

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

Table of Contents.....	1
1. Executive Summary.....	2
2. Project Introduction.....	2
3. Methodology.....	3
4. Phase-by-Phase Analysis.....	3
Phase 1: Wireshark – Network Traffic Capture & Analysis.....	3
Phase 2: pfSense and Snort- Firewall & Policy Enforcement.....	9
Testing.....	11
Snort Intrusion Detection System.....	12
Phase 3: Wazuh – Security Event Monitoring & Response.....	13
Deployment of wazuh agent on victim machine.....	13
Setting up syslog to forward pfSense logs to wazuh.....	13
Sending syslog-ng Logs to Remote Server.....	18
Hardening pfSense.....	19
Bruteforce protection.....	22
Attack Process.....	22
Privilege Escalation.....	24
Malware detection.....	25
Wazuh FIM setup:.....	25
Getting VirusTotal API key:.....	25
Enable the VirusTotal integration:.....	26
Testing.....	27

5. Final Findings & Impact.....	28
6. Recommendations.....	29
7. Conclusion.....	29
8. References.....	29
9. Appendices.....	30

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

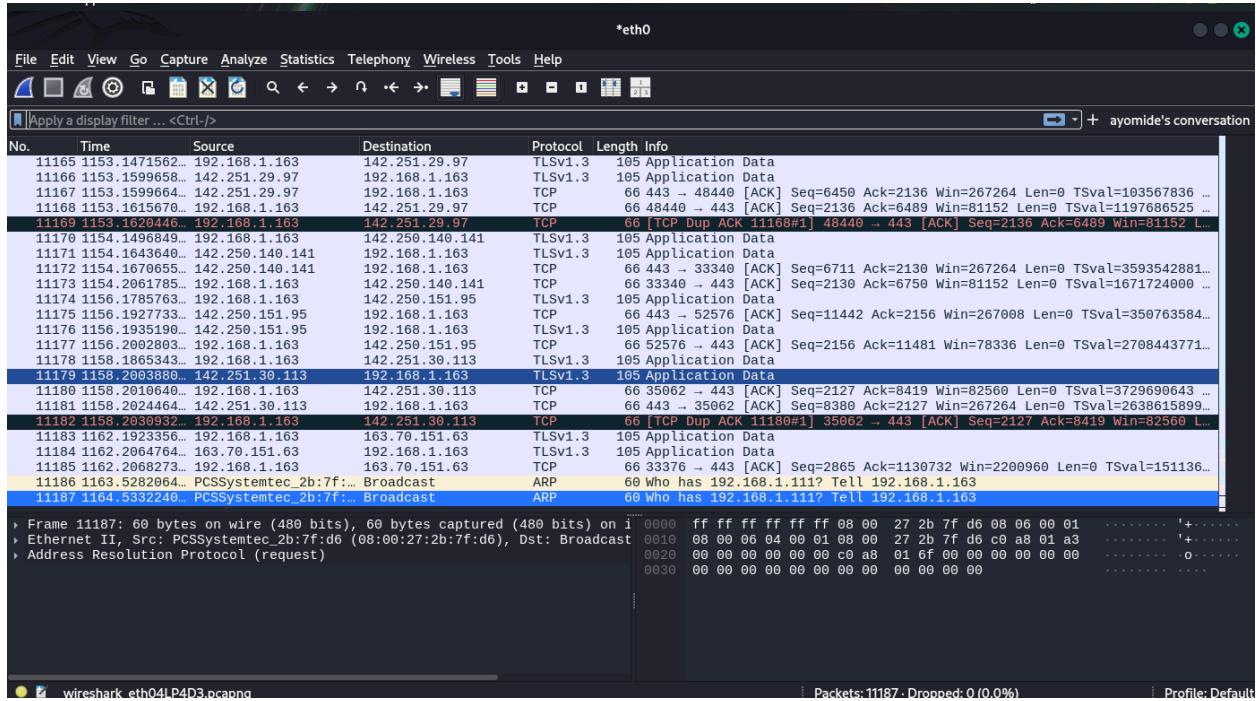
and outgoing network traffic based on configured rules. These tools provided a demonstration of how all these tools can work together to protect Femlogy's network from inside and outside threats.

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

No. Time Source dns Destination Protocol Length Info

12	31.154228272	192.168.1.163	192.168.1.1	DNS	106	Standard query 0xc302 A content-signature-2.0
13	31.157922555	192.168.1.163	192.168.1.1	DNS	106	Standard query 0x7ca4 AAAA content-signature
15	31.1546060346	192.168.1.163	192.168.1.163	DNS	192	Standard query response 0xc302 A content-signature
16	31.1547829067	192.168.1.1	192.168.1.163	DNS	204	Standard query response 0x7ca4 AAAA content-signature
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.firefox
83	37.808669573	192.168.1.1	192.168.1.163	DNS	206	Standard query response 0xf4df A detectportal.firefox
84	37.822080590	192.168.1.1	192.168.1.163	DNS	218	Standard query response 0x3700 AAAA detectportal.firefox
92	41.874463216	192.168.1.163	192.168.1.1	DNS	108	Standard query 0xd19 A firefox.settings.ser
93	41.904940336	192.168.1.163	192.168.1.1	DNS	108	Standard query 0x5e48 AAAA firefox.settings.ser
94	41.9356051141	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 OPT
96	42.097013654	192.168.1.1	192.168.1.163	DNS	160	Standard query response 0xd19 A firefox.set
97	42.097014445	192.168.1.1	192.168.1.163	DNS	172	Standard query response 0x5e48 AAAA firefox.set
100	42.134405085	192.168.1.1	192.168.1.163	DNS	155	Standard query response 0xe48c A ads.mozilla.org
102	42.139971208	192.168.1.1	192.168.1.163	DNS	232	Standard query response 0x98b6 AAAA ads.mozilla.org
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.ads.prod
141	42.848481869	192.168.1.163	192.168.1.1	DNS	98	Standard query 0x93b3 A services.addons.mozilla.org
142	42.849427207	192.168.1.163	192.168.1.1	DNS	98	Standard query 0x3e25 AAAA services.addons.mozilla.org
152	42.953106346	192.168.1.1	192.168.1.163	DNS	210	Standard query response 0x3e25 AAAA services.addons.mozilla.org
160	43.005062483	192.168.1.1	192.168.1.163	DNS	162	Standard query response 0x93b3 A services.addons.mozilla.org
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 (0:00.000000 - 0:00.000000) on interface eth0
Ethernet II, Src: PCSsystemtec_0f:72:a9 (08:00:27:0f:72:a9), Dst: 192.168.1.1 (08:00:27:00:00:01)
Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.1
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 ...5.
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 tal f
0050 00 1c 00 01 c0 0c 00 05 00 01 00 00 00 3c 00 1e ...
0060 00 64 65 74 65 63 74 70 6f 72 74 61 6c 04 70 72 detec
0070 6f 64 06 6d 6f 7a 61 77 73 03 6e 65 74 00 c0 36 od-mo
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 detec

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

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 /
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 /
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 /
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 /
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 (0:00.000000 - 0:00.000000) on interface eth0
Ethernet II, Src: PCSsystemtec_2b:7f:d6 (08:00:27:2b:7f:d6), Dst: 192.168.1.163 (08:00:27:00:00:01)
Internet Protocol Version 4, Src: 192.168.1.163, Dst: 34.107.221.82
Transmission Control Protocol, Src Port: 49840, Dst Port: 80, Sequence Number: 1, Acknowledgment Number: 2, Flags: S
Hypertext Transfer Protocol

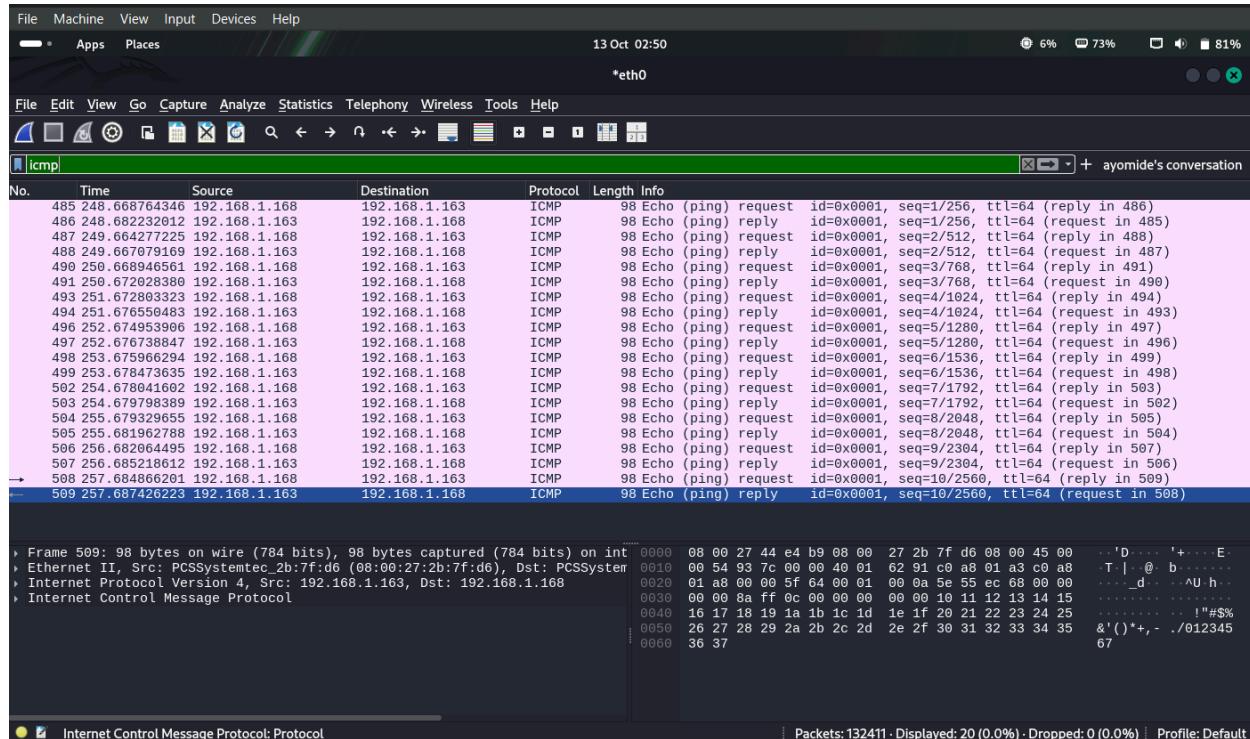
0000 08 00 27 0f 72 a9 08 00 27 2b 7f d6 08 00 45 00 ...'+.
0010 01 61 22 38 40 00 40 06 55 56 c0 a8 01 a3 22 6b a"8@
0020 dd 52 c2 b0 00 50 6a 9b 99 0d 0f 83 67 61 80 18 R...
0030 01 f6 bf 5b 00 00 01 01 08 0a 85 73 b2 49 fe 81 ...[
0040 2a 71 47 45 54 20 2f 63 61 6e 6f 6e 69 63 61 6c *qGET
0050 2e 68 74 6d 6c 20 48 54 54 50 2f 31 2e 31 0d 0a .html
0060 48 6f 73 74 3a 20 64 65 74 65 63 74 70 6f 72 74 Host:
0070 61 6c 2e 66 69 72 65 66 6f 78 2e 63 6f 6d 0d 0a al.fi
0080 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 User-
0090 6c 6c 61 2f 35 2e 30 20 28 58 31 31 3b 20 55 62 lla/5

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.



Nmap 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.

```
(nifise㉿nifise)-[~]$ nmap -sS -p- 192.168.1.163
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-13 02:27 BST
Nmap scan report for 192.168.1.163
Host is up (0.0026s latency).
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 08:00:27:2B:7F:D6 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 174.05 seconds
```

This is the result of the nmap on wireshark:

*eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src == 192.168.1.163 and ip.dst == 192.168.1.168 and tcp.flags.syn and tcp.flags.ack

No. Time Source Destination Protocol Length Info

533	278.092384078	192.168.1.163	192.168.1.168	TCP	60 139 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
524	278.095776912	192.168.1.163	192.168.1.168	TCP	60 53 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
529	278.099149578	192.168.1.163	192.168.1.168	TCP	60 199 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
530	278.099150260	192.168.1.163	192.168.1.168	TCP	60 111 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
531	278.099150420	192.168.1.163	192.168.1.168	TCP	60 80 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
535	278.101777840	192.168.1.163	192.168.1.168	TCP	60 554 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
536	278.102967263	192.168.1.163	192.168.1.168	TCP	60 1025 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
537	278.102968375	192.168.1.163	192.168.1.168	TCP	60 8888 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
539	278.107436284	192.168.1.163	192.168.1.168	TCP	60 995 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
540	278.107437015	192.168.1.163	192.168.1.168	TCP	60 21 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
544	278.109685903	192.168.1.163	192.168.1.168	TCP	60 143 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
548	278.109685271	192.168.1.163	192.168.1.168	TCP	60 22 → 38840 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
550	278.114918942	192.168.1.163	192.168.1.168	TCP	60 3306 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
551	278.114919694	192.168.1.163	192.168.1.168	TCP	60 587 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
552	278.115627426	192.168.1.163	192.168.1.168	TCP	60 25 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
553	278.115628157	192.168.1.163	192.168.1.168	TCP	60 23 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
558	278.119229860	192.168.1.163	192.168.1.168	TCP	60 445 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
559	278.119230491	192.168.1.163	192.168.1.168	TCP	60 5900 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
560	278.119230621	192.168.1.163	192.168.1.168	TCP	60 1723 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
561	278.119230741	192.168.1.163	192.168.1.168	TCP	60 443 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
565	278.121756428	192.168.1.163	192.168.1.168	TCP	60 113 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
566	278.121756929	192.168.1.163	192.168.1.168	TCP	60 3389 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
569	278.124184787	192.168.1.163	192.168.1.168	TCP	60 110 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
570	278.124185638	192.168.1.163	192.168.1.168	TCP	60 256 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
574	278.1279902702	192.168.1.163	192.168.1.168	TCP	60 993 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
575	278.127990393	192.168.1.163	192.168.1.168	TCP	60 8080 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
576	278.128634020	192.168.1.163	192.168.1.168	TCP	60 135 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
579	278.131199913	192.168.1.163	192.168.1.168	TCP	60 1720 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
580	278.131200064	192.168.1.163	192.168.1.168	TCP	60 37297 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
581	278.131200794	192.168.1.163	192.168.1.168	TCP	60 21917 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
585	278.134863163	192.168.1.163	192.168.1.168	TCP	60 52672 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
586	278.134863874	192.168.1.163	192.168.1.168	TCP	60 15455 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
587	278.135648293	192.168.1.163	192.168.1.168	TCP	60 46135 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
589	278.143247975	192.168.1.163	192.168.1.168	TCP	60 64717 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
593	278.147142124	192.168.1.163	192.168.1.168	TCP	60 36776 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
594	278.147142726	192.168.1.163	192.168.1.168	TCP	60 44296 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
595	278.147890705	192.168.1.163	192.168.1.168	TCP	60 46642 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
600	278.158414634	192.168.1.163	192.168.1.168	TCP	60 39913 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
601	278.159187391	192.168.1.163	192.168.1.168	TCP	60 11223 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
602	278.160898461	192.168.1.163	192.168.1.168	TCP	60 24797 → 38840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

```

Frame 523: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on ...
Ethernet II, Src: PCSSysteme_2b:7f:d6 (08:00:27:2b:7f:d6), Dst: PCSSy ...
Internet Protocol Version 4, Src: 192.168.1.163, Dst: 192.168.1.168 ...
Transmission Control Protocol, Src Port: 139, Dst Port: 38840, Seq: 1, ...

```

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 “tcp.flags.syn==1” means to give the packet where there was a connection to the three way handshake.

*eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src == 192.168.1.163 and tcp.flags.syn == 1 and tcp.flags.ack == 1

No. Time Source Destination Protocol Length Info

548	278.112851271	192.168.1.163	192.168.1.168	TCP	60 22 → 38840 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
-----	---------------	---------------	---------------	-----	---

```
(nifise㉿nifise) [~]
└─$ sudo hping3 -c 100 -S -p 80 192.168.1.163
20 10.944961857 192.168.1.169 90.155.12.1
21 10.944961857 192.168.1.169 192.168.1.1
22 10.944961857 192.168.1.169 192.168.1.1
23 21.146536997 192.168.1.1 192.168.1.1
24 21.147093554 192.168.1.1 192.168.1.1
25 21.148646396 192.168.1.163 192.168.1.1
26 21.154245690 192.168.1.163 192.168.1.1
27 21.155131459 192.168.1.1 192.168.1.1
28 21.258713711 192.168.1.1 192.168.1.1
29 21.300657627 192.168.1.163 192.168.1.1
30 21.481866011 192.168.1.169 217.154.60
31 21.500791156 217.154.60.177 192.168.1.1
```

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”.

tcp.flags.syn and tcp.flags.ack						
No.	Time	Source	Destination	Protocol	Length	Info
7	5.235520522	192.168.1.168	192.168.1.163	TCP	54	2978 → 80 [SYN] Seq=0 Win=512 Len=0
8	5.237224966	192.168.1.163	192.168.1.168	TCP	60	80 → 2978 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
9	6.236566629	192.168.1.168	192.168.1.163	TCP	54	2979 → 80 [SYN] Seq=0 Win=512 Len=0
10	6.238477567	192.168.1.163	192.168.1.168	TCP	60	80 → 2979 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
11	7.237924687	192.168.1.168	192.168.1.163	TCP	54	2980 → 80 [SYN] Seq=0 Win=512 Len=0
12	7.240935618	192.168.1.163	192.168.1.168	TCP	60	80 → 2980 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
13	8.238833559	192.168.1.168	192.168.1.163	TCP	54	2981 → 80 [SYN] Seq=0 Win=512 Len=0
14	8.240594193	192.168.1.163	192.168.1.168	TCP	60	80 → 2981 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
15	9.240163709	192.168.1.168	192.168.1.163	TCP	54	2982 → 80 [SYN] Seq=0 Win=512 Len=0
16	9.242770657	192.168.1.163	192.168.1.168	TCP	60	80 → 2982 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
17	10.241529036	192.168.1.168	192.168.1.163	TCP	54	2983 → 80 [SYN] Seq=0 Win=512 Len=0
18	10.243297282	192.168.1.163	192.168.1.168	TCP	60	80 → 2983 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
23	11.243009728	192.168.1.168	192.168.1.163	TCP	54	2984 → 80 [SYN] Seq=0 Win=512 Len=0
24	11.243949334	192.168.1.163	192.168.1.168	TCP	60	80 → 2984 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
25	12.244619630	192.168.1.168	192.168.1.163	TCP	54	2985 → 80 [SYN] Seq=0 Win=512 Len=0
26	12.246060865	192.168.1.163	192.168.1.168	TCP	60	80 → 2985 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
27	13.246589710	192.168.1.168	192.168.1.163	TCP	54	2986 → 80 [SYN] Seq=0 Win=512 Len=0
28	13.2482090556	192.168.1.163	192.168.1.168	TCP	60	80 → 2986 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
29	14.247431499	192.168.1.168	192.168.1.163	TCP	54	2987 → 80 [SYN] Seq=0 Win=512 Len=0
30	14.249742554	192.168.1.163	192.168.1.168	TCP	60	80 → 2987 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
31	15.248646897	192.168.1.168	192.168.1.163	TCP	54	2988 → 80 [SYN] Seq=0 Win=512 Len=0
32	15.250563423	192.168.1.163	192.168.1.168	TCP	60	80 → 2988 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
34	16.250286634	192.168.1.168	192.168.1.163	TCP	54	2989 → 80 [SYN] Seq=0 Win=512 Len=0
35	16.252583154	192.168.1.163	192.168.1.168	TCP	60	80 → 2989 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
37	17.2539098156	192.168.1.168	192.168.1.163	TCP	54	2990 → 80 [SYN] Seq=0 Win=512 Len=0
38	17.255541681	192.168.1.163	192.168.1.168	TCP	60	80 → 2990 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
40	18.263332198	192.168.1.168	192.168.1.163	TCP	54	2991 → 80 [SYN] Seq=0 Win=512 Len=0
41	18.265388232	192.168.1.163	192.168.1.168	TCP	60	80 → 2991 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
42	19.267436773	192.168.1.168	192.168.1.163	TCP	54	2992 → 80 [SYN] Seq=0 Win=512 Len=0
43	19.269232197	192.168.1.163	192.168.1.168	TCP	60	80 → 2992 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
44	20.270534897	192.168.1.168	192.168.1.163	TCP	54	2993 → 80 [SYN] Seq=0 Win=512 Len=0
45	20.271910860	192.168.1.163	192.168.1.168	TCP	60	80 → 2993 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
46	21.271261243	192.168.1.168	192.168.1.163	TCP	54	2994 → 80 [SYN] Seq=0 Win=512 Len=0
47	21.273849834	192.168.1.163	192.168.1.168	TCP	60	80 → 2994 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
48	22.272674630	192.168.1.168	192.168.1.163	TCP	54	2995 → 80 [SYN] Seq=0 Win=512 Len=0
49	22.274141250	192.168.1.163	192.168.1.168	TCP	60	80 → 2995 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
50	23.274462743	192.168.1.168	192.168.1.163	TCP	54	2996 → 80 [SYN] Seq=0 Win=512 Len=0
51	23.275605258	192.168.1.163	192.168.1.168	TCP	60	80 → 2996 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
52	24.282456709	192.168.1.168	192.168.1.163	TCP	54	2997 → 80 [SYN] Seq=0 Win=512 Len=0
53	24.299379336	192.168.1.163	192.168.1.168	TCP	60	80 → 2997 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
54	25.310895862	192.168.1.168	192.168.1.163	TCP	54	2998 → 80 [SYN] Seq=0 Win=512 Len=0

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 as closed.

Phase 2: pfSense and Snort- Firewall & Policy Enforcement

The screenshot shows the pfSense Firewall Rules configuration. The top navigation bar has tabs for Floating, WAN, and LAN, with WAN selected. The main table lists firewall rules:

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
X 0/16.70 MiB	*	RFC 1918 networks	*	*	*	*	*	*	Block private networks	
X 0/7.80 MiB	*	Reserved Not assigned by IANA	*	*	*	*	*	*	Block bogon networks	
X 0/0 B	IPv4 TCP	Medusa_Ransomware	*	WAN address	80 (HTTP)	*	none		Blocks Lockbit ransomware IP	
X 0/0 B	IPv4 TCP	Lockbit_ransomware	*	WAN address	80 (HTTP)	*	none			
0/0 B	IPv4 ICMP	any	*	*	*	*	*	none	Reject ICMP protocol	

Below the table are buttons for Add, Delete, Toggle, Copy, Save, and Separator.

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.

The screenshot shows the pfSense Alias configuration for 'Lockbit_ransomware'. The top navigation bar has tabs for Firewall / Aliases / Edit, with Edit selected. The main form has fields for Name, Description, and Type:

Name	Lockbit_ransomware
The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and _".	
Description	Blocks Lockbit ransomware IP
A description may be entered here for administrative reference (not parsed).	
Type	Host(s)

The 'Host(s)' section contains a table of blocked IP addresses:

IP or FQDN	Entry added	Delete
3.64.4.198	Mon, 13 Oct 2025 15:26:20 +0000	
3.67.161.133	Mon, 13 Oct 2025 15:26:20 +0000	
185.215.113.66	Mon, 13 Oct 2025 15:33:06 +0000	
13.107.4.52	Mon, 13 Oct 2025 15:33:06 +0000	
92.51.2.22	Mon, 13 Oct 2025 15:33:06 +0000	

At the bottom are buttons for Save, Export to file, and Add Host.

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_Ransomware	Host(s)	18.158.58.205, 18.197.239.109	Blocks ip from medus ransomware						
Phobos_ransomware	Host(s)	194.165.16.4, 45.9.74.14, 147.78.47.224, 185.202.0.111	Blocks ip from phobos ransomware						

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_Ransomware	*	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	<u>any</u>	*	*	*	*	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.

The screenshot shows the pfBlockerNG web interface under the 'IP' tab. The 'GeoIP' sub-tab is selected. A table titled 'GeoIP Summary' lists various geographical regions with their corresponding actions and logging settings. The table has columns for Name, Description, Action, and Logging. Most entries have 'Deny Inbound' as the action and 'Enabled' for logging, except for Europe which is set to 'Disabled'. A 'Save' button is located at the bottom right of the table.

Name	Description	Action	Logging
Top Spammers	GeolP Top Spammers	Deny Inbound	Enabled
Africa	GeolP Africa	Deny Inbound	Enabled
Antarctica	GeolP Antarctica	Deny Inbound	Enabled
Asia	GeolP Asia	Deny Inbound	Enabled
Europe	GeolP Europe	Disabled	Enabled
North America	GeolP North America	Deny Inbound	Enabled
Oceania	GeolP Oceania	Deny Inbound	Enabled
South America	GeolP South America	Deny Inbound	Enabled
Proxy and Satellite	GeolP Proxy and...	Disabled	Enabled

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.

```

└─(nifise㉿nifise)-[~]
$ ping -c4 192.168.1.163
PING 192.168.1.163 (192.168.1.163) 56(84) bytes of data.
From 192.168.1.168 icmp_seq=1 Destination Host Unreachable
From 192.168.1.168 icmp_seq=2 Destination Host Unreachable
From 192.168.1.168 icmp_seq=3 Destination Host Unreachable
From 192.168.1.168 icmp_seq=4 Destination Host Unreachable

--- 192.168.1.163 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3090ms
pipe 3

└─(nifise㉿nifise)-[~]
$ ping -c4 192.168.1.163
PING 192.168.1.163 (192.168.1.163) 56(84) bytes of data.
From 192.168.1.168 icmp_seq=1 Destination Host Unreachable
From 192.168.1.168 icmp_seq=2 Destination Host Unreachable
From 192.168.1.168 icmp_seq=3 Destination Host Unreachable
From 192.168.1.168 icmp_seq=4 Destination Host Unreachable

--- 192.168.1.163 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3070ms
pipe 3

└─(nifise㉿nifise)-[~]
$ 

```

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.

The screenshot shows the pfSense Snort configuration interface. The top navigation bar includes links for Services, Snort, and Interfaces. Below the navigation is a toolbar with buttons for Global Settings, Updates, Alerts, Blocked, Pass Lists, Suppress, IP Lists, SID Mgmt, Log Mgmt, and Sync. A red underline highlights the "Snort Interfaces" tab. The main content area is titled "Interface Settings Overview". It displays a table with one row for the WAN interface (em0). The columns are: Interface, Snort Status, Pattern Match, Blocking Mode, Description, and Actions. The "Interface" column shows "WAN (em0)". The "Snort Status" column shows a green checkmark and two blue circular icons. The "Pattern Match" column shows "AC-BNFA". The "Blocking Mode" column shows "DISABLED". The "Description" column shows "WAN". The "Actions" column contains three icons: a pencil, a copy, and a delete. At the bottom right of the table are "Add" and "Delete" buttons.

Interface	Snort Status	Pattern Match	Blocking Mode	Description	Actions
WAN (em0)	✓ ↻ ⟳	AC-BNFA	DISABLED	WAN	-pencil -copy -delete

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
        Main PID: 1134 (wazuh-agent)
           CPU: 0.000 CPU(s) total
          Tasks: 1 (since Mon 2025-12-22 15:49:57 UTC)
         Memory: 1.0M
            CPU: 0.000 CPU(s) total
          Tasks: 1 (since Mon 2025-12-22 15:49:57 UTC)
```

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/>

✓	syslog-ng	sysutils	1.16.2	Syslog-ng syslog server. This service is not intended to replace the default pfSense syslog server but rather acts as an independent syslog server.		
Package Dependencies:						
syslog-ng-4.8.1_3 logrotate-3.13.0_2						

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

<u>Default Log Directory</u>	/var/syslog-ng	Enter default log directory (no trailing slash)
<u>Default Log File</u>	default.log	Enter default log file
<u>Archive Frequency</u>	Daily	Select the frequency to archive (rotate) log files
Compress Archives	<input checked="" type="checkbox"/> Select this option to compress archived log files	
Compress Type	Gzip	Select the type of compression for archived log files
Max Archives	30	Enter the number of max archived log files
Include SCL	<input type="checkbox"/> Include syslog-ng standard configuration library (SCL)	
 Save		

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 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 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 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

Remote Logging Options

Enable Remote Logging	<input checked="" type="checkbox"/> Send log messages to remote syslog server
Source Address	<input type="button" value="Default (any)"/>
This option will allow the logging daemon to bind to a single IP address, rather than all IP addresses. If a single IP is picked, remote syslog servers must all be of that IP type. To mix IPv4 and IPv6 remote syslog servers, bind to all interfaces.	
NOTE: If an IP address cannot be located on the chosen interface, the daemon will bind to all addresses.	
IP Protocol	<input type="button" value="IPv4"/>
This option is only used when a non-default address is chosen as the source above. This option only expresses a preference; if an IP address of the selected type is not found on the chosen interface, the other type will be tried.	
Remote log servers	<input type="text" value="192.168.1.164:514"/> <input type="text" value="IP[:port]"/> <input type="text" value="IP[:port]"/>
Remote Syslog Contents	<input checked="" type="checkbox"/> Everything <input type="checkbox"/> System Events <input type="checkbox"/> Firewall Events <input type="checkbox"/> DNS Events (Resolver/unbound, Forwarder/dnsmasq, filterdns) <input type="checkbox"/> DHCP Events (DHCP Daemon, DHCP Relay, DHCP Client) <input type="checkbox"/> PPP Events (PPPoE WAN Client, L2TP WAN Client, PPTP WAN Client) <input type="checkbox"/> General Authentication Events <input type="checkbox"/> Captive Portal Events <input type="checkbox"/> VPN Events (IPsec, OpenVPN, L2TP, PPPoE Server) <input type="checkbox"/> Gateway Monitor Events <input type="checkbox"/> Routing Daemon Events (RADVD, UPnP, RIP, OSPF, BGP) <input type="checkbox"/> Network Time Protocol Events (NTP Daemon, NTP Client) <input type="checkbox"/> Wireless Events (hostapd)
Syslog sends UDP datagrams to port 514 on the specified remote syslog server, unless another port is specified. Be sure to set <i>syslogd</i> on the remote server to accept syslog messages from pfSense.	

Configure Wazuh syslog input

Edit the `/var/ossec/etc/ossec.cfg` on the Wazuh Manager 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)); };
```

The screenshot shows the 'Advanced' tab of the Wazuh UI for managing Syslog-NG objects. The 'General Options' section includes fields for 'Object Name' (set to 'DST_WAZUH_SYSLOG'), 'Object Type' (set to 'Destination'), and 'Object Parameters' (containing the configuration code). The 'Description' field is empty. At the bottom is a blue 'Save' button.

General Options
Object Name: DST_WAZUH_SYSLOG Enter the object name
Object Type: Destination Select the object type
Object Parameters: { network("192.168.1.164" transport(udp) localip(192.168.1.1)); }; Enter the object parameters
Description: Enter the description for this item

Save

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); };
```

General Advanced Log Viewer

General Options

<u>Object Name</u>	DST_WAZUH_SYSLOG
Enter the object name	
<u>Object Type</u>	Log
Select the object type	
<u>Object Parameters</u>	{ source(_DEFAULT); destination(DST_WAZUH_SYSLOG); };
Enter the object parameters	
<u>Description</u>	
Enter the description for this item	
<input type="button" value="Save"/>	

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-nginx	Syslog-nginx 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.

Login Protection

Threshold	<input type="text" value="10"/>	Block attackers when their cumulative attack score exceeds threshold. Most attacks have a score of 10.
Blocktime	<input type="text" value="20"/>	Block attackers for initially blocktime seconds after exceeding threshold. Subsequent blocks increase by a factor of 1.5. Attacks are unblocked at random intervals, so actual block times will be longer.
Detection time	<input type="text" value="1800"/>	Remember potential attackers for up to detection_time seconds before resetting their score.
Pass list	<input type="text" value="Address"/> / <input type="text" value="128"/>	Addresses added to the pass list will bypass login protection.
Add address	+ Add address	

That should do it.

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

```
Message from syslogd ...
<32>1 2025-12-22T20:29:04.176562+00:00 pfSense.home.arpa php-fpm 434 -- /index.php: webConfigurator authentication error for user 'fuck' from: 192.168.1.168

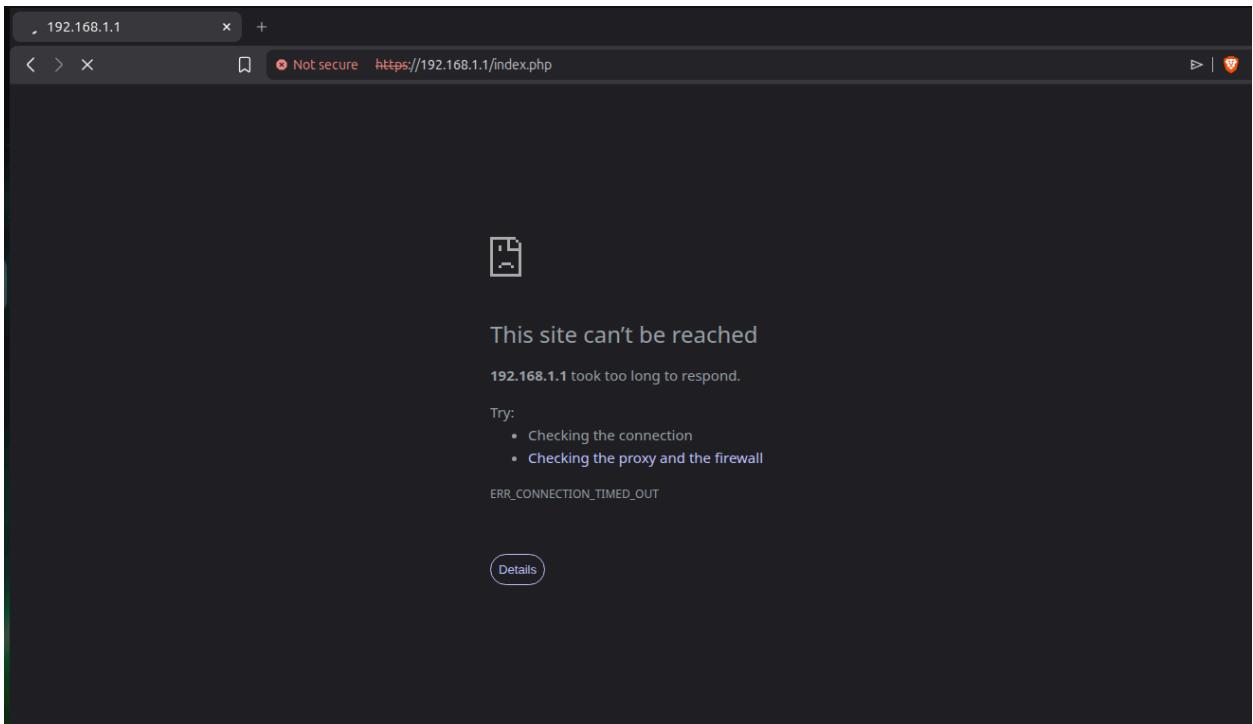
Message from syslogd@pfSense at Dec 22 21:27:06 ...
php-fpm[2782]: /index.php: webConfigurator authentication error for user 'admin' from: 192.168.1.168

Message from syslogd@pfSense at Dec 22 21:27:32 ...
php-fpm[2782]: /index.php: webConfigurator authentication error for user 'hello' from: 192.168.1.168
```

As well as wazuh:

```
21:27:07.200325 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG auth.emergency, length: 1
20
21:27:07.219632 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG auth.notice, length: 107
21:27:07.219633 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG auth.info, length: 127
```

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



Other pfSense logs being sent to wazuh:

```
[root@wazuh-server wazuh-user]# tcpdump -n udp port 514
dropped privs to tcpdump
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
21:24:06.786090 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG kernel.info, length: 36
21:24:06.786091 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG kernel.info, length: 67
21:24:06.810669 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG auth.info, length: 60
21:24:07.212194 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG local5.error, length: 165
21:24:07.400286 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG local5.info, length: 279
21:24:32.623075 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG local0.info, length: 137
21:24:36.992035 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG daemon.info, length: 50
21:25:00.301320 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG cron.info, length: 143
21:25:45.844415 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG local0.info, length: 199
21:25:46.846509 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG local0.info, length: 199
21:25:47.846115 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG local0.info, length: 199
21:25:48.846376 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG local0.info, length: 199
21:25:49.865757 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG local0.info, length: 199
21:25:50.865932 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG local0.info, length: 199
21:25:52.867622 IP 192.168.1.1.syslog > 192.168.1.164.syslog: SYSLOG local0.info, length: 199
```

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
```

```
nifise@nifise:~$ sudo hydra -t 4 -l nifise -P my_passwords.txt 192.168.1.163 ssh
[sudo] password for nifise:
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in milita
ry or secret service organizations, or for illegal purposes (this is non-binding, th
ese *** ignore laws and ethics anyway).
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-12-22 23:42:54
[DATA] max 4 tasks per 1 server, overall 4 tasks, 11 login tries (l:1/p:11), ~3 tri
es per task
[DATA] attacking ssh://192.168.1.163:22/14 → 443 [ACK] Seq=1930 Ack=140325 W
[22][ssh] host: 192.168.1.163    login:nifise 4 password: nifise
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-12-22 23:43:03
```

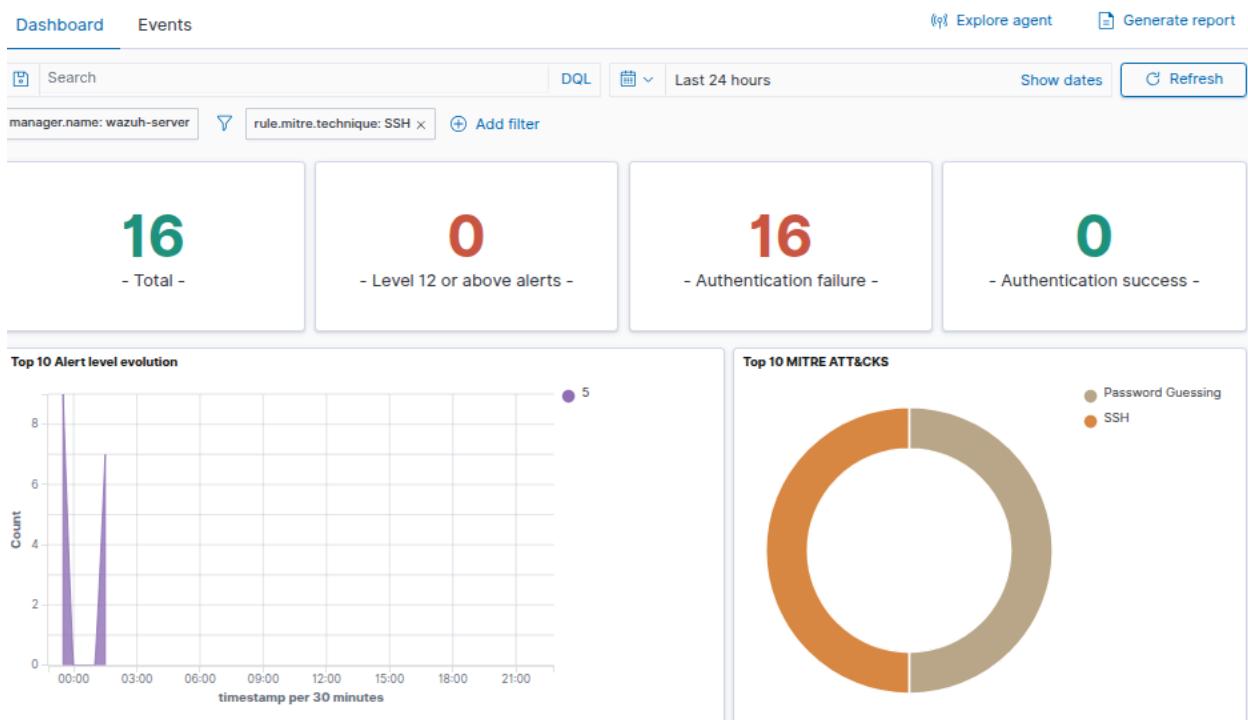
Wireshark Network Traffic Analysis

Filter: ip.addr==192.168.1.168 and ip.addr==192.168.1.163 and ssh

No.	Length	Time	Protocol	Source	Destination	Info
89	405.2044...	SSHv2	192.168.1.168	192.168.1.163		Client: Protocol (SSH-2.0-libssh_0.11.3)
109	405.2399...	SSHv2	192.168.1.163	192.168.1.168		Server: Protocol (SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.14)
1186	405.2483...	SSHv2	192.168.1.163	192.168.1.168		Server: Key Exchange Init
1042	405.2517...	SSHv2	192.168.1.168	192.168.1.163		Client: Key Exchange Init
114	405.2936...	SSHv2	192.168.1.168	192.168.1.168		Client: Elliptic Curve Diffie-Hellman Key Exchange Init
558	405.3046...	SSHv2	192.168.1.163	192.168.1.168		Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys, E
82	405.3094...	SSHv2	192.168.1.168	192.168.1.163		Client: New Keys
110	405.3517...	SSHv2	192.168.1.168	192.168.1.163		Client: Encrypted packet (len=44)
110	405.3527...	SSHv2	192.168.1.163	192.168.1.168		Server: Encrypted packet (len=44)
134	405.3530...	SSHv2	192.168.1.168	192.168.1.163		Client: Encrypted packet (len=68)
118	405.3573...	SSHv2	192.168.1.163	192.168.1.168		Server: Encrypted packet (len=52)
118	405.3576...	SSHv2	192.168.1.168	192.168.1.163		Client: Encrypted packet (len=52)
89	405.5799...	SSHv2	192.168.1.168	192.168.1.163		Client: Protocol (SSH-2.0-libssh_0.11.3)
89	405.5800...	SSHv2	192.168.1.168	192.168.1.163		Client: Protocol (SSH-2.0-libssh_0.11.3)
89	405.5808...	SSHv2	192.168.1.168	192.168.1.163		Client: Protocol (SSH-2.0-libssh_0.11.3)
89	405.5817...	SSHv2	192.168.1.168	192.168.1.163		Client: Protocol (SSH-2.0-libssh_0.11.3)
109	405.6068...	SSHv2	192.168.1.163	192.168.1.168		Server: Protocol (SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.14)
1042	405.6075...	SSHv2	192.168.1.168	192.168.1.163		Client: Key Exchange Init
109	405.6099...	SSHv2	192.168.1.163	192.168.1.168		Server: Protocol (SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.14)
1042	405.6105...	SSHv2	192.168.1.168	192.168.1.163		Client: Key Exchange Init
109	405.6119...	SSHv2	192.168.1.163	192.168.1.168		Server: Protocol (SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.14)
1042	405.6123...	SSHv2	192.168.1.168	192.168.1.163		Client: Key Exchange Init
109	405.6130...	SSHv2	192.168.1.163	192.168.1.168		Server: Protocol (SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.14)
1042	405.6133...	SSHv2	192.168.1.168	192.168.1.163		Client: Key Exchange Init
1186	405.6141...	SSHv2	192.168.1.163	192.168.1.168		Server: Key Exchange Init

Frame 6352: 89 bytes on wire (712 bits), 89 bytes captured (712 bits) on interface eth0, duration 0.000000 seconds, capture rate 0.000000 BPS on wire

0000 08 00 27 59 04 0f 08 00 27 06 30 | Packets: 10197 - Displayed: 72 (0.7%) | Profile: Default



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

nifise@nifise:~\$ ping 192.168.1.168
PING 192.168.1.168 (192.168.1.168) 56(84) bytes of data.
^C 0
--- 192.168.1.168 ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9237ms

Top 5 agents Alerts evolution - Top 5 agents

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.

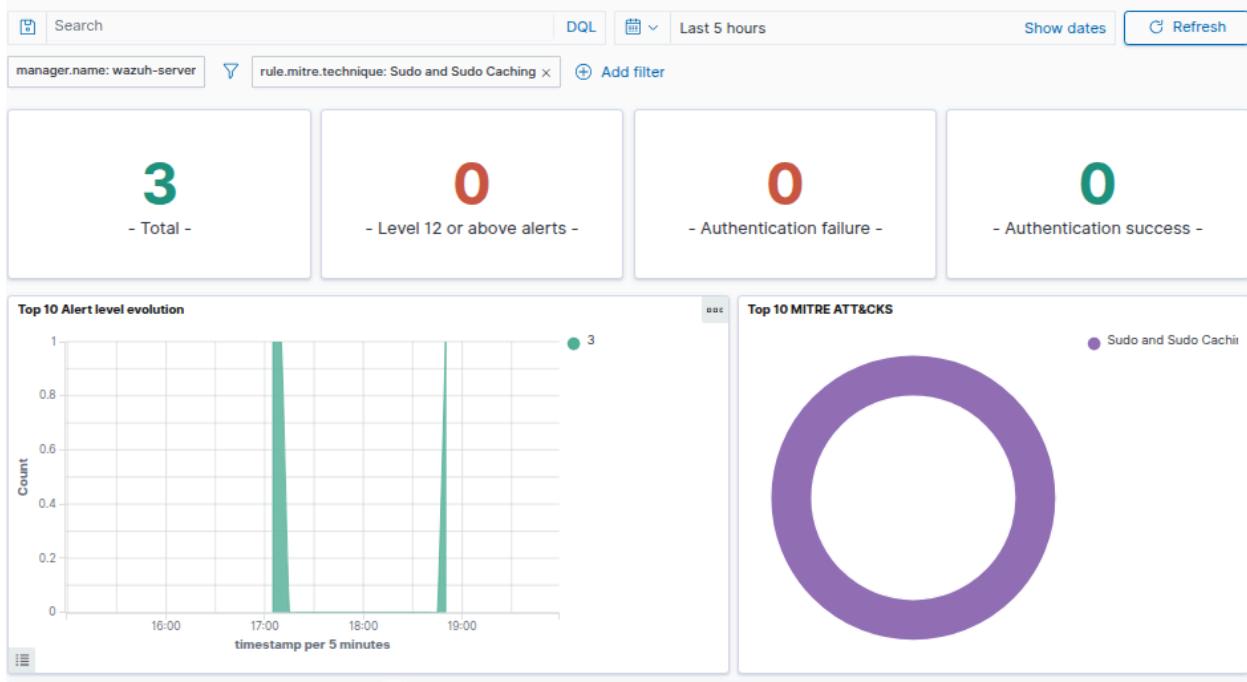
Privilege Escalation

```
nifise@nifise:~$ sudo -i  
[sudo] password for nifise:  
root@nifise:~#
```

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:

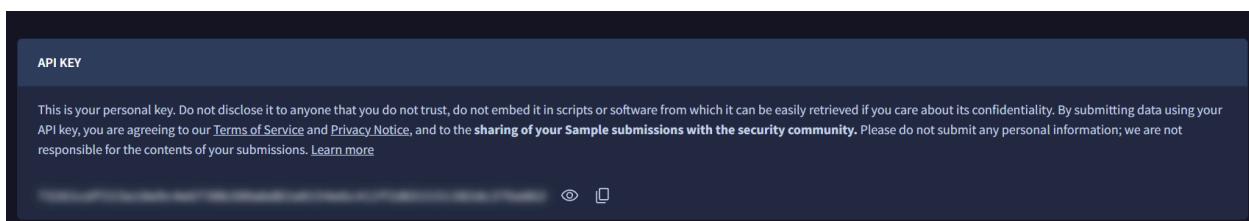
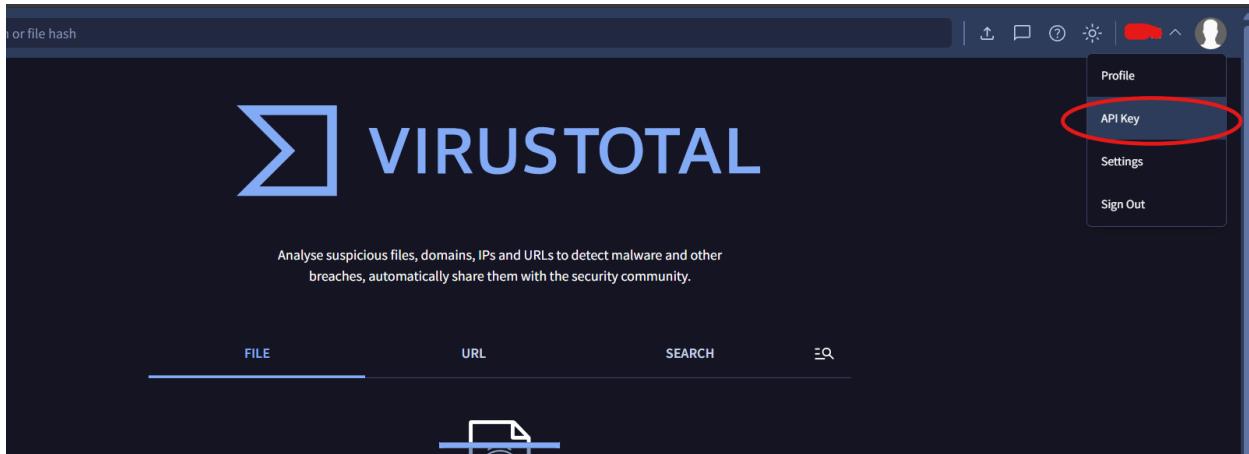
I checked if the File Integrity Monitoring is enabled on the wazuh agent by going to the `/var/ossec/etc/ossec.conf` and searching for `<syscheck>` block. `<disabled>` has to be set to `no`.

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

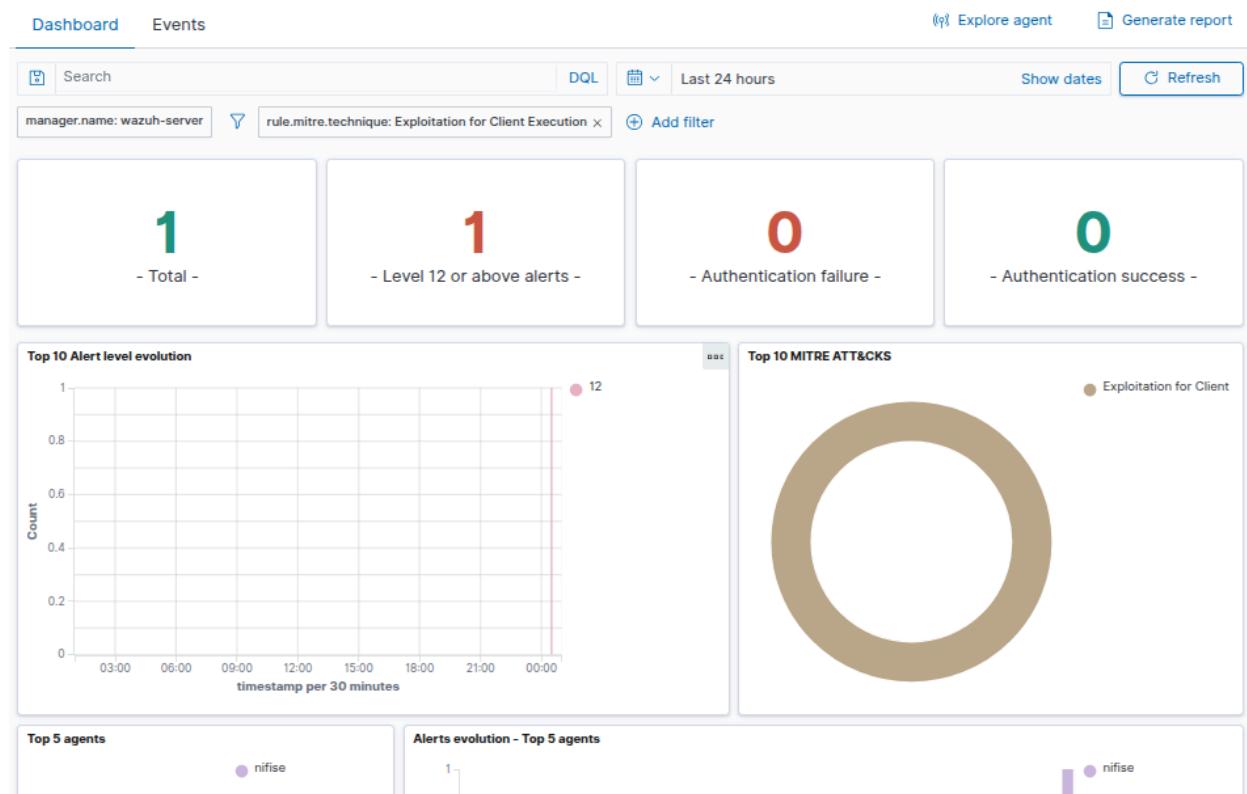
Now, we can download a malicious file on the endpoint in the monitored folder.

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 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:

1 hit					
Dec 24, 2025 @ 00:56:00.000 - Dec 24, 2025 @ 00:56:30.000					
Export Formatted	Reset view	653 available fields	Columns	Density	1 fields sorted
timestamp	agent.name	rule.description	rule.level	rule.id	
Dec 24, 2025 @ 00:56:26.123	nifise	VirusTotal: Alert - /home/nifise/Downloads/suspicious-file.exe - 6...	12	87105	

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.