

# Home Lab

## *Firewall and SIEM Configuration*

In this lab, a **network infrastructure** was designed and deployed with a focus on **system defense and security monitoring**.

The environment includes a **firewall and router based on pfSense**, which manage the routing of external traffic toward internal devices, while enforcing filtering policies to ensure both security and proper network functionality.

```
FreeBSD/amd64 (pfSense.home.arp) (ttyv0)

pfSense - Netgate Device ID: 3d8e396099ba79e3db65

*** Welcome to pfSense 2.7.2-RELEASE (amd64) on pfSense ***

      WAN (wan)      -> em0          -> v4/DHCP4: 192.168.1.203/24
                                v6/DHCP6: 2a0c:5a81:bb11:d100:20c:29ff:fe90:a39f/64
      LAN (lan)       -> em1          -> v4: 192.168.100.1/24
      DMZ (opt1)     -> em2          -> v4: 192.168.200.1/24
      DMZ2 (opt2)    -> em3          -> v4: 192.168.250.1/24

  0) Logout (SSH only)          9) pfTop
  1) Assign Interfaces          10) Filter Logs
  2) Set interface(s) IP address 11) Restart webConfigurator
  3) Reset webConfigurator password 12) PHP shell + pfSense tools
  4) Reset to factory defaults   13) Update from console
  5) Reboot system               14) Enable Secure Shell (sshd)
  6) Halt system                 15) Restore recent configuration
  7) Ping host                   16) Restart PHP-FPM
  8) Shell
```

The network topology was divided into three main segments:

- **LAN Subnet (192.168.100.0/24)**: contains a single Windows workstation configured with the static IP address **192.168.100.10**. This subnet represents the secure, internal corporate environment.
- **DMZ Subnet (192.168.200.0/24)**: hosts a **honeypot** designed to collect data from potential attackers. The system, with IP **192.168.200.10**, is intentionally exposed to external connections, simulating a vulnerable service.
- **DMZ2 Subnet (192.168.250.0/24)**: includes a Linux-based machine with IP **192.168.250.10**, running both an **Apache2 web server** and a **Suricata IDS**, configured to generate alerts upon detecting suspicious traffic or behavior.

All systems in the lab have an **Elastic Stack (ELK) agent** installed, enabling real-time

monitoring and data collection.

Through this integration, event logs are centralized, allowing for anomaly detection and the identification of attack patterns across the network segments.

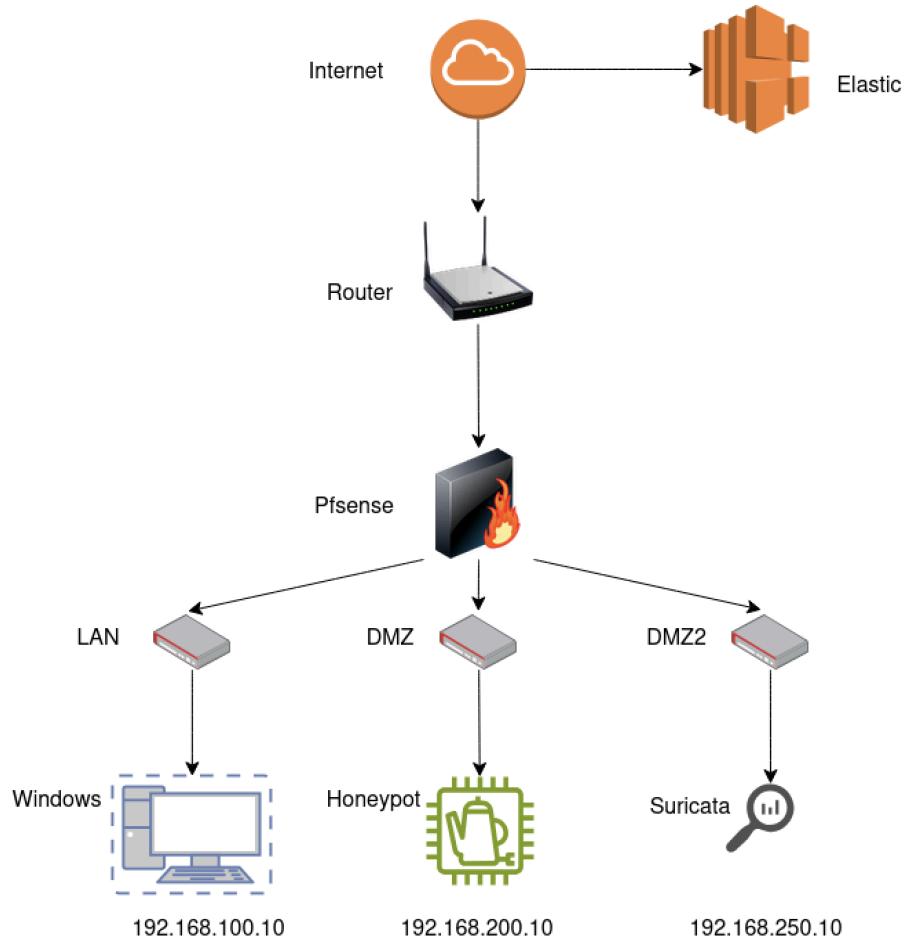
For the **SIEM component**, **Kibana** was used as the main visualization and analysis interface.

Each host's Elastic Agent was configured with its own log collection policy and operational parameters, facilitating **event tracking and correlation** between devices and network zones.

## *pfSense Firewall Configuration*

The **pfSense firewall** serves as the central control point for managing traffic between the internal subnets and the external network (WAN).

Its configuration aims to ensure proper **network segmentation, isolation of exposed services, and protection of internal assets** from potential threats.

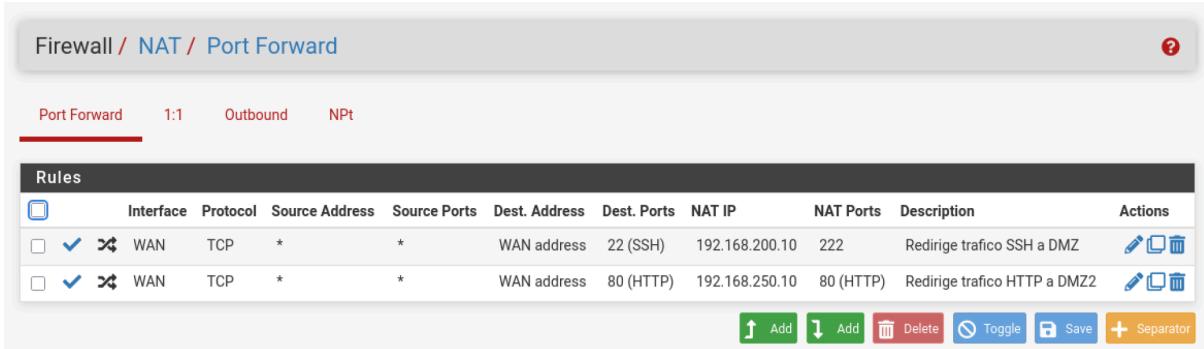


# NAT Rules and Traffic Redirection

Two NAT (Network Address Translation) rules were created to redirect incoming traffic from the **WAN interface** to the appropriate devices within the DMZ zones:

- **HTTP Traffic (port 80):** redirected from the WAN interface to **192.168.250.10** (the Apache web server located in DMZ2).  
This rule allows external users to access the web service hosted in the lab environment.
- **SSH Traffic (external port 22 → internal port 222):** redirected from the WAN interface to **192.168.200.10** (the honeypot in DMZ).  
This setup simulates an intentionally exposed vulnerable service, allowing attackers to interact with the honeypot and generate behavioral data for later analysis.

The NAT rules automatically generate the corresponding rules on the **WAN interface**, ensuring that connections are processed according to the defined security policies.



The screenshot shows the pfSense configuration interface for port forwarding. The top navigation bar includes links for Firewall, NAT, and Port Forward. The Port Forward tab is selected. Below the tabs, there are three categories: 1:1, Outbound, and NPt. The main area is titled "Rules" and displays a table of two entries. The columns are: Interface, Protocol, Source Address, Source Ports, Dest. Address, Dest. Ports, NAT IP, NAT Ports, Description, and Actions. The first rule maps WAN TCP port 22 (SSH) to 192.168.200.10 port 222, with the description "Redirige trafico SSH a DMZ". The second rule maps WAN TCP port 80 (HTTP) to 192.168.250.10 port 80 (HTTP), with the description "Redirige trafico HTTP a DMZ2". At the bottom of the table are buttons for Add, Delete, Toggle, Save, and Separator.

	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions
<input type="checkbox"/>	✓ ✘ WAN	TCP	*	*	WAN address	22 (SSH)	192.168.200.10	222	Redirige trafico SSH a DMZ	
<input type="checkbox"/>	✓ ✘ WAN	TCP	*	*	WAN address	80 (HTTP)	192.168.250.10	80 (HTTP)	Redirige trafico HTTP a DMZ2	

## Rules per Subnet

LAN Subnet (192.168.100.0/24)

- Allow **SSH traffic** to DMZ (192.168.200.10) for honeypot administration.
- Block all other traffic to DMZ and DMZ2 to prevent direct communication between segments.
- Allow outbound **DNS (UDP 53)**, **HTTP (TCP 80)**, and **HTTPS (TCP 443)** traffic.

These rules are placed at the end of the list since pfSense applies firewall policies from top to bottom.

Floating	WAN	LAN	DMZ	DMZ2							
Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	4/1.17 MiB	*	*	*	LAN Address	443 80	*	*		Anti-Lockout Rule	   
<input type="checkbox"/>	<input checked="" type="checkbox"/> 0/11 KiB	IPv4 TCP	192.168.100.10	*	192.168.200.10	22 (SSH)	*	none		Permite acceso por SSH hacia DMZ	   
<input type="checkbox"/>	<input checked="" type="checkbox"/> 0/572 B	IPv4 *	LAN subnets	*	DMZ subnets	*	*	none		Bloquea todo trafico hacia DMZ	   
<input type="checkbox"/>	<input checked="" type="checkbox"/> 0/0 B	IPv4 *	LAN subnets	*	DMZ2 subnets	*	*	none		Bloquea todo trafico hacia DMZ2	   
<input type="checkbox"/>	<input checked="" type="checkbox"/> 27/26.74 MiB	IPv4 TCP	LAN subnets	*	*	webs	*	none		Permite Trafico HTTP y HTTPS	   
<input type="checkbox"/>	<input checked="" type="checkbox"/> 3/74 KiB	IPv4 UDP	LAN subnets	*	*	53 (DNS)	*	none		Permite trafico DNS	   

### DMZ Subnet (192.168.200.0/24)

- Allow incoming traffic from the **WAN address** (pfSense public IP) to **192.168.200.10** on port **222/TCP**.
- Allow SSH access from the **LAN** (192.168.100.10) exclusively for honeypot management.
- Block all traffic to the **LAN** and **DMZ2** networks, maintaining isolation between zones.
- Allow **DNS (UDP)**, **HTTP (TCP)**, and **HTTPS (TCP)** traffic at the end of the rule set.

Floating	WAN	LAN	DMZ	DMZ2							
Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	<input checked="" type="checkbox"/> 0/0 B	IPv4 TCP	WAN address	*	192.168.200.10	222	*	none		Permite trafico a cowrie	   
<input type="checkbox"/>	<input checked="" type="checkbox"/> 0/0 B	IPv4 TCP	192.168.100.10	*	192.168.200.10	22 (SSH)	*	none		Permite acceso por SSH desde LAN	   
<input type="checkbox"/>	<input checked="" type="checkbox"/> 0/1 KIB	IPv4 *	DMZ subnets	*	LAN subnets	*	*	none		Bloquea todo el trafico hacia LAN	   
<input type="checkbox"/>	<input checked="" type="checkbox"/> 0/0 B	IPv4 *	DMZ subnets	*	DMZ2 subnets	*	*	none		Bloquea todo el trafico hacia DMZ2	   
<input type="checkbox"/>	<input checked="" type="checkbox"/> 7/84 KiB	IPv4 UDP	DMZ subnets	*	*	53 (DNS)	*	none		Permite trafico DNS	   
<input type="checkbox"/>	<input checked="" type="checkbox"/> 25/3.90 MiB	IPv4 TCP	DMZ subnets	*	*	webs	*	none		Permite trafico HTTP y HTTPS	   

### DMZ2 Subnet (192.168.250.0/24)

- Allow traffic from the **WAN address** to **192.168.250.10** (the Apache web server).
- Block all traffic to the **LAN** and **DMZ** networks to ensure complete separation between environments.
- Allow **DNS (UDP)**, **HTTP (TCP)**, and **HTTPS (TCP)** traffic as the final rule in the chain.

Rules (Drag to Change Order)										
States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	WAN address	*	192.168.250.10	80 (HTTP)	*	none	Permite trafico HTTP desde WAN	
<input type="checkbox"/>	✗ 0/1020 B	IPv4 *	DMZ2 subnets	*	LAN subnets	*	*	none	Rechaza todo el trafico hacia LAN	
<input type="checkbox"/>	✗ 0/0 B	IPv4 *	DMZ2 subnets	*	DMZ subnets	*	*	none	Rechaza todo el trafico hacia DMZ	
<input type="checkbox"/>	✓ 12/328 KiB	IPv4 UDP	*	*	*	53 (DNS)	*	none	Permite trafico DNS	
<input type="checkbox"/>	✓ 18/13.67 MiB	IPv4 TCP	*	*	*	webs	*	none	Permite trafico HTTP y HTTPS	

Add Add Delete Toggle Copy Save Separator

## Result

This configuration ensures:

- A **controlled and segmented traffic flow** between networks.
- Only **necessary services** (web and honeypot) are accessible from the outside.
- Internal hosts remain isolated and protected** from external access.

By applying these policies, the network architecture follows the principles of **defense in depth** and **network segmentation**, both essential concepts within Blue Team operations.

## Suricata

In the DMZ2 subnet, where the Apache web server is hosted, the intrusion detection system **Suricata** was deployed. The IDS was configured in **passive mode (detection only)** so that it generates alerts when detecting anomalous behavior or attack patterns without blocking network traffic.

```

dani@kali: /etc/suricata/rules

File: suricata-kc-http.rules

1 #alert tcp any any -> any any (msg:"Trafico detectado"; sid:1;)
2 #alert tcp any any -> 192.168.50.210 22 (msg:"Trafico SSH detectado"; sid:2; classtype:attempted-a
3 dmin;)
4 #alert tcp any any -> any any (msg:"Archivo PDF descargado"; flow:established,to_client; fileext:-
5 pdf"; sid:3; classtype:file-download;)
6 alert http any any -> 192.168.250.10 80 (\ 
7     msg:"Trafico HTTP ENTRANTE"; \
8     classtype:web-application-activity; sid:001; rev:1;)
9 
10 # Genera alerta cuando se hace fuzzing y se hacen 25 conexiones en 30s.
11 alert tcp any any -> 192.168.250.10 80 (\ 
12     msg:"Intento de Fuzzing Web"; \
13     flow:established,to_server; \
14     threshold:type threshold, track_by_src, count 25, seconds 30; \
15     classtype:web-application-attack; sid:002; rev:1;)
16 
17 # Genera nueva alerta por cada 60s. Trackeada por src es por IP e inicia el contador en 1.
18 alert http any any -> 192.168.250.10 80 (\ 
19     msg:"Iniciando contador Login"; \
20     flow:established,to_server; \
21     content:"POST"; http_method; \
22     content:"/login.php"; http_uri; \
23     nocase; \
24     threshold:type limit, track by_src,count 1, seconds 60; \
25     classtype:web-application-activity; sid:003; rev:1;)
26 
27 # Genera alerta al superar 5 intentos de esa IP en menos de 60s.
28 alert http any any -> 192.168.250.10 80 (\ 
29     msg:"Ataque de fuerza bruta. Mas de 5 intentos en 60s"; \
30     flow:established,to_server; \
31     content:"POST"; http_method; \
32     content:"/login.php"; http_uri; \
33     nocase; \
34     threshold:type threshold, track by_src, count 5, seconds 60; \
35     classtype:web-application-attack; sid:004; rev:1;)

```

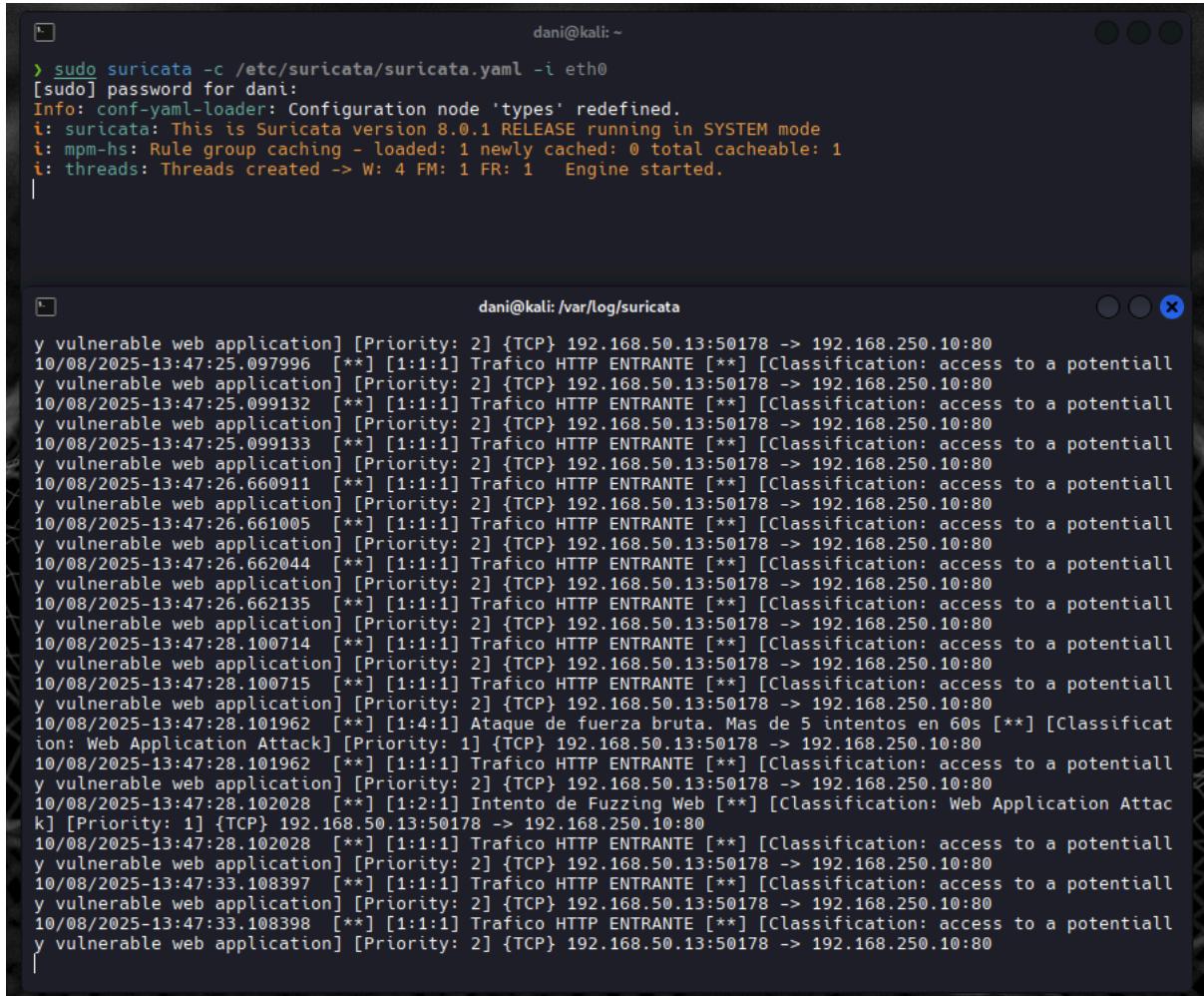
The main objective of this implementation is to detect possible attack attempts against the web service and record this activity for later analysis in the SIEM. Suricata uses a set of custom rules designed to cover different web attack scenarios and suspicious activity patterns.

## Configured rules

A specific set of detection rules was created with the following criteria:

- **HTTP traffic:** an alert is triggered each time an HTTP connection to the Apache server is detected. This allows maintaining traceability of every incoming request.
- **Authentication attempts:** when a login event is detected on the web application, a variable is defined to act as a counter linked to the source IP address.
- **Brute-force detection:** if the counter reaches five consecutive failed login attempts, an alert is generated, indicating a possible brute-force or credential stuffing attack.
- **Fuzzing or scanning detection:** when an abnormal number of requests are detected within a short time period, Suricata triggers an alert indicating a

possible fuzzing or enumeration attempt against the service.



The screenshot shows two terminal windows. The top window is titled 'dani@kali: ~' and displays the command 'sudo suricata -c /etc/suricata/suricata.yaml -i eth0'. It shows Suricata version 8.0.1 running in SYSTEM mode with rule group caching enabled. The bottom window is titled 'dani@kali: /var/log/suricata' and shows a log entry from October 8, 2025, at 13:47:25.099132. The log entry indicates a potential vulnerable web application attack (HTTP ENTRANTE) from 192.168.50.13 to 192.168.250.10:80. The classification is 'access to a potentially vulnerable web application'.

```
> sudo suricata -c /etc/suricata/suricata.yaml -i eth0
[sudo] password for dani:
Info: conf-yaml-loader: Configuration node 'types' redefined.
i: suricata: This is Suricata version 8.0.1 RELEASE running in SYSTEM mode
i: mpm-hs: Rule group caching - loaded: 1 newly cached: 0 total cacheable: 1
i: threads: Threads created -> W: 4 FM: 1 FR: 1 Engine started.

dani@kali: /var/log/suricata

y vulnerable web application] [Priority: 2] {TCP} 192.168.50.13:50178 -> 192.168.250.10:80
10/08/2025-13:47:25.097996 [**] [1:1:1] Trafico HTTP ENTRANTE [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.50.13:50178 -> 192.168.250.10:80
10/08/2025-13:47:25.099132 [**] [1:1:1] Trafico HTTP ENTRANTE [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.50.13:50178 -> 192.168.250.10:80
10/08/2025-13:47:25.099133 [**] [1:1:1] Trafico HTTP ENTRANTE [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.50.13:50178 -> 192.168.250.10:80
10/08/2025-13:47:26.660911 [**] [1:1:1] Trafico HTTP ENTRANTE [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.50.13:50178 -> 192.168.250.10:80
10/08/2025-13:47:26.661005 [**] [1:1:1] Trafico HTTP ENTRANTE [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.50.13:50178 -> 192.168.250.10:80
10/08/2025-13:47:26.662044 [**] [1:1:1] Trafico HTTP ENTRANTE [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.50.13:50178 -> 192.168.250.10:80
10/08/2025-13:47:26.662135 [**] [1:1:1] Trafico HTTP ENTRANTE [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.50.13:50178 -> 192.168.250.10:80
10/08/2025-13:47:28.100714 [**] [1:1:1] Trafico HTTP ENTRANTE [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.50.13:50178 -> 192.168.250.10:80
10/08/2025-13:47:28.100715 [**] [1:1:1] Trafico HTTP ENTRANTE [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.50.13:50178 -> 192.168.250.10:80
10/08/2025-13:47:28.101962 [**] [1:4:1] Ataque de fuerza bruta. Mas de 5 intentos en 60s [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 192.168.50.13:50178 -> 192.168.250.10:80
10/08/2025-13:47:28.101962 [**] [1:1:1] Trafico HTTP ENTRANTE [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.50.13:50178 -> 192.168.250.10:80
10/08/2025-13:47:28.102028 [**] [1:2:1] Intento de Fuzzing Web [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 192.168.50.13:50178 -> 192.168.250.10:80
10/08/2025-13:47:28.102028 [**] [1:1:1] Trafico HTTP ENTRANTE [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.50.13:50178 -> 192.168.250.10:80
10/08/2025-13:47:33.108397 [**] [1:1:1] Trafico HTTP ENTRANTE [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.50.13:50178 -> 192.168.250.10:80
10/08/2025-13:47:33.108398 [**] [1:1:1] Trafico HTTP ENTRANTE [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.168.50.13:50178 -> 192.168.250.10:80
```

These rules are stored in the corresponding configuration file and managed through the policies defined in Kibana for centralized forwarding and visualization. The generated alerts are logged in **EVE JSON format**, allowing direct integration with the **Elastic Agent** for delivery to the ELK stack.

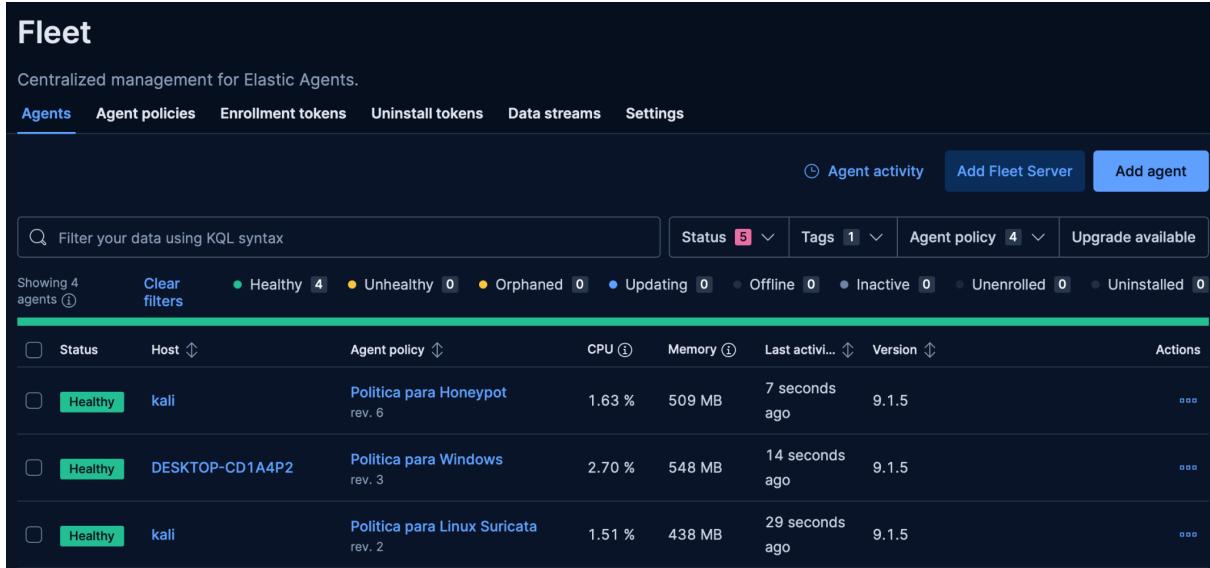
## Expected results

Proper configuration of Suricata allows for:

- Continuous monitoring of HTTP traffic directed to the web server.
- Identification of intrusion attempts or anomalous behaviors in real time.
- Event correlation within the SIEM to detect broader attack patterns.
- Improved defensive capabilities of the environment through early threat detection.

# Kibana

For centralized management and analysis of security events, **Kibana** was used as the main interface of the **SIEM based on the ELK Stack (Elastic Stack)**. From Kibana, all logs sent by the different **Elastic Agents** installed on the lab devices can be visualized, allowing analysts to correlate events, build dashboards, and proactively detect anomalies.



The screenshot shows the Kibana Fleet interface. At the top, there's a header with tabs for 'Agents' (which is selected), 'Agent policies', 'Enrollment tokens', 'Uninstall tokens', 'Data streams', and 'Settings'. Below the header are buttons for 'Agent activity', 'Add Fleet Server', and 'Add agent'. A search bar with placeholder text 'Filter your data using KQL syntax' is followed by filters for 'Status' (5 healthy, 1 tagged), 'Agent policy' (4 assigned), and 'Upgrade available'. A summary bar indicates 4 agents total, with counts for Healthy (4), Unhealthy (0), Orphaned (0), Updating (0), Offline (0), Inactive (0), Unenrolled (0), and Uninstalled (0). The main table lists three agents:

Status	Host	Agent policy	CPU	Memory	Last activi...	Version	Actions
Healthy	kali	Política para Honeypot rev. 6	1.63 %	509 MB	7 seconds ago	9.1.5	...
Healthy	DESKTOP-CD1A4P2	Política para Windows rev. 3	2.70 %	548 MB	14 seconds ago	9.1.5	...
Healthy	kali	Política para Linux Suricata rev. 2	1.51 %	438 MB	29 seconds ago	9.1.5	...

## Agent policies and integrations

Three custom agent policies were created, each one tailored to the type of device and its specific role within the network. Every policy includes integrations designed to collect and forward the right set of logs for each system:

- **Suricata Policy:** collects and forwards alerts generated by the IDS deployed in DMZ2, including event fields such as source, destination, and detected signatures.
- **Windows Policy:** gathers logs from the Windows operating system (security, system events, PowerShell executions, etc.) located in the LAN subnet.
- **Honeypot Policy (FileStream):** captures executed commands and recorded activity from the honeypot.

Once created, the policies were assigned to their respective agents. After installation, each agent started sending data continuously to Elasticsearch for indexing and correlation.

The screenshot shows the Fleet interface with a banner at the top stating, "We've added new privileges that let you define more granularly who can view or edit Fleet agents, policies, and settings. Learn more." Below the banner, the title "Fleet" is displayed, followed by the subtitle "Centralized management for Elastic Agents." A navigation bar includes tabs for "Agents", "Agent policies" (which is selected), "Enrollment tokens", "Uninstall tokens", "Data streams", and "Settings". Below the navigation bar is a search bar with placeholder text "Filter your data using KQL syntax" and a "Reload" button. To the right of the search bar is a "Create agent policy" button. The main content area displays a table of agent policies:

Name	Last updated on	Unprivileged / Privileged	Integrations	Actions
Política para Honeypot rev. 6	Oct 10, 2025	0 / 1 (1)	2	...
Política para Windows rev. 3	Oct 07, 2025	0 / 1 (1)	2	...
Política para Linux Suricata rev. 2	Oct 07, 2025	0 / 1 (1)	2	...
Elastic Cloud agent policy rev. 4 Default agent policy for agents hosted on Elastic Cloud	Oct 07, 2025	1 / 0 (1)	2	...

## Data discovery and analysis

Using Kibana's **Discover** tab, an initial review of generated events was performed. Custom **data views** were configured to simplify log reading and analysis based on each agent's origin.

- **Suricata data view.** Includes the following fields:

- @timestamp
- source.ip
- destination.ip
- suricata.eve.alert.signature\_id
- suricata.eve.alert.signature
- log.file.path

These fields allow identifying which alerts were triggered, the originating IP address, and the time they occurred. By filtering on `suricata.eve.alert.signature`, only actual Suricata detections are displayed.

	@timestamp	source.ip	destination.ip	suricata.eve.alert.signature	log-file.path
□	Oct 8, 2025 @ 19:47:33.108	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:33.108	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:28.102	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:28.102	192.168.50.13	192.168.250.10	2 Intento de Fuzzing Web	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:28.101	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:28.101	192.168.50.13	192.168.250.10	4 Ataque de fuerza bruta. Mas de 5 intentos en 6s	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:28.100	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:28.100	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:26.602	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:26.602	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:26.661	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:26.660	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:25.099	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:25.099	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:25.097	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:25.097	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:22.848	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:22.848	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:22.839	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:22.839	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:20.586	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:20.586	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:20.583	192.168.50.13	192.168.250.10	3 Iniciando contador Login	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:20.583	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:20.522	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:20.522	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:15.515	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 19:47:15.515	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 10:45:03.319	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json
□	Oct 8, 2025 @ 10:45:03.319	192.168.50.13	192.168.250.10	1 Trafico HTTP ENTRANTE	/var/log/suricata/eve.json

- **Honeypot data view.** Includes:

- @timestamp
- log.file.path
- message

By filtering the keyword `CMD` in the `message` field, it is possible to see which commands were executed by attackers inside the honeypot, helping analyze their techniques and behavior.

	@timestamp	log.file.path	message
□	Oct 8, 2025 @ 12:20:29.048	/home/k3p4/honeypot.log	2025-10-08T10:13:31+0000 [HoneyPotSSHTransport,1,192.168.50.13] :0: exit
□	Oct 8, 2025 @ 12:20:29.039	/home/k3p4/honeypot.log	2025-10-08T10:13:28+0000 [HoneyPotSSHTransport,1,192.168.50.13] :0: hola
□	Oct 8, 2025 @ 12:20:29.039	/home/k3p4/honeypot.log	2025-10-08T10:13:25+0000 [HoneyPotSSHTransport,1,192.168.50.13] :0: cd ..
□	Oct 8, 2025 @ 12:20:29.019	/home/k3p4/honeypot.log	2025-10-08T10:13:12+0000 [HoneyPotSSHTransport,0,192.168.50.13] :0: exit
□	Oct 8, 2025 @ 12:20:29.017	/home/k3p4/honeypot.log	2025-10-08T10:13:07+0000 [HoneyPotSSHTransport,0,192.168.50.13] :0: f
□	Oct 8, 2025 @ 12:20:29.017	/home/k3p4/honeypot.log	2025-10-08T10:13:02+0000 [HoneyPotSSHTransport,0,192.168.50.13] :0: d
□	Oct 8, 2025 @ 12:20:29.017	/home/k3p4/honeypot.log	2025-10-08T10:13:02+0000 [HoneyPotSSHTransport,0,192.168.50.13] :0: s
□	Oct 8, 2025 @ 12:20:29.015	/home/k3p4/honeypot.log	2025-10-08T10:12:49+0000 [HoneyPotSSHTransport,0,192.168.50.13] :0: tes

- **Windows data view.** Allows tracking activities executed on the Windows machine within the LAN. In this example, events related to PowerShell were filtered to detect potentially suspicious or automated operations.

The screenshot shows the Kibana search interface with the following details:

- Selected fields:** @timestamp, host.ip, winlog.task, message, winlog.event\_id, host.name.
- Popular fields:** message, winlog.event\_id, winlog.task, host.name, host.ip.
- Available fields:** @timestamp, agent.ephemeral\_id, agent.id, agent.name, agent.type, agent.version, data\_stream.dataset, data\_stream.namespace, data\_stream.type, dataset.name, dataset.namespace, dataset.type.
- Documents (173):**
  - Oct 10, 2025 @ 16:45:42.053: [fe80::bcc8:8233 Ciclo de vida :21c:fe79\_ del motor 192.168.100.10] - El estado del motor ha cambiado de No... (400)
  - Oct 10, 2025 @ 16:45:42.020: [fe80::bcc8:8233 Ciclo de vida :21c:fe79\_ del proveedor 192.168.100.10] - El proveedor "Variable" está Started. ... (600)
  - Oct 10, 2025 @ 16:45:42.020: [fe80::bcc8:8233 Ciclo de vida :21c:fe79\_ del proveedor 192.168.100.10] - El proveedor "Function" está Started. ... (600)
  - Oct 10, 2025 @ 16:45:42.020: [fe80::bcc8:8233 Ciclo de vida :21c:fe79\_ del proveedor 192.168.100.10] - El proveedor "FileSystem" está Started. ... (600)
  - Oct 10, 2025 @ 16:45:42.008: [fe80::bcc8:8233 Ciclo de vida :21c:fe79\_ del proveedor 192.168.100.10] - El proveedor "Environment" está Started. ... (600)
  - Oct 10, 2025 @ 16:45:42.008: [fe80::bcc8:8233 Ciclo de vida :21c:fe79\_ del proveedor 192.168.100.10] - El proveedor "Alias" está Started. ... (600)
  - Oct 10, 2025 @ 16:45:42.008: [fe80::bcc8:8233 Ciclo de vida :21c:fe79\_ del proveedor 192.168.100.10] - El proveedor "Registry" está Started. ... (600)
  - Oct 10, 2025 @ 16:42:12.855: [fe80::bcc8:8233 Ejecutando :21c:fe79\_ canalización 192.168.100.10] - CommandInvocation(Add-Type): "Add-Type" ... (4103)
  - Oct 10, 2025 @ 16:42:12.851: [fe80::bcc8:8233 Ejecutando :21c:fe79\_ canalización 192.168.100.10] - CommandInvocation(Add-Type): "Add-Type" ... (4103)
  - Oct 10, 2025 @ 16:42:12.844: [fe80::bcc8:8233 Detalles de :21c:fe79\_ ejecución de la 192.168.100.10] - Detalles de ejecución de la canalización ... (800)
  - Oct 10, 2025 @ 16:42:12.844: [fe80::bcc8:8233 Detalles de :21c:fe79\_ ejecución de la 192.168.100.10] - Detalles de ejecución de la canalización ... (800)

## Dashboards and visualization

A **custom dashboard** was created in Kibana to visually represent the most relevant security events. Among the configured metrics are:

- The total count of Suricata-generated alerts.
- Top 10 source and destination IP addresses.
- Time-based evolution of events by detection signature.

These visualizations help correlate activities across subnets, identify attack patterns, and assess the effectiveness of both IDS and firewall rules.

The screenshot shows the Splunk Security Dashboard with the following details:

- Discover:** KC
- Fields:** From, Destination, Message, SID, Numero de Conexiones.
- Data:**

From	Destination	Message	SID	Numero de Conexiones
192.168.50.13	192.168.250.10	Traffic HTTP ENTRANTE	1	98
192.168.50.13	192.168.250.10	Iniciando contador Login	3	6
192.168.50.13	192.168.250.10	Ataque de fuerza bruta. Mas de 5 intentos	4	3
192.168.50.13	192.168.250.10	Intento de Fuzzing Web	2	3

## Results and conclusions

Thanks to the integration of **Elastic + Suricata + pfSense**, the lab setup enables:

- Real-time detection and analysis of security incidents.
- Centralization of security information across all subnets.
- Observation and correlation of attacker interactions captured by the honeypot with IDS alerts.
- A global, unified view of the environment through dynamic dashboards.

Together, this deployment represents a **functional and educational Blue Team environment**, ideal for understanding how defensive components interact within a modern infrastructure.

This lab allowed me to gain a deeper understanding of how a **stateful firewall** operates and the importance of **rule hierarchy** for effective traffic control. It also helped me recognize the significance of properly configured security policies and how **event correlation between the firewall, IDS, and SIEM** is essential for effective incident detection and analysis within a corporate network.