

Setup Active Defense Syslog

- source code / Build Your Own

Contents

Introduce Active Defense Syslog	1
Install Active Defense Syslog System.....	3
Assume you have an Ubuntu server installed and enabled SSH with full access	3
Install Django Framework	4
Create virtual environment and install Django	4
Create web applications and install dependent libraries.....	5
Copy the source code files and install.....	5
Setup Palo Firewall and forward the logs	8
Setup syslog at Palo firewall.....	8
Create blacklist address object	9
Create the Active Defense policies.....	10
• The First policy needs to be at the top position.....	10
• The Second policy needs to be at the bottom position	10
• Apply the ActiveDefense syslog to all other Internet facing policy such as the Globalprotect one as below	11
Setup Active Defense Syslog	11
Login to GUI and change password or create your own account	11
Setup SMTP server and recipients for notification. System will email recipients if new IP address added to blacklist (optional)	11
Find and add the syslog file location.....	11
Tune the Active Defense settings to meet your need under mane “ActiveDefense Settings”	12
Start the system and verify running.....	13
Limitation	14

Introduce Active Defense Syslog

This Active Defense application is an Internet traffic forced syslog system. It was built base on Palo Alto firewall and it mainly looks at traffic coming from internet and find out any IP / port scanning and vulnerability attack attempt activities and instructs the firewall to perform an explicit block action.

An example of how policy configuration makes use of this Active Defense system. The top policy to block IP addresses that match the Blacklist address object which provided by Active Defense system. The bottom policy will feed all un-matched / implicit blocked traffic, plus all other public facing policies' logs to the system for data analysing where a Blacklist-IP will be produced.

	NAME	TAGS	TYPE	Source		Destination		APPLICATI...	SERVICE	ACTION
				ZONE	ADDRESS	ZONE	ADDRESS			
1	Active Defense Deny	none	universal	UNTRUST	ActiveDefense Blacklist	any	any	any	any	Reset Both
13	Active Defense sensor	none	universal	UNTRUST	any	UNTRUST	any	any	any	Reset Both

An example of list of bad guys. And the detail of why they got blacklisted. Firewall will block these public IPs by rule 1 above.

Blacklist Records

Type	Blacklisted IP
Vulnerability Scan v1	184.105.247.252
Port Scan Attack v1	45.142.193.118
Port Scan Attack v1	173.234.107.200
Port Scan Attack v1	51.161.172.223
Port Scan Attack v1	185.91.127.81
Port Scan Attack v1	165.154.205.78
Port Scan Attack v1	203.50.229.44

Syslog Records

Logfile: LAB-FW-01

Hostname: LAB-FW-01

Source Country: United States

Source IP: 184.105.247.252 Blacklist it

Source Port: 41890

User Account:

Destination Country: Australia

Destination IP:

Destination Port: 443

Threat Name: Palo Alto Networks GlobalProtect OS Command Injection Vulnerability(95187)

Threat Type: vulnerability

















Severity: critical

Rule Name: GlobalProtect_Portal

Action: reset-both

Log Type: THREAT

Last Seen: 2025/02/20 11:49:43

	RECEIVE TIME	TYPE	FROM ZONE	SOURCE	TO ZONE	TO PORT	APPLICATION	ACTION	RULE	SESSION END REASON	SOURCE COUNTRY
	02/20 12:29:35	drop	UNTRUST	203.50.229.44	UNTRUST	0	ping	drop	Active Defense Deny	policy-deny	Australia
	02/20 12:29:33	drop	UNTRUST	165.154.205.78	UNTRUST	8060	not-applicable	reset-both	Active Defense Deny	policy-deny	Singapore
	02/20 12:29:26	drop	UNTRUST	51.161.172.223	UNTRUST	4821	not-applicable	reset-both	Active Defense Deny	policy-deny	Canada
	02/20 12:29:21	drop	UNTRUST	173.234.107.200	UNTRUST	16066	not-applicable	reset-both	Active Defense Deny	policy-deny	Australia
	02/20 12:29:10	drop	UNTRUST	173.234.107.200	UNTRUST	16066	not-applicable	reset-both	Active Defense Deny	policy-deny	Australia
	02/20 12:28:35	drop	UNTRUST	203.50.229.44	UNTRUST	0	ping	drop	Active Defense Deny	policy-deny	Australia
	02/20 12:28:20	drop	UNTRUST	165.154.205.78	UNTRUST	1027	not-applicable	reset-both	Active Defense Deny	policy-deny	Singapore
	02/20 12:28:04	drop	UNTRUST	45.142.193.118	UNTRUST	39940	not-applicable	reset-both	Active Defense Deny	policy-deny	Romania
	02/20 12:27:45	drop	UNTRUST	173.234.107.200	UNTRUST	16066	not-applicable	reset-both	Active Defense Deny	policy-deny	Australia
	02/20 12:27:35	drop	UNTRUST	203.50.229.44	UNTRUST	0	ping	drop	Active Defense Deny	policy-deny	Australia
	02/20 12:27:30	drop	UNTRUST	165.154.205.78	UNTRUST	8600	not-applicable	reset-both	Active Defense Deny	policy-deny	Singapore
	02/20 12:27:21	drop	UNTRUST	45.142.193.118	UNTRUST	39905	not-applicable	reset-both	Active Defense Deny	policy-deny	Romania
	02/20 12:27:12	drop	UNTRUST	51.161.172.223	UNTRUST	4821	not-applicable	reset-both	Active Defense Deny	policy-deny	Canada
	02/20 12:26:52	drop	UNTRUST	173.234.107.200	UNTRUST	16066	not-applicable	reset-both	Active Defense Deny	policy-deny	Australia
	02/20 12:26:48	drop	UNTRUST	185.91.127.81	UNTRUST	12983	not-applicable	reset-both	Active Defense Deny	policy-deny	Germany
	02/20 12:26:42	drop	UNTRUST	51.161.172.223	UNTRUST	4821	not-applicable	reset-both	Active Defense Deny	policy-deny	Canada

Finally, user can tune the settings of how Defense engine to run. Such as the number of ports per certain mins to be defined as port scan activity. Blacklist public IP that triggered Critical-level or High-level vulnerability attempts. Keeping firewall logs for number of months and remove blacklist IP after number of months without being detected with any bad activity.

Defense case1 - Port Scan Attack:

If an external public IP trying reach the same company owned public IP on multiple destination ports, more than **15 ports** within **10 mins**. This external public IP will be add to Blacklist database.

[Change](#)

Defense case2 - Port Scan Attack:

Not Active!

[Change](#)

Defense case3 - Vulnerability Attack

If the firewall reported a **"Critical"** vulnerability event from an external public IP via any policy. This public IP will be blacklisted

[Change](#)

The system keeps 3 months of record of logs

[Change](#)

Remove inactive blacklisted IPs after 1 months

[Change](#)

Install Active Defense Syslog System

Assume you have an Ubuntu server installed and enabled SSH with full access

The system is using Rsyslog application which is available out-of-box with Ubuntu server.

- Edit file "*sudo nano /etc/rsyslog.conf*" and add below 3x lines into the end of the config file. Save and exit.

```
$template firewall, "/var/log/Firewall/%HOSTNAME%.log"
```

```
if $fromhost != 'Whatever-the-server-hostname-is' then ?firewall
```

```
& stop
```

- Create a folder for storing logs and give full access permission to all users

```
sudo mkdir /var/log/Firewall
```

```
sudo chmod 777 /var/log/Firewall
```

- Restart and verify Rsyslog service.

```
sudo systemctl restart rsyslog.service
```

```
sudo systemctl status rsyslog.service
```

```

● rsyslog.service - System Logging Service
   Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-02-25 02:12:02 UTC; 12s ago
     TriggeredBy: ● syslog.socket
        Docs: man:rsyslogd(8)
              man:rsyslog.conf(5)
              https://www.rsyslog.com/doc/
   Process: 1844 ExecStartPre=/usr/lib/rsyslog/reload-apparmor-profile (code=exited, status=0/SUCCESS)
  Main PID: 1849 (rsyslogd)
    Tasks: 4 (limit: 2214)
   Memory: 1.3M (peak: 4.9M)
      CPU: 112ms
   CGroup: /system.slice/rsyslog.service
           └─1849 /usr/sbin/rsyslogd -n -iNONE

```

Install Django Framework

sudo apt-get update

sudo apt-get install python3-django

sudo apt-get install python3-pip python3-venv

Create virtual environment and install Django

cd /

sudo mkdir Automation

sudo chmod 777 Automation

python3 -m venv Automation

cd Automation

source bin/activate

pip3 install Django

```

user@activedefense:/$ sudo mkdir Automation
user@activedefense:/$ sudo chmod 777 Automation
user@activedefense:/$ python3
python3          python3.12          python3.12-config  python3-config
user@activedefense:/$ python3 -m venv Automation
user@activedefense:/$ ls
Automation  bin.usr-is-merged  cdrom  etc  lib  lib.usr-is-merged  media  op
bin         boot              dev    home  lib64  lost+found  mnt    pr
user@activedefense:/$ cd Automation/
user@activedefense:/Automation$ ls
bin  include  lib  lib64  pyenv.cfg
user@activedefense:/Automation$ source bin/activate
(Automation) user@activedefense:/Automation$ pip3 install django
Collecting django
  Downloading Django-5.1.6-py3-none-any.whl.metadata (4.2 kB)
Collecting asgiref<4,>=3.8.1 (from django)
  Downloading asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from django)
  Downloading sqlparse-0.5.3-py3-none-any.whl.metadata (3.9 kB)
Downloading Django-5.1.6-py3-none-any.whl (8.3 MB)
 8.3/8.3 MB 6.4 MB/s eta 0:00:00
Downloading asgiref-3.8.1-py3-none-any.whl (23 kB)
Downloading sqlparse-0.5.3-py3-none-any.whl (44 kB)
 44.4/44.4 kB 3.7 MB/s eta 0:00:00
Installing collected packages: sqlparse, asgiref, django
Successfully installed asgiref-3.8.1 django-5.1.6 sqlparse-0.5.3
(Automation) user@activedefense:/Automation$

```

Create web applications and install dependent libraries

- Perform below commands under directory /Automation within the virtual environment

django-admin startproject ActiveDefense .

python3 manage.py startapp Login

python3 manage.py startapp Syslog

pip3 install pathlib netifaces datetime apscheduler sqlalchemy django-sslserver

pip3 install python-dateutil

Copy the source code files and install

We will use sftp to copy the source files into created directories from above

If you are using windows computer you can use MoxaXterm free application to do this. Linux user can perform this task natively using Terminal.

- Edit the database encryption key from source code file "ActiveDefense/setting.py" to whatever

```

"""
Django settings for ActiveDefence project.

Generated by 'django-admin startproject' using Django 5.1.2.

For more information on this file, see
https://docs.djangoproject.com/en/5.1/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/5.1/ref/settings/
"""

from pathlib import Path
#import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/5.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'cv[REDACTED]2%'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

```

- Start from the directory where you have downloaded source files. SFTP to the server

```
cd /Automation
```

```
put -R ActiveDefense
```

```
put -R Syslog
```

```
put -R Login
```

```
put -R static
```

```
put -R WebTemplates
```

```
put -R Cert          #if it's internal access only then it's ok to use this shared cert
                    #for https server. Or please replace it with your own SSL cert
```

```
put run.sh          #server start-up script. It will run from crontab
```

```
put myiptable.sh    #redirect https port 8000 to 443
```

```
exit
```

- Compile / create Database
 - Temporary rename the app.py file to avoid error during the database initial setup

```
cd /Automation/Syslog
```

```
mv app.py app.py.tmp
```

```
cd ..
```

- create Database

```
python3 manage.py makemigrations
```

```
(Automation) user@activedefense:/Automation$ python3 manage.py makemigrations
Migrations for 'Syslog':
  Syslog/migrations/0001_initial.py
    + Create model Blacklist
    + Create model DefenseSetting
    + Create model EmailSetting
    + Create model JobLock
    + Create model Logfile
    + Create model Recipient
    + Create model PATrafficLog
(Automation) user@activedefense:/Automation$ cd ActiveDefense/
```

python3 manage.py migrate

```
(Automation) user@activedefense:/Automation$ python3 manage.py migrate
Operations to perform:
  Apply all migrations: Syslog, admin, auth, contenttypes, sessions
Running migrations:
  Applying Syslog.0001_initial... OK
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
(Automation) user@activedefense:/Automation$
```

- Create superuser

python3 manage.py createsuperuser

```
(Automation) user@activedefense:/Automation$ python3 manage.py createsuperuser
Username (leave blank to use 'user'): user
Email address:
Password:
Password (again):
Superuser created successfully.
(Automation) user@activedefense:/Automation$
```

- Rename the app.py.tmp back to app.py

cd /Automation/Syslog/

mv apps.py.tmp apps.py

cd ..

- Setup application auto-start

sudo crontab -e (select nano editor)

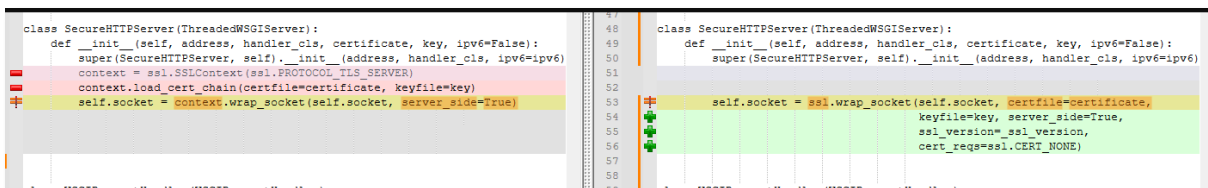
add below line to the bottom and save / exit

@reboot /bin/bash -c "/Automation/myiptable.sh && /Automation/run.sh"

- Fixing a known bug introduced at python3.12 running sslserver (optional if running different python version)

Refer: <https://github.com/teddziuba/django-sslserver/pull/109>

- Modify the /Automation/lib/python3.12/site-packages/sslserver/management/commands/runsslserver.py as below (from right to left)



- Alternative: override the runsslserver.py using the one provided in source folder to the server vis SFTP

sftp to the server

cd /Automation

put -R lib #similar procedure as above steps

- Set correct timezone before Rebooting the server.

sudo timedatectl set-timezone <local timezone such as Australia/Sydney>

sudo reboot now

- The system will startup automatically. Access the application using:

<https://server-ip-address>

username password has been created as superuser above

Setup Palo Firewall and forward the logs

Setup syslog at Palo firewall

Device → Server Profiles → Syslog → Add new

Add the IP address of the Defense system as syslog server. Copy below log format string to both Thread and Traffic log type

```
SrcLocation=$srcloc; SrcIP=$src; SrcPort=$sport; SrcUser=$srcuser; DstLocation=$dstloc; DstIP=$dst;
DstPort=$dport; Action=$action; RuleName=$rule; RuleID=$rule_uuid;
TimeReceived=$time_received; ThreatName=$threat_name; ThreatID=$threatid; Severity=$severity;
Subtype=$subtype; Type= $type;
```

Syslog Server Profile ⓘ

Name: 10.254.109.20

Servers | **Custom Log Format**

LOG TYPE	CUSTOM FORMAT
Config	Default
System	Default
Threat	SrcLocation=\$srcloc; SrcIP=\$src; SrcPort=\$sport; SrcUser=\$srcuser; DstLocation=\$dstloc; DstIP=\$dst; DstPort=\$dport; Action=\$action; RuleName=\$rule; RuleID=\$rule_uid; TimeReceived=\$time_received; ThreatName=\$threat_name; ThreatID=\$threatid; Severity=\$severity; Subtype=\$subtype; Type=\$type;
Traffic	SrcLocation=\$srcloc; SrcIP=\$src; SrcPort=\$sport; SrcUser=\$srcuser; DstLocation=\$dstloc; DstIP=\$dst; DstPort=\$dport; Action=\$action; RuleName=\$rule; RuleID=\$rule_uid; TimeReceived=\$time_received; ThreatName=\$threat_name; ThreatID=\$threatid; Severity=\$severity; Subtype=\$subtype; Type=\$type;
URL	Default
Data	Default

☐ Escaping

Escaped Characters:

Escape Character:

OK Cancel

Objects → Log Forwarding → Add new

Log Forwarding Profile ⓘ

Name: ActiveDefense

Description:

2 items → ×

<input type="checkbox"/>	NAME	LOG TYPE	FILTER	FORWARD METHOD	BUILT-IN ACTIONS
<input type="checkbox"/>	traffic	traffic	All Logs	<u>SysLog</u> • 10.254.109.20	
<input type="checkbox"/>	Threat	threat	All Logs	<u>SysLog</u> • 10.254.109.20	

+ Add - Delete ↺ Clone

OK Cancel

Create [blacklist address object](#)

Objects → External Dynamic Lists → Add new

Type: IP List

Source: <https://<server-IP>/files/blacklist.txt>

Check for update: Every 5 mins

External Dynamic Lists

Name

ActiveDefense Blacklist

Create List

List Entries And Exceptions

Type

IP List

Description

Source

https://10.254.109.20/files/blacklist.txt

Server Authentication

Certificate Profile

None

Check for updates

Every five minutes

Test Source URL

OK

Cancel

Create the Active Defense policies

- The First policy needs to be at the top position

Source Zone: Internet/Untrust

Source: address: <ActiveDefense Blacklist created at above>

Destination Zone: any

Destination Address: any

Action: Deny

Log forward: <ActiveDefense Syslog>

Note: best practice is to forward the blocked logs. System will know if any known blacklisted IPs are still attacking you. System will reset its time within the block-window (default 1 month). Otherwise, system will remove it after block-window regardless. Eventually it will get block again as a new IP

NAME	TAGS	TYPE	ZONE	ADDRESS	ZONE	ADDRESS	APPLICATI...	SERVICE	ACTION	PROFILE	OPTIONS
Active Defense Deny	none	universal	UNTRUST	ActiveDefense Blacklist	any	any	any	any	Reset Both	none	

- The Second policy needs to be at the bottom position.

This policy needs to be at the bottom whenever new policy adds in the future. Basically, it captures all un-matched traffic and forward the logs to the syslog.

Source Zone: Internet / Untrust

Source Address: any

Destination Zone: any

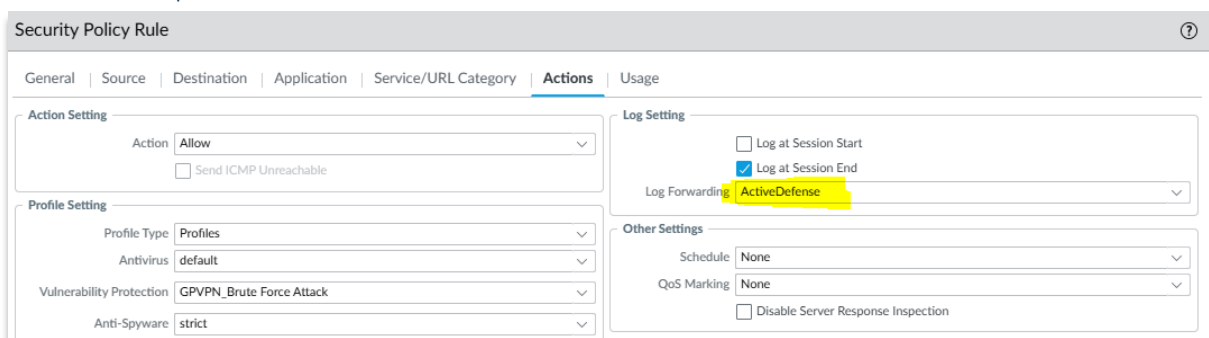
Action: Deny or reset

Log forward: <ActiveDefense Syslog>

Note: We need to forward all un-matched traffic that ActiveDefense can tell if anyone trying to do port scanning on you. If you don't have this catch-all rule. ActiveDefense can only block vulnerability detected IP.

NAME	TAGS	TYPE	ZONE	ADDRESS	ZONE	ADDRESS	APPLICATI...	SERVICE	ACTION	PROFILE	OPTIONS
Active Defense sensor	none	universal	UNTRUST	any	UNTRUST	any	any	any	Reset Both	none	

- Apply the ActiveDefense syslog to all other Internet facing policy such as the Globalprotect one as below



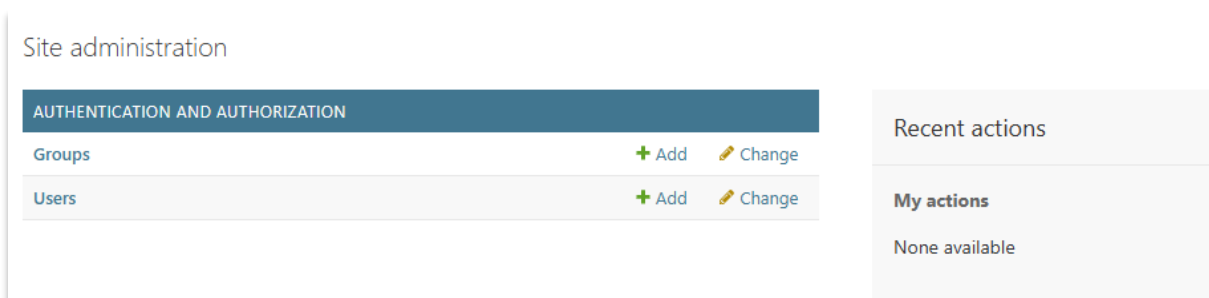
Note: ActiveDefense will find out if anyone try to do vulnerability attack to you and blacklist them. This type of policy usually has security Profile attached such as Vulnerability Protection etc.

Commit all above changes and firewall is completed.

Setup Active Defense Syslog

Login to GUI and change password or create your own account

<https://<server-ip>/admin>



Setup SMTP server and recipients for notification. System will email recipients if new IP address added to blacklist (optional)

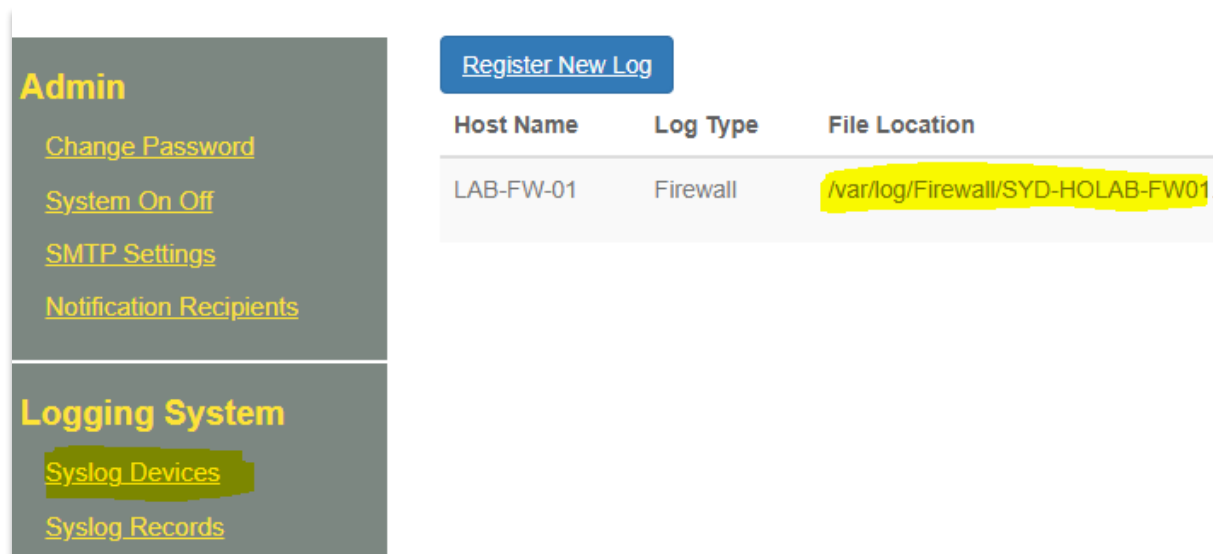
Find and add the syslog file location

New log data will be saved at /var/log/Firewall/ directory before converting into Database records. The log file name is the hostname of the firewall with extension ".log"

You can SSH to the server with provided (above) and find out the log files.

Command: `ls /var/log/Firewall/`

Register the log file and path to the system



The screenshot shows the 'ActiveDefense Settings' interface. On the left, there are two main sections: 'Admin' and 'Logging System'. The 'Admin' section includes links for 'Change Password', 'System On Off', 'SMTP Settings', and 'Notification Recipients'. The 'Logging System' section includes links for 'Syslog Devices' and 'Syslog Records'. On the right, there is a 'Register New Log' button and a table with the following columns: 'Host Name', 'Log Type', and 'File Location'. The table contains one entry: 'LAB-FW-01' for 'Firewall' with the file location '/var/log/Firewall/SYD-HOLAB-FW01'.

Host Name	Log Type	File Location
LAB-FW-01	Firewall	/var/log/Firewall/SYD-HOLAB-FW01

Tune the Active Defense settings to meet your need under mane “ActiveDefense Settings”

- Defense case 1 – port scanning (multiple ports)

Default setting. It is targeting a single public IP being trying with more than 10 ports in the last 10mins window. You can relax it by increase number of ports or lower down the time. Such as 20+ ports or in 8 mins. Be aware that the defence engineer runs every 5 mins. If time set to lower than 5 mins. It will skip logs

- Defense case 2 – Port scanning (multiple IPs)

Default setting: system will find out if someone try to find out open ports on all of your public IPs. Such as targeting https port (443) on all your public IPs. It set to 10 of your public IP being tried on single port in the last 10 mins. You can relax it by increase the number of public IPs or reduce time. If you don't own multiple public IPs. You can disable this by setting the time to 0.

- Defense case 3 – Vulnerability detection

By default. It blacklists any external public IP who had triggered “Critical” vulnerability alerts. You can make it more aggressive to set the value to 2. It will also blacklist someone triggered “High” vulnerability alert. Set to 0 to disable this function if you don't have Threat protection on your firewall.

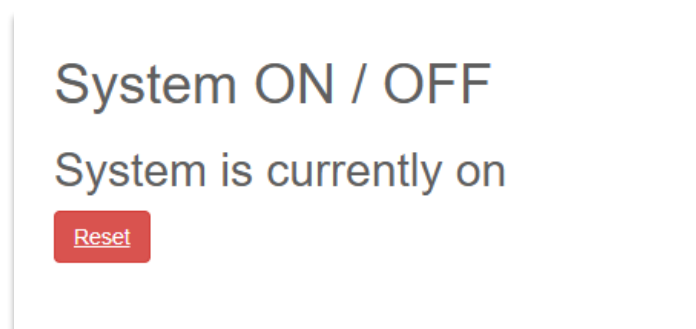
- The other two settings are Database keeping.

By default, it set to keep 3 month's log records and release the blacklisted IP if it has been doing bad thing in a month.

Start the system and verify running

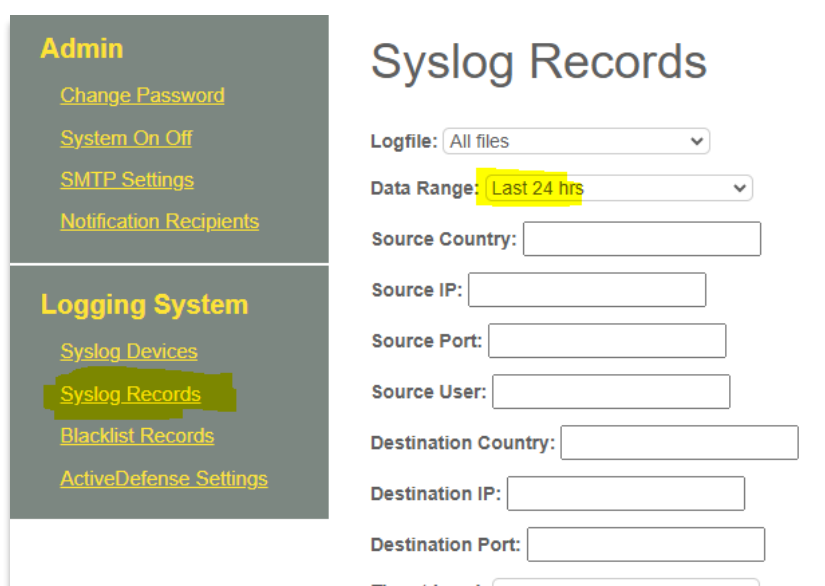
“System On Off” is to start or restart the scheduled jobs when need. If you can see any new traffic logs in “Syslog Records”. Do a “Reset” and “start”. It will restart the defense engine. The logs will always be received even the engine is not running. It will keep in /var/log/Firewall. the log file will grow until the engine runs again. Logs will convert into DB and the log file size gets reduce.

The engine will not auto start when you restart the server. You will need to login to the GUI and start the defense system manually or “reset / start” it every time you rebooted the server.



To verify it's running. You can simply look at the last 24 hours logs. confirm by the time of the record

Or SSH to the server and look at the log file size. A .tmp file will be created when the system converts the logs into Database. Then the log file will get reduced and the .tmp file should be gone after the scheduled job completed. It runs every 5 mins.



Admin

[Change Password](#)

[System On Off](#)

[SMTP Settings](#)

[Notification Recipients](#)

Logging System

[Syslog Devices](#)

[Syslog Records](#)

[Blacklist Records](#)

[ActiveDefense Settings](#)

Syslog Records

Time	SrcLocation	SrcIP	Dis
2025/02/20 18:54:15	United States	206.168.34.139	20
2025/02/20 18:54:15	United States	147.185.132.231	20
2025/02/20 18:54:15	United States	147.185.132.245	20
2025/02/20 18:54:10	Netherlands	93.174.93.12	20
2025/02/20 18:54:10	United States	172.206.147.236	20

Limitation

If you have large amount to logs and the server can not process it within 5 mins. Then you need a faster system to keep up with the log growing. Spread to multiple servers for multiple firewalls. Or external the scheduling time to every 10mins for example. It will require edit to the source code:

/Automation/Syslog/scheduler.py

```
def run():
    global scheduler
    if not job_exists('FletchLog_id01'):
        scheduler.add_job(FletchLog, 'interval', minutes = 5, max_instances = 1, misfire_grace_time=60, id = 'FletchLog_id01', replace_e
        time.sleep(30)
    if not job_exists('defense_id01'):
        scheduler.add_job(defense.run, 'interval', minutes = 5, max_instances = 1, misfire_grace_time=60, id = 'defense_id01', replace_e
    if not job_exists('remove_old_blacklist_id01'):
        scheduler.add_job(remove_old_blacklist, CronTrigger(hour = 0, minute = 0, timezone = 'Australia/Sydney'), max_instances = 1, mis
    if not job_exists('delete_old_logs_id01'):
        scheduler.add_job(delete_old_logs, CronTrigger(hour = 1, minute = 0, timezone = 'Australia/Sydney'), max_instances = 1, misfire_
    if not scheduler.running:
        scheduler.start()
    job_lock, created = JobLock.objects.get_or_create(job_name='System_ON_OFF')
    job_lock.is_running = True
    job_lock.save()
```

Note: the “sleep time” is the gap between two jobs. Database gets lock every action. Job gets delay automatically when DB gets locked. Graceful period of 1 mins before the job gives up and wait for next run.