove Duplicates)
Consolidated Domain/IP Block List
Firewall (Block Domains/IPs)

Objective

In this lab, you will perform the following tasks:

- Explore and comprehend intelligence feed blocklists
- Understand how a python script can use intelligence feeds to create a domain blocklist and or IP blocklist from intelligence feeds
- Execute python scripts to create blocklists
- Configure the firewall appliance to use the blocklists to block traffic from malicious sites

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.

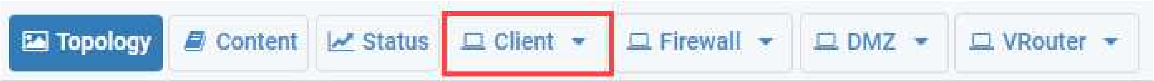
Virtual Machine	IP Address	Account (if needed)	Password (if needed)
Client	192.168.1.20	`lab-user`	`Pal0Alt0!`
DMZ	192.168.50.10	`root`	`Pal0Alt0!`
Firewall	192.168.1.254	`admin`	`Pal0Alt0!`

1 Threat Intelligence

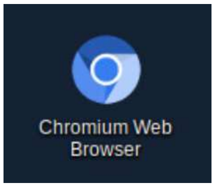
1.0 Load Lab Configuration

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

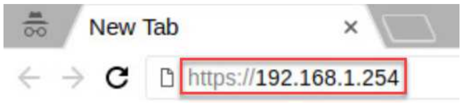
1. Click on the **Client** tab to access the client PC.



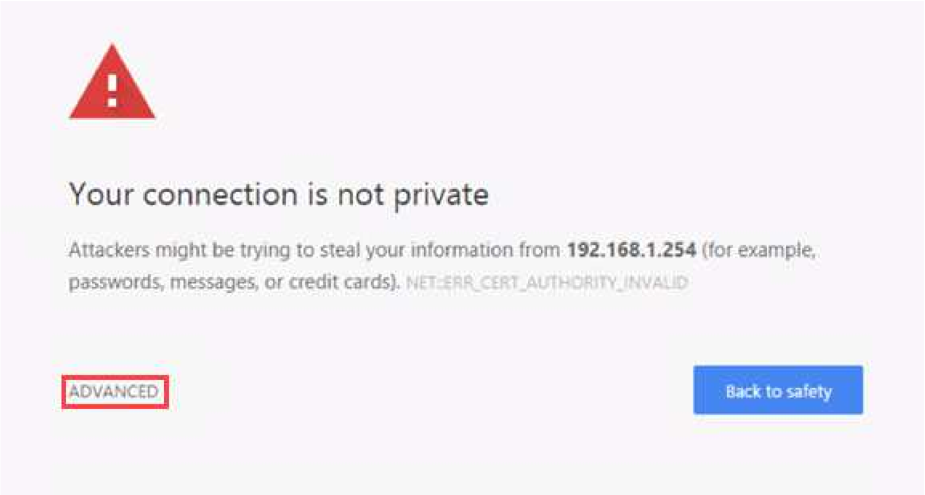
2. Log in to the client PC with the username `lab-user` and password `Pal0Alt0!`.
3. Double-click the **Chromium** icon located on the desktop.



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.

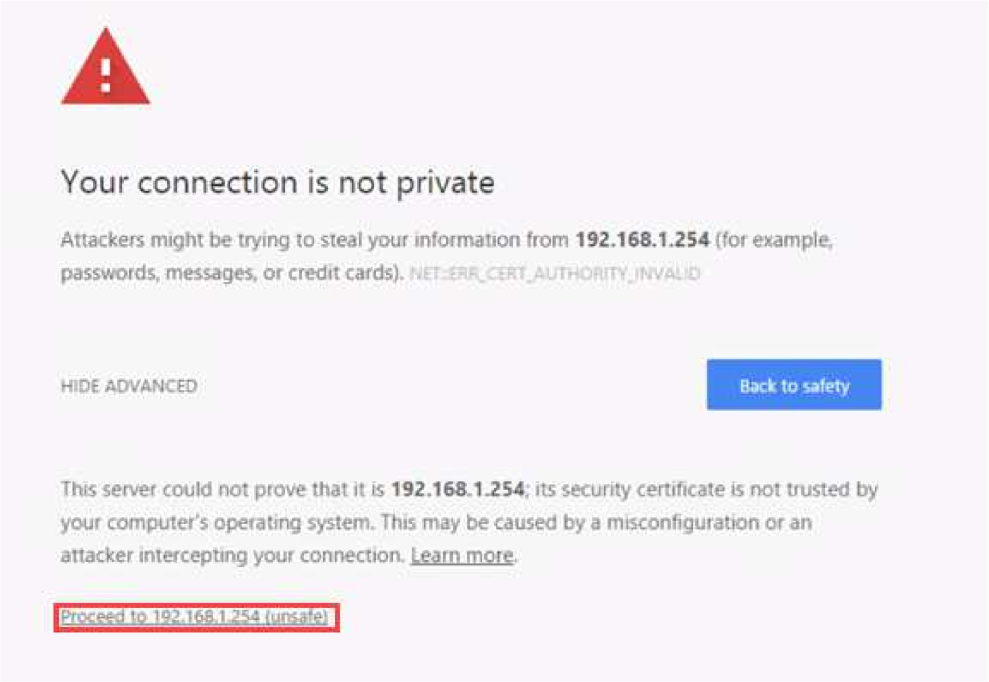


Your connection is not private

Attackers might be trying to steal your information from **192.168.1.254** (for example, passwords, messages, or credit cards). NET::ERR_CERT_AUTHORITY_INVALID

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)**.



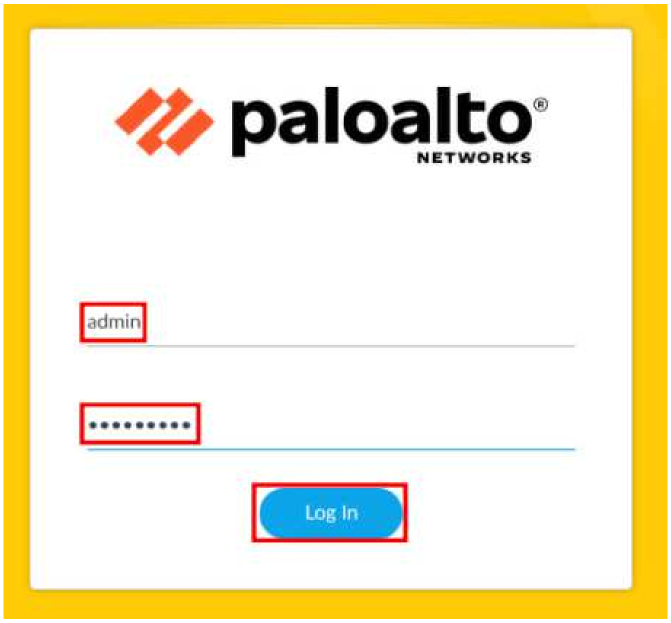
Your connection is not private

Attackers might be trying to steal your information from **192.168.1.254** (for example, passwords, messages, or credit cards). NET::ERR_CERT_AUTHORITY_INVALID

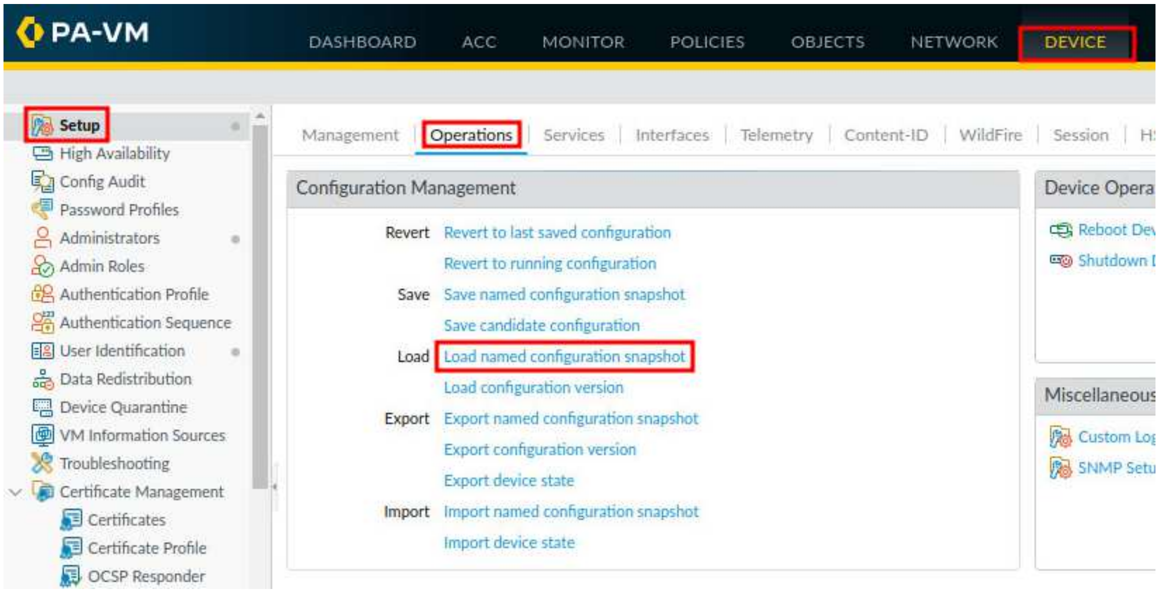
This server could not prove that it is **192.168.1.254**; its security certificate is not trusted by your computer's operating system. This may be caused by a misconfiguration or an attacker intercepting your connection. Learn more.

Proceed to 192.168.1.254 (unsafe)

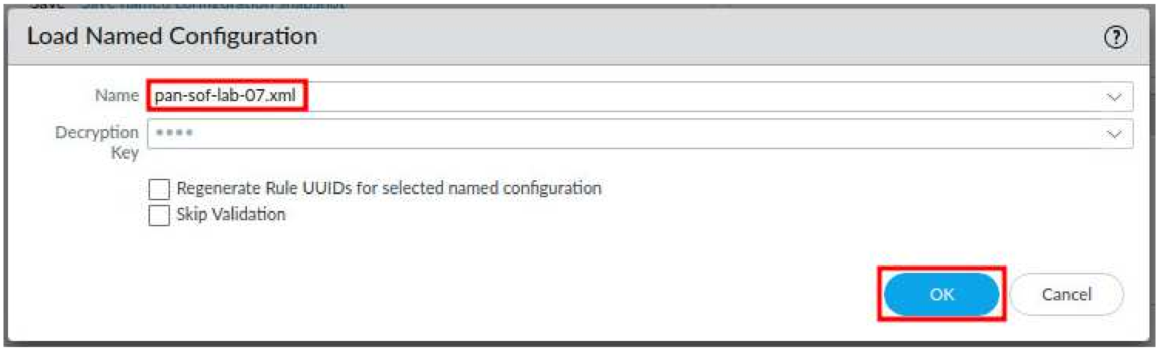
7. Log in to the Firewall web interface as username `admin`, password `Pal0Alt0!`.



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



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



Load Named Configuration

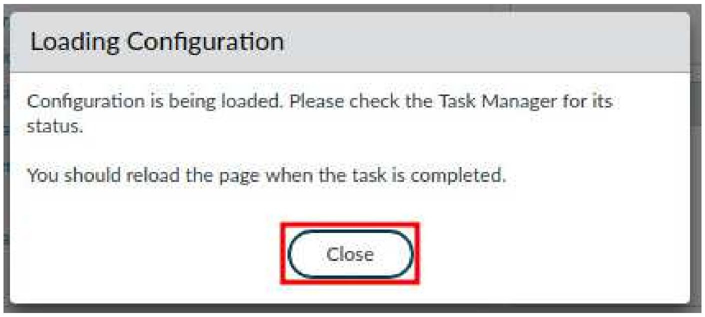
Name: pan-sof-lab-07.xml

Decryption Key: ••••

□ Regenerate Rule UUIDs for selected named configuration

□ Skip Validation

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.

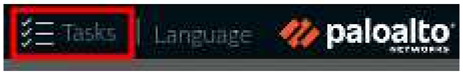


Loading Configuration

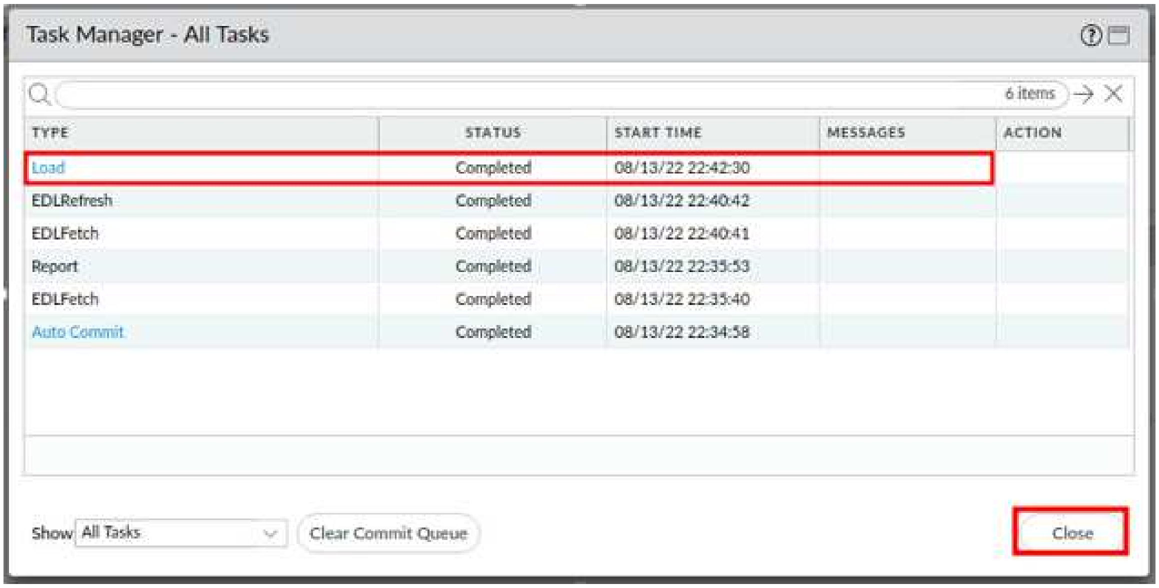
Configuration is being loaded. Please check the Task Manager for its status.

You should reload the page when the task is completed.

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

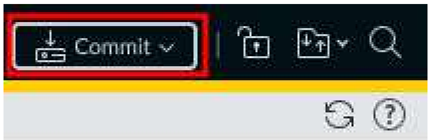


12. In the *Task Manager – All Tasks* window, verify that the *Load* type has successfully completed. Click **Close**.

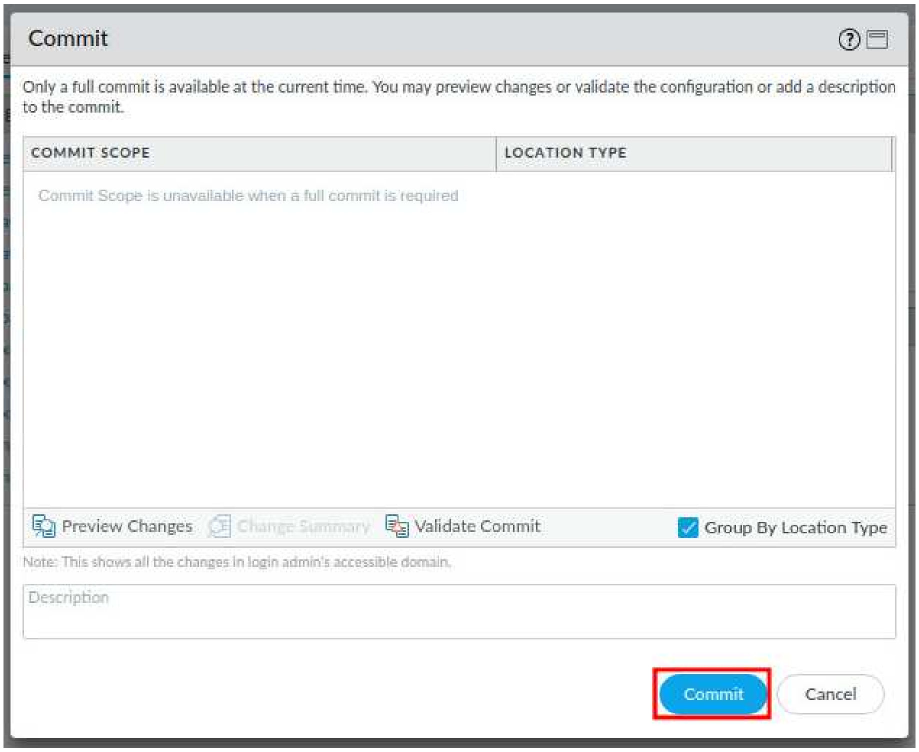


TYPE	STATUS	START TIME	MESSAGES	ACTION
Load	Completed	08/13/22 22:42:30		
EDLRefresh	Completed	08/13/22 22:40:42		
EDLFetch	Completed	08/13/22 22:40:41		
Report	Completed	08/13/22 22:35:53		
EDLFetch	Completed	08/13/22 22:35:40		
Auto Commit	Completed	08/13/22 22:34:58		

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.



Commit

Only a full commit is available at the current time. You may preview changes or validate the configuration or add a description to the commit.

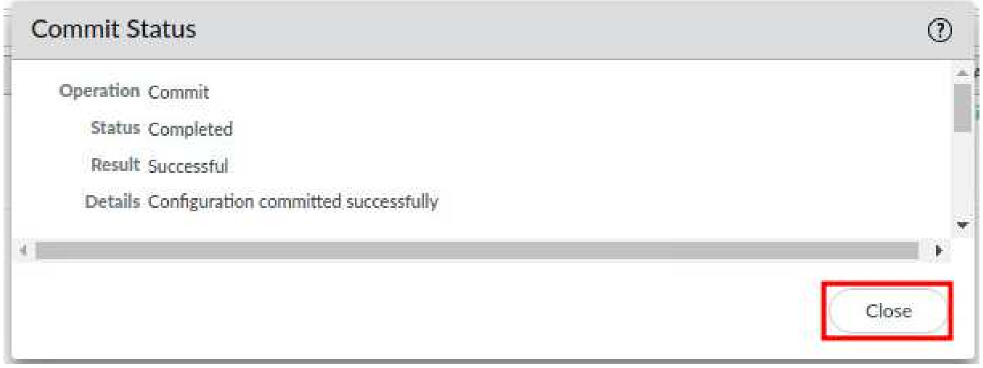
COMMIT SCOPE	LOCATION TYPE
Commit Scope is unavailable when a full commit is required	

Preview Changes Change Summary Validate Commit ☑ Group By Location Type

Note: This shows all the changes in login admin's accessible domain.

Description

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



Commit Status

Operation Commit
Status Completed
Result Successful
Details Configuration committed successfully

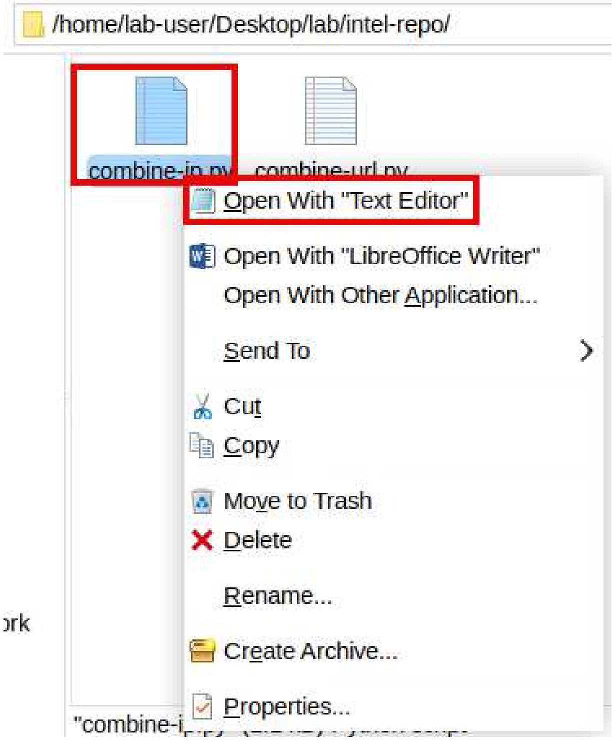
The commit process takes changes made to the Firewall and copies them to the running configuration, which will activate all configuration changes since the last commit.

1.1 Examine and Run the IP Blocking List Intelligence Script

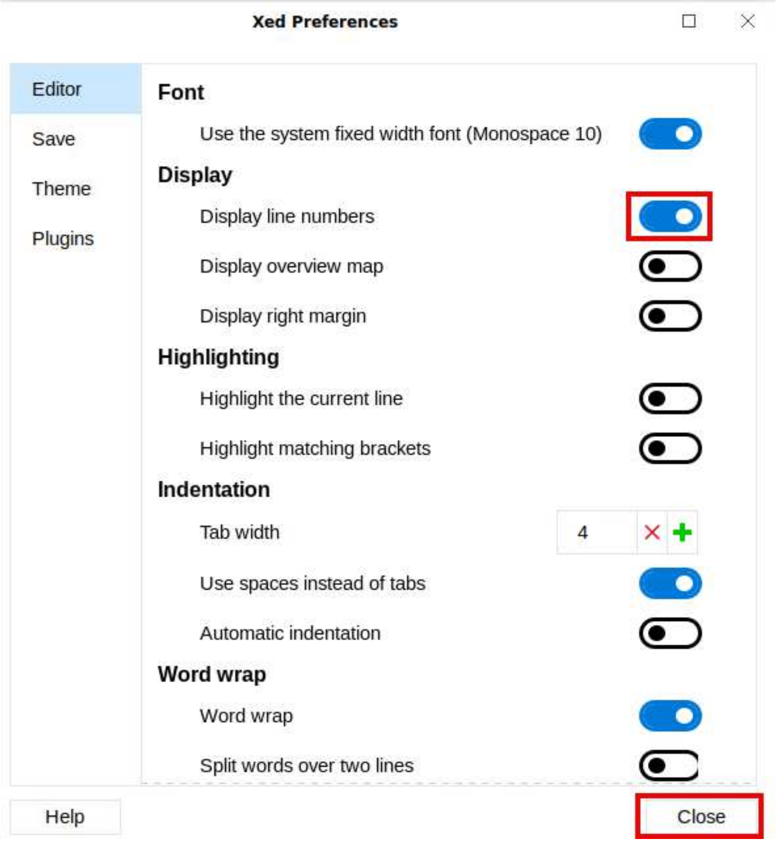
In this section, you will examine and run the *combine-ip.py* python script. This script will use three cybersecurity intelligence feeds to create one IP blocklist that will be posted on a localhost website.

1. While on the *Client*, open the **lab** folder on the Desktop. Then, navigate to the **intel-repo** folder within it.

2. Right-click the **combine-ip.py** and open it with the text editor.



3. In the new text editor window, navigate to **Edit** and enable **Display line numbers** to help you examine the python script. Click **Close** in the *Xed Preferences* window.



Font

Use the system fixed width font (Monospace 10) ☑

Display

Display line numbers ☑

Display overview map ☐

Display right margin ☐

Highlighting

Highlight the current line ☐

Highlight matching brackets ☐

Indentation

Tab width 4

Use spaces instead of tabs ☑

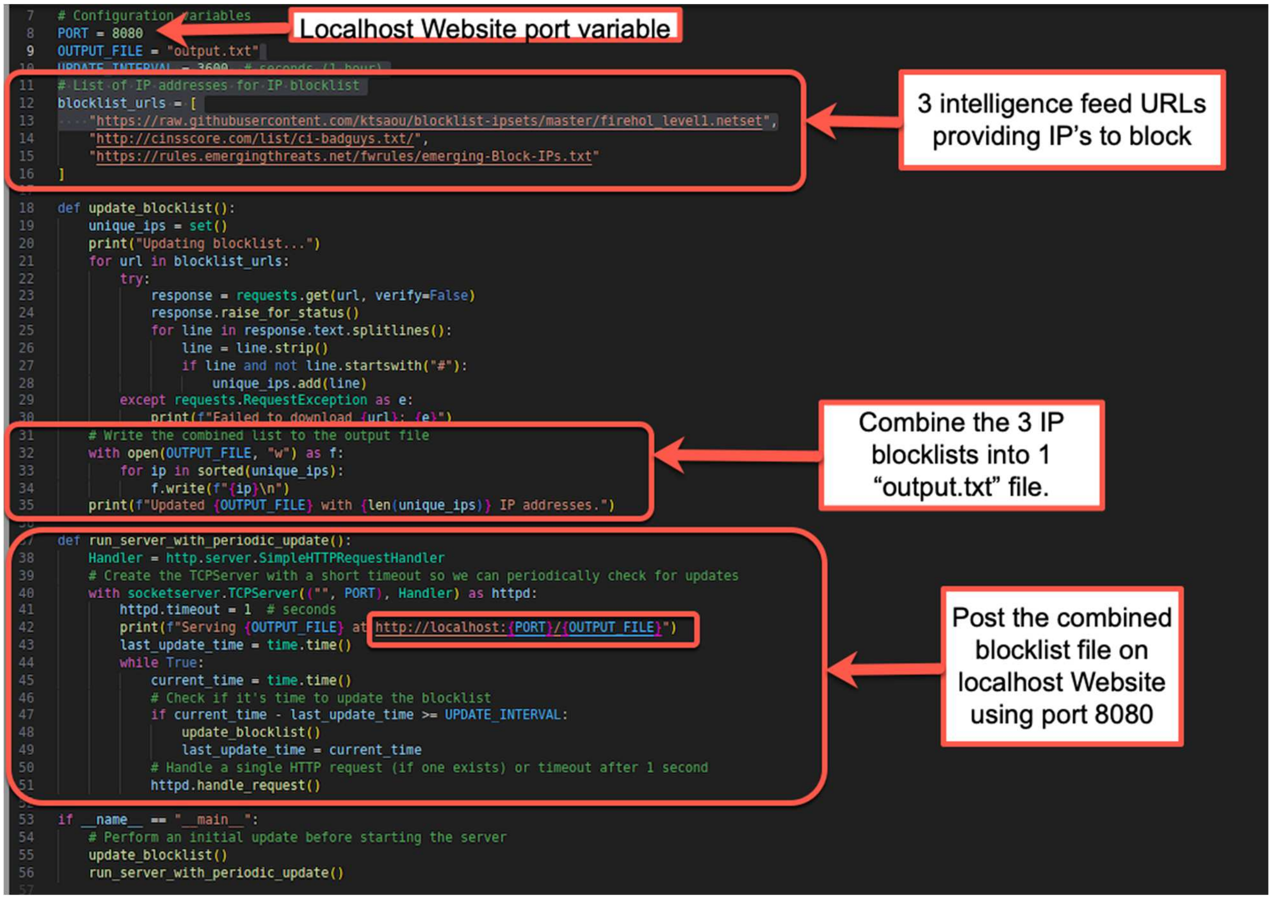
Automatic indentation ☐

Word wrap

Word wrap ☑

Split words over two lines ☐

4. Examine the *combine-ip.py* python script using the screenshot below as your guide. Lines 12-16 provide the URL addresses for the intelligence feeds that will provide the IP blocklists. These are IP addresses of known bad Internet hosts identified via the intelligence sources. Lines 31-35 combine the IP blocklists from the three intelligence repositories into one *output.txt* file. Lines 37-51 post the *output.txt* on a localhost website using port *8080,* which was listed as a *PORT* variable in line 8 of the script.



```
 7  # Configuration variables
 8  PORT = 8080
 9  OUTPUT_FILE = "output.txt"
10  UPDATE_INTERVAL = 3600  # seconds (1 hour)
11  # List of IP addresses for IP blocklist
12  blocklist_urls = [
13      "https://raw.githubusercontent.com/ktsaou/blocklist-ipsets/master/firehol_level1.netset",
14      "http://cinsscore.com/list/ci-badguys.txt/",
15      "https://rules.emergingthreats.net/fwrules/emerging-Block-IPs.txt"
16  ]
17
18  def update_blocklist():
19      unique_ips = set()
20      print("Updating blocklist...")
21      for url in blocklist_urls:
22          try:
23              response = requests.get(url, verify=False)
24              response.raise_for_status()
25              for line in response.text.splitlines():
26                  line = line.strip()
27                  if line and not line.startswith("#"):
28                      unique_ips.add(line)
29          except requests.RequestException as e:
30              print(f"Failed to download {url}: {e}")
31      # Write the combined list to the output file
32      with open(OUTPUT_FILE, "w") as f:
33          for ip in sorted(unique_ips):
34              f.write(f"{ip}\n")
35      print(f"Updated {OUTPUT_FILE} with {len(unique_ips)} IP addresses.")
36
37  def run_server_with_periodic_update():
38      Handler = http.server.SimpleHTTPRequestHandler
39      # Create the TCPServer with a short timeout so we can periodically check for updates
40      with socketserver.TCPServer(("", PORT), Handler) as httpd:
41          httpd.timeout = 1  # seconds
42          print(f"Serving {OUTPUT_FILE} at http://localhost:{PORT}/{OUTPUT_FILE}")
43          last_update_time = time.time()
44          while True:
45              current_time = time.time()
46              # Check if it's time to update the blocklist
47              if current_time - last_update_time >= UPDATE_INTERVAL:
48                  update_blocklist()
49                  last_update_time = current_time
50              # Handle a single HTTP request (if one exists) or timeout after 1 second
51              httpd.handle_request()
52
53  if __name__ == "__main__":
54      # Perform an initial update before starting the server
55      update_blocklist()
56      run_server_with_periodic_update()
57
```

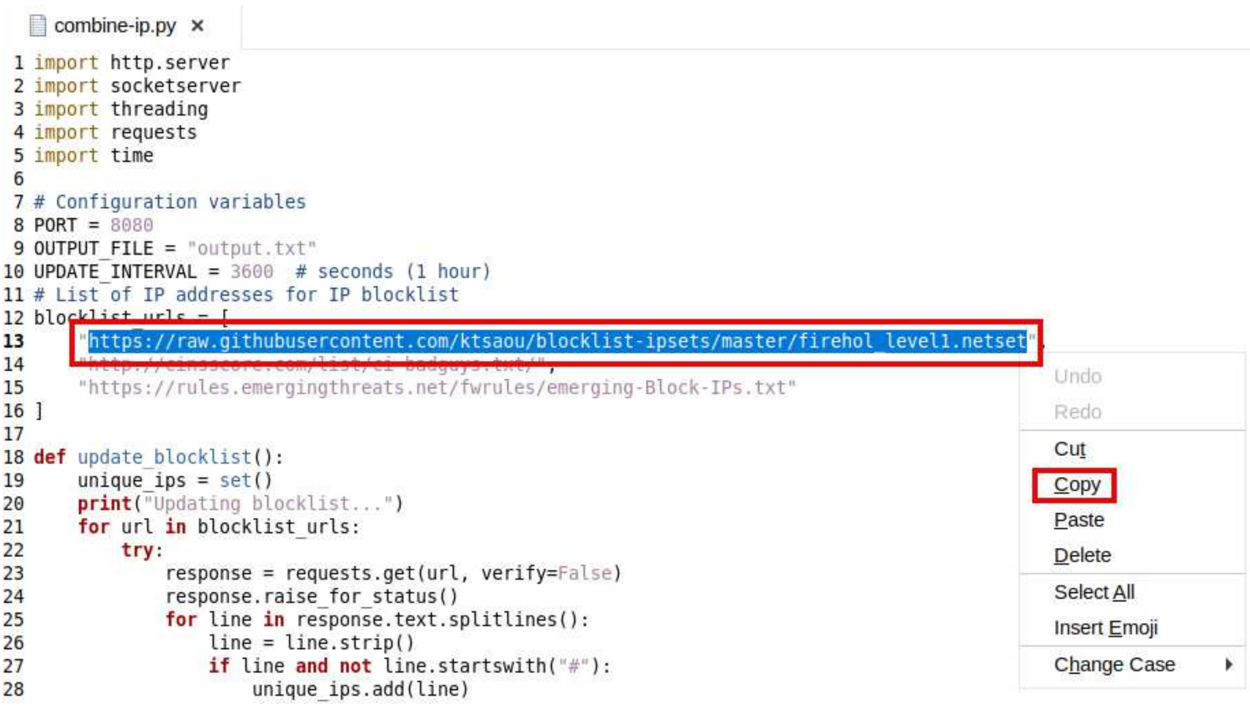
Localhost Website port variable

3 intelligence feed URLs providing IP's to block

Combine the 3 IP blocklists into 1 "output.txt" file.

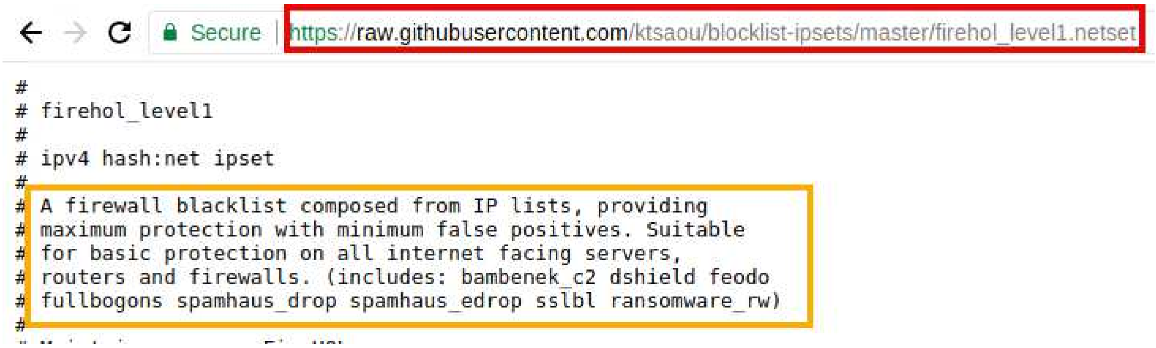
Post the combined blocklist file on localhost Website using port 8080

5. Copy the first blocklist URL listed from line 13, **https://raw.githubusercontent.com/ktsaou/blocklist-ipsets/master/firehol_level1.netset**.



```
1  import http.server
2  import socketserver
3  import threading
4  import requests
5  import time
6
7  # Configuration variables
8  PORT = 8080
9  OUTPUT_FILE = "output.txt"
10 UPDATE_INTERVAL = 3600  # seconds (1 hour)
11 # List of IP addresses for IP blocklist
12 blocklist_urls = [
13     "https://raw.githubusercontent.com/ktsaou/blocklist-ipsets/master/firehol_level1.netset",
14     "http://cinsscore.com/list/ci-badguys.txt/",
15     "https://rules.emergingthreats.net/fwrules/emerging-Block-IPs.txt"
16 ]
17
18 def update_blocklist():
19     unique_ips = set()
20     print("Updating blocklist...")
21     for url in blocklist_urls:
22         try:
23             response = requests.get(url, verify=False)
24             response.raise_for_status()
25             for line in response.text.splitlines():
26                 line = line.strip()
27                 if line and not line.startswith("#"):
28                     unique_ips.add(line)
```

6. Navigate to the *Chromium* web browser, paste the URL into a new open tab, and access the website with a list of bad IP addresses. Note the description of this blocklist:



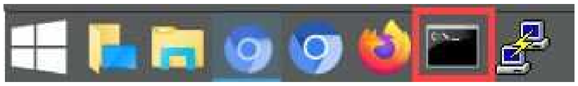
https://raw.githubusercontent.com/ktsaou/blocklist-ipsets/master/firehol_level1.netset

```
#
# firehol_level1
#
# ipv4 hash:net ipset
#
# A firewall blacklist composed from IP lists, providing
# maximum protection with minimum false positives. Suitable
# for basic protection on all internet facing servers,
# routers and firewalls. (includes: bambenek_c2 dshield feodo
# fullbogons spamhaus_drop spamhaus_edrop sslbl ransomware_rw)
#
```

"A firewall blacklist composed from IP lists, providing
maximum protection with minimum false positives. Suitable
for basic protection on all internet facing servers,
routers and firewalls. (includes: bambenek_c2 dshield feodo
fullbogons spamhaus_drop spamhaus_edrop sslbl ransomware_rw)"

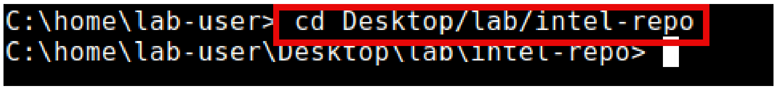
After you have completed your review, close the *combine-ip.py* text file and be sure not to save any accidental changes you may have made to this file.

7. Open the **Xfce Terminal** by clicking on the **Terminal** icon.



8. In the terminal, change to the *intel-repo* directory where the python scripts are located.

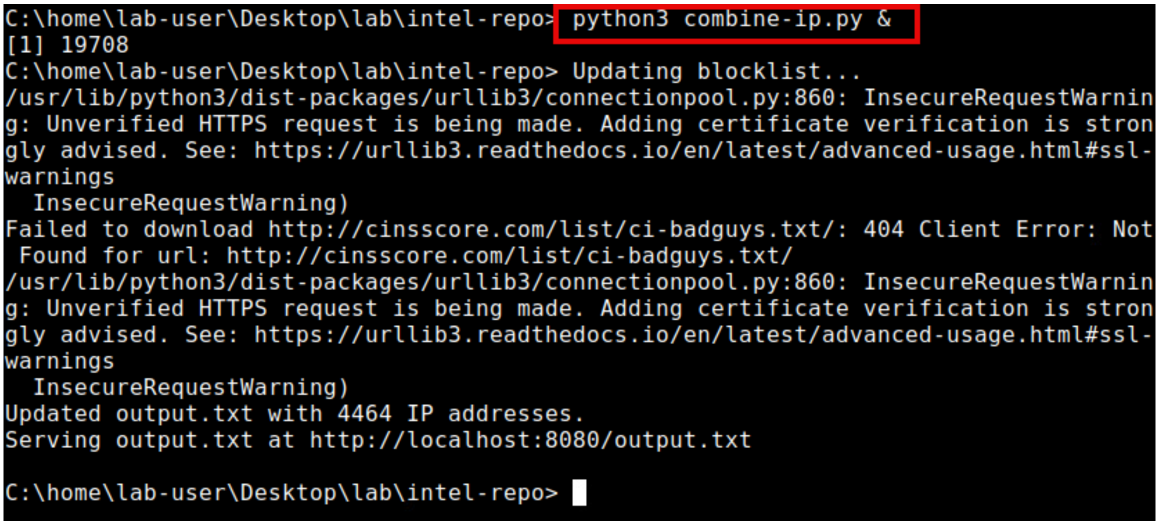
```
C:\home\lab-user> cd Desktop/lab/intel-repo
```



```
C:\home\lab-user> cd Desktop/lab/intel-repo
C:\home\lab-user\Desktop\lab\intel-repo> 
```

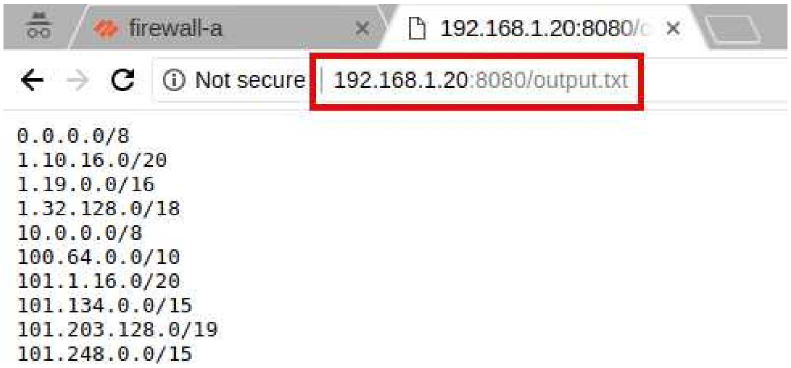
9. Run the **combine-ip.py** python script. After entering the command, press the **Enter** key once more to return to the prompt. Keep the terminal window open and uninterrupted.

```
C:\home\lab-user\Desktop\lab\intel-repo> python3 combine-ip.py &
```



```
C:\home\lab-user\Desktop\lab\intel-repo> python3 combine-ip.py &
[1] 19708
C:\home\lab-user\Desktop\lab\intel-repo> Updating blocklist...
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:860: InsecureRequestWarnin
g: Unverified HTTPS request is being made. Adding certificate verification is stron
gly advised. See: https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-
warnings
  InsecureRequestWarning)
Failed to download http://cinsscore.com/list/ci-badguys.txt/: 404 Client Error: Not
 Found for url: http://cinsscore.com/list/ci-badguys.txt/
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:860: InsecureRequestWarnin
g: Unverified HTTPS request is being made. Adding certificate verification is stron
gly advised. See: https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-
warnings
  InsecureRequestWarning)
Updated output.txt with 4464 IP addresses.
Serving output.txt at http://localhost:8080/output.txt

C:\home\lab-user\Desktop\lab\intel-repo> 
```

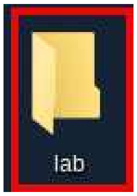
10. Navigate to the **Chromium** web browser, open a new tab, and enter the following URL to view the consolidated IP blocklist: `http://192.168.1.20:8080/output.txt`.



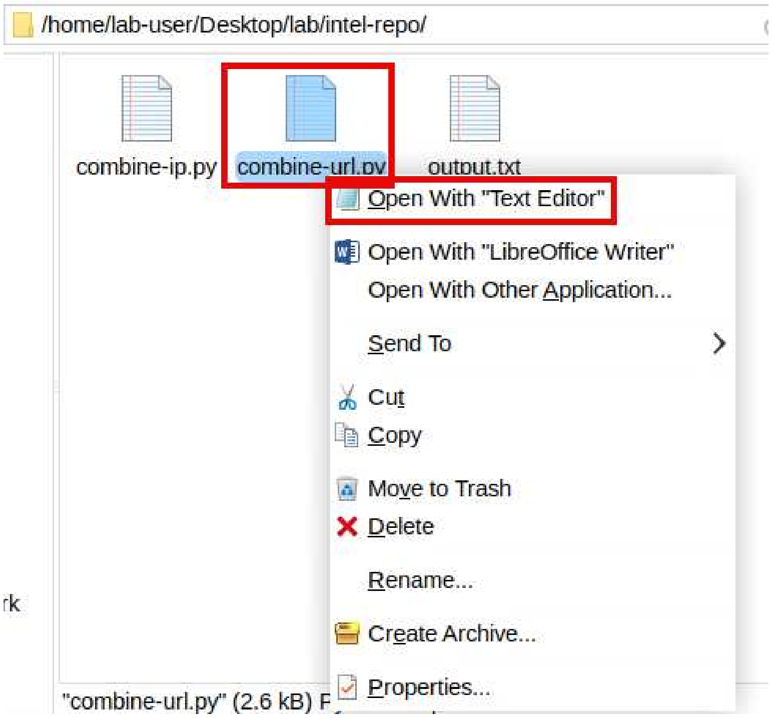
```
0.0.0.0/8
1.10.16.0/20
1.19.0.0/16
1.32.128.0/18
10.0.0.0/8
100.64.0.0/10
101.1.16.0/20
101.134.0.0/15
101.203.128.0/19
101.248.0.0/15
```

1.2 Examine and Run the Domain Blocking List Intelligence Feed Script

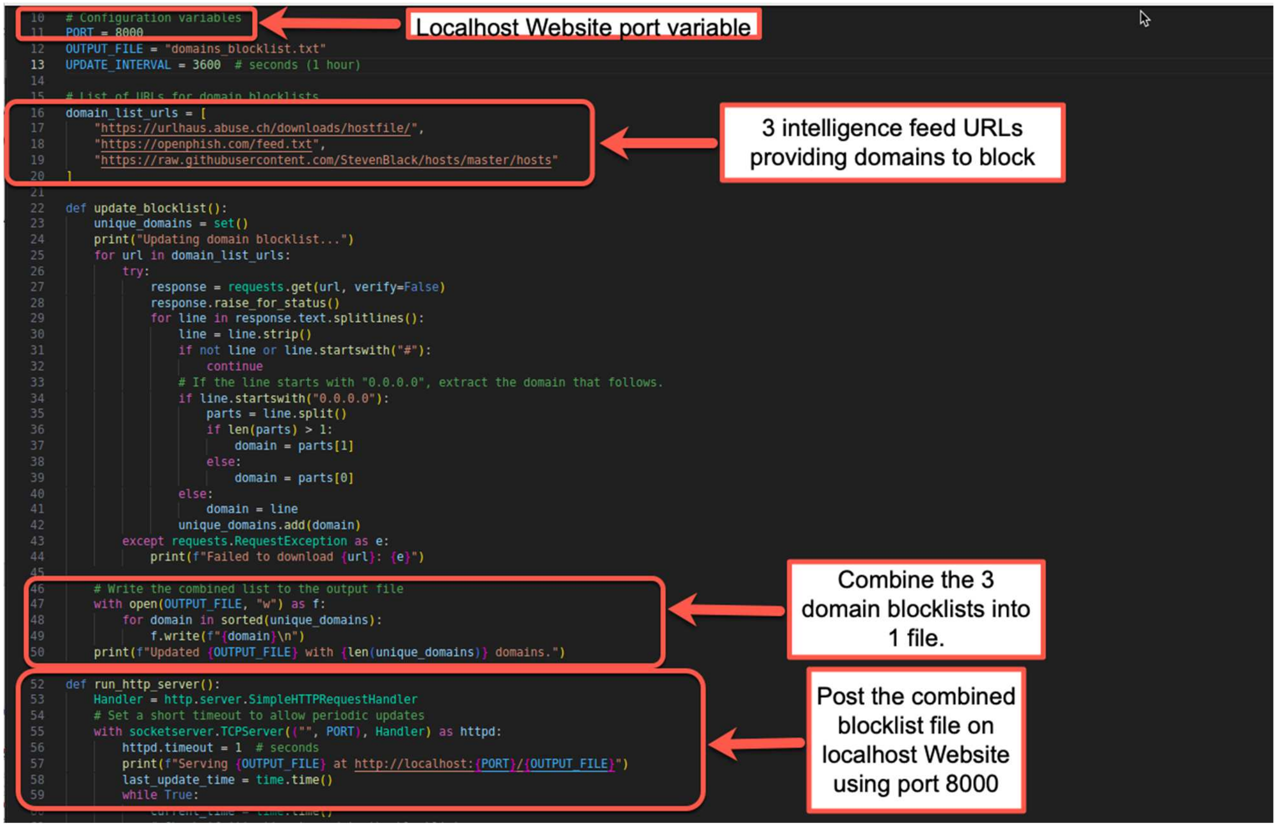
1. While on the *Client*, open the **lab** folder on the Desktop. Then, navigate to the **intel-repo** folder.



2. Right-click the **combine-url.py** and open it with the text editor.



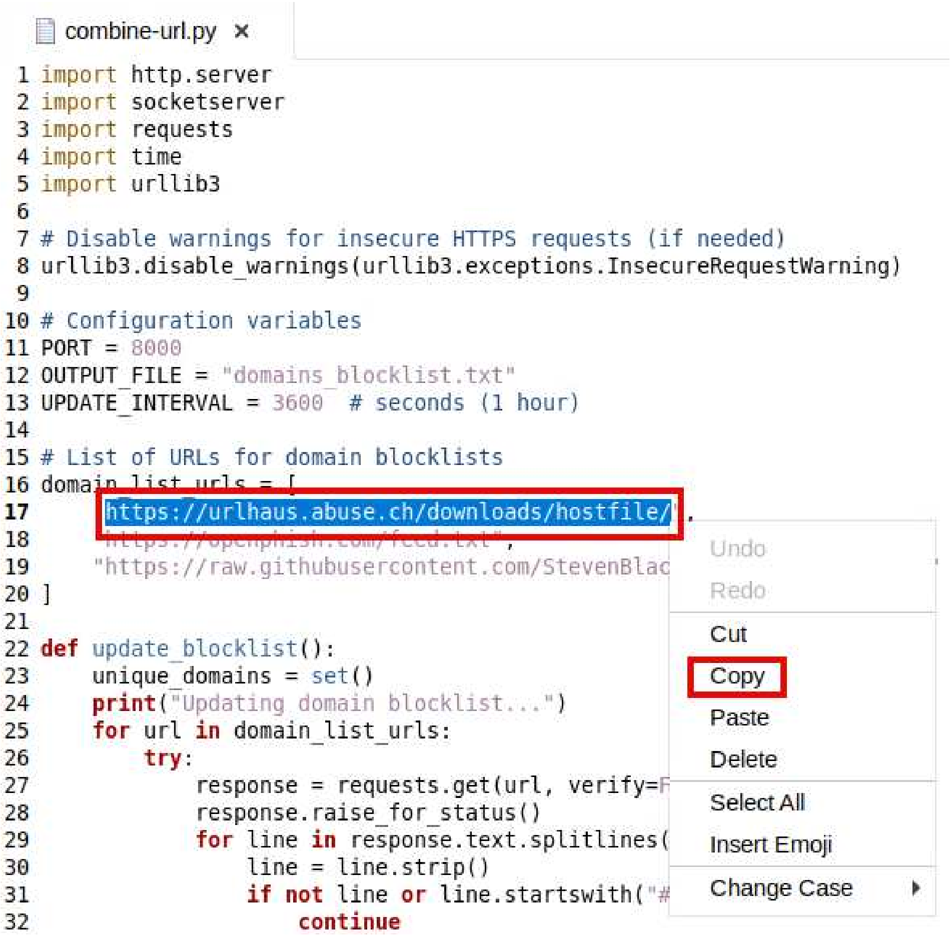
3. Examine the *combine-url.py* python script using the screenshot below as your guide. Note that the format for this domain name blocklist script is almost identical to the IP blocklist script. The main difference is that this script will post the domain blocklist on a localhost website using port *8000* instead of port *8080*.



```python
# Configuration variables
PORT = 8000
OUTPUT_FILE = "domains_blocklist.txt"
UPDATE_INTERVAL = 3600  # seconds (1 hour)

# List of URLs for domain blocklists
domain_list_urls = [
    "https://urlhaus.abuse.ch/downloads/hostfile/",
    "https://openphish.com/feed.txt",
    "https://raw.githubusercontent.com/StevenBlack/hosts/master/hosts"
]

def update_blocklist():
    unique_domains = set()
    print("Updating domain blocklist...")
    for url in domain_list_urls:
        try:
            response = requests.get(url, verify=False)
            response.raise_for_status()
            for line in response.text.splitlines():
                line = line.strip()
                if not line or line.startswith("#"):
                    continue
                # If the line starts with "0.0.0.0", extract the domain that follows.
                if line.startswith("0.0.0.0"):
                    parts = line.split()
                    if len(parts) > 1:
                        domain = parts[1]
                    else:
                        domain = parts[0]
                else:
                    domain = line
                unique_domains.add(domain)
        except requests.RequestException as e:
            print(f"Failed to download {url}: {e}")

    # Write the combined list to the output file
    with open(OUTPUT_FILE, "w") as f:
        for domain in sorted(unique_domains):
            f.write(f"{domain}\n")
    print(f"Updated {OUTPUT_FILE} with {len(unique_domains)} domains.")

def run_http_server():
    Handler = http.server.SimpleHTTPRequestHandler
    # Set a short timeout to allow periodic updates
    with socketserver.TCPServer(("", PORT), Handler) as httpd:
        httpd.timeout = 1  # seconds
        print(f"Serving {OUTPUT_FILE} at http://localhost:{PORT}/{OUTPUT_FILE}")
        last_update_time = time.time()
        while True:
            current_time = time.time()
```

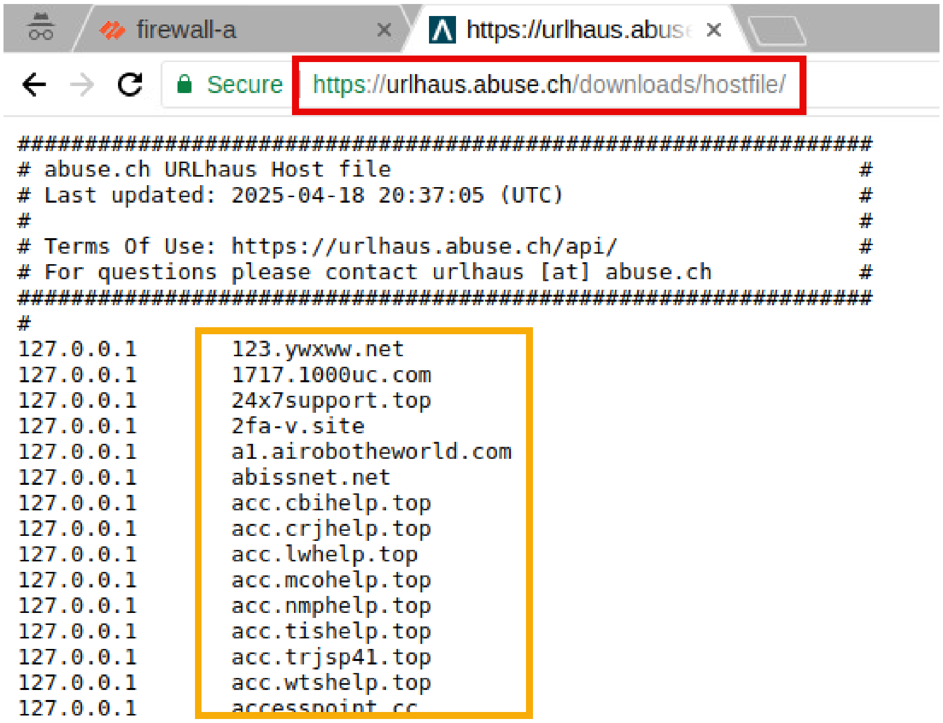
Localhost Website port variable
3 intelligence feed URLs providing domains to block
Combine the 3 domain blocklists into 1 file.
Post the combined blocklist file on localhost Website using port 8000

4. Copy the first blocklist URL listed in line 17, **https://urlhaus.abuse.ch/downloads/hostfile/**.



```
1 import http.server
2 import socketserver
3 import requests
4 import time
5 import urllib3
6
7 # Disable warnings for insecure HTTPS requests (if needed)
8 urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)
9
10 # Configuration variables
11 PORT = 8000
12 OUTPUT_FILE = "domains_blocklist.txt"
13 UPDATE_INTERVAL = 3600  # seconds (1 hour)
14
15 # List of URLs for domain blocklists
16 domain_list_urls = [
17     "https://urlhaus.abuse.ch/downloads/hostfile/",
18     "https://openphish.com/feed.txt",
19     "https://raw.githubusercontent.com/StevenBlac
20 ]
21
22 def update_blocklist():
23     unique_domains = set()
24     print("Updating domain blocklist...")
25     for url in domain_list_urls:
26         try:
27             response = requests.get(url, verify=F
28             response.raise_for_status()
29             for line in response.text.splitlines(
30                 line = line.strip()
31                 if not line or line.startswith("#
32                     continue
```

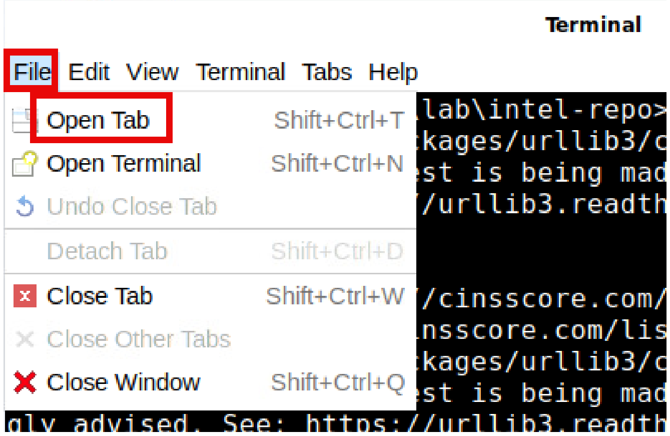
5. Navigate to the **Chromium** web browser, paste the URL into a new open tab, and access the website with a list of bad domains. Note that this blocklist contains domain names.



```
################################################################
# abuse.ch URLhaus Host file                                   #
# Last updated: 2025-04-18 20:37:05 (UTC)                      #
#                                                              #
# Terms Of Use: https://urlhaus.abuse.ch/api/                  #
# For questions please contact urlhaus [at] abuse.ch           #
################################################################
#
127.0.0.1       123.ywxww.net
127.0.0.1       1717.1000uc.com
127.0.0.1       24x7support.top
127.0.0.1       2fa-v.site
127.0.0.1       a1.airobotheworld.com
127.0.0.1       abissnet.net
127.0.0.1       acc.cbihelp.top
127.0.0.1       acc.crjhelp.top
127.0.0.1       acc.lwhelp.top
127.0.0.1       acc.mcohelp.top
127.0.0.1       acc.nmphelp.top
127.0.0.1       acc.tishelp.top
127.0.0.1       acc.trjsp41.top
127.0.0.1       acc.wtshelp.top
127.0.0.1       accesspoint.cc
```

After you have completed your review, close the *combine-url.py* text file and be sure not to save any accidental changes you may have made to this file.

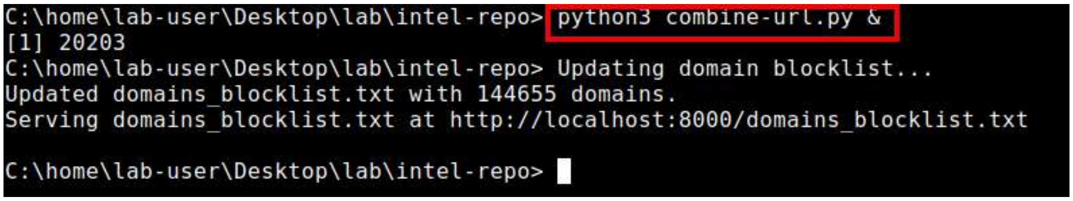
11. Navigate to your previously opened terminal and select **File > Open Tab**.



```
lab\intel-repo>
kages/urllib3/c
st is being mad
/urllib3.readth

/cinsscore.com/
nsscore.com/lis
kages/urllib3/c
st is being mad
gly advised. See: https://urllib3.readth
```

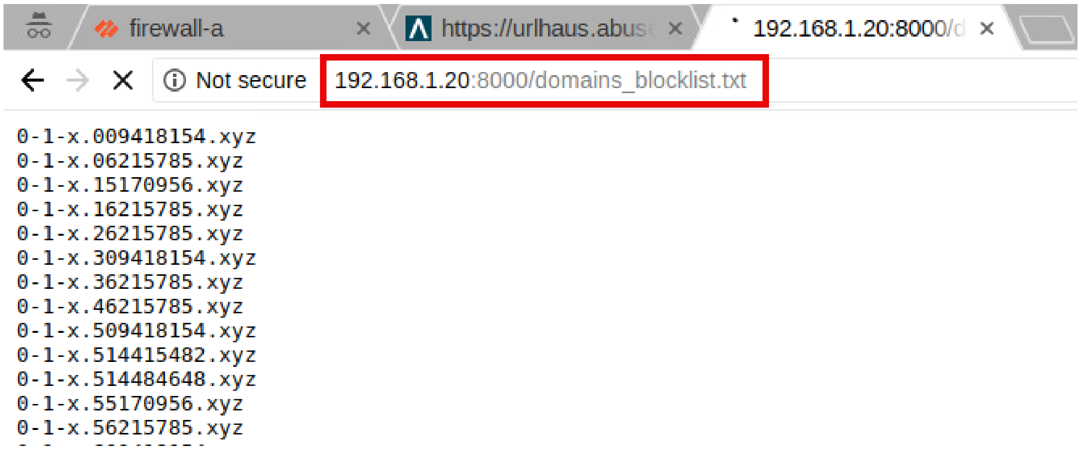
12. Run the **combine-url.py** python script. After entering the command, press the **Enter** key once more to return to the prompt. Keep the terminal window open and uninterrupted.

```
C:\home\lab-user\Desktop\lab\intel-repo> python3 combine-url.py &
```



```
C:\home\lab-user\Desktop\lab\intel-repo> python3 combine-url.py &
[1] 20203
C:\home\lab-user\Desktop\lab\intel-repo> Updating domain blocklist...
Updated domains_blocklist.txt with 144655 domains.
Serving domains_blocklist.txt at http://localhost:8000/domains_blocklist.txt

C:\home\lab-user\Desktop\lab\intel-repo> 
```

6. Navigate to the **Chromium** web browser, open a new tab, and enter the following URL to view the consolidated domain name blocklist: `http://192.168.1.20:8000/domains_blocklist.txt`.

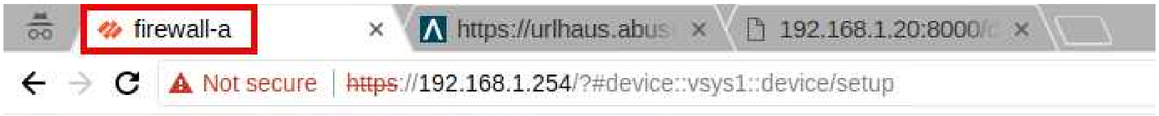


```
0-1-x.009418154.xyz
0-1-x.06215785.xyz
0-1-x.15170956.xyz
0-1-x.16215785.xyz
0-1-x.26215785.xyz
0-1-x.309418154.xyz
0-1-x.36215785.xyz
0-1-x.46215785.xyz
0-1-x.509418154.xyz
0-1-x.514415482.xyz
0-1-x.514484648.xyz
0-1-x.55170956.xyz
0-1-x.56215785.xyz
```

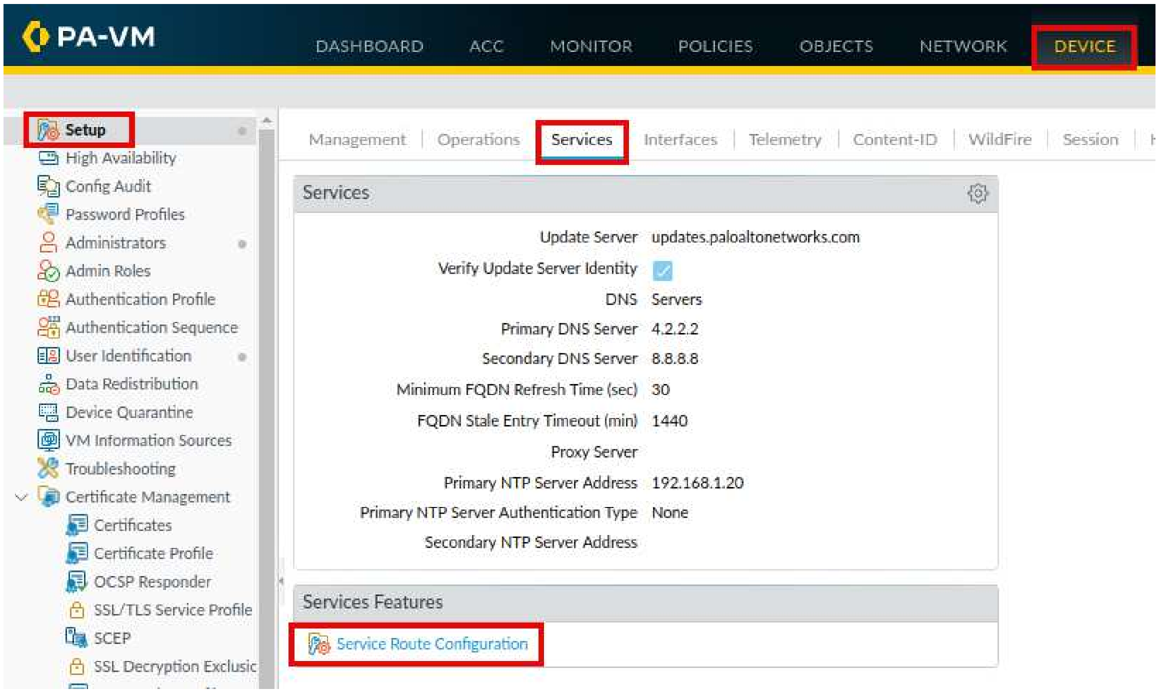
1.3 Configure an External Dynamic List (EDL) on the Firewall Appliance Using the Python Script Blocklists

In this section, you will configure an *External Dynamic List (EDL)* on the firewall to use the python-scripted blocklists and then use the EDL in a security policy rule to block incoming traffic.

1. In the *Chromium* web browser, click on the **firewall-a** tab to return to the firewall web interface..



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



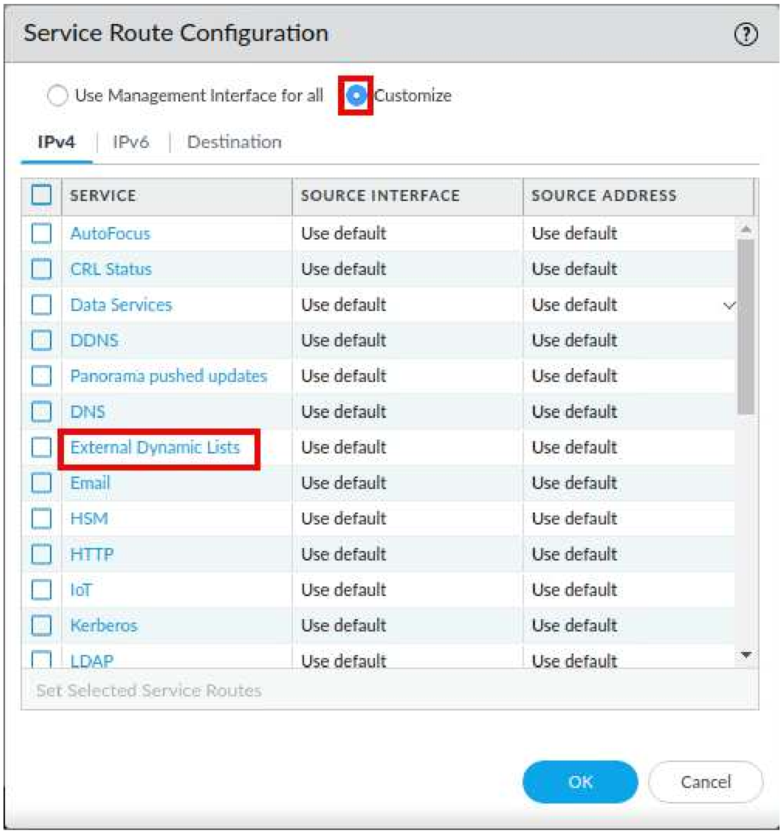
Services

Update Server	updates.paloaltonetworks.com
Verify Update Server Identity	☑
DNS	Servers
Primary DNS Server	4.2.2.2
Secondary DNS Server	8.8.8.8
Minimum FQDN Refresh Time (sec)	30
FQDN Stale Entry Timeout (min)	1440
Proxy Server	
Primary NTP Server Address	192.168.1.20
Primary NTP Server Authentication Type	None
Secondary NTP Server Address	

Services Features

Service Route Configuration

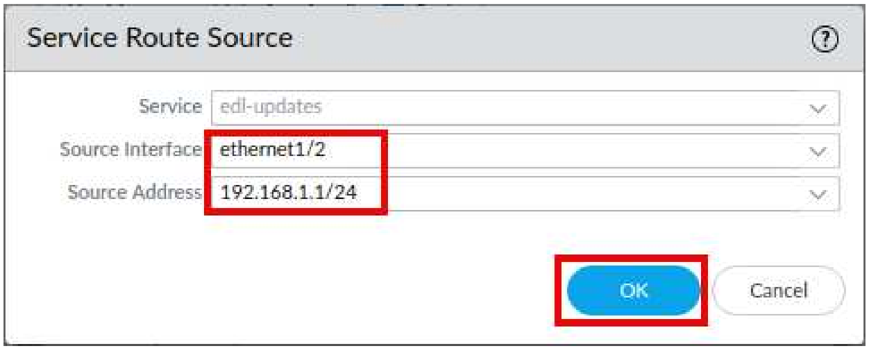
3. In the *Service Route Configuration* dialog box, select **Customize**. Click on **External Dynamic Lists**.


Service Route Configuration

○ Use Management Interface for all ● Customize

□	SERVICE	SOURCE INTERFACE	SOURCE ADDRESS
□	AutoFocus	Use default	Use default
□	CRL Status	Use default	Use default
□	Data Services	Use default	Use default
□	DDNS	Use default	Use default
□	Panorama pushed updates	Use default	Use default
□	DNS	Use default	Use default
□	External Dynamic Lists	Use default	Use default
□	Email	Use default	Use default
□	HSM	Use default	Use default
□	HTTP	Use default	Use default
□	IoT	Use default	Use default
□	Kerberos	Use default	Use default
□	LDAP	Use default	Use default

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.**



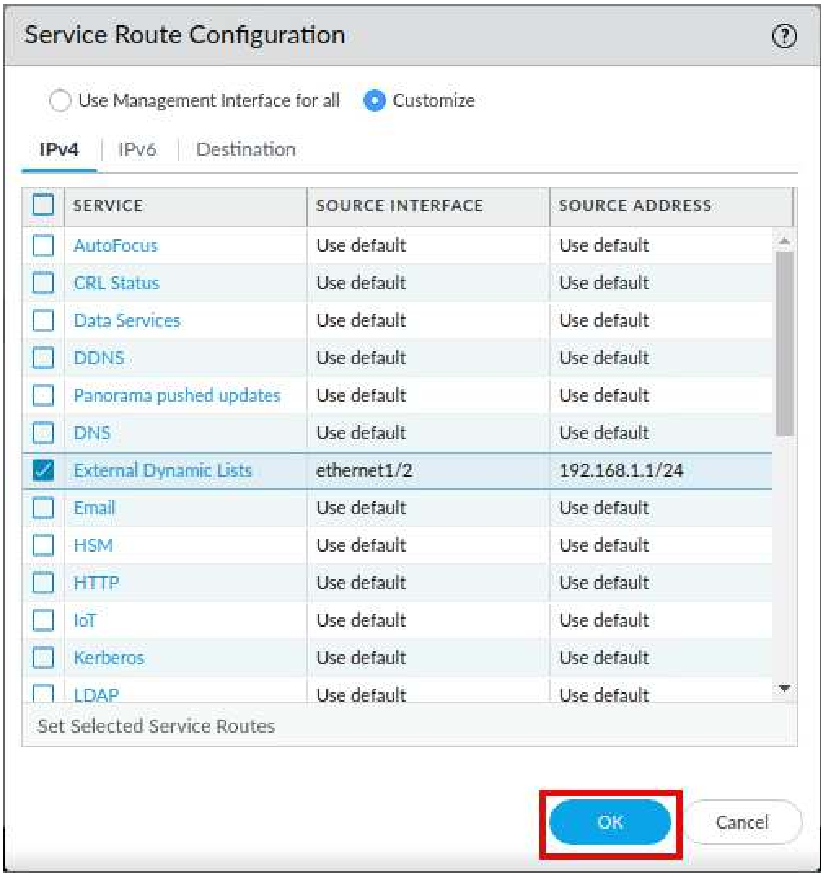
Service Route Source

Service: edl-updates

Source Interface: ethernet1/2

Source Address: 192.168.1.1/24

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



Service Route Configuration

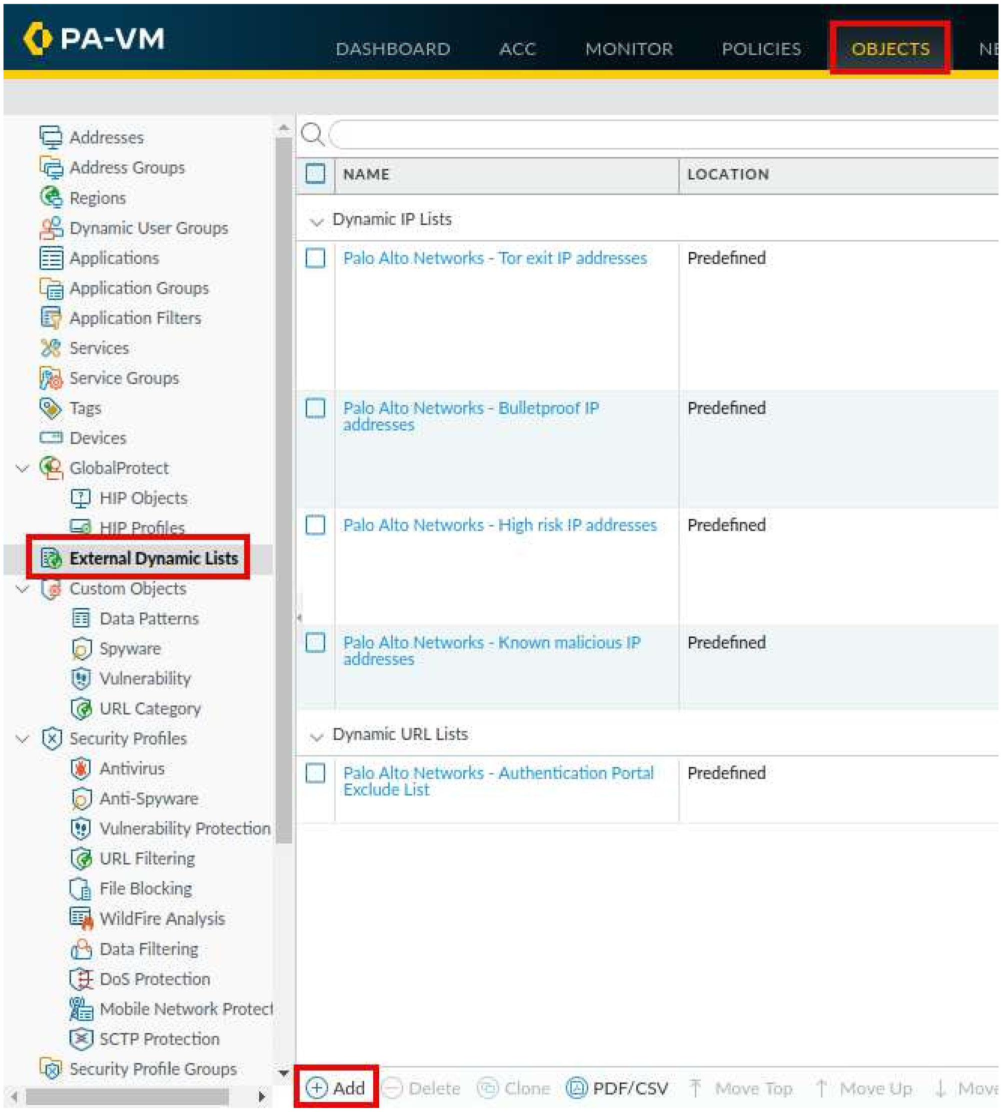
○ Use Management Interface for all ● Customize

IPv4 | IPv6 | Destination

□	SERVICE	SOURCE INTERFACE	SOURCE ADDRESS
□	AutoFocus	Use default	Use default
□	CRL Status	Use default	Use default
□	Data Services	Use default	Use default
□	DDNS	Use default	Use default
□	Panorama pushed updates	Use default	Use default
□	DNS	Use default	Use default
☑	External Dynamic Lists	ethernet1/2	192.168.1.1/24
□	Email	Use default	Use default
□	HSM	Use default	Use default
□	HTTP	Use default	Use default
□	IoT	Use default	Use default
□	Kerberos	Use default	Use default
□	LDAP	Use default	Use default

Set Selected Service Routes

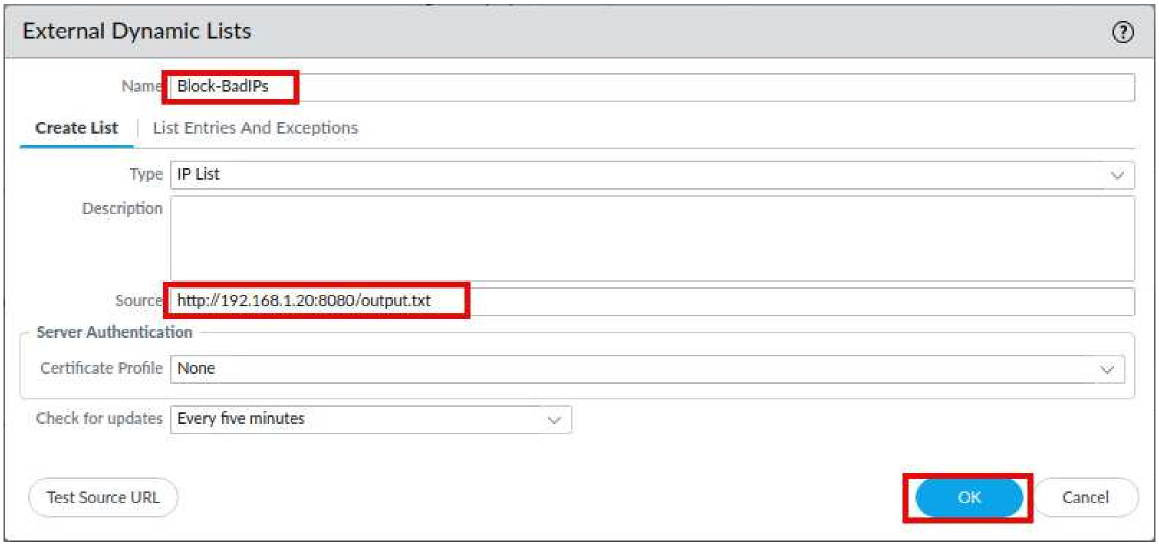
6. Navigate to **Objects > External Dynamic Lists** and click **Add**.



NAME	LOCATION
Dynamic IP Lists	
Palo Alto Networks - Tor exit IP addresses	Predefined
Palo Alto Networks - Bulletproof IP addresses	Predefined
Palo Alto Networks - High risk IP addresses	Predefined
Palo Alto Networks - Known malicious IP addresses	Predefined
Dynamic URL Lists	
Palo Alto Networks - Authentication Portal Exclude List	Predefined

7. If a notice appears regarding appending ending tokens to entries, select **Do not show this message again** and click **Cancel** to continue.

8. In the *External Dynamic Lists* window, type `Block-BadIPs` in the *Name* field, and enter `http://192.168.1.20:8080/output.txt` for the *Source*. Click **OK**.



External Dynamic Lists

Name: Block-BadIPs

Type: IP List

Description:

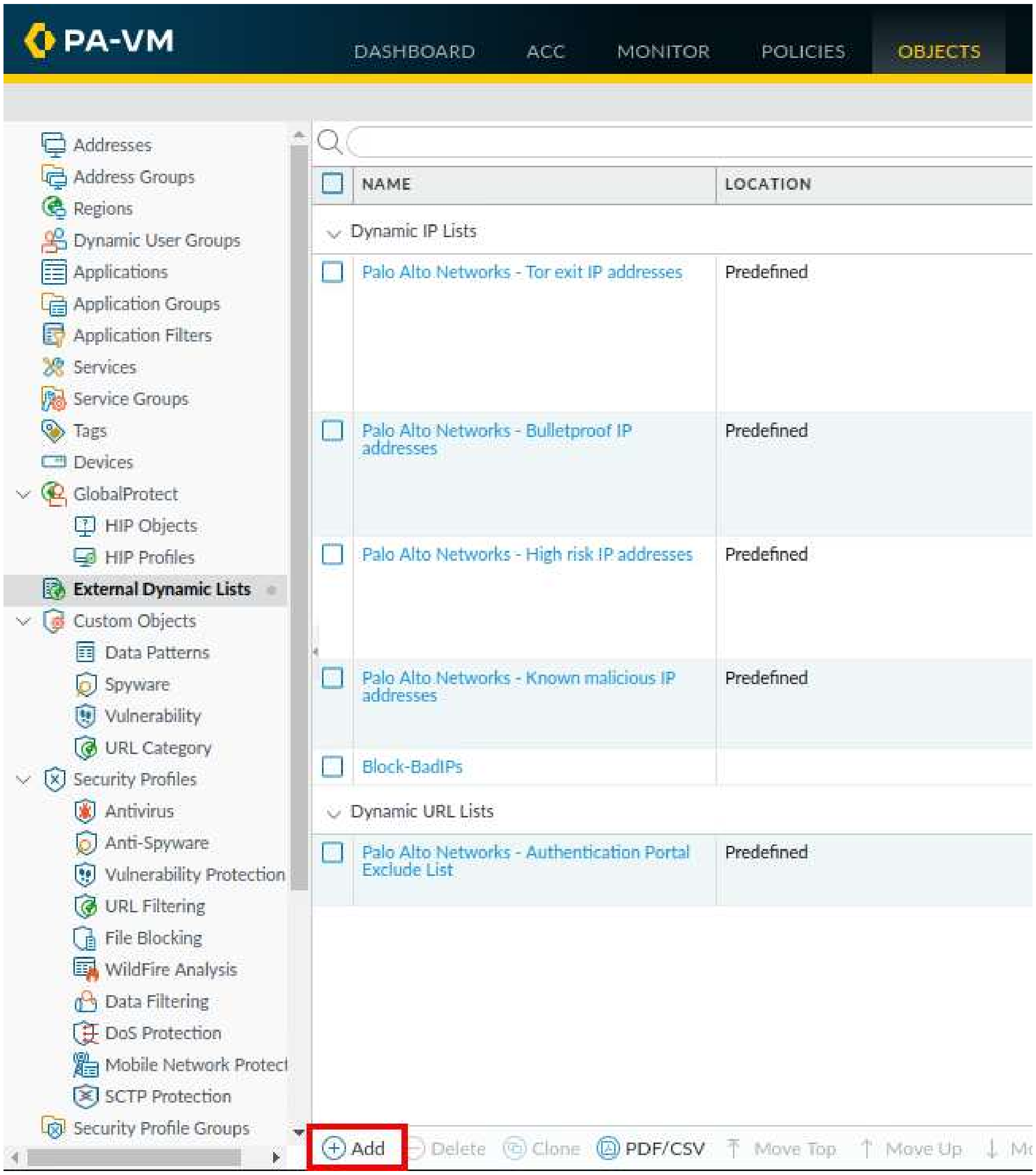
Source: http://192.168.1.20:8080/output.txt

Server Authentication

Certificate Profile: None

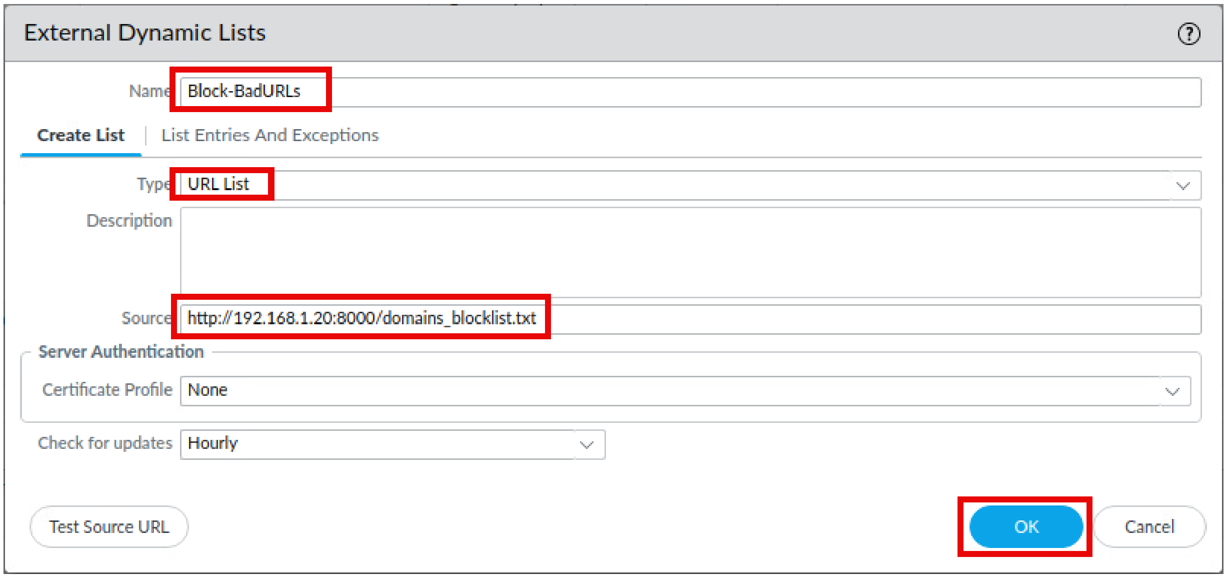
Check for updates: Every five minutes

9. On the *External Dynamic Lists* page, click **Add** again.



NAME	LOCATION
Dynamic IP Lists	
Palo Alto Networks - Tor exit IP addresses	Predefined
Palo Alto Networks - Bulletproof IP addresses	Predefined
Palo Alto Networks - High risk IP addresses	Predefined
Palo Alto Networks - Known malicious IP addresses	Predefined
Block-BadIPs	
Dynamic URL Lists	
Palo Alto Networks - Authentication Portal Exclude List	Predefined

10. In the *External Dynamic Lists* window, type `Block-BadURLs` in the *Name* field, select **URL List** from the *Type* drop-down list, and enter `http://192.168.1.20:8000/domains_blocklist.txt` for the *Source*. Click **OK**.



External Dynamic Lists

Name: Block-BadURLs

Type: URL List

Description:

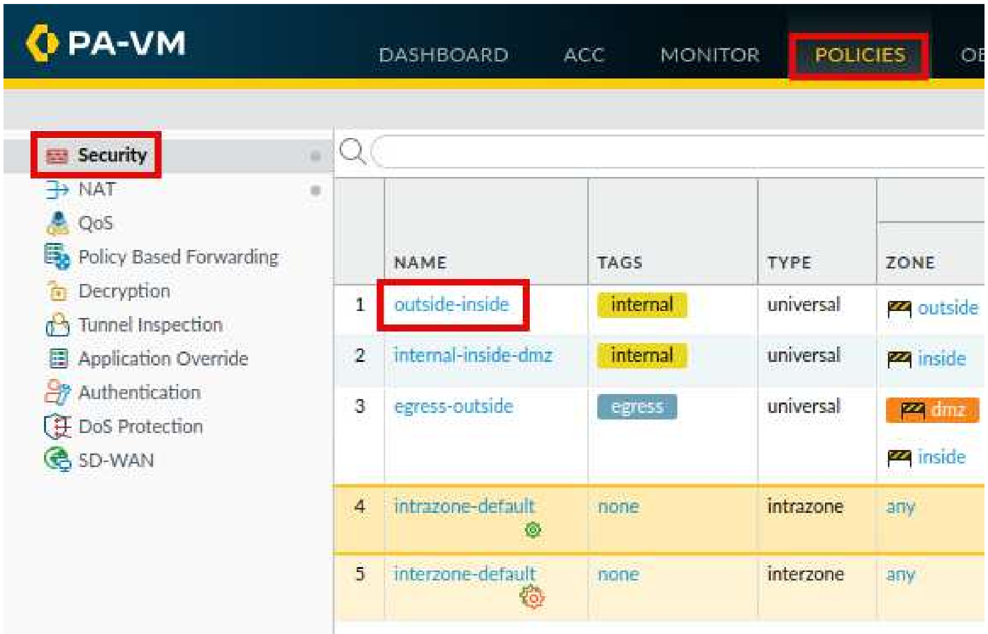
Source: http://192.168.1.20:8000/domains_blocklist.txt

Server Authentication

Certificate Profile: None

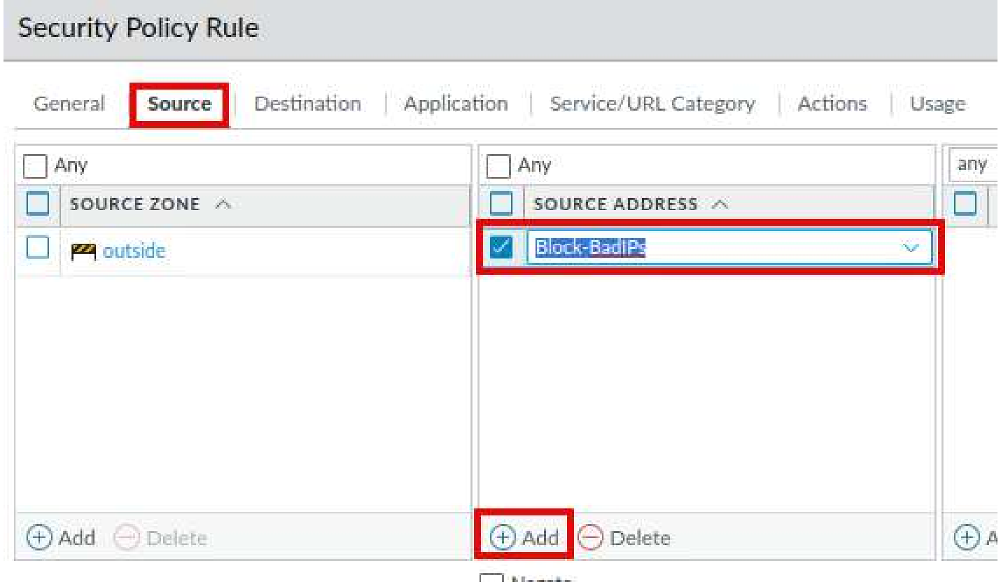
Check for updates: Hourly

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

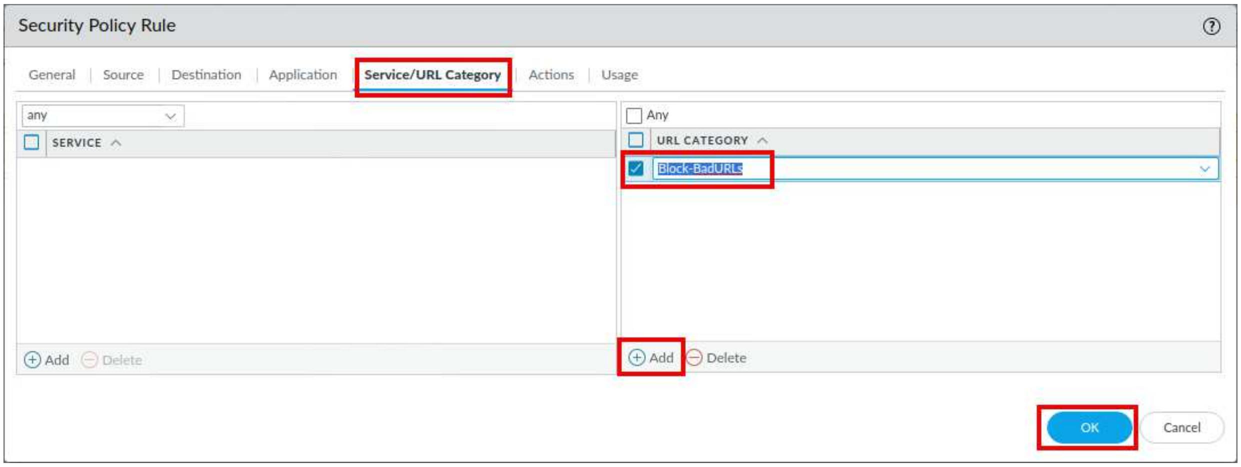


	NAME	TAGS	TYPE	ZONE
1	outside-inside	internal	universal	outside
2	internal-inside-dmz	internal	universal	inside
3	egress-outside	egress	universal	dmz inside
4	intrazone-default	none	intrazone	any
5	interzone-default	none	interzone	any

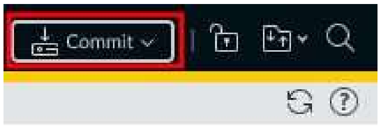
12. In the *Security Policy Rule* window, select the **Source** tab. In the *Source Address* box, click **Add** and select **Block-BadIPs** from the dropdown menu.



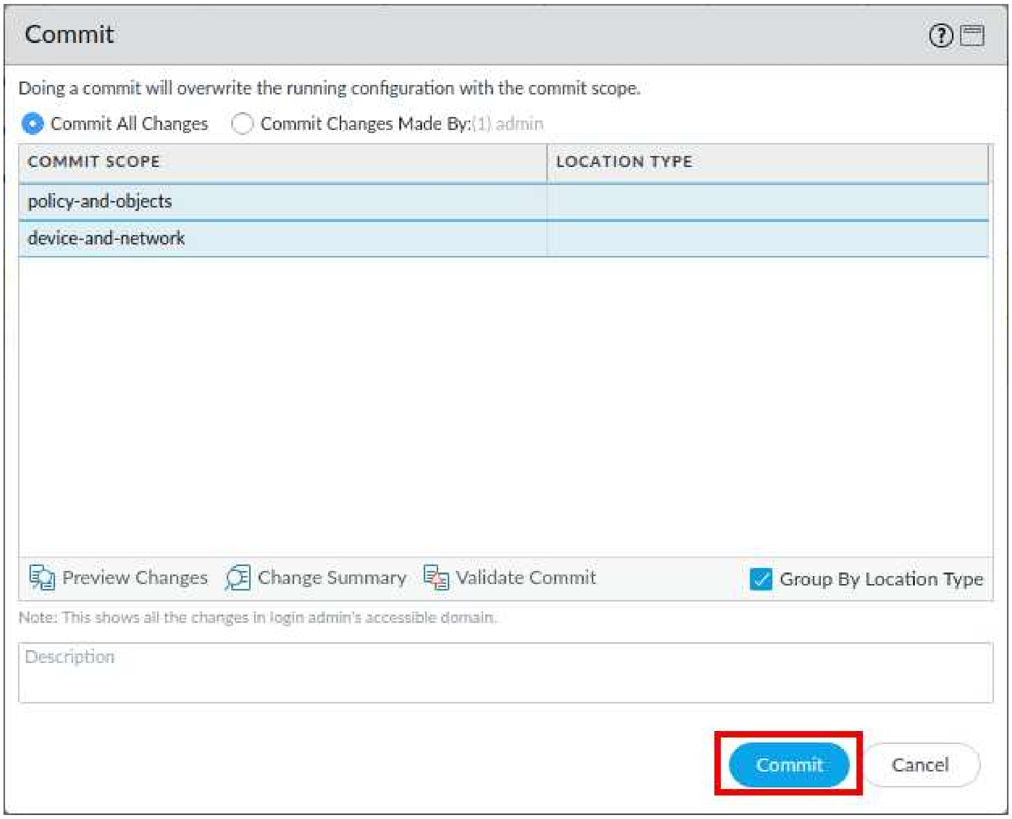
13. In the *Security Policy Rule* window, select the **Service/URL Category** tab. In the *URL Category* box, click **Add** and select **Block-BadURLs** from the dropdown menu. Click **OK** to save changes and to close the window.



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



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



Commit

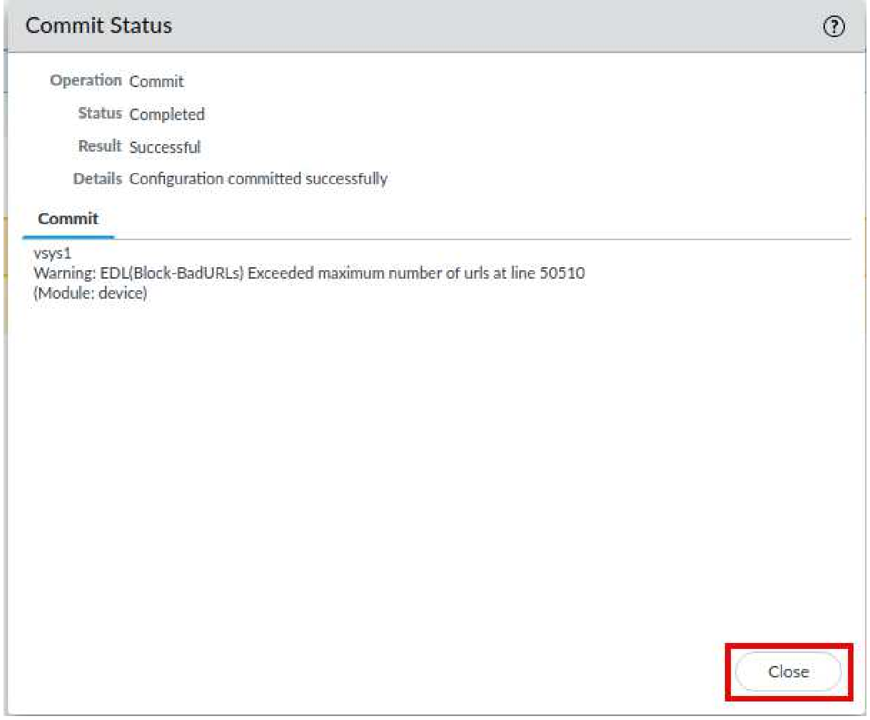
Doing a commit will overwrite the running configuration with the commit scope.

● Commit All Changes ○ Commit Changes Made By:(1) admin

COMMIT SCOPE	LOCATION TYPE
policy-and-objects	
device-and-network	

Note: This shows all the changes in login admin's accessible domain.

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

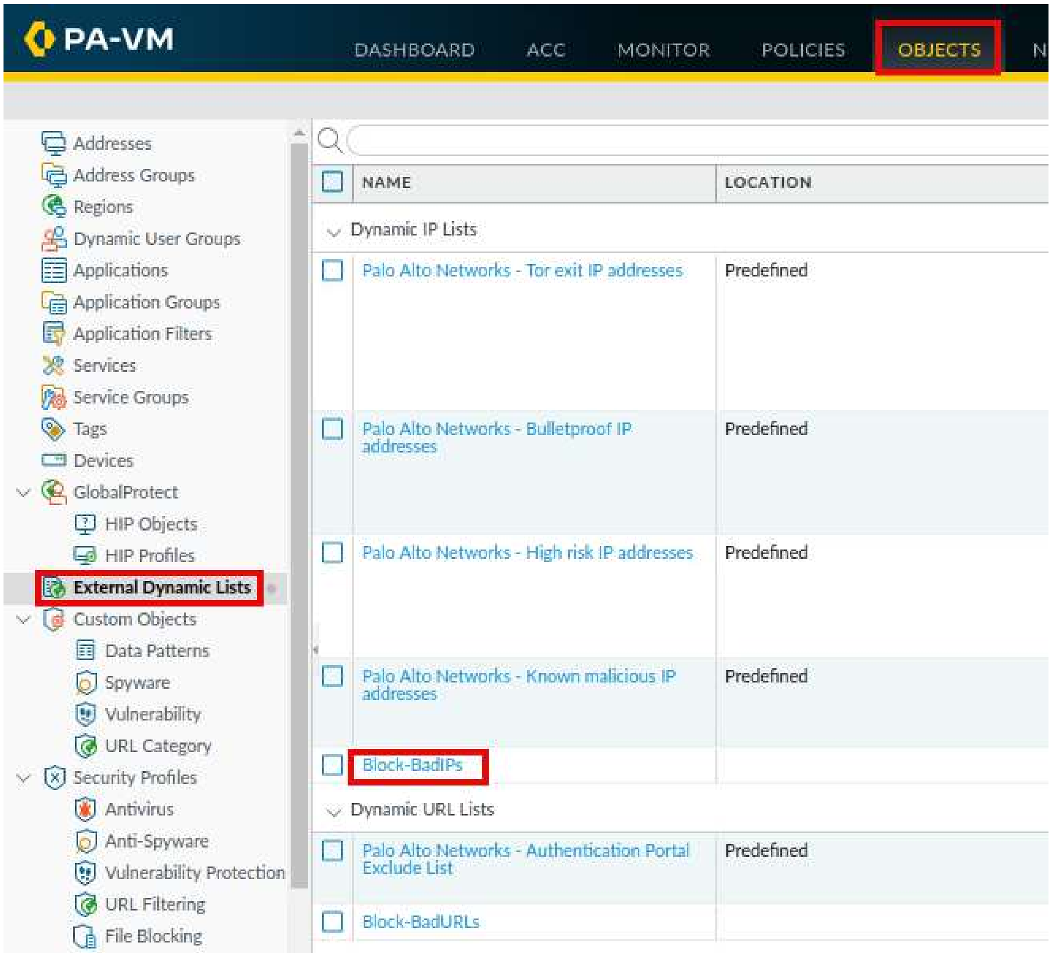


Commit Status

Operation Commit
Status Completed
Result Successful
Details Configuration committed successfully

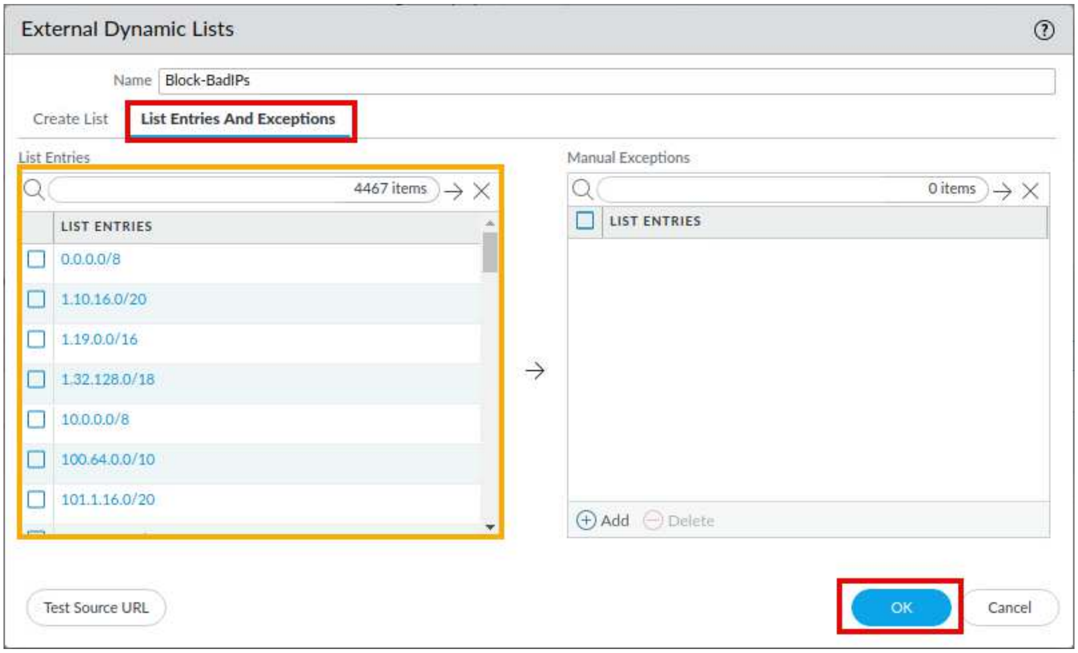
```
vsys1
Warning: EDL(Block-BadURLs) Exceeded maximum number of urls at line 50510
(Module: device)
```

17. Navigate to **Objects > External Dynamic Lists** and click the **Block-BadIPs** list.



NAME	LOCATION
Dynamic IP Lists	
Palo Alto Networks - Tor exit IP addresses	Predefined
Palo Alto Networks - Bulletproof IP addresses	Predefined
Palo Alto Networks - High risk IP addresses	Predefined
Palo Alto Networks - Known malicious IP addresses	Predefined
Block-BadIPs	
Dynamic URL Lists	
Palo Alto Networks - Authentication Portal Exclude List	Predefined
Block-BadURLs	

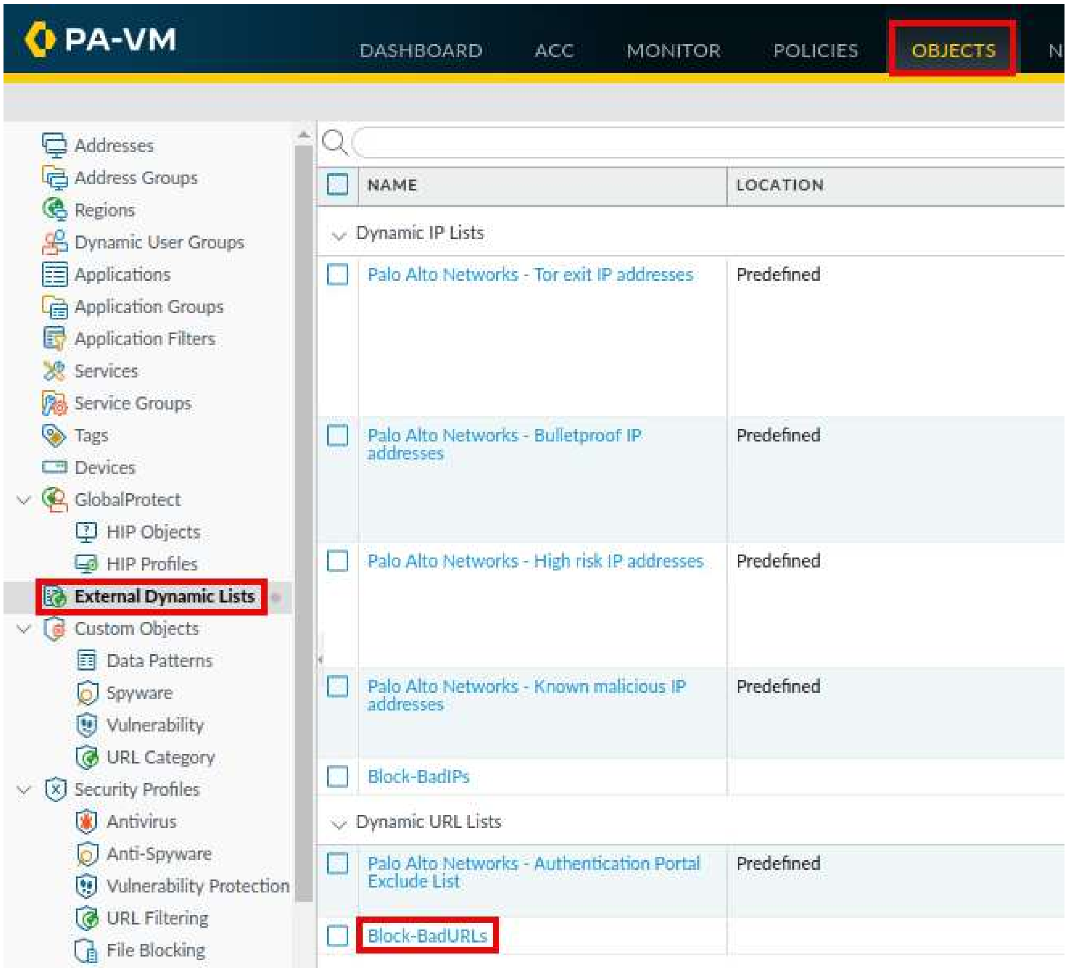
18. In the *External Dynamic Lists* window, select **List Entries and Exceptions** and observe the *IP block list indicators* feeding the *Firewall*. Click **OK**.



Name: Block-BadIPs

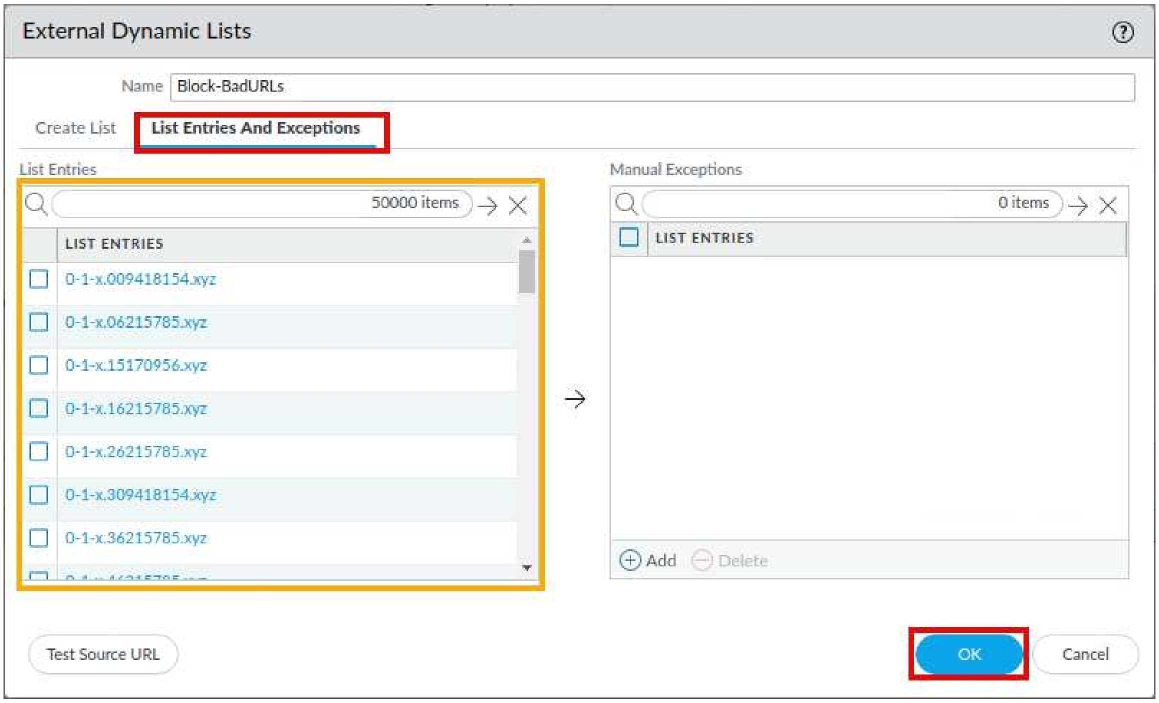
LIST ENTRIES
0.0.0.0/8
1.10.16.0/20
1.19.0.0/16
1.32.128.0/18
10.0.0.0/8
100.64.0.0/10
101.1.16.0/20

19. Navigate to **Objects > External Dynamic Lists** and click the **Block-BadURLs** list.



☐	NAME	LOCATION
	Dynamic IP Lists	
☐	Palo Alto Networks - Tor exit IP addresses	Predefined
☐	Palo Alto Networks - Bulletproof IP addresses	Predefined
☐	Palo Alto Networks - High risk IP addresses	Predefined
☐	Palo Alto Networks - Known malicious IP addresses	Predefined
☐	Block-BadIPs	
	Dynamic URL Lists	
☐	Palo Alto Networks - Authentication Portal Exclude List	Predefined
☐	Block-BadURLs	

20. In the *External Dynamic Lists* window, select **List Entries and Exceptions** and observe the *URL block list indicators* feeding the *Firewall*. Click **OK**.



External Dynamic Lists

Name: Block-BadURLs

Create List | List Entries And Exceptions

List Entries (50000 items)

□	LIST ENTRIES
□	0-1-x.009418154.xyz
□	0-1-x.06215785.xyz
□	0-1-x.15170956.xyz
□	0-1-x.16215785.xyz
□	0-1-x.26215785.xyz
□	0-1-x.309418154.xyz
□	0-1-x.36215785.xyz

Manual Exceptions (0 items)

□	LIST ENTRIES

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