



Network Infrastructure and Security

Project Report

A Multi-Zone Network with an Integrated Security Operations Center

Author:

Hadir Ben Arbia

Academic Supervisor:

Firas Bouallegue

August 12, 2025

A comprehensive lab environment built in GNS3 with VMware, FortiGate, Windows Server, Ubuntu, Suricata, Wazuh, and Shuffle SOAR

Contents

1	Introduction to Prologic	5
1.1	History of Prologic	5
1.2	Activities of Prologic Tunisia	5
1.3	Prologic's Partners	5
1.4	Service Domains	5
2	Introduction	6
2.1	Technology Selection and Justification	6
3	Network Topology Design	7
3.1	Components	7
3.2	Network Zones	8
4	FortiGate Firewall Configuration	8
4.1	Interface and DHCP Configuration	8
4.2	Security Policies and Intra-Zone Hardening	9
5	Active Directory Environment Setup	10
5.1	Domain Controller (DC) Setup on Windows Server 2022	10
5.1.1	Initial Server Preparation	10
5.1.2	Role Installation and Promotion	11
5.2	Ubuntu Client Integration into the Domain	11
6	Inline Intrusion Prevention System (Suricata)	11
6.1	Inline Placement and Bridge Configuration	11
6.2	Kernel and iptables Configuration	12
6.3	Suricata Configuration and Testing	12
7	Security Operations Center (SOC) and DMZ Implementation	12
7.1	DMZ Web Server Deployment	12
7.2	Wazuh SIEM Deployment	13
7.3	Shuffle SOAR Deployment	13
7.4	Use Case: Automated Attacker Blocking	13
8	Functional Validation: Attack Simulation	14
8.1	Test Scenario	14
8.2	Attack Steps	14
8.2.1	Attack Preparation	14
8.2.2	Executing the Hydra Command	14
8.3	Results and Validation of the Automated Response Chain	15
9	Network Access Control with PacketFence	16
9.1	Deployment and Configuration	16
9.2	Role in the Security Architecture	17

10 Future Perspective and Enhancements	17
10.1 Full Zero Trust Network Access (ZTNA) Implementation	17
10.2 Advanced SOAR Playbooks and Threat Intelligence	18
10.3 Hybrid Cloud Integration	18
10.4 Advanced Threat Simulation and Purple Teaming	19
11 Conclusion	19

List of Figures

1	PROLOGIC's Global Technology Partners.	5
2	Overview of Service Domains Offered by PROLOGIC.	6
3	GNS3 Network Topology	8
4	suricata testing.	12
5	Hydra attack	15
6	wazuh detection.	15
7	webhook data	16

Listings

1	FortiGate LAN Interface and DHCP Server Configuration	8
2	Final Consolidated Firewall Policy Set	10
3	Commands to join the Active Directory domain	11
4	Key nmcli commands for bridge creation	11
5	Enabling IP Forwarding	12
6	iptables Rule for NFQUEUE	12
7	Wazuh Webhook Configuration in ossec.conf	13
8	Brute-force attack command using Hydra	14

1 Introduction to Prologic

1.1 History of Prologic

PROLOGIC Tunisia is recognized today as one of the undisputed leaders in the IT equipment and services market, drawing on 30 years of extensive experience in sales, consulting, services, and enterprise software. Founded in 1985 by experts with a passion for new technologies, PROLOGIC has grown to become a key player in the industry. [1]

1.2 Activities of Prologic Tunisia

The company's operations are organized into three complementary business divisions:

- **SMART:** A wholesale distributor of computer and telephony equipment.
- **PROLOGIC:** The core division focused on sales, system integration, consulting, and professional services.
- **SIMOP:** A dedicated division for providing maintenance and after-sales support services.

1.3 Prologic's Partners

To deliver services that consistently meet client demands, PROLOGIC Tunisia has forged strategic partnerships with leading global companies, as shown in Figure 1.



Figure 1: PROLOGIC's Global Technology Partners.

1.4 Service Domains

Leveraging its beneficial partnerships, PROLOGIC provides a wide range of services, detailed in Figure 2, which include:

- Software solutions tailored to respect each client's budget.
- Professional expertise in implementing solutions from major technology brands.
- A comprehensive selection of IT equipment and services.
- A commitment to delivering superior quality of service.



Figure 2: Overview of Service Domains Offered by PROLOGIC.

2 Introduction

This report details the design, implementation, and configuration of a comprehensive network security lab. The primary objective was to build a realistic, multi-zone network environment using industry-standard tools and practices to create a functional cyber range. The project encompasses firewall configuration, enterprise identity management with Active Directory, Linux client integration, the deployment of an inline Intrusion Prevention System (IPS), and a fully functional Security Operations Center (SOC) capable of automated incident response. The entire topology was virtualized using a combination of GNS3 and VMware Workstation to leverage the strengths of both platforms.

2.1 Technology Selection and Justification

The selection of technologies for this project was deliberate, aiming to mirror a modern, yet practical, enterprise environment while leveraging powerful open-source tools. Each component was chosen for its specific capabilities and relevance in the industry.

- **GNS3 and VMware Workstation:** This hybrid approach was chosen to get the best of both worlds. GNS3 excels at emulating complex network topologies and running specialized network operating systems like FortiGate's FortiOS. VMware Workstation provides superior performance and stability for running full server and desktop operating systems like Windows Server and Ubuntu. Using them together allows for the creation of a high-fidelity lab that would be difficult to achieve with either tool alone.
- **FortiGate Firewall:** As a leading commercial Next-Generation Firewall (NGFW), FortiGate was selected to provide experience with an industry-standard enterprise security appliance. Unlike open-source alternatives like pfSense, FortiGate offers a

comprehensive Unified Threat Management (UTM) feature set and a robust API, which was critical for the automated response integration with the SOAR platform.

- **Windows Server Active Directory:** AD is the undisputed industry standard for identity and access management in corporate networks. Its inclusion was non-negotiable for simulating a realistic enterprise environment. Alternatives like Samba or OpenLDAP, while capable, do not reflect the infrastructure found in the vast majority of businesses today.
- **Suricata IPS:** For the inline threat prevention layer, the open-source Suricata was chosen over alternatives like Snort. Suricata is renowned for its high performance, native multi-threading capabilities, and modern architecture. Its ability to run in inline ‘NFQUEUE’ mode allows it to actively block threats in real-time, moving beyond passive detection to active prevention.
- **Wazuh (SIEM):** While the Elastic Stack (ELK) is a powerful platform for log management, Wazuh was selected because it is a security-focused solution built upon it. It comes pre-packaged with security-specific detection rules, agent management, vulnerability scanning, and compliance modules out-of-the-box. This makes it a complete Security Information and Event Management (SIEM) and eXtended Detection and Response (XDR) platform, significantly accelerating the deployment of a functional SOC compared to building one from a general-purpose tool.
- **Shuffle (SOAR):** To implement security automation, the open-source Shuffle platform was chosen. Its primary advantage is its intuitive, visual workflow editor, which simplifies the creation of complex automation playbooks. Unlike more fragmented solutions that may require separate tools for incident management and automation, Shuffle provides a unified, vendor-agnostic platform focused on orchestrating actions across different security tools via their APIs, making it ideal for the project’s goal of automated remediation.

3 Network Topology Design

The foundation of the project is a segmented network topology designed to separate traffic based on security requirements.

3.1 Components

- **GNS3:** The core network emulation platform.
- **VMware Workstation:** Used to host the virtual machines.
- **FortiGate Firewall:** The central security appliance for routing, policies, and DHCP.
- **Windows Server 2022:** Deployed as the Active Directory Domain Controller (DC) and DNS server.
- **Ubuntu Desktop:** Deployed as a client workstation joined to the AD domain.
- **Suricata IPS:** Deployed on an Ubuntu VM as a transparent inline IPS.

- **DMZ Web Server:** An Ubuntu server hosting a public-facing service.
- **SOC Server:** An Ubuntu server hosting Wazuh (SIEM) and Shuffle (SOAR).

3.2 Network Zones

The topology is divided into three main zones, controlled by the FortiGate firewall:

1. **WAN (port1):** Connects to the internet via the GNS3 NAT cloud.
2. **LAN (port2):** The internal, trusted network hosting the Domain Controller and client workstations. IP Subnet: 192.168.10.0/24.
3. **DMZ (port3):** A demilitarized zone for public-facing services. IP Subnet: 192.168.30.0/24.

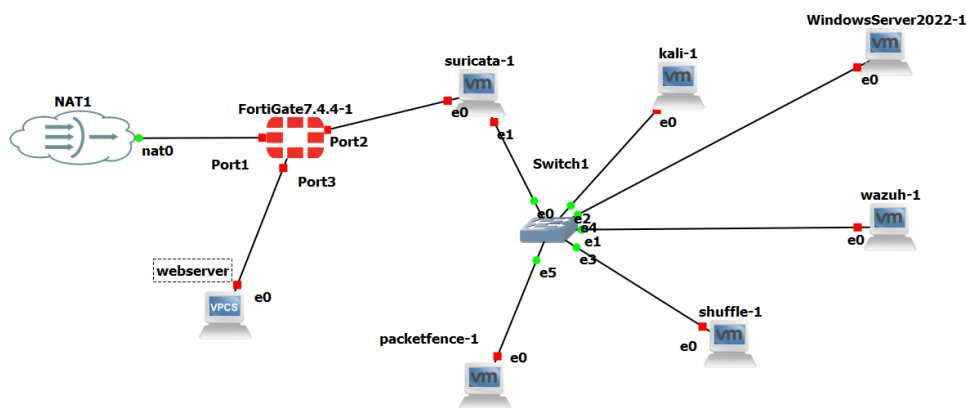


Figure 3: GNS3 Network Topology .

4 FortiGate Firewall Configuration

4.1 Interface and DHCP Configuration

The FortiGate interfaces were configured to define the network zones and provide DHCP services to the LAN and DMZ.

```

1
2 config system interface
3     edit "port1"
4         set vdom "root"
5         set mode dhcp
6         set allowaccess ping
7         set type physical
8         set alias "WAN"
9         set snmp-index 1
10    next
11    edit "port2"
12        set vdom "root"

```

```
13      set ip 192.168.10.1 255.255.255.0
14      set allowaccess ping https ssh http
15      set type physical
16      set alias "lan"
17      set snmp-index 2
18  next
19  edit "port3"
20      set vdom "root"
21      set ip 192.168.30.1 255.255.255.0
22      set allowaccess ping
23      set type physical
24      set alias "dmz"
25      set snmp-index 3
26  next
27
28 config system dhcp server
29  edit 20
30      set default-gateway 192.168.10.1
31      set netmask 255.255.255.0
32      set interface "port2"
33      config ip-range
34          edit 1
35              set start-ip 192.168.10.100
36              set end-ip 192.168.10.150
37          next
38      end
39      config reserved-address
40          edit 1
41              set ip 192.168.10.20
42              set mac 00:0c:29:92:5f:92
43              set description "domain-controller-dc01"
44          next
45          edit 2
46              set ip 192.168.10.40
47              set mac 00:0c:29:7c:98:27
48              set description "shuffle"
49          next
50          edit 3
51              set ip 192.168.10.50
52              set mac 00:0c:29:2f:ff:0f
53              set description "Ubuntu-wazuh-client"
54          next
55          edit 4
56              set ip 192.168.10.10
57              set mac 00:0c:29:8b:9e:25
58              set description "packetfence"
59          next
60      end
61      set dns-server1 8.8.8.8
62  next
63 end
```

Listing 1: FortiGate LAN Interface and DHCP Server Configuration

4.2 Security Policies and Intra-Zone Hardening

Due to the 3-policy limit of the trial license, a consolidated policy set was created.

```
1
2 config firewall policy
3     edit 1
4         set name "INTRA-LAN-ALLOW-AD-to-DC"
5         set uuid 6c05560c-5be8-51f0-3bf6-958c45d1f62f
6         set srcintf "port2"
7         set dstintf "port2"
8         set action accept
9         set srcaddr "all"
10        set dstaddr "ADMINSERVER" "Wazuh_Manager"
11        set schedule "always"
12        set service "activeDirectory-required" "PING" "SYSLOG" "Wazuh-
Agent-Comm" "Wazuh-Agent-Reg"
13        set logtraffic all
14        set comments "Allows AD traffic and PING to the DC"
15    next
16    edit 2
17        set name "LAN-to-WAN-Outbound"
18        set uuid c322dda6-5be8-51f0-44de-113cf37527ad
19        set srcintf "port2"
20        set dstintf "port1"
21        set action accept
22        set srcaddr "all"
23        set dstaddr "all"
24        set schedule "always"
25        set service "ALL"
26        set nat enable
27        set comments "Combined rule for LAN/DMZ to WAN/DMZ"
28    next
29    edit 3
30        set name "INTRA-LAN-DENY-ALL"
31        set uuid 01c2bebe-5be9-51f0-cba3-6daf73a25e06
32        set srcintf "port2"
33        set dstintf "port2"
34        set srcaddr "all"
35        set dstaddr "all"
36        set schedule "always"
37        set service "ALL"
38        set logtraffic all
39        set comments "Blocks all other traffic within the LAN"
40    next
41 end
```

Listing 2: Final Consolidated Firewall Policy Set

5 Active Directory Environment Setup

5.1 Domain Controller (DC) Setup on Windows Server 2022

The Windows Server 2022 VM was configured as the primary Domain Controller for the `project.local` domain.

5.1.1 Initial Server Preparation

1. **VMware Tools** were installed for performance optimization.

2. A static IP of 192.168.10.20 was set on the network adapter, with the Preferred DNS server set to 127.0.0.1 (itself).
3. The server was renamed to DC01.

5.1.2 Role Installation and Promotion

1. The *Active Directory Domain Services* role was installed via Server Manager.
2. The server was promoted to a Domain Controller. Key choices included:
 - Selected **"Add a new forest"**.
 - Set the root domain name to `project.local`.
 - Ensured **DNS Server** and **Global Catalog (GC)** options were enabled.
3. The server rebooted to complete the promotion.

5.2 Ubuntu Client Integration into the Domain

The Ubuntu Desktop client was successfully joined to the `project.local` domain.

```
1 # Install required packages
2 sudo apt update
3 sudo apt install realmd sssd sssd-tools adcli samba-common-bin
4
5 # Join the domain (prompts for Administrator password)
6 sudo realm join --user=Administrator project.local
7
8 # Permit all domain users to log in
9 sudo realm permit --all
```

Listing 3: Commands to join the Active Directory domain

6 Inline Intrusion Prevention System (Suricata)

To enhance security, a Suricata IPS was deployed inline between the LAN switch and the FortiGate firewall.

6.1 Inline Placement and Bridge Configuration

The Suricata VM was equipped with two network interfaces and configured as a transparent Layer 2 bridge. On Ubuntu Desktop, this was achieved using `nmcli` as Network-Manager is in control.

```
1 # Create bridge interface
2 sudo nmcli connection add type bridge con-name br0-conn ifname br0
3
4 # Configure bridge to have no IP address
5 sudo nmcli connection modify br0-conn ipv4.method "disabled"
6
7 # Add physical interfaces as slaves to the bridge
8 sudo nmcli connection add type bridge-slave con-name br0-slave-eth0
   ifname eth0 master br0
```

```
9 sudo nmcli connection add type bridge-slave con-name br0-slave-eth1
   ifname eth1 master br0
```

Listing 4: Key nmcli commands for bridge creation

6.2 Kernel and iptables Configuration

For the bridge to pass traffic, IP forwarding had to be enabled in the kernel.

```
1 # Enable in current session
2 sudo sysctl -w net.ipv4.ip_forward=1
3
4 # Make permanent by adding to /etc/sysctl.conf
5 net.ipv4.ip_forward=1
```

Listing 5: Enabling IP Forwarding

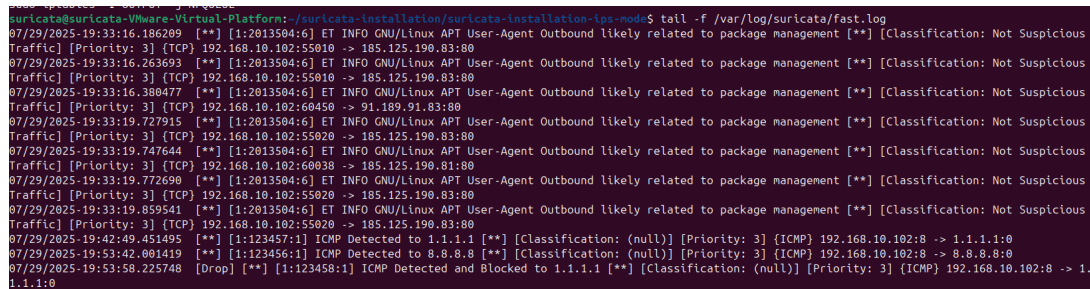
An iptables rule was added to send all forwarded traffic through Suricata for inspection.

```
1 sudo iptables -I FORWARD -i br0 -o br0 -j NFQUEUE --queue-num 0
```

Listing 6: iptables Rule for NFQUEUE

6.3 Suricata Configuration and Testing

The `/etc/suricata/suricata.yaml` file was configured to run in `nfqueue` mode. The IPS was started as a daemon and tested by generating a specific ICMP packet, which was successfully detected and dropped, confirming correct operation.



```
suricata@suricata-Virtual-Platform: /suricata-installation/suricata-installation-ips-mwds$ tail -f /var/log/suricata/fast.log
07/29/2025-19:33:16.186289  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound likely related to package management [**] [Classification: Not Suspicious
Traffic] [Priority: 3] [TCP] 192.168.10.102:55010 -> 185.125.190.83:80
07/29/2025-19:33:16.263693  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound likely related to package management [**] [Classification: Not Suspicious
Traffic] [Priority: 3] [TCP] 192.168.10.102:55010 -> 185.125.190.83:80
07/29/2025-19:33:16.380477  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound likely related to package management [**] [Classification: Not Suspicious
Traffic] [Priority: 3] [TCP] 192.168.10.102:60450 -> 91.189.91.83:80
07/29/2025-19:33:19.727915  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound likely related to package management [**] [Classification: Not Suspicious
Traffic] [Priority: 3] [TCP] 192.168.10.102:55020 -> 185.125.190.83:80
07/29/2025-19:33:19.747644  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound likely related to package management [**] [Classification: Not Suspicious
Traffic] [Priority: 3] [TCP] 192.168.10.102:60038 -> 185.125.190.81:80
07/29/2025-19:33:19.772690  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound likely related to package management [**] [Classification: Not Suspicious
Traffic] [Priority: 3] [TCP] 192.168.10.102:55020 -> 185.125.190.83:80
07/29/2025-19:33:19.859541  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound likely related to package management [**] [Classification: Not Suspicious
Traffic] [Priority: 3] [TCP] 192.168.10.102:55020 -> 185.125.190.83:80
07/29/2025-19:42:49.451495  [**] [1:123457:1] ICMP Detected to 1.1.1.1 [**] [Classification: (null)] [Priority: 3] [ICMP] 192.168.10.102:8 -> 1.1.1.1:0
07/29/2025-19:53:42.001419  [**] [1:123456:1] ICMP Detected to 8.8.8.8 [**] [Classification: (null)] [Priority: 3] [ICMP] 192.168.10.102:8 -> 8.8.8.8:0
07/29/2025-19:53:58.225748  [Drop] [**] [1:123458:1] ICMP Detected and Blocked to 1.1.1.1 [**] [Classification: (null)] [Priority: 3] [ICMP] 192.168.10.102:8 -> 1.1.1.1:0
```

Figure 4: suricata testing.

7 Security Operations Center (SOC) and DMZ Implementation

To create a complete security ecosystem, a DMZ was established for public services, and a SOC was deployed for centralized monitoring and automated response.

7.1 DMZ Web Server Deployment

An Ubuntu Server was deployed in the DMZ zone (192.168.30.0/24). An Apache2 web server was installed to simulate a public-facing application. The FortiGate was configured with a Virtual IP (VIP) and a firewall policy to allow external traffic from the WAN to reach this server on port 80, while blocking all direct access from the DMZ to the internal LAN.

7.2 Wazuh SIEM Deployment

Wazuh was installed on a dedicated Ubuntu server to function as the SIEM. It centralizes and analyzes logs from all critical assets in the infrastructure.

- **Installation:** The Wazuh manager was installed following the official documentation.
- **Agent Deployment:** Wazuh agents were installed on the Windows Domain Controller, the LAN Ubuntu client, and, crucially, on the **DMZ Web Server**. This provides deep visibility into host-level events (e.g., file changes, failed logins, command execution) across all security zones.

7.3 Shuffle SOAR Deployment

Shuffle, an open-source Security Orchestration, Automation, and Response (SOAR) platform, was installed on the same SOC server using Docker.

- **Installation:** Deployed via `docker-compose`.
- **Troubleshooting:** Initial setup required increasing the kernel parameter `vm.max_map_count` to 262144 for Elasticsearch to run correctly.
- **Integration:** An API key for the FortiGate was generated and stored securely in Shuffle's workflow variables to allow for automated actions.

7.4 Use Case: Automated Attacker Blocking

A fully automated incident response workflow was created to demonstrate the power of the integrated SOC.

1. **Detection:** An attacker attempts a web application attack against the DMZ server. The Wazuh agent on the server detects this malicious activity (e.g., multiple 404 errors, SQL injection attempts) and generates a high-severity alert.
2. **Triggering:** The Wazuh manager is configured with a webhook integration in its `ossec.conf` file. Upon receiving an alert matching the severity level (e.g., level 7 or higher), it automatically forwards the alert data in JSON format to a specific Shuffle workflow endpoint, as shown in the configuration below.

```
1 <integration>
2   <name>shuffle</name>
3   <hook_url>http://192.168.10.40:3001/api/v1/hooks/webhook_76ce9c05
4     -e042-4cc6-bedd-64513cf068e7</hook_url>
5   <level>7</level>
6   <alert_format>json</alert_format>
7 </integration>
```

Listing 7: Wazuh Webhook Configuration in `ossec.conf`

3. **Orchestration:** The Shuffle workflow is triggered by the incoming webhook. It parses the JSON alert to extract the attacker's source IP address.

4. **Automated Response:** Shuffle executes a pre-configured action that makes a REST API call to the FortiGate firewall. This API call creates a new address object for the attacker's IP and adds it to a "Blocklist" group, which is used in a high-priority firewall policy to deny all traffic from that source.

8 Functional Validation: Attack Simulation

To validate the effectiveness of the integrated security chain (SIEM/SOAR), a brute-force attack simulation was performed. This test aimed to demonstrate the system's ability to detect a threat, trigger an automated analysis, and apply a remediation measure without human intervention.

8.1 Test Scenario

The scenario simulates one of the most common attacks against internal infrastructure: a brute-force attempt against an authentication service.

- **Attacker:** The Shuffle SOAR virtual machine (192.168.10.40).
- **Attack Tool:** Hydra, a popular online password-cracking tool.
- **Target:** The SSH (Secure Shell) service on the Wazuh Manager virtual machine (192.168.10.50).
- **Attack Vector:** Attempting to guess the password for the `root` user using a short list of common passwords.

This scenario is particularly relevant because it allows the Wazuh server to act as both the victim and the detection system by analyzing its own authentication logs.

8.2 Attack Steps

The attack was initiated from the Shuffle VM's terminal.

8.2.1 Attack Preparation

1. The Hydra tool was installed on the attacker VM.

```
sudo apt install hydra
```

2. A simple password dictionary was created for the test.

```
echo -e "password\nPassword123\nadmin\nroot" > passwords.txt
```

8.2.2 Executing the Hydra Command

The attack was launched with the following command, generating multiple failed SSH login attempts.

Listing 8: Brute-force attack command using Hydra

```
hydra -l root -P passwords.txt 192.168.10.50 ssh -vV
```

```

shuffle@shuffle-VMware-Virtual-Platform:~$ hydra -l wazuh -P passwords.txt 192.168.10.50 ssh -vV
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-08-01 09:32:01
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 6 tasks per 1 server, overall 6 tasks, 6 login tries (l:1/p:6), ~1 try per task
[DATA] attacking ssh://192.168.10.50:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://wazuh@192.168.10.50:22
[INFO] Successful, password authentication is supported by ssh://192.168.10.50:22
[ATTEMPT] target 192.168.10.50 - login "wazuh" - pass "password" - 1 of 6 [child 0] (0/0)
[ATTEMPT] target 192.168.10.50 - login "wazuh" - pass "Password123" - 2 of 6 [child 1] (0/0)
[ATTEMPT] target 192.168.10.50 - login "wazuh" - pass "admin" - 3 of 6 [child 2] (0/0)
[ATTEMPT] target 192.168.10.50 - login "wazuh" - pass "123456" - 4 of 6 [child 3] (0/0)
[ATTEMPT] target 192.168.10.50 - login "wazuh" - pass "fortinet" - 5 of 6 [child 4] (0/0)
[ATTEMPT] target 192.168.10.50 - login "wazuh" - pass "wazuh1234" - 6 of 6 [child 5] (0/0)
[22][ssh] host: 192.168.10.50 login: wazuh password: wazuh1234
[STATUS] attack finished for 192.168.10.50 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-08-01 09:32:03
shuffle@shuffle-VMware-Virtual-Platform:~$

```

Figure 5: Hydra attack .

8.3 Results and Validation of the Automated Response Chain

The test was a comprehensive success, validating every link in the security chain.

1. Step 1: Detection by Wazuh

Seconds after the attack began, the Wazuh agent (analyzing itself) detected the failed login attempts. A high-level alert (Rule ID 5712: *sshd: multiple authentication failures*) was generated and appeared in the Wazuh security events dashboard, correctly identifying the attacker's source IP (192.168.10.40).

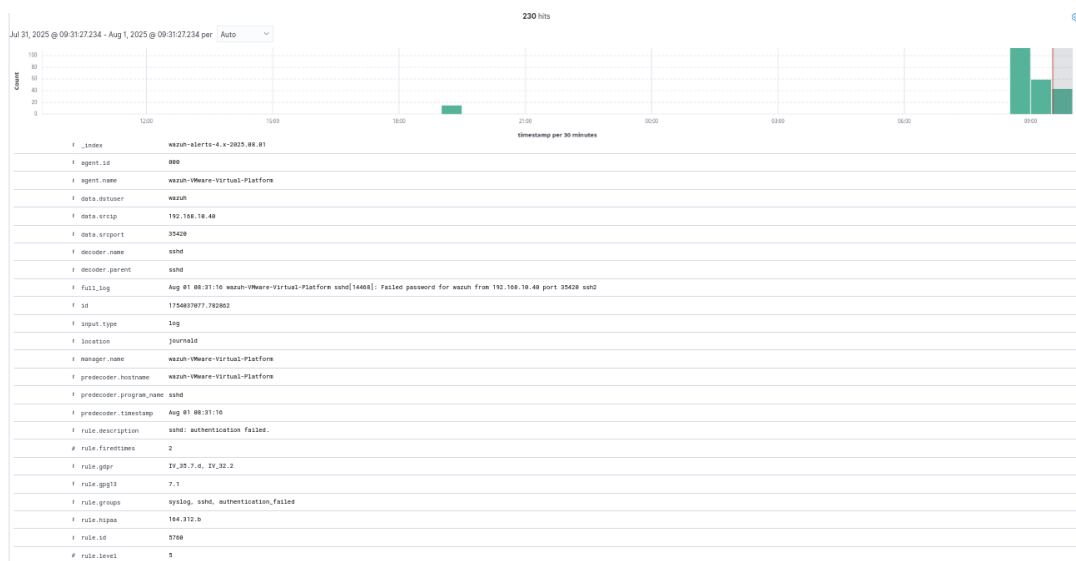


Figure 6: wazuh detection.

2. Step 2: Webhook Trigger to Shuffle

As configured in `ossec.conf`, the Wazuh alert automatically triggered a call to Shuffle's webhook, passing the full alert details in JSON format.

3. Step 3: Playbook Execution in Shuffle

In the Shuffle interface, a new execution of the "Wazuh Alert -> FortiGate Block" workflow was observed. The playbook correctly received the webhook data, extracted the source IP address (`rule.data.srcip`), and proceeded to the next step.

4. Step 4: Remediation by FortiGate

The playbook's final action was an API call to the FortiGate firewall. Verification in

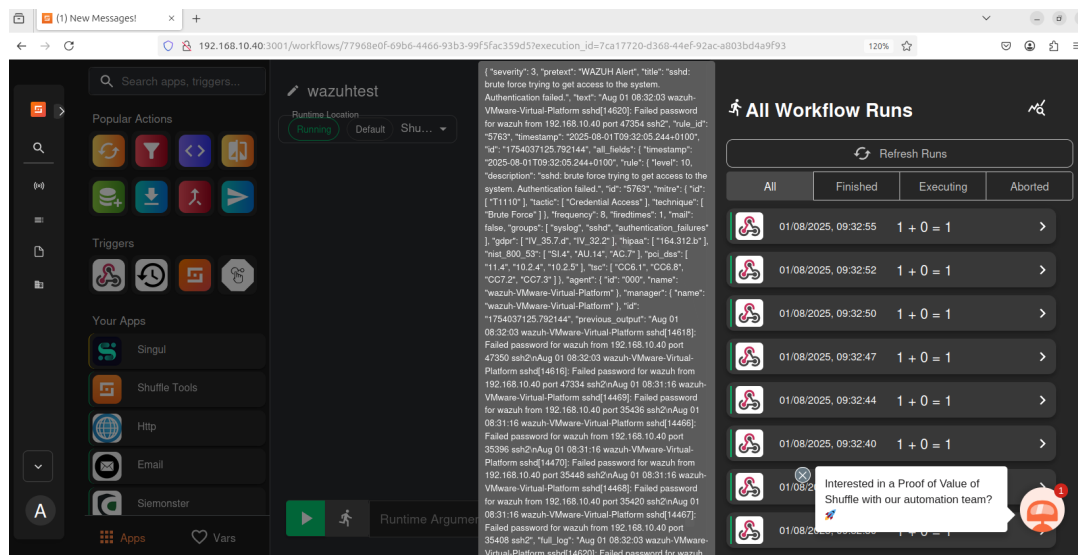


Figure 7: webhook data

the FortiGate UI under **Policy & Objects > Addresses** confirmed the automatic creation of a new Address object.

- **Object Name:** SOAR-Block-192.168.10.40
- **IP Address:** 192.168.10.40/32

This object, when added to a "deny" firewall policy, effectively blocks all future communication from the attacker.

This end-to-end test demonstrates that the implemented SOC architecture is capable of moving from simple passive detection to an **active, automated defense**, reducing the response time to a threat from minutes (or hours) to mere seconds.

9 Network Access Control with PacketFence

To implement a Zero Trust security model at the network edge, PacketFence, a powerful open-source Network Access Control (NAC) solution, was deployed. The primary objective of PacketFence in this architecture is to act as a gatekeeper for the internal LAN, ensuring that only registered and compliant devices are granted network access.

9.1 Deployment and Configuration

The PacketFence ZEN (Zero Effort NAC) virtual appliance was chosen for its streamlined installation process. It was deployed within the GNS3 topology on the LAN segment and configured with the following parameters:

- **Static IP Address:** 192.168.10.10/24
- **Gateway:** 192.168.10.1 (FortiGate LAN interface)
- **DNS Server:** 192.168.10.20 (Active Directory DC)

The initial configuration was completed through the PacketFence administrative interface. A key integration step was to harmonize PacketFence with the existing network services. To avoid conflicts with the FortiGate, which serves as the authoritative DHCP server for the LAN, the internal DHCP service within PacketFence was proactively disabled using the `systemctl` command. This ensures a clear separation of duties between the firewall and the NAC platform.

Following the configuration, all necessary PacketFence services—including the MariaDB database, Redis cache, the core application API, and web services—were successfully started and verified. The system is now fully operational and accessible for administration via its secure web interface.

9.2 Role in the Security Architecture

With PacketFence successfully integrated, it serves a critical function in the network security chain:

1. **Device Registration:** Any new, unknown device attempting to connect to the LAN will be isolated and redirected to a captive portal, where the user must register the device before gaining access.
2. **Compliance Enforcement:** PacketFence can be configured to run security scans on connecting devices to ensure they meet a baseline security posture (e.g., antivirus enabled, OS up-to-date).
3. **Future Integration with AD:** The platform is now ready for further integration with Active Directory to enable 802.1X authentication, providing robust, identity-based network access control for both wired and wireless connections.

The successful deployment of PacketFence adds a critical layer of internal security, moving beyond perimeter defense to control what happens inside the trusted network.

10 Future Perspective and Enhancements

The current architecture provides a robust foundation for network security and operations. However, the modular nature of the lab allows for significant future enhancements that would further align it with modern enterprise security challenges. The following points outline potential avenues for development.

10.1 Full Zero Trust Network Access (ZTNA) Implementation

The deployment of PacketFence is the first step towards a Zero Trust model. The next logical evolution would be to fully leverage its capabilities:

- **802.1X Authentication:** Integrate PacketFence with Active Directory as a RADIUS server. This would enforce identity-based network access control for both wired and (future) wireless clients, ensuring that only authenticated users from the `project.local` domain can access the network.

- **Device Posture Assessment:** Configure PacketFence to perform compliance checks on connecting devices. For example, it could verify that a Windows client has its firewall enabled and that an Ubuntu client is running an up-to-date kernel before granting full network access.
- **Dynamic VLAN Assignment:** Implement policies that dynamically assign devices to different VLANs based on their identity and compliance status. A non-compliant device could be automatically moved to a "quarantine" VLAN with restricted internet access for remediation.

10.2 Advanced SOAR Playbooks and Threat Intelligence

The current SOAR use case focuses on automated blocking. This can be expanded to create a more intelligent and context-aware response system:

- **Threat Intelligence Enrichment:** Enhance the Shuffle workflow to, upon receiving an alert, automatically query external threat intelligence platforms (like VirusTotal or AbuseIPDB) for the attacker's IP address. This enriched data can help an analyst quickly determine if the IP is a known bad actor, adding valuable context to the incident.
- **Host-Level Remediation:** Develop playbooks that go beyond network blocking. For instance, a workflow could use SSH or WinRM to connect to a compromised host (as identified by Wazuh) and execute a command to isolate it from the network by disabling its network interface, providing a more surgical response than a full IP block.
- **Integration with Ticketing Systems:** Connect Shuffle to an incident management platform like TheHive or Jira. A high-severity alert could automatically create a detailed incident ticket, assign it to an analyst, and populate it with all the relevant information from Wazuh and threat intelligence lookups.

10.3 Hybrid Cloud Integration

Modern enterprise networks are rarely confined to on-premises infrastructure. A critical next step would be to extend the lab into a hybrid cloud environment:

- **Site-to-Site VPN:** Establish an IPsec VPN tunnel from the FortiGate firewall to a Virtual Private Cloud (VPC) in a public cloud provider like AWS or Azure.
- **Cloud Workload Monitoring:** Deploy a virtual machine within the cloud VPC, install a Wazuh agent on it, and configure security group rules to allow it to communicate back to the on-premises Wazuh manager. This would extend visibility and security monitoring to cloud assets.
- **Securing Cloud-Native Services:** Explore the integration of logs from cloud-native services (e.g., AWS CloudTrail, Azure Activity Log) into the Wazuh SIEM to create a single pane of glass for both on-premises and cloud security events.

10.4 Advanced Threat Simulation and Purple Teaming

To continuously validate and improve the defensive stack, more sophisticated attack scenarios can be simulated:

- **Lateral Movement Testing:** Simulate an attacker who has compromised the DMZ web server and is attempting to pivot to the internal LAN. This would rigorously test the effectiveness of the FortiGate's intra-zone and inter-zone firewall policies.
- **Data Exfiltration Simulation:** Attempt to exfiltrate sensitive data from the LAN to an external server and monitor whether Suricata's rules or Wazuh's File Integrity Monitoring (FIM) can detect and alert on the activity.
- **Purple Team Exercises:** Use the lab as a "purple team" environment, where an attacker (red team) executes a technique while a defender (blue team) observes the security tool stack in real-time to see if the activity is detected, alerted on, and properly remediated. This collaborative approach is invaluable for tuning detection rules and closing security gaps.

11 Conclusion

This project successfully culminated in the creation of a robust, multi-zone, and secure virtual network lab. It demonstrates a segmented architecture with centralized authentication, proactive inline threat prevention, and a comprehensive security monitoring and response capability. The integration of the DMZ, SIEM, and SOAR platforms elevates the lab from a simple network to a dynamic cyber range. The automated response workflow proves the effectiveness of a defense-in-depth strategy where multiple security layers work in concert. The final environment serves as an excellent platform for advanced cybersecurity testing and defense strategy validation.