Femlalogy SOC Report

Project Title: Threat Detection & Incident Response Using Wireshark, pfSense, and Wazuh
Organization: Femlalogy
Analyst: Agboola David Oluwanifise
Role: Security Operations Center (SOC) Analyst
Submission Date: 10/15/2025

Table of Contents

---

1. Executive Summary

The purpose of this project was on how the cybersecurity posture of Femlalogy could be strengthened through the deployment and integration of three key tools, namely, Wazuh, Wireshark and pfSense.Our task was to simulate real world attacks, detect network anomalies,, enforce firewall rules and respond to security incident effectively and timely.

Using Wireshark, suspicious traffic such as Nmap and ping sweeps were captured and analysed by the team. Also, we pfSense integrated with Snort, the team configured and enforced network and firewall policies, thereby blocking ransomware-related IPs, as well as malicious ICMP traffic.

Lastly, Wazuh provided us with a focused security event correlation and monitoring, thereby resulting in actionable alerts and incidents reports. This project depicts the importance of protecting organizational assets against the world of changing cyber threats, proactive monitoring and how relevant the various levels of defense is to an organization.

---

2. Project Introduction

Cybersecurity in today's world remains one of the critical aspects of any organization’s success in this digital age. Therefore, this project aimed to build and access a SOC for Femalogy designed to detect, monitor and respond to cyber threats directed at the organization in real time. By deploying tools such as Wazuh, pfSense and Wireshark, this simulation was done to test the readiness of the organization should a cyber threat occur. Wireshark was used for packet capturing and analysis, while Wazuh was used as the platform that coordinated correlation and reporting. pfSense on the other hand was our barrier between our internal and external network, thereby filtering incoming

and outgoing network traffic based on configured rules. These tools provided a demonstration of how all these tools can work together to protect Femlogy's network from inside and outside threats.

---

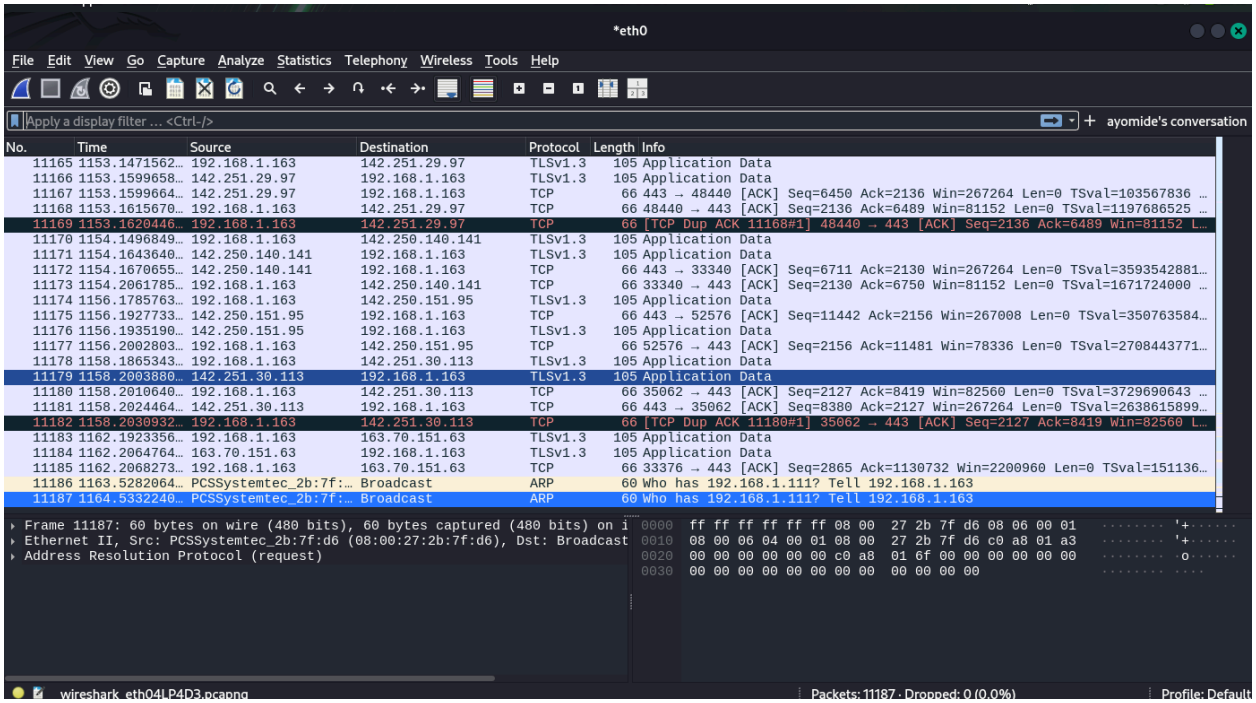
3. Methodology

Wireshark was used to capture all of the suspicious traffic i.e. nmap scan and ping sweeps. pfSense and Snort were configured to block ICMP packets, and as intrusion detection. Wazuh was used to track logs and create alerts, and generate an executive report. Kali linux was the attacker machine and ubuntu was the victim agent and the wazuh agent.

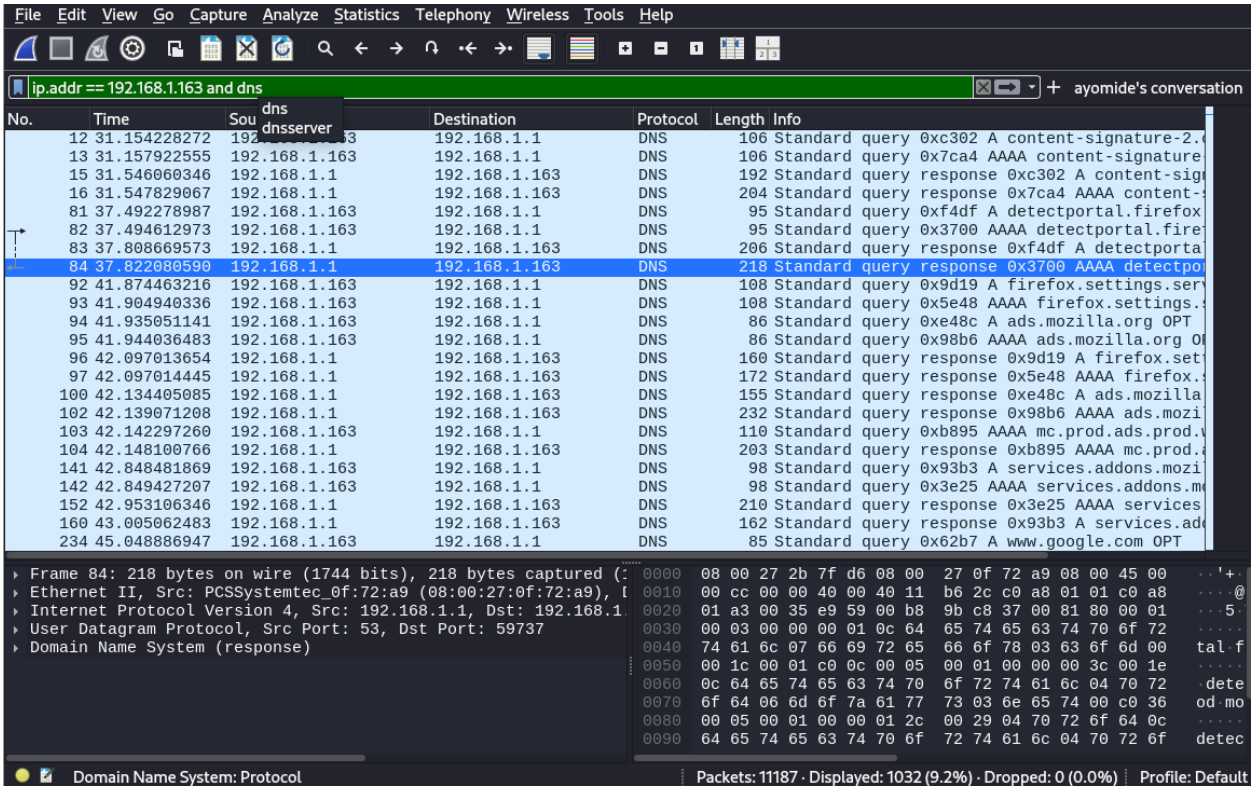
---

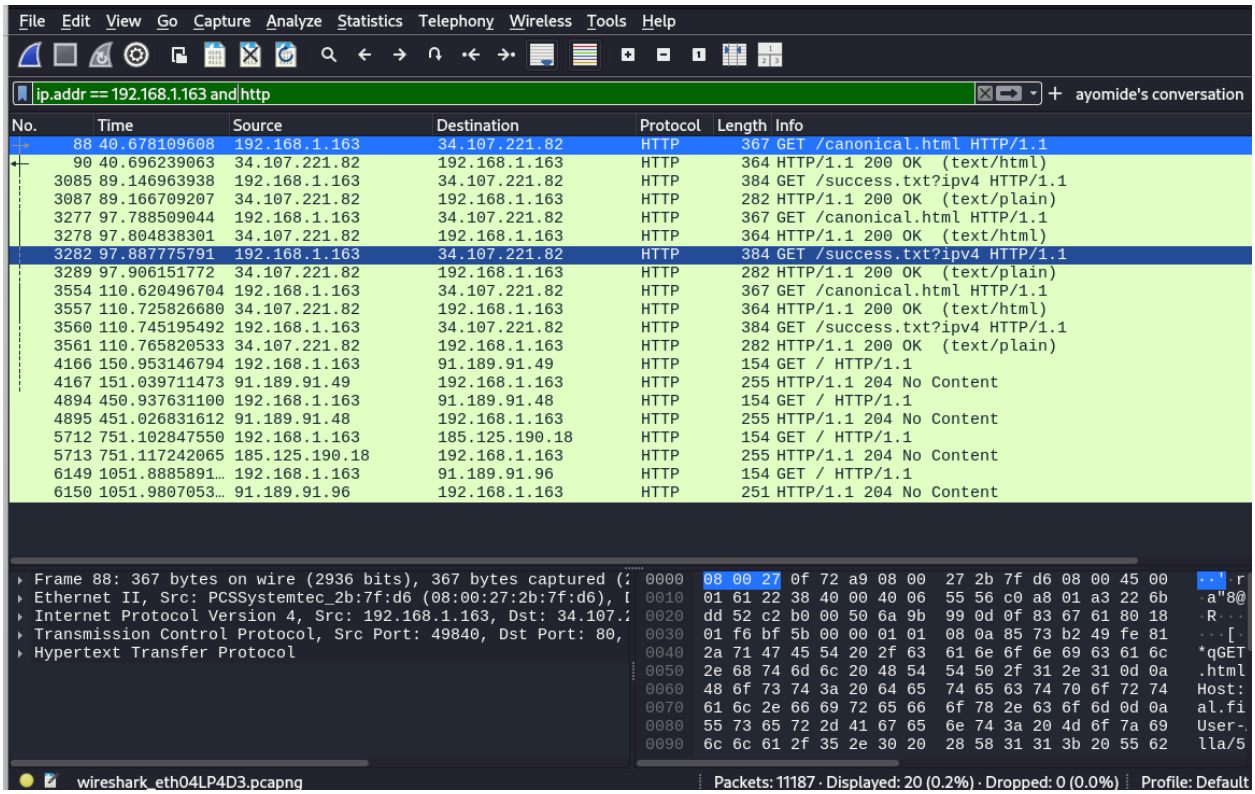
4. Phase-by-Phase Analysis

Phase 1: Wireshark – Network Traffic Capture & Analysis



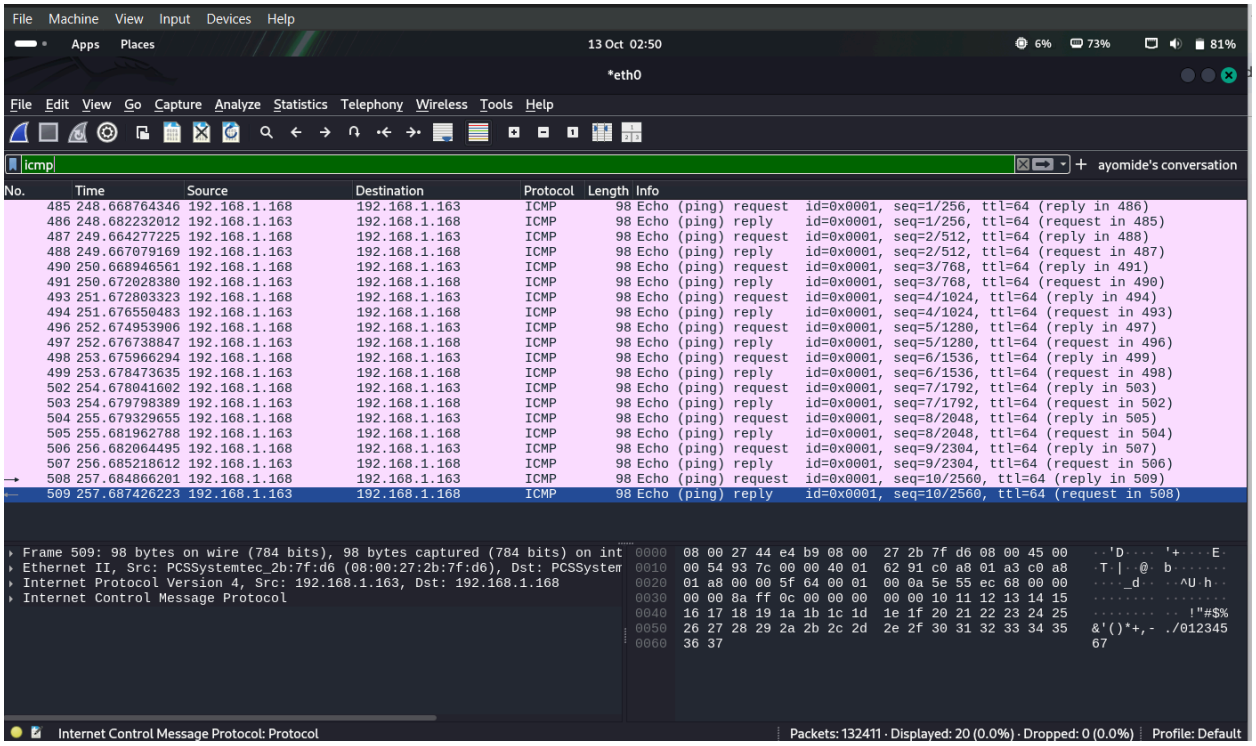
The screenshot above displays the normal traffic on wireshark on kali with no filters to show there was an internet connection.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr == 192.168.1.163 and dns
dns
dnsserver
ayomide's conversation
No. Time Source Destination Protocol Length Info
12 31.154228272 192.168.1.163 192.168.1.1 DNS 106 Standard query 0xc302 A content-signature-2.
13 31.157922555 192.168.1.163 192.168.1.1 DNS 106 Standard query 0x7ca4 AAAA content-signature
15 31.546060346 192.168.1.1 192.168.1.163 DNS 192 Standard query response 0xc302 A content-sig
16 31.547829067 192.168.1.1 192.168.1.163 DNS 204 Standard query response 0x7ca4 AAAA content-
81 37.492278987 192.168.1.163 192.168.1.1 DNS 95 Standard query 0xf4df A detectportal.firefox
82 37.494612973 192.168.1.163 192.168.1.1 DNS 95 Standard query 0x3700 AAAA detectportal.fire
83 37.808669573 192.168.1.1 192.168.1.163 DNS 206 Standard query response 0xf4df A detectporta
84 37.822080590 192.168.1.1 192.168.1.163 DNS 218 Standard query response 0x3700 AAAA detectpo
92 41.874463216 192.168.1.163 192.168.1.1 DNS 108 Standard query 0x9d19 A firefox.settings.ser
93 41.904940336 192.168.1.163 192.168.1.1 DNS 108 Standard query 0x5e48 AAAA firefox.settings.
94 41.935051141 192.168.1.163 192.168.1.1 DNS 86 Standard query 0xe48c A ads.mozilla.org OPT
95 41.944036483 192.168.1.163 192.168.1.1 DNS 86 Standard query 0x98b6 AAAA ads.mozilla.org O
96 42.097013654 192.168.1.1 192.168.1.163 DNS 160 Standard query response 0x9d19 A firefox.set
97 42.097014445 192.168.1.1 192.168.1.163 DNS 172 Standard query response 0x5e48 AAAA firefox.
100 42.134405085 192.168.1.1 192.168.1.163 DNS 155 Standard query response 0xe48c A ads.mozilla
102 42.139071208 192.168.1.1 192.168.1.163 DNS 232 Standard query response 0x98b6 AAAA ads.mozi
103 42.142297260 192.168.1.163 192.168.1.1 DNS 110 Standard query 0xb895 AAAA mc.prod.ads.prod.
104 42.148100766 192.168.1.1 192.168.1.163 DNS 203 Standard query response 0xb895 AAAA mc.prod.
141 42.848481869 192.168.1.163 192.168.1.1 DNS 98 Standard query 0x93b3 A services.addons.mozi
142 42.849427207 192.168.1.163 192.168.1.1 DNS 98 Standard query 0x3e25 AAAA services.addons.m
152 42.953106346 192.168.1.1 192.168.1.163 DNS 210 Standard query response 0x3e25 AAAA services
160 43.005062483 192.168.1.1 192.168.1.163 DNS 162 Standard query response 0x93b3 A services.ad
234 45.048886947 192.168.1.163 192.168.1.1 DNS 85 Standard query 0x62b7 A www.google.com OPT
Frame 84: 218 bytes on wire (1744 bits), 218 bytes captured (
Ethernet II, Src: PCSSystemtec_0f:72:a9 (08:00:27:0f:72:a9), [
Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1
User Datagram Protocol, Src Port: 53, Dst Port: 59737
Domain Name System (response)
0000 08 00 27 2b 7f d6 08 00 27 0f 72 a9 08 00 45 00
0010 00 cc 00 00 40 00 40 11 b6 2c c0 a8 01 01 c0 a8
0020 01 a3 00 35 e9 59 00 b8 9b c8 37 00 81 80 00 01
0030 00 03 00 00 00 01 0c 64 65 74 65 63 74 70 6f 72
0040 74 61 6c 07 66 69 72 65 66 6f 78 03 63 6f 6d 00
0050 00 1c 00 01 c0 0c 00 05 00 01 00 00 00 3c 00 1e
0060 0c 64 65 74 65 63 74 70 6f 72 74 61 6c 04 70 72
0070 6f 64 06 6d 6f 7a 61 77 73 03 6e 65 74 00 c0 36
0080 00 05 00 01 00 00 01 2c 00 29 04 70 72 6f 64 0c
0090 64 65 74 65 63 74 70 6f 72 74 61 6c 04 70 72 6f
Domain Name System: Protocol
Packets: 11187 · Displayed: 1032 (9.2%) · Dropped: 0 (0.0%)
Profile: Default

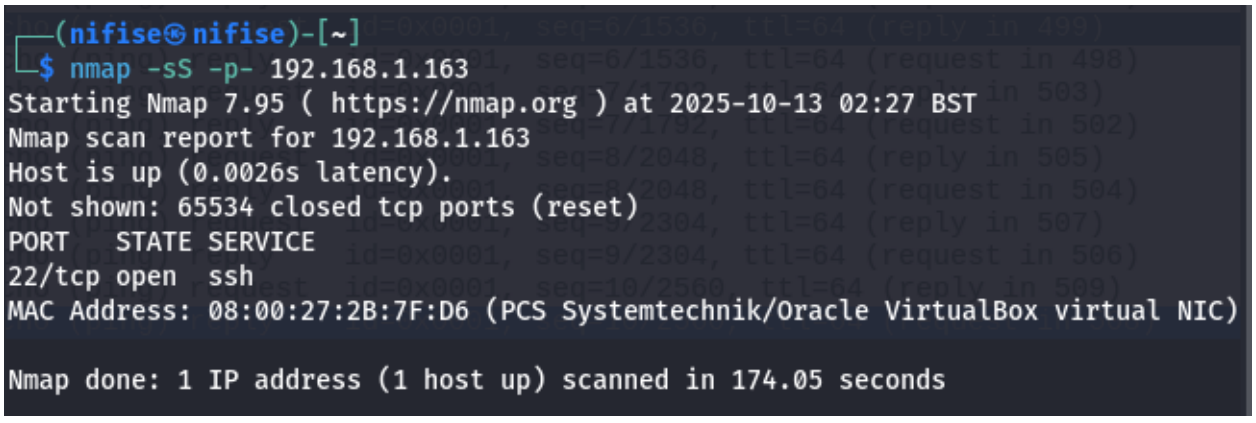

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr == 192.168.1.163 and http
ayomide's conversation
No. Time Source Destination Protocol Length Info
88 40.678109608 192.168.1.163 34.107.221.82 HTTP 367 GET /canonical.html HTTP/1.1
90 40.696239063 34.107.221.82 192.168.1.163 HTTP 364 HTTP/1.1 200 OK (text/html)
3085 89.146963938 192.168.1.163 34.107.221.82 HTTP 384 GET /success.txt?ipv4 HTTP/1.1
3087 89.166709207 34.107.221.82 192.168.1.163 HTTP 282 HTTP/1.1 200 OK (text/plain)
3277 97.788509044 192.168.1.163 34.107.221.82 HTTP 367 GET /canonical.html HTTP/1.1
3278 97.804838301 34.107.221.82 192.168.1.163 HTTP 364 HTTP/1.1 200 OK (text/html)
3282 97.887775791 192.168.1.163 34.107.221.82 HTTP 384 GET /success.txt?ipv4 HTTP/1.1
3289 97.906151772 34.107.221.82 192.168.1.163 HTTP 282 HTTP/1.1 200 OK (text/plain)
3554 110.620496704 192.168.1.163 34.107.221.82 HTTP 367 GET /canonical.html HTTP/1.1
3557 110.725826680 34.107.221.82 192.168.1.163 HTTP 364 HTTP/1.1 200 OK (text/html)
3560 110.745195492 192.168.1.163 34.107.221.82 HTTP 384 GET /success.txt?ipv4 HTTP/1.1
3561 110.765820533 34.107.221.82 192.168.1.163 HTTP 282 HTTP/1.1 200 OK (text/plain)
4166 150.953146794 192.168.1.163 91.189.91.49 HTTP 154 GET / HTTP/1.1
4167 151.039711473 91.189.91.49 192.168.1.163 HTTP 255 HTTP/1.1 204 No Content
4894 450.937631100 192.168.1.163 91.189.91.48 HTTP 154 GET / HTTP/1.1
4895 451.026831612 91.189.91.48 192.168.1.163 HTTP 255 HTTP/1.1 204 No Content
5712 751.102847550 192.168.1.163 185.125.190.18 HTTP 154 GET / HTTP/1.1
5713 751.117242065 185.125.190.18 192.168.1.163 HTTP 255 HTTP/1.1 204 No Content
6149 1051.8885891… 192.168.1.163 91.189.91.96 HTTP 154 GET / HTTP/1.1
6150 1051.9807053… 91.189.91.96 192.168.1.163 HTTP 251 HTTP/1.1 204 No Content
Frame 88: 367 bytes on wire (2936 bits), 367 bytes captured (
Ethernet II, Src: PCSSystemtec_2b:7f:d6 (08:00:27:2b:7f:d6), [
Internet Protocol Version 4, Src: 192.168.1.163, Dst: 34.107.
Transmission Control Protocol, Src Port: 49840, Dst Port: 80,
Hypertext Transfer Protocol
0000 08 00 27 0f 72 a9 08 00 27 2b 7f d6 08 00 45 00
0010 01 61 22 38 40 00 40 06 55 56 c0 a8 01 a3 22 6b
0020 dd 52 c2 b0 00 50 6a 9b 99 0d 0f 83 67 61 80 18
0030 01 f6 bf 5b 00 00 01 01 08 0a 85 73 b2 49 fe 81
0040 2a 71 47 45 54 20 2f 63 61 6e 6f 6e 69 63 61 6c
0050 2e 68 74 6d 6c 20 48 54 54 50 2f 31 2e 31 0d 0a
0060 48 6f 73 74 3a 20 64 65 74 65 63 74 70 6f 72 74
0070 61 6c 2e 66 69 72 65 66 6f 78 2e 63 6f 6d 0d 0a
0080 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69
0090 6c 6c 61 2f 35 2e 30 20 28 58 31 31 3b 20 55 62
wireshark_eth04LP4D3.pcapng
Packets: 11187 · Displayed: 20 (0.2%) · Dropped: 0 (0.0%)
Profile: Default

The image above shows the filters used to display the traffic for when the victim IP 192.168.1.163 visited google.com on mozilla firefox browser.

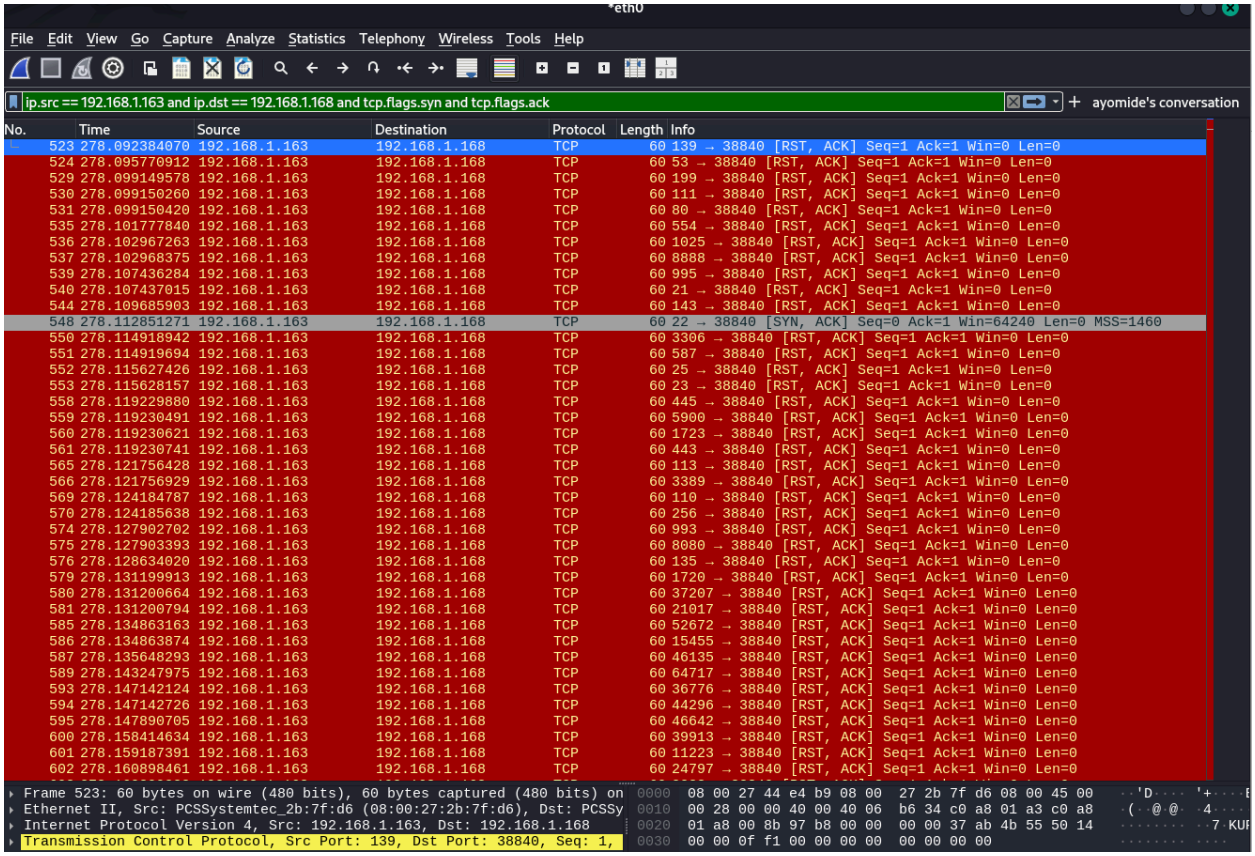
Ping flood was ran to simulate a Denial Of Service (DOS) attack on the victim machine, Using the “icmp” keyword filter in wireshark, it displayed only the ping which was launched from the attacker onto the victim IP.



Nnmap was ran to scan from my kali using the -sS- to perform a stealth scan where Nmap sends SYN packets to start connections but never completes the TCP handshake and the -p- to scan all the ports.

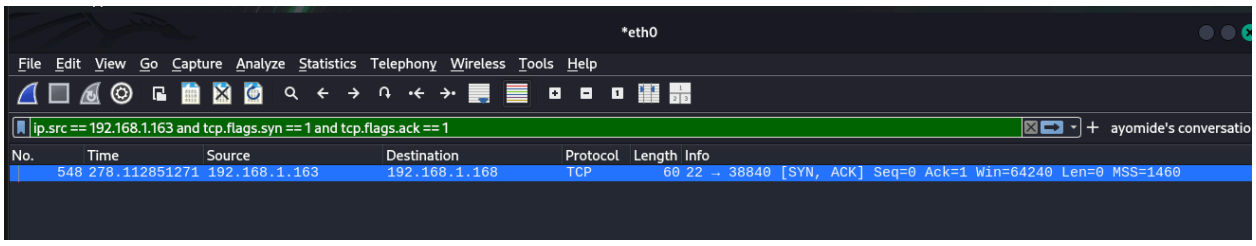


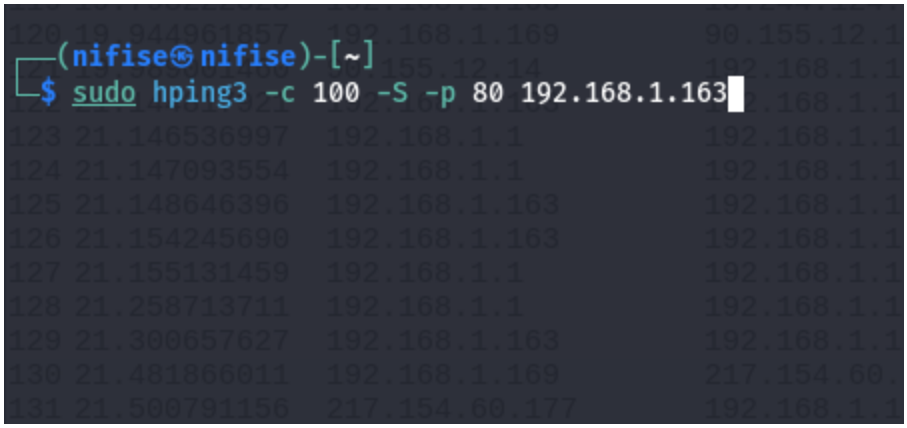
This is the result of the nmap on wireshark:



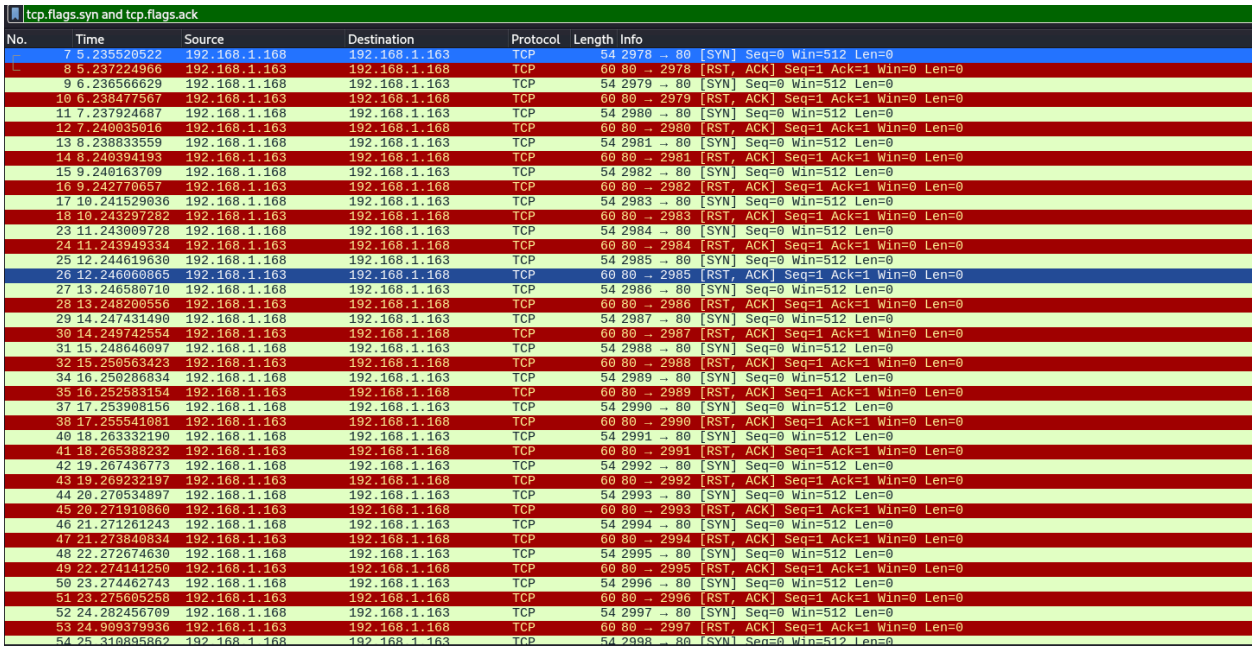
The filter “ip.src == 192.168.1.163 and ip.dst == 192.168.1.168 and tcp.flags.syn and tcp.flags.ack” displays the TCP handshake between the victim machine (ip.src) and the attacker machine (ip.dst). The red portrays the ports which are closed on the victim machine.

And these where there results from the nmap of the only open port, where i filtered the nmap scan using “ip.src == <victim_ip> and tcp.flags.syn == 1 and tcp.flags.ack == 1”. The “tcps.flags.syn==1” means to give the packet where there was a connection to the three way handshake.



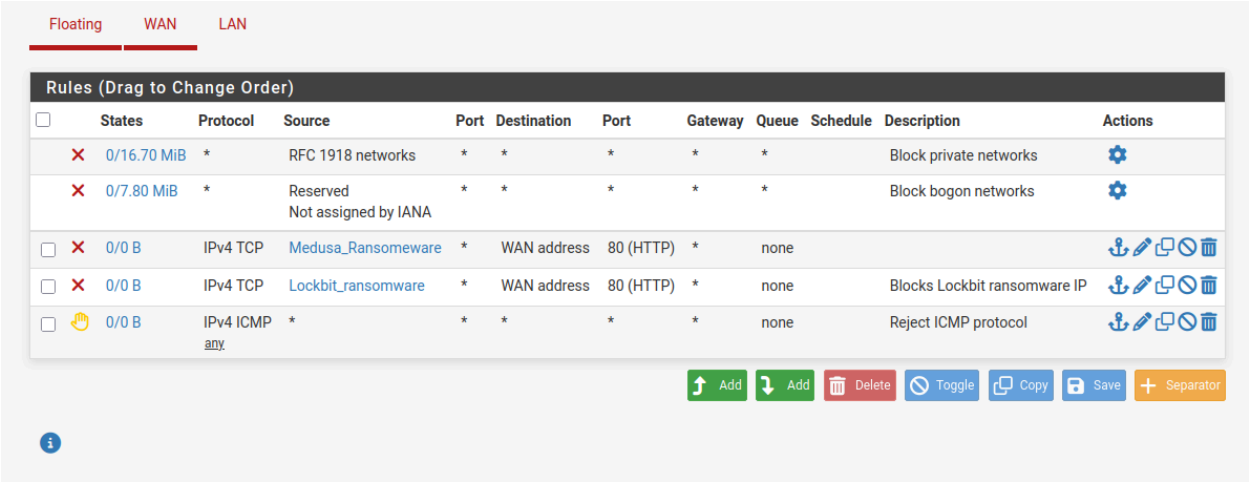


A hping3 ping was ran on the http port 80 to detect anomalous HTTP requests.Proceeded to filtered the attack using “tcp.flags.syn and tcp.flags.ack”.

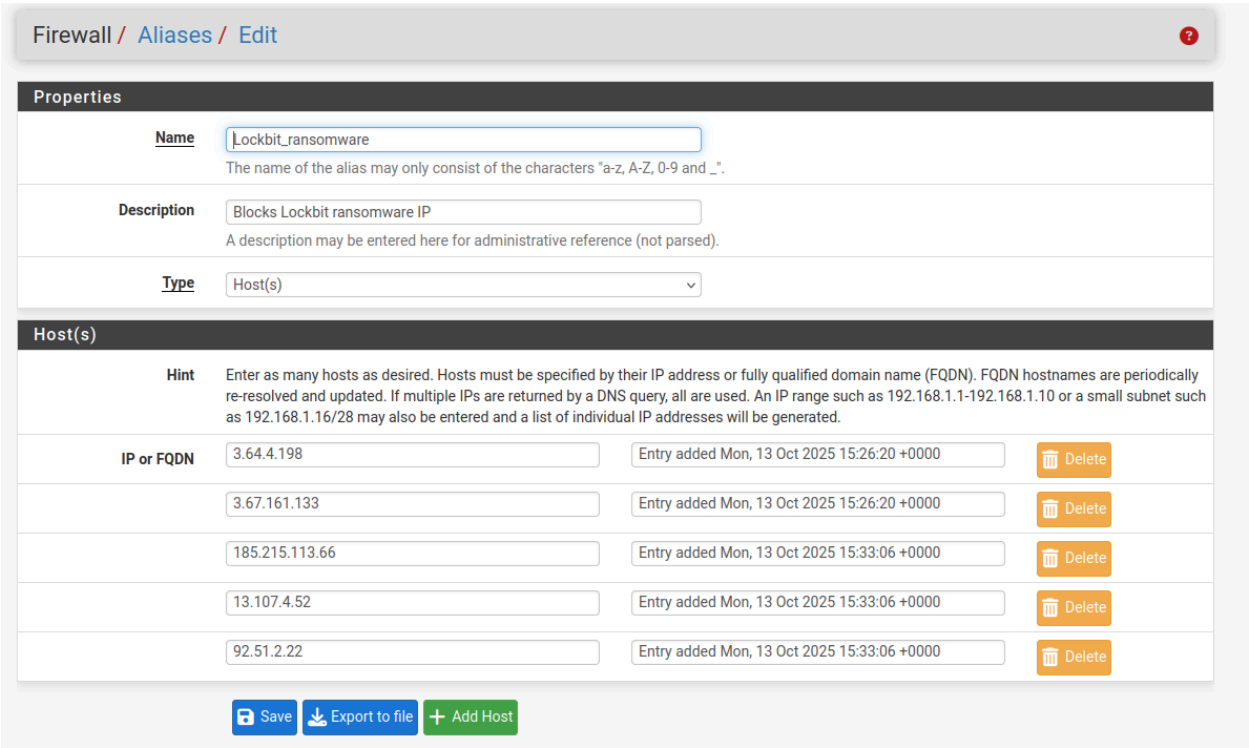


The attacker, 192.168.1.168 repeatedly attempted TCP three-way handshakes to 192.168.1.163 at port 80. The victim responded with a refused connection (RST) for each attempt, indicating no service was listening on port 80. This is consistent with the Nmap scan results showing port 80 as closed.

Phase 2: pfSense and Snort– Firewall & Policy Enforcement



The above screenshot displays the firewall rules which block popular ransomware groups which are Medusa and Lockbit on port 80. It also displays the rule that blocks ICMP packets from happening in the network.



These are the lists IP addresses from Lockbit ransomware group blocked by the firewall.

IP Ports URLs All

Firewall Aliases IP

Name	Type	Values	Description	Actions
Lockbit_ransomware	Host(s)	3.64.4.198, 3.67.161.133, 185.215.113.66, 13.107.4.52, 92.51.2.22	Blocks Lockbit ransomware IP	
Medusa_Ransomeware	Host(s)	18.158.58.205, 18.197.239.109	Blocks ip from medus ransomware	
Phobos_ransomeware	Host(s)	194.165.16.4, 45.9.74.14, 147.78.47.224, 185.202.0.111	Blocks ip from phobos ransomware	

Add Import

The above screenshot shows the list of IP addresses blocked by the firewall and grouped by the threats.

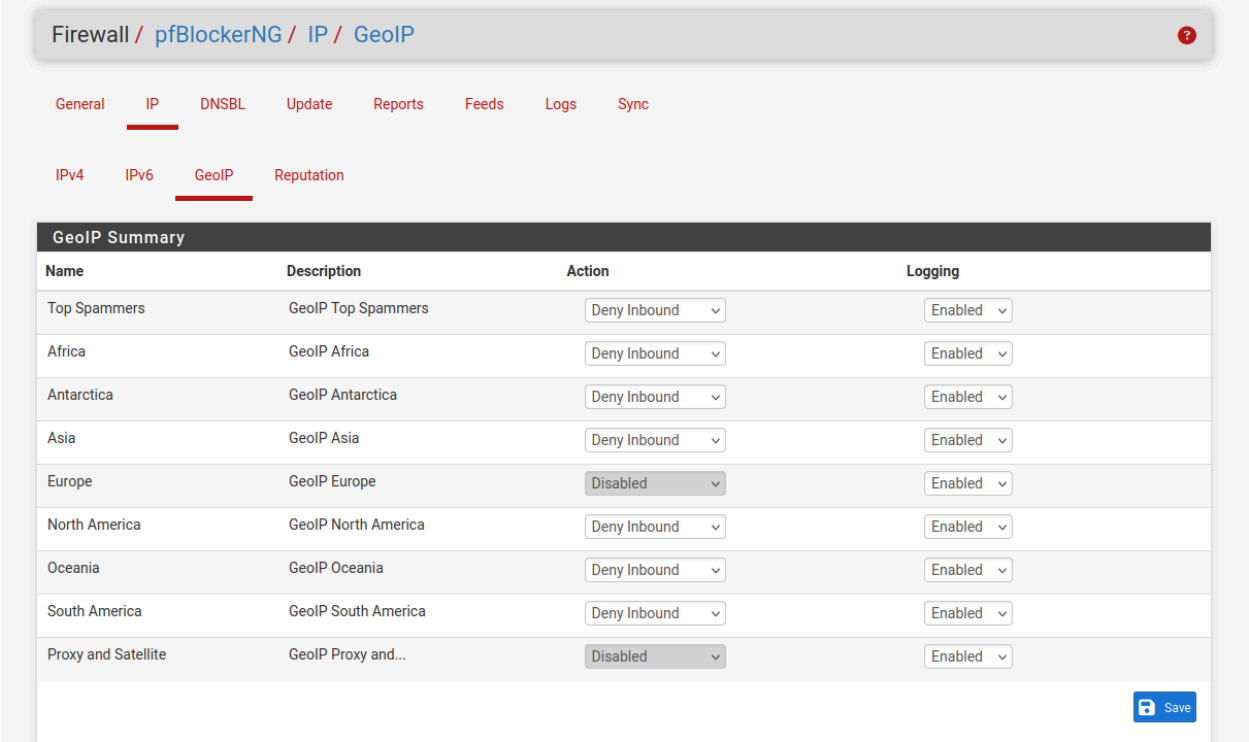
Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
	0/20.93 MiB	*	RFC 1918 networks	*	*	*	*	*		Block private networks	
	0/9.77 MiB	*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogon networks	
	0/0 B	IPv4 TCP	Medusa_Ransomeware	*	WAN address	80 (HTTP)	*	none			
	0/0 B	IPv4 TCP	Lockbit_ransomware	*	WAN address	80 (HTTP)	*	none		Blocks Lockbit ransomware IP	
	0/0 B	IPv4 ICMP any	*	*	*	*	*	none		Reject ICMP protocol	
	0/0 B	IPv4 TCP	*	*	*	22 (SSH)	*	none			

Add Add Delete Toggle Copy Save Separator

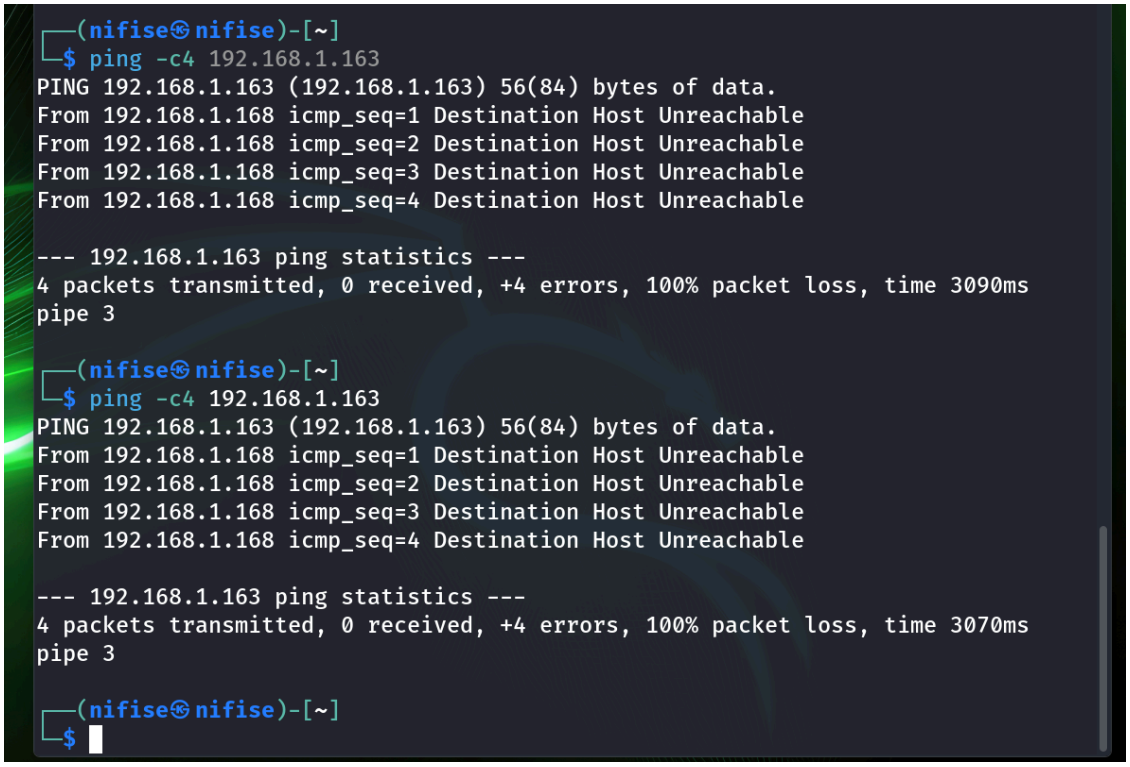
The screenshot above shows the updated firewall rules where SSH is blocked on the WAN.

Below is the firewall rule GeoIP blocking that restricts access from high-risk countries. Blocking most continent's inbound traffic heavily reduces the attack surface and ensures all legitimate activity comes from inside your lab.



Testing

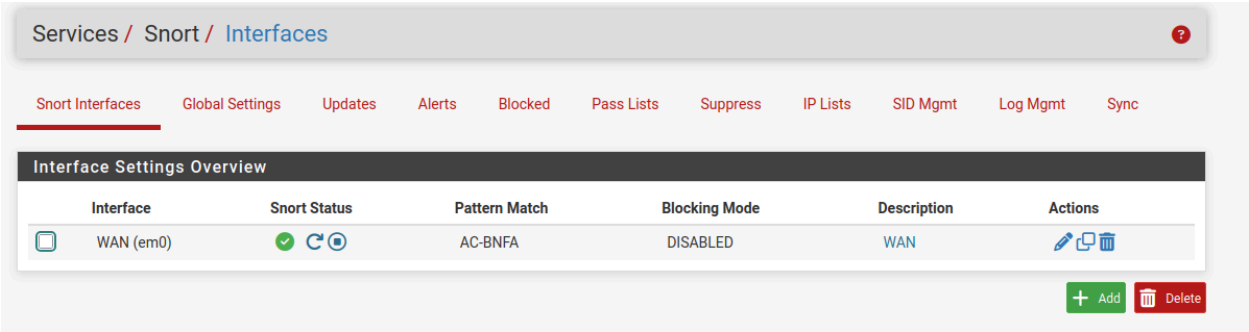
Here is the test for the blocked ICMP firewall rule in the network. The ip pinged was the host network and thanks to the rule the pings were dropped, proving the effectiveness of the rule. In this case, the attacker was behind the firewall with the victim.



When trying to do something similar with an attacker outside of the wirewall, traffic failed to be routed by pfSense.

Snort Intrusion Detection System

Below's screenshot shows the IDS rule which detects offenders and raises alerts in pfsense when there is an nmap or a ping flood on the victim.



Phase 3: Wazuh – Security Event Monitoring & Response

Wazuh is an open-source security monitoring platform designed for threat detection, integrity monitoring, incident response, and compliance.

Deployment of wazuh agent on victim machine

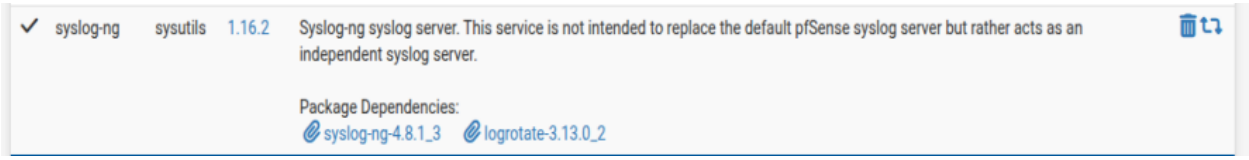
```
root@nifise:/home/nifise# nano /var/ossec/etc/ossec.conf
root@nifise:/home/nifise# sudo systemctl start wazuh-agent
root@nifise:/home/nifise# sudo systemctl status wazuh-agent
● wazuh-agent.service - Wazuh agent
     Loaded: loaded (/usr/lib/systemd/system/wazuh-agent.service; enabled; pres>
     Active: active (running) since Mon 2025-12-22 15:49:57 UTC; 1min 51s ago
```

ID	Status	IP address	Version	Group	Operating system	Cluster node	Registration date	Last keep alive
001	active	192.168.1.163	Wazuh v4.14.1	default	Ubuntu 24.04.3 LTS	node01	Dec 22, 2025 @ 15:49:49.000	Dec 22, 2025 @ 16:05:09.000

Setting up syslog to forward pfSense logs to wazuh

Navigate to System > Package Manager > Available Packages and search for syslog the only result will be the syslog-ng package. Click on the install button to add it to the firewall.

https://devopstales.github.io/linux/wazuh-pfsense-syslog/



Navigate to Services > Syslog-ng > Settings Tab and set the syslog-ng on the GUI as the image below and click on the save button.

General Options

Enable
☑ Select this option to enable syslog-ng

Interface Selection
LAN
WAN
loopback
Select interfaces you want to listen on

Default Protocol
UDP
Select the default protocol you want to listen on

CA
Select Certificate Authority for TLS protocol.
You can use it in your object definition as ca-dir('/var/etc/syslog-ng/ca.d') option of tls( ).

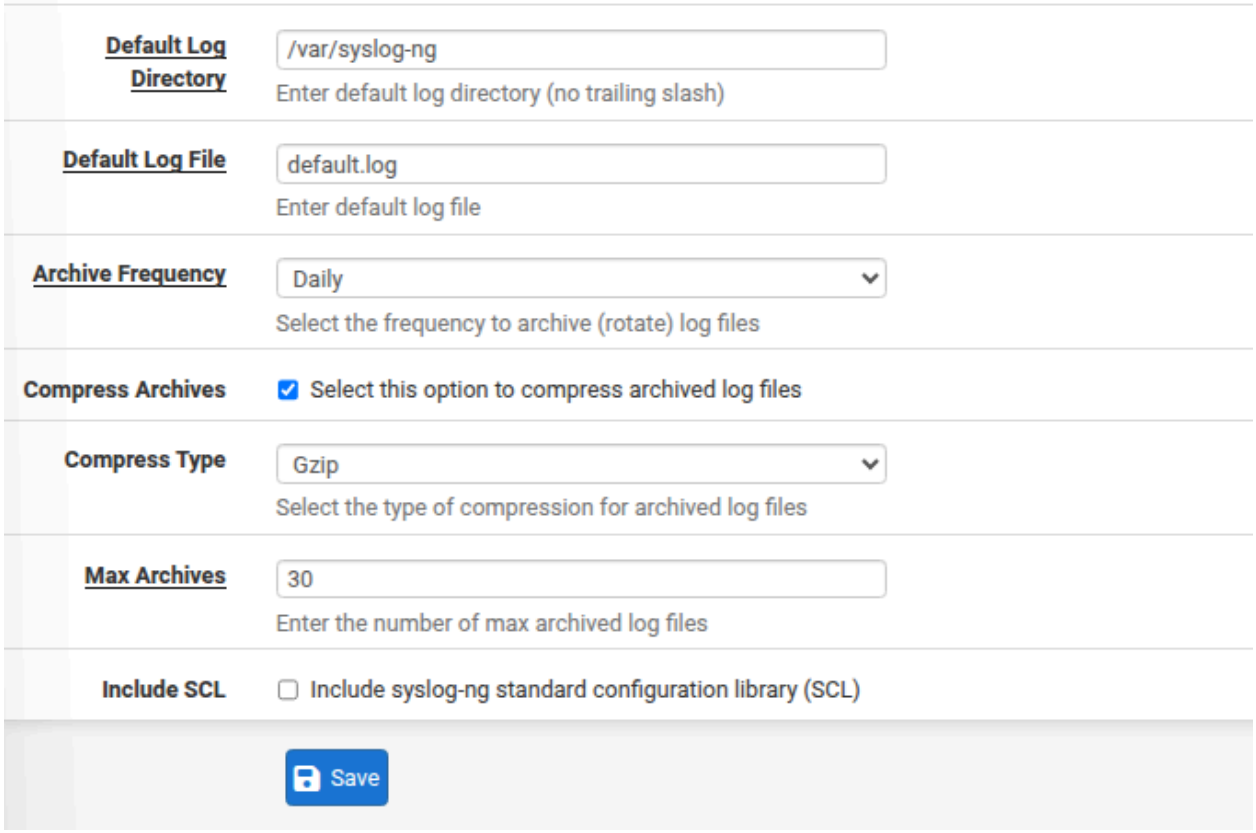
Certificate
GUI default (68eced1cf11ea)
Select server certificate for TLS protocol.
You can use it in your object definition as key-file('/var/etc/syslog-ng/syslog-ng.key') and cert-file('/var/etc/syslog-ng/syslog-ng.cert') options of tls( ).

Default Port
5140
Enter default port number you want to listen on

Default Log Directory
/var/syslog-ng
Enter default log directory (no trailing slash)

Default Log File
default.log
Enter default log file

Archive Frequency
Daily
Select the frequency to archive (rotate) log files



Navigate to Status > System Logs > Settings Tab and at the button check the Enable Remote Logging checkbox.
Set the settings as in the picture below and click the save button.

Packages Settings

General Logging Options

Log Message Format

BSD (RFC 3164, default)

The format of syslog messages written to disk locally and sent to remote syslog servers (if enabled).
Changing this value will only affect new log messages.

Forward/Reverse Display

☑ Show log entries in reverse order (newest entries on top)

GUI Log Entries

500

This is only the number of log entries displayed in the GUI. It does not affect how many entries are contained in the actual log files.

Raw Logs

☐ Show raw filter logs

If this is checked, filter logs are shown as generated by the packet filter, without any formatting. This will reveal more detailed information, but it is more difficult to read.

Where to show rule descriptions

Display as column

Show the applied rule description below or in the firewall log rows.
Displaying rule descriptions for all lines in the log might affect performance with large rule sets.

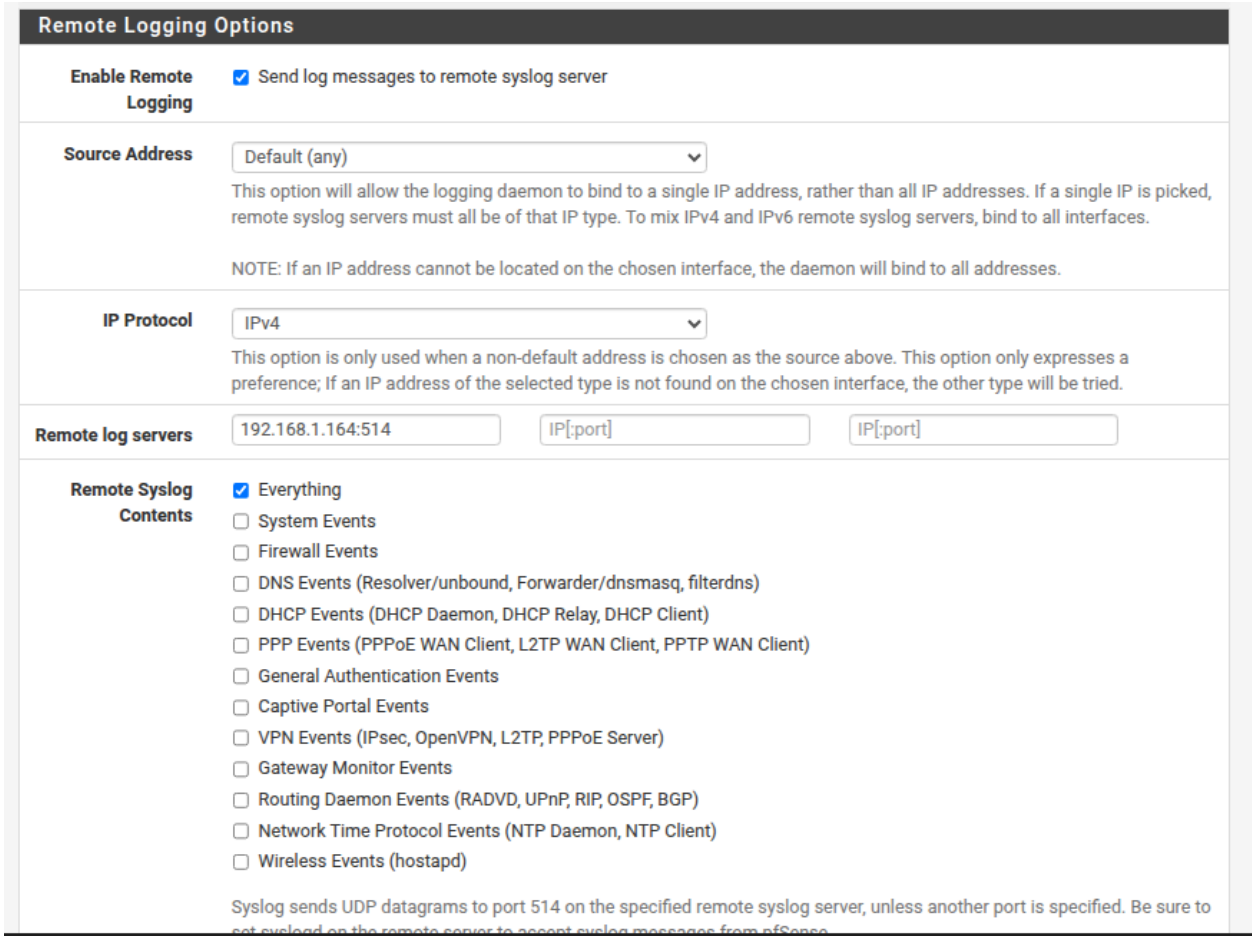
Local Logging

☐ Disable writing log files to the local disk

WARNING: This will also disable Login Protection!

Reset Log Files

Reset Log Files



Configure Wazuh syslog input

Edit the /var/ossec/etc/ossec.cfg on the Wazuh Manger with

nano /var/ossec/etc/ossec.conf

Paste:

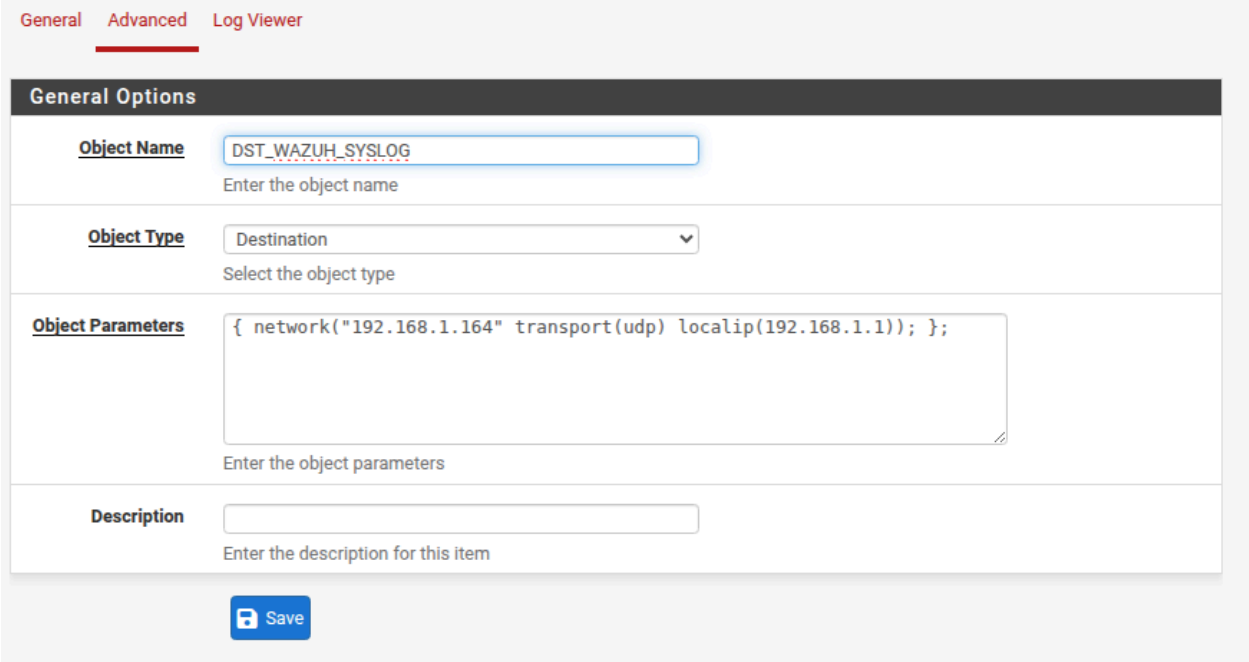
```
<!-- pfsense syslog input -->
  <remote>
    <connection>syslog</connection>
    <port>5514</port>
    <protocol>tcp</protocol>
    <allowed-ips>pfSense_IP</allowed-ips>
    <local_ip>wazuh_IP</local_ip>
  </remote>
  <remote>
    <connection>syslog</connection>
    <port>5514</port>
    <protocol>udp</protocol>
    <allowed-ips>pfSense_IP</allowed-ips>
    <local_ip>wazuh_IP</local_ip>
```

</remote>

Sending syslog-ng Logs to Remote Server

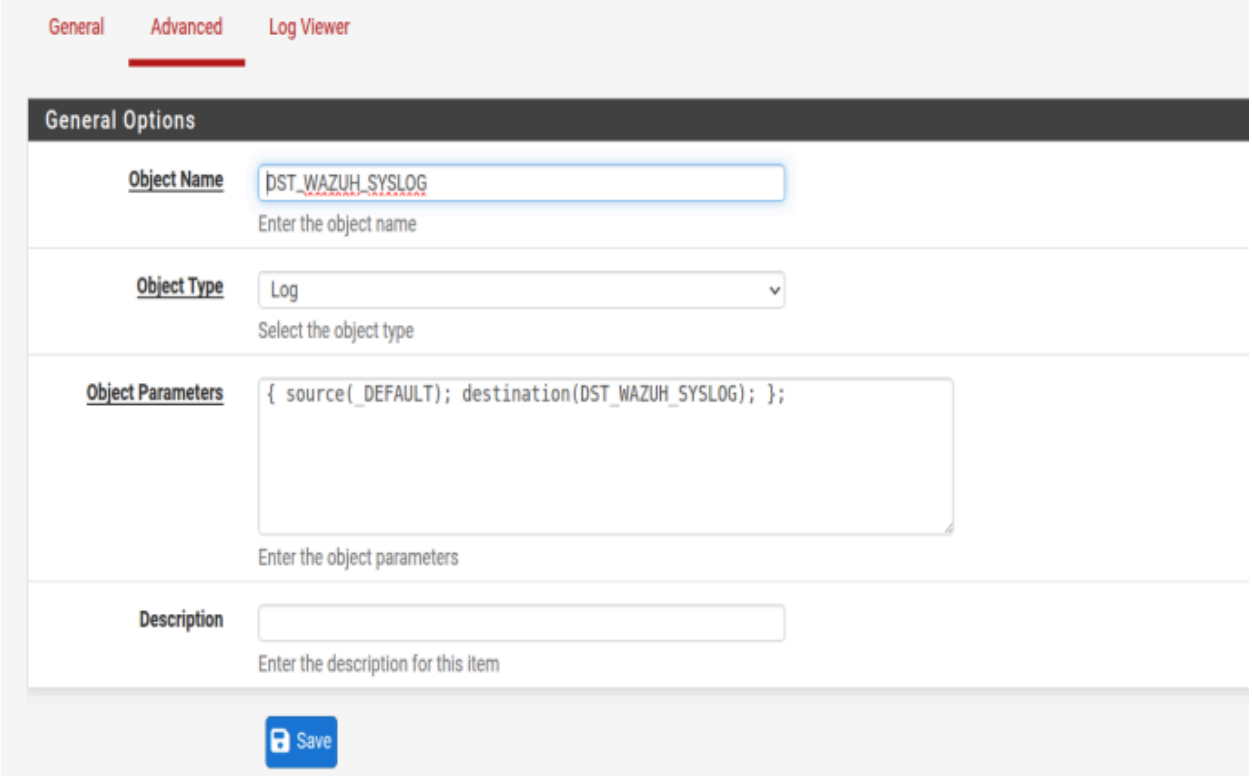
First, we need to add a new destination entry named DST_WAZUH_SYSLOG. Navigate to Services > Syslog-ng > Advanced Tab and add a new destination as in the picture below.

{ network("wazuh_IP" transport(udp) localip(192.168.1.1)); };

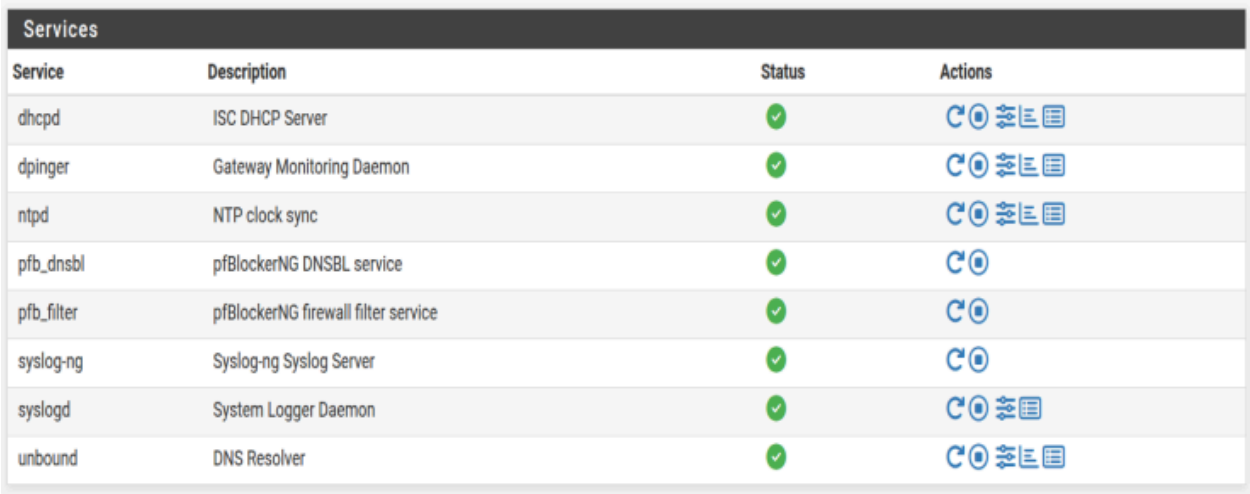


After adding the destination we need to connect with the remote server adding a new log object as seen below.

{ source(_DEFAULT); destination(DST_WAZUH_SYSLOG); };



Check if the Service is running.



Services

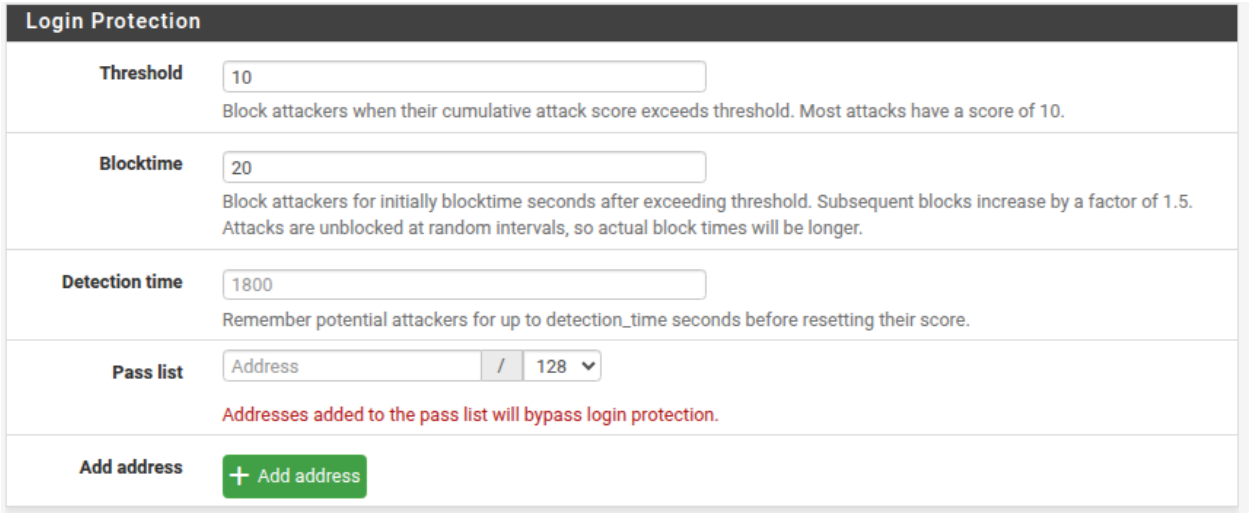
Service	Description	Status	Actions
dhcpd	ISC DHCP Server		
dpinger	Gateway Monitoring Daemon		
ntpd	NTP clock sync		
pfb_dnsbl	pfBlockerNG DNSBL service		
pfb_filter	pfBlockerNG firewall filter service		
syslog-ng	Syslog-ng Syslog Server		
syslogd	System Logger Daemon		
unbound	DNS Resolver		

Hardening pfSense

Enabling login protection on pfSense GUI: This gives clear logs for GUI brute force that Wazuh can alert on. System > Advanced > Admin Access.

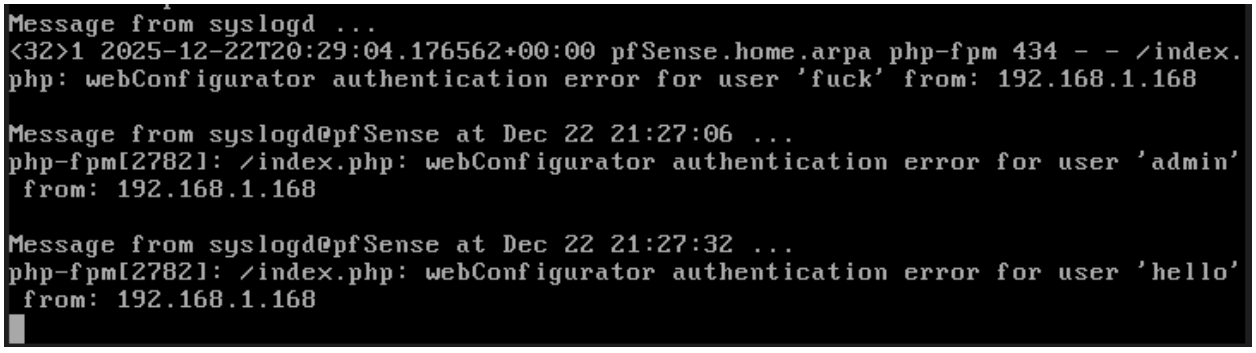
In Login Protection:

- Set Threshold (max failed logins before block) to something like 10.
- Set Block Time to at least 20.
- Save.

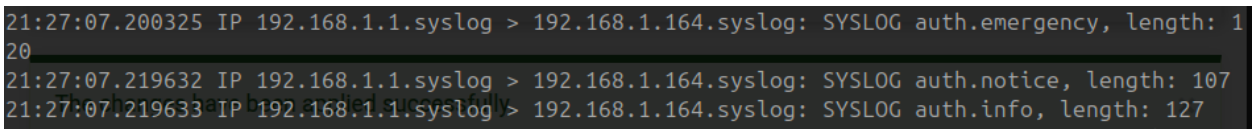


That should do it.

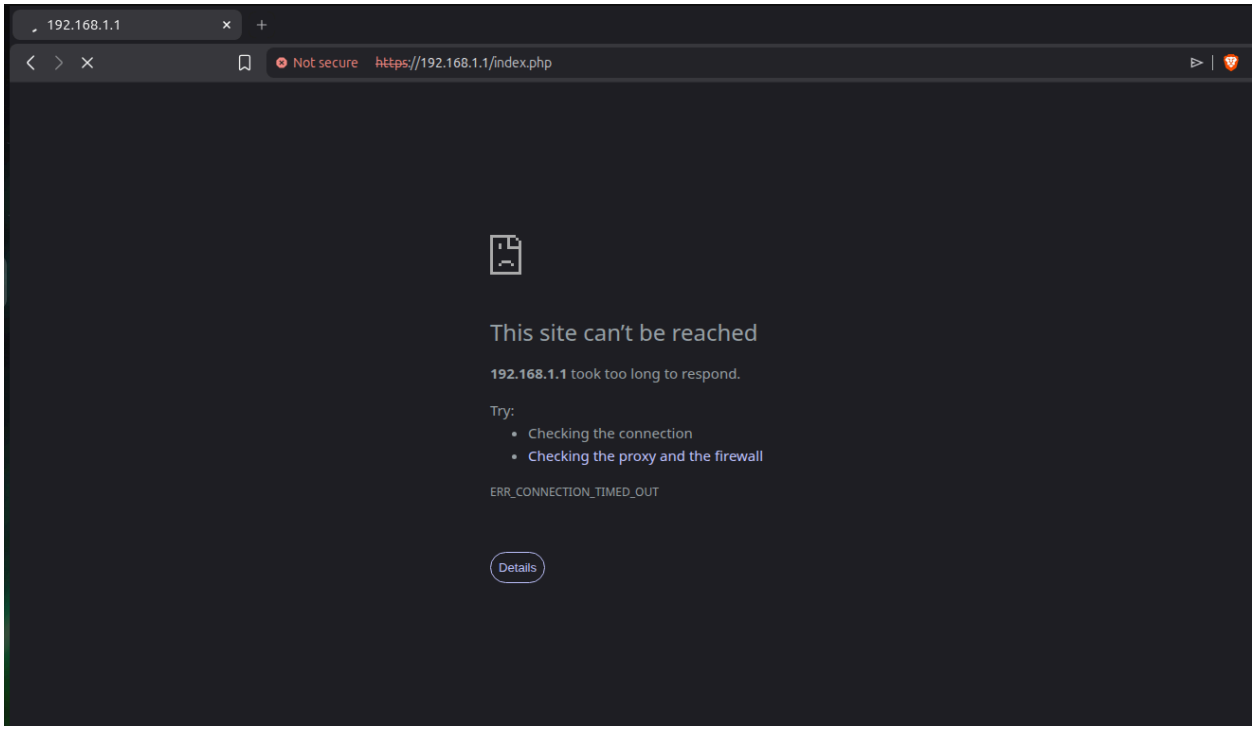
So after multiple failed attempts of logging in, the syslog showed multiple authentication errors.



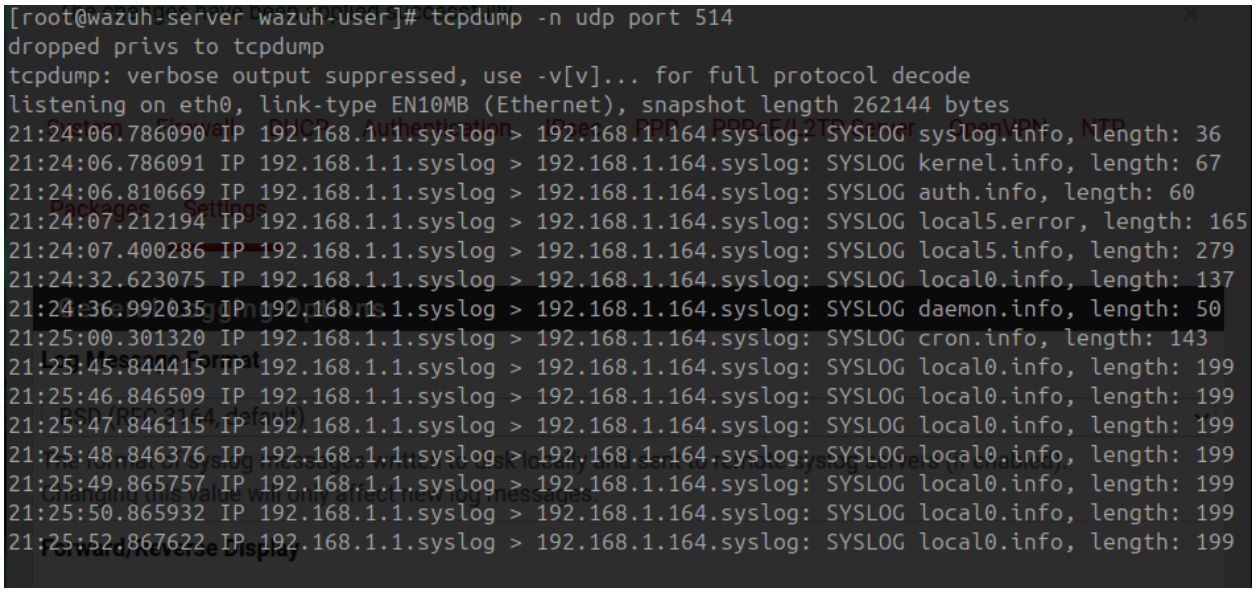
As well as wazuh:



And the web GUI stopped working until the assigned time was over.



Other pfSense logs being sent to wazuh:



Bruteforce protection

We will be bruteforcing the ssh port. Wazuh comes with a set of default scripts used in Active Response which works with Linux/Unix operating systems and uses its iptables to block malicious IP addresses.

Open /var/ossec/etc/ossec.conf and add

```
<ossec_config>
  <active-response>
    <disabled>no</disabled>
    <command>firewall-drop</command>
    <location>local</location>
    <rules_id>5763</rules_id>
    <timeout>180</timeout>
  </active-response>
</ossec_config>
```

Then restart wazuh manager.

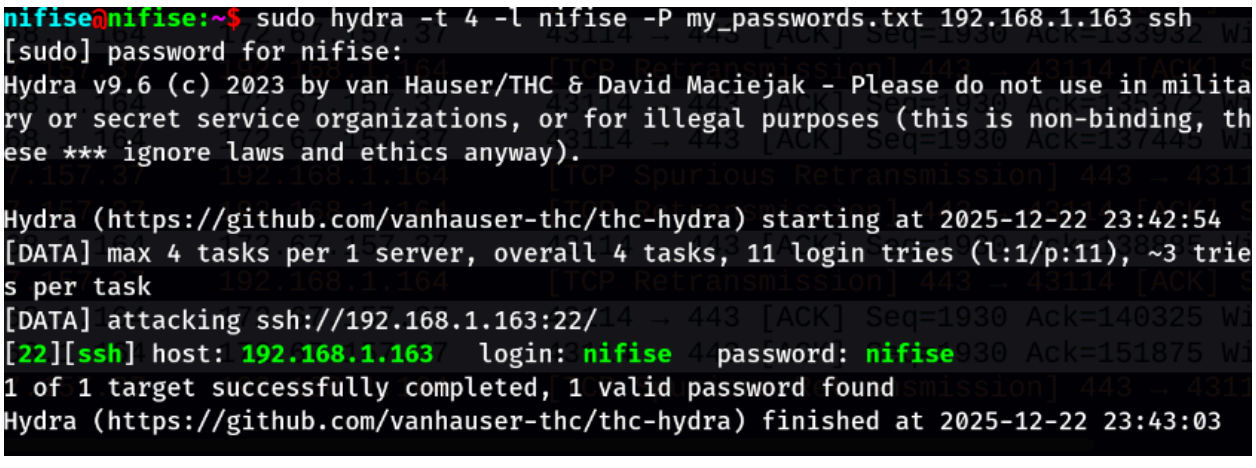
The wazuh configuration enables an active response that automatically blocks the IP address with pfSense when the rule 5763 has been broken. Rule 5763 triggers when multiple authentication failures occur.

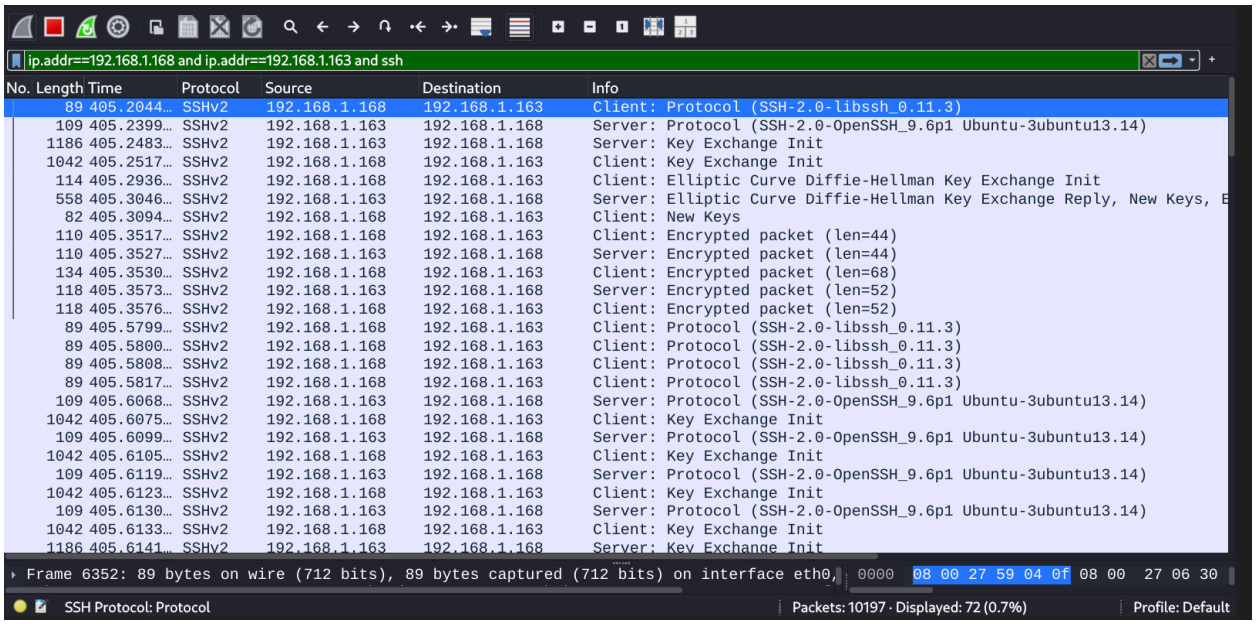
Attack Process

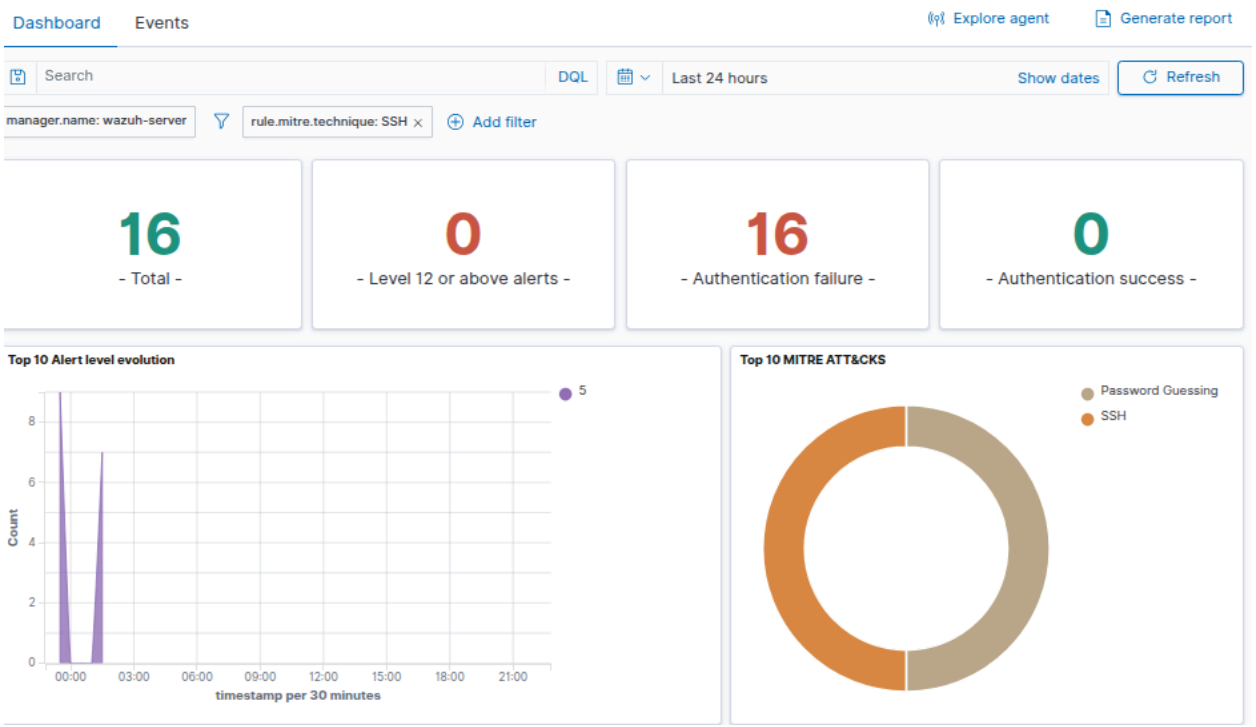
Ping the victim machine to ensure it is up.

If it is up, then go on to use hydra to brute force.

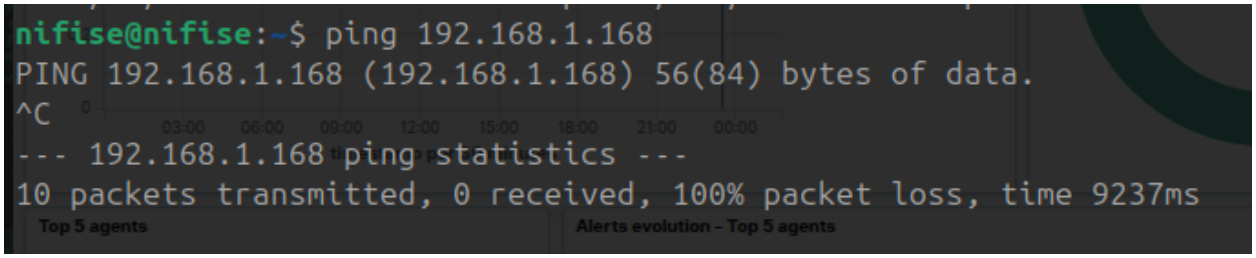
sudo hydra -t 4 -l <VICTIM_USERNAME> -P <PASSWD_LIST.txt> <VICTIM_IP> ssh







Wazuh logged the bruteforce and blocked the attacker IP for the duration of the timeout via firewall-drop, then removed the block.



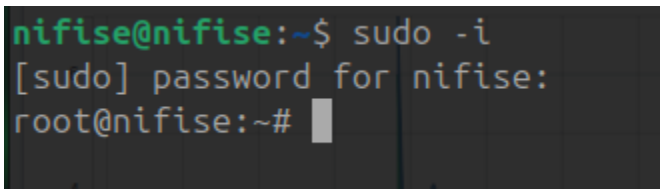
As shown in the screenshot above, wazuh blocked the IP address of the attacker machine temporarily.

Looking at the wazuh agent logs using

sudo cat /var/ossec/logs/active-responses.log

The logs shown mean the brute-force was successfully detected and alerted on. Wazuh triggered Active Response and temporarily blocked the attacker IP according to your configuration.

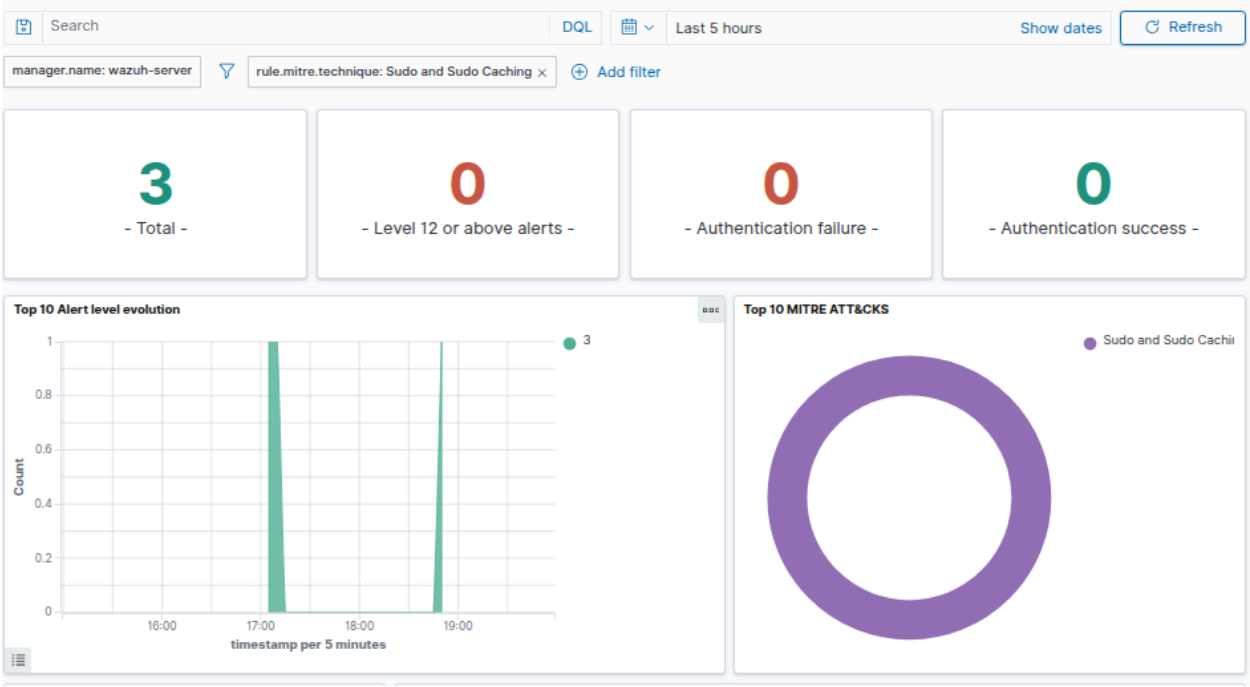
Privilege Escalation



Escalating my privilege with the command to root user with

sudo -i

By default, wazuh already alerts privilege escalation of users in the network so not much configuration was needed.



Malware detection

Wazuh File Integrity Monitoring (FIM) can monitor changes in a directory and using an external VirusTotal API, VirusTotal can scan the files in the directory. Configuring an active response module similar to the one used for bruteforce protection to remove the files that virustotal detects as malicious.

Wazuh FIM setup:

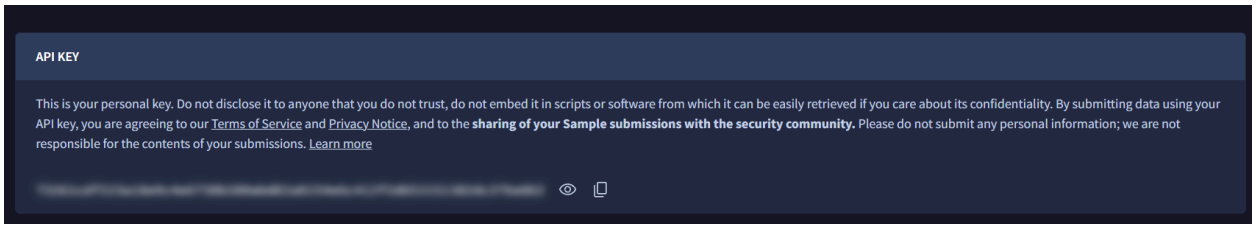
I checked if the File Integrity Monitoring is enabled on the wazuh agent by going to the **/var/ossec/etc/ossec.conf** and searching for **<syscheck>** block. **<disabled>** has to be set to **no**.

In the same block, mention the directory you want to be monitored:

<directories realtime="yes">/home/nifise/Downloads</directories> <!-- My rule -->

Getting VirusTotal API key:

Go to the virustotal website and join the community https://www.virustotal.com/gui/home/upload.



Copy the API key.

Enable the VirusTotal integration:

SSH into the wazuh server and open the **ossec.conf** on the manager. Then scroll down to the end </ossec_config> and right above it add:

```
<integration>
  <name>virustotal</name>
  <api_key>YOUR_API_KEY_HERE</api_key> <!-- Replace with your VirusTotal API key -->
  <group>syscheck</group>
  <alert_format>json</alert_format>
</integration>
```

When everything is set, restart the manager and agent.

sudo systemctl restart wazuh-manager

sudo systemctl restart wazuh-agent

Testing

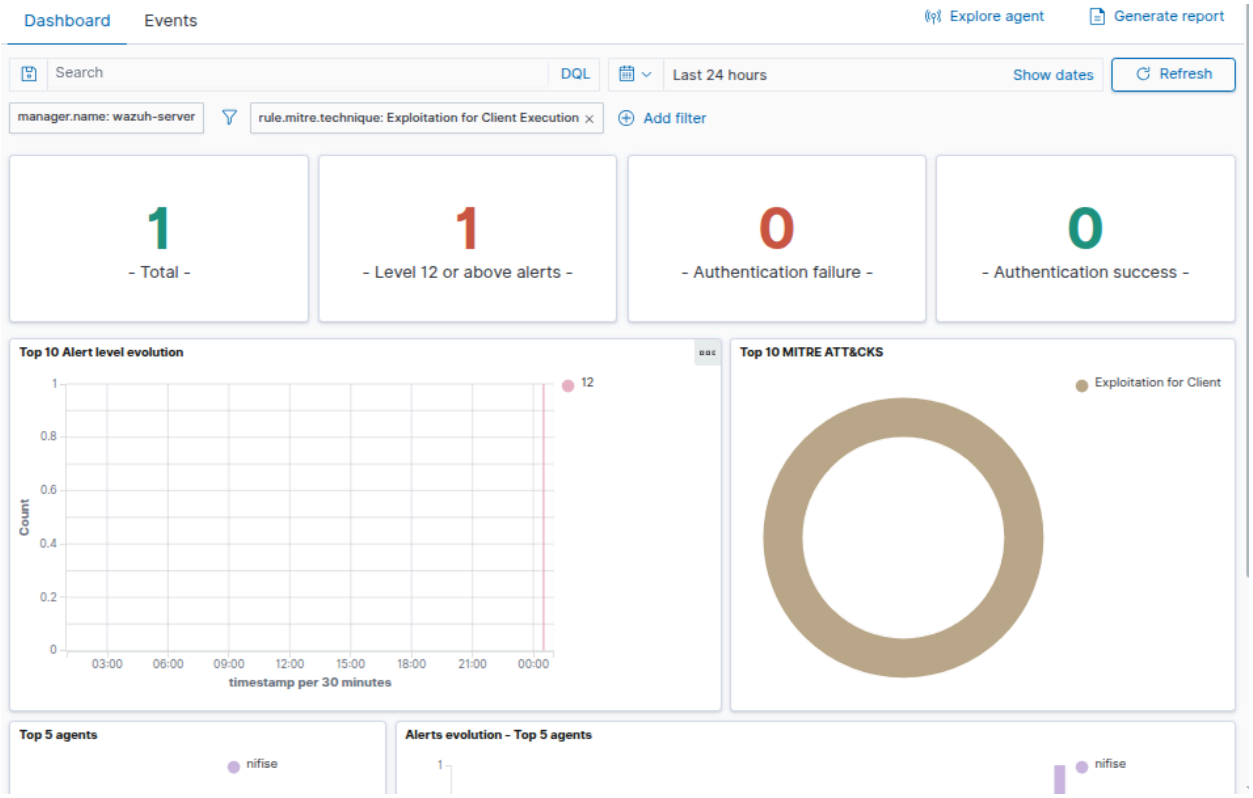
Now, we can download a malicious file on the endpoint in the monitored folder.

This should be downloaded in a non productive environment as it is for testing purposes only.

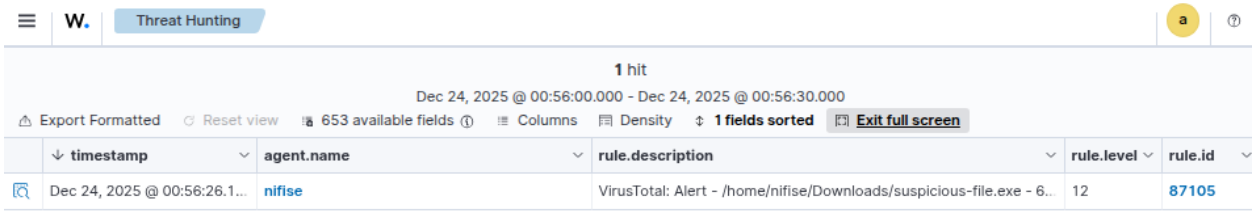
sudo curl -Lo /home/nifise/Downloads/suspicious-file.exe https://secure.eicar.org/eicar.com

```
nifise@nifise:~$ sudo curl -Lo /home/nifise/Downloads/suspicious-file.exe https://secure.eicar.org/eicar.com
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    68  100    68    0     0    298      0 --:--:-- --:--:-- --:--:--   299
```

Wazuh Identified the malware and identifying it as a high level alert:



Events Tab:



After it has been alerted, the next phase of action is to disconnect the device from the network and delete the malware.

---

5. Final Findings & Impact

The engagement confirmed that Femalogy was susceptible to multiple attempts by an attacker to gain access to a victim's information and identify possible vulnerabilities like open ports. By implementing GeoIP blocking and inbound firewall rules, most malicious scans were successfully filtered by pfSense, the firewall, before reaching the internal LAN. Snort's Intrusion Detection System alerted the network of the malicious traffic and wazuh was supposed to log the events. This report is incomplete due to some connectivity issues between the attacker and the victim machine. Kali, being outside of the machine, had a different default gateway than that of Ubuntu which was the firewall. And pfSense had issues trying to route traffic from the attacker to the victim, despite being on the same subnet mask. In the process of trying to solve these issues, they took more time than expected, therefore, making the report incomplete.

After multiple troubleshooting efforts, the attacker machine has been unable to access the pfSense, despite being on the same LAN. To complete the project, I decided to make the attacker machine an insider threat as it is very realistic, because most real-world breaches come from inside (lateral movement, privilege escalation, insider threat, compromised endpoint, etc.).

From my research, I discovered how to prevent brute-force attacks on devices in the network, and that Wazuh could use a method called active response to block an IP address on the firewall level. I also discovered how to integrate software like VirusTotal with Wazuh to detect and analyse malicious files in the network. Moreover, I learnt how to send pfSense logs to Wazuh using syslog and that wazuh can be ssh'd into and the importance of blocking ssh on the firewall..

At the time of concluding this report, my SOC analyst skills have been improved upon.

---

6. Recommendations

Based on findings, the following are recommended for the organization to regularly check the firewall for alerts and constantly update rules. To minimize exposure of networks from high risk countries, geoIP blocking should be enabled to protect the network. Restricting SSH to only VPN- tunnel connections should also be done. Isolation of infected devices to prevent malware lateral movement and privilege escalation. Hardening pfSense log-in by limiting how many times credentials can be used in a specific time.

---

7. Conclusion

This confirmed that Femalogy's network was exposed to external reconnaissance attempts. Through the use of GeoIP blocking, strict inbound firewall rules, IDSs with Snort and Wazuh's logging, the attack surface was reduced, and malicious activity was detected and logged. The improved security measures effectively strengthened the organization's ability to identify itself and respond to threats. Also, by restricting remote access, enhancing firewall rules, and maintaining constant logging, Femalogy now has an effectively secured network and capable SOC group to protect the network from outsiders.

---

8. References

- Wireshark Documentation
- pfSense and IDS/IPS Configuration - https://docs.netgate.com/pfsense/en/latest/packages/snort/index.html
- Wazuh Official Guide - https://documentation.wazuh.com/current/installation-guide/wazuh-agent/index.html
- MITRE ATT&CK Framework
- Logs and dashboards from the lab environment

- Raw logs, alert data, and full packet captures
- How-to-send-pfsense-log-to-wazuh Repository https://github.com/oyelaa99/how-to-send-pfsense-log-to-wazuh-
- 🕵️WAZUH📊 12. pfSense 📈 MONITORING | SYSLOG | INDEX PATTERN https://www.youtube.com/watch?v=y1Zjs5L3PT8
- Useful Wazuh Rules and Capabilities for Threat Detection https://medium.com/@ismapersonal97/useful-wazuh-rules-and-capabilities-for-threat-detection-e2cc0debabde
- VirusTotal Integration https://documentation.wazuh.com/current/user-manual/capabilities/malware-detection/virus-total-integration.html
-

---

9. Appendices

The screenshots were added above with explanations.