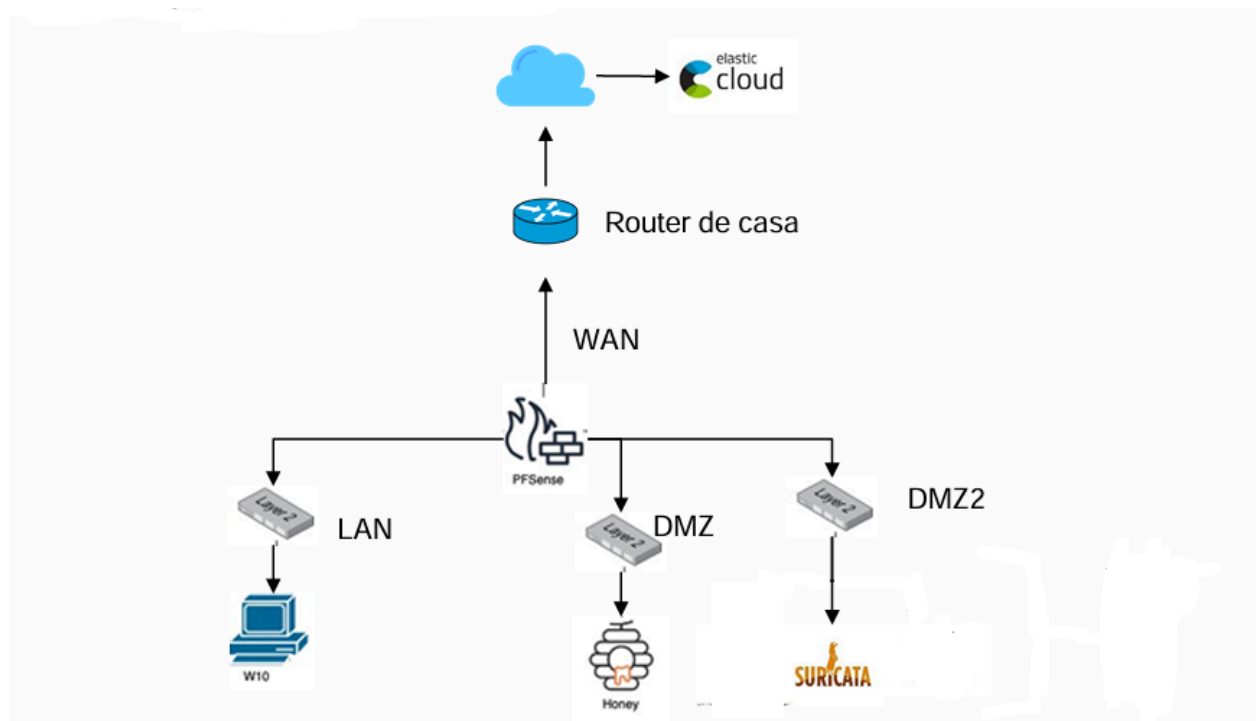


ADRIÁN LÓPEZ FERNÁNDEZ

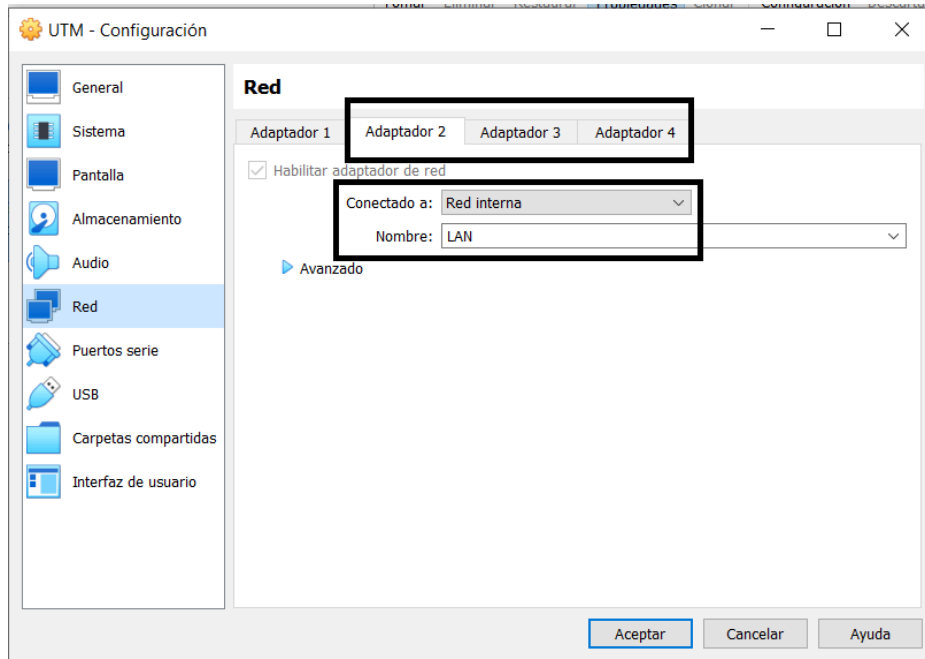
BLUE TEAM

Con el objetivo de explorar las capacidades y ventajas de un SIEM basado en Elastic Cloud, se ha diseñado y desplegado una infraestructura de red que permita gestionar, centralizar y analizar eventos de seguridad de manera eficiente. Este trabajo documenta el proceso de implementación, desde la planificación inicial hasta la puesta en marcha del sistema con la siguiente estructura:



Para ello, se ha procedido a crear un **UTM (pfSense)** que nos proporcionará un servidor DNS, DHCP y Firewall.

pfSense se ha instalado utilizando VirtualBox, configurando tres redes internas: **LAN**, **DMZ** y **DMZ2**.



A través del administrador de pfSense, se han asignado **direcciones IP y rangos** para cada red, configurándolas como **puertas de enlace**, además de habilitar los servicios de **DNS y DHCP**.

Las direcciones utilizadas son las siguientes:

Red LAN: 192.168.100.1/24

Red DMZ: 192.168.200.1/24

Red DMZ2: 192.168.250.1/24

```

UTM (UTM pfsense configurado) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
pfSense 2.7.2-RELEASE amd64 20231206-2010
Bootup complete

FreeBSD/amd64 (UTM.keepcoding.local) (ttyv0)

VirtualBox Virtual Machine - Netgate Device ID: 7d986aefdfaa5e3b7d4

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

WAN (wan)      -> em0      -> v4/DHCP4: 192.168.0.15/24
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

Enter an option:

```

A continuación, se ha configurado el **DNS Resolver** para que pfSense gestione los paquetes DNS y pueda **resolver las direcciones IP de los sitios web mediante sus dominios**. Además, se ha habilitado el **reenvío** (forwarding) para delegar en otros servidores DNS en caso de fallo de los servidores primarios.

General DNS Resolver Options	
Enable	<input checked="" type="checkbox"/> Enable DNS resolver
Listen Port	<input type="text" value="53"/> <small>The port used for responding to DNS queries. It should normally be left blank unless another service needs to bind to TCP/UDP port 53.</small>
Enable SSL/TLS Service	<input type="checkbox"/> Respond to incoming SSL/TLS queries from local clients <small>Configures the DNS Resolver to act as a DNS over SSL/TLS server which can answer queries from clients which also support DNS over TLS. Activating this option disables automatic interface response routing behavior, thus it works best with specific interface bindings.</small>
SSL/TLS Certificate	<input type="text" value="GUI default (677ee48096c3a)"/> <small>The server certificate to use for SSL/TLS service. The CA chain will be determined automatically.</small>
SSL/TLS Listen Port	<input type="text" value="853"/> <small>The port used for responding to SSL/TLS DNS queries. It should normally be left blank unless another service needs to bind to TCP/UDP port 853.</small>
Network Interfaces	<div><div>All</div><div>WAN</div><div>LAN</div><div>DMZ</div><div>DMZ2</div></div> <small>Interface IP addresses used by the DNS Resolver for responding to queries from clients. If an interface has both IPv4 and IPv6 addresses, both are used. Queries to addresses not selected in this list are discarded. The default behavior is to respond to queries on every available IPv4 and IPv6 address.</small>
Outgoing Network Interfaces	<div><div>All</div><div>WAN</div><div>LAN</div><div>DMZ</div><div>DMZ2</div></div> <small>Utilize different network interface(s) that the DNS Resolver will use to send queries to authoritative servers and receive their replies. By default all interfaces are used.</small>
Strict Outgoing Network Interface Binding	<input type="checkbox"/> Do not send recursive queries if none of the selected Outgoing Network Interfaces are available. <small>By default the DNS Resolver sends recursive DNS requests over any available interfaces if none of the selected Outgoing Network Interfaces are available. This option makes the DNS Resolver refuse recursive queries.</small>
System Domain Local Zone Type	<input type="text" value="Transparent"/> <small>The local-zone type used for the pfSense system domain (System General Setup Domain). Transparent is the default.</small>
DNSSEC	<input type="checkbox"/> Enable DNSSEC Support
Python Module	<input type="checkbox"/> Enable Python Module <small>Enable the Python Module.</small>
DNS Query	<input checked="" type="checkbox"/> Enable Forwarding Mode

También se ha configurado el **servidor DHCP** con los siguientes rangos de direcciones IP dinámicas para los dispositivos conectados a cada red:

LAN: 192.168.100.100 - 192.168.100.200

DMZ: 192.168.200.100 - 192.168.200.150 (Además de una IP estática)

DMZ2: 192.168.250.100 - 192.168.250.150 (Además de una IP estática)

General DHCP Options

DHCP Backend ISC DHCP

Enable ☒ Enable DHCP server on LAN interfaceBOOTP ☐ Ignore BOOTP queriesDeny Unknown Clients Allow all clients

When set to **Allow all clients**, any DHCP client will get an IP address within this scope/range on this interface. If set to **Allow known clients from any interface**, any DHCP client with a MAC address listed in a static mapping on **any** scope(s)/interface(s) will get an IP address. If set to **Allow known clients from only this interface**, only MAC addresses listed in static mappings on this interface will get an IP address within this scope/range.

Ignore Denied Clients ☐ Ignore denied clients rather than reject
This option is not compatible with failover and cannot be enabled when a Failover Peer IP address is configured.

Ignore Client Identifiers ☐ Do not record a unique identifier (UID) in client lease data if present in the client DHCP request
This option may be useful when a client can dual boot using different client identifiers but the same hardware (MAC) address. Note that the resulting server behavior violates the official DHCP specification.

Primary Address Pool

Subnet 192.168.100.0/24

Subnet Range 192.168.100.1 - 192.168.100.254

Address Pool Range 192.168.100.100 192.168.100.200
From To

The specified range for this pool must not be within the range configured on any other address pool for this interface.

Additional Pools [+ Add Address Pool](#)

If additional pools of addresses are needed inside of this subnet outside the above range, they may be specified here.

Server Options

WINS Servers WINS Server 1WINS Server 2DNS Servers 192.168.100.11.1.1.18.8.8.8

Other DHCP Options

Gateway 192.168.100.1

The default is to use the IP address of this firewall interface as the correct gateway for the network. Enter "none" for no gateway.

A partir de este momento se pasa a configurar cada uno de los dispositivos que irán contenidos en las distintas subredes.

Para implementar el **Honeypot** en la subred DMZ, se ha asignado una **dirección IP estática** a la máquina virtual que lo alberga, asegurándose de que esté fuera del rango definido para la subred:

pfSense
COMMUNITY EDITION

System ▾ Interfaces ▾ Firewall ▾ Services ▾ VPN ▾ Status ▾ Diagnostics ▾ Help ▾

↻

Status / DHCP Leases

↻ ⚙️ 📊 📄 ?

ISC DHCP has reached end-of-life and will be removed in a future version of pfSense. Visit [System > Advanced > Networking](#) to switch DHCP backend.

Search

Search Term

All ▾

🔍 Search

↺ Clear

Enter a search string or *nix regular expression to filter entries.

Leases

	IP Address	MAC Address	Hostname	Description	Start	End	Actions
👤 ↑	192.168.200.99	08:00:27:ad:25:87	kali		n/a	n/a	➕ ✎

Lease Utilization

Interface	Pool Start	Pool End	Used	Capacity	Utilization
No leases are in use					

➕ Show All Configured Leases

🗑️ Clear All DHCP Leases

A continuación, se ha configurado un **NAT Port Forwarding** para **redirigir el puerto 22 de la WAN hacia el Honeypot**, permitiendo así el acceso desde el exterior.

Firewall / NAT / Port Forward

?

The changes have been applied successfully. The firewall rules are now reloading in the background.
[Monitor](#) the filter reload progress.

Port Forward 1:1 Outbound NPT

Rules

<input type="checkbox"/>	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)	192.168.200.99	22 (SSH)	Honeypot SSH	✎ 📄 🗑️

↑ Add

↓ Add

🗑️ Delete

🔄 Toggle

💾 Save

➕ Separator

Legend





























▶








 Pass

⚙️

 Linked rule

Asimismo, se ha **reforzado la seguridad** de la subred DMZ, **aislándola de la red LAN y la red DMZ2**, permitiendo únicamente el envío de logs al SIEM a través de Internet, el DNS Resolver, la función de ping y la conexión SSH para el Honeypot.

Rules (Drag to Change Order)											
<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✗	0/0 B	IPv4 *	DMZ subnets	*	LAN subnets	*	*	none		   
<input type="checkbox"/>	✗	0/0 B	IPv4 *	DMZ subnets	*	DMZ2 subnets	*	*	none		   
<input type="checkbox"/>	✓	0/0 B	IPv4 TCP	DMZ subnets	*	*	Web	*	none		    
<input type="checkbox"/>	✓	0/0 B	IPv4 TCP/UDP	DMZ subnets	*	*	53 (DNS)	*	none		    
<input type="checkbox"/>	✓	0/0 B	IPv4 ICMP echoreq	*	*	*	*	*	none		    
<input type="checkbox"/>	✓	0/0 B	IPv4 TCP	*	*	192.168.200.99	22 (SSH)	*	none		    

 Add
  Add
  Delete
  Toggle
  Copy
  Save
  Separator

Se ha probado la conexión SSH al Honeypot Cowrie desde CMD en la **máquina host**, quedando los registros de los logs almacenados en el Honeypot.

```

C:\> Símbolo del sistema

Microsoft Windows [Versión 10.0.19045.5247]
(c) Microsoft Corporation. Todos los derechos reservados.









C:\Users\Adrián>ssh root@192.168.0.15
The authenticity of host '192.168.0.15 (192.168.0.15)' can't be established.
ED25519 key fingerprint is SHA256:AE3jD0kF6p7zYPf3b91cKpi/COXu3W1/kCA7bNOQbro.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.15' (ED25519) to the list of known hosts.
root@192.168.0.15's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@svr04:~# ls
root@svr04:~# timed out waiting for input: auto-logout
Connection to 192.168.0.15 closed by remote host.
Connection to 192.168.0.15 closed.
    
```

```
kali-linux-2024.3-virtualbox-amd64 (UTM Pfense configurado e instalado) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
1 2 3 4
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ docker logs honeypot-ssh
/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:105: CryptographyDeprecationWarning: TripleDES has been moved to cr
yptography.hazmat.decrepit.ciphers.algorithms.TripleDES and will be removed from cryptography.hazmat.primitives.ciphers.algorithms in 48.0.0.
b"3des-cbc": (algorithms.TripleDES, 24, modes.CBC),
/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:112: CryptographyDeprecationWarning: TripleDES has been moved to cr
yptography.hazmat.decrepit.ciphers.algorithms.TripleDES and will be removed from cryptography.hazmat.primitives.ciphers.algorithms in 48.0.0.
b"3des-ctr": (algorithms.TripleDES, 24, modes.CTR),
2025-01-18T14:38:12+0000 [-] Reading configuration from ['/cowrie/cowrie-git/etc/cowrie.cfg.dist']
2025-01-18T14:38:12+0000 [-] Python Version 3.11.2 (main, Sep 14 2024, 03:00:30) [GCC 12.2.0]
2025-01-18T14:38:12+0000 [-] Twisted Version 24.10.0
2025-01-18T14:38:12+0000 [-] Cowrie Version 2.6.1
2025-01-18T14:38:12+0000 [-] Loaded output engine: jsonlog
2025-01-18T14:38:12+0000 [twisted.scripts._twistd_unix.UnixAppLogger#info] twistd 24.10.0 (/cowrie/cowrie-env/bin/python3 3.11.2) starting up.
2025-01-18T14:38:12+0000 [twisted.scripts._twistd_unix.UnixAppLogger#info] reactor class: twisted.internet.epollreactor.EPollReactor.
2025-01-18T14:38:12+0000 [-] CowrieSSHFactory starting on 2222
2025-01-18T14:38:12+0000 [cowrie.ssh.factory.CowrieSSHFactory#info] Starting factory <cowrie.ssh.factory.CowrieSSHFactory object at 0x7f8d19148910
>
2025-01-18T14:38:12+0000 [-] Generating new RSA keypair...
2025-01-18T14:38:13+0000 [-] Generating new ECDSA keypair...
2025-01-18T14:38:13+0000 [-] Generating new ed25519 keypair...
2025-01-18T14:38:13+0000 [-] Ready to accept SSH connections
2025-01-18T14:41:07+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha1
2025-01-18T14:41:07+0000 [cowrie.ssh.factory.CowrieSSHFactory] No moduli, no diffie-hellman-group-exchange-sha256
2025-01-18T14:41:07+0000 [cowrie.ssh.factory.CowrieSSHFactory] New connection: 192.168.0.16:62494 (172.17.0.2:2222) [session: 1472d4f238ba]
2025-01-18T14:41:07+0000 [HoneyPotSSHTransport,0,192.168.0.16] Remote SSH version: SSH-2.0-OpenSSH_for_Windows_9.5
2025-01-18T14:41:07+0000 [HoneyPotSSHTransport,0,192.168.0.16] SSH client hassh fingerprint: 701158e75b508e76f0410d5d22ef9df0
2025-01-18T14:41:07+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] kex alg=b'curve25519-sha256' key alg=b'ssh-ed25519'
2025-01-18T14:41:07+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] outgoing: b'aes128-ctr' b'hmac-sha2-256' b'none'
2025-01-18T14:41:07+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] incoming: b'aes128-ctr' b'hmac-sha2-256' b'none'
2025-01-18T14:41:17+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] NEW KEYS
2025-01-18T14:41:17+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-userauth'
2025-01-18T14:41:17+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' trying auth b'none'
2025-01-18T14:41:37+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' trying auth b'password'
2025-01-18T14:41:37+0000 [HoneyPotSSHTransport,0,192.168.0.16] Could not read etc/userdb.txt, default database activated
2025-01-18T14:41:37+0000 [HoneyPotSSHTransport,0,192.168.0.16] login attempt [b'root'/b'pepe'] succeeded
2025-01-18T14:41:37+0000 [HoneyPotSSHTransport,0,192.168.0.16] Initialized emulated server as architecture: linux-x64-lsb
2025-01-18T14:41:37+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' authenticated with b'password'
2025-01-18T14:41:37+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-connection'
2025-01-18T14:41:37+0000 [cowrie.ssh.connection.CowrieSSHConnection#debug] got channel b'session' request
2025-01-18T14:41:37+0000 [cowrie.ssh.session.HoneyPotSSHSession#info] channel open
2025-01-18T14:41:37+0000 [cowrie.ssh.connection.CowrieSSHConnection#debug] got global b'no-more-sessions@openssh.com' request
2025-01-18T14:41:37+0000 [twisted.conch.ssh.session#info] Handling pty request: b'xterm-256color' (30, 120, 640, 480)
2025-01-18T14:41:37+0000 [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotSSHTransport,0,192.168.0.16] Terminal Size: 120 30
2025-01-18T14:41:37+0000 [twisted.conch.ssh.session#info] Getting shell
2025-01-18T14:41:48+0000 [HoneyPotSSHTransport,0,192.168.0.16] CMD: ls
2025-01-18T14:41:48+0000 [HoneyPotSSHTransport,0,192.168.0.16] Command found: ls
(kali@kali)-[~]
$
```

























Una vez completado este paso, se ha asignado una nueva IP estática a la máquina que aloja **Suricata** en la subred DMZ2.

Leases							
	IP Address	MAC Address	Hostname	Description	Start	End	Actions
 	192.168.250.99	08:00:27:b7:70:14	kali2		n/a	n/a	 
 	192.168.200.99	08:00:27:ad:25:87	kali		n/a	n/a	 

Habiendo hecho esto, se han **definido las reglas** de la DMZ2 para **aislarla nuevamente** de la LAN y la DMZ, garantizando su funcionamiento seguro y evitando el tráfico no autorizado.

Floating WAN LAN DMZ DMZ2

Rules (Drag to Change Order)

<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✗ 0/0 B	IPv4 *	DMZ2 subnets	*	DMZ subnets	*	*	none			   
<input type="checkbox"/>	✗ 0/0 B	IPv4 *	DMZ2 subnets	*	LAN subnets	*	*	none			   
<input type="checkbox"/>	✓ 11/408 KiB	IPv4 TCP/UDP	DMZ2 subnets	*	*	Web	*	none		Regla trafico web	   
<input type="checkbox"/>	✓ 9/139 KiB	IPv4 UDP	*	*	*	53 (DNS)	*	none			   
<input type="checkbox"/>	✓ 0/0 B	IPv4 ICMP echoreq	*	*	*	*	*	none			   
<input type="checkbox"/>	✗ 0/0 B	IPv4 *	DMZ2 subnets	*	*	*	*	none			   

 Add  Add  Delete  Toggle  Copy  Save  Separator

A continuación, se ha instalado Suricata en Kali y se han configurado **reglas de monitorización para IDS**, enfocándose en el **tráfico web** y generando una alerta para la **descarga de archivos PDF**.

```
GNU nano 8.3 classification.config
#
# config classification:shortname,short description,priority
#
config classification: not-suspicious,Not Suspicious Traffic,3
config classification: unknown,Unknown Traffic,3
config classification: bad-unknown,Potentially Bad Traffic, 2
config classification: attempted-recon,Attempted Information Leak,2
config classification: successful-recon-limited,Information Leak,2
config classification: successful-recon-largescale,Large Scale Information Leak,2
config classification: attempted-dos,Attempted Denial of Service,2
config classification: successful-dos,Denial of Service,2
config classification: attempted-user,Attempted User Privilege Gain,1
config classification: unsuccessful-user,Unsuccessful User Privilege Gain,1
config classification: successful-user,Successful User Privilege Gain,1
config classification: attempted-admin,Attempted Administrator Privilege Gain,1
config classification: successful-admin,Successful Administrator Privilege Gain,1

# NEW CLASSIFICATIONS
config classification: rpc-portmap-decode,Decode of an RPC Query,2
config classification: shellcode-detect,Executable code was detected,1
config classification: string-detect,A suspicious string was detected,3
config classification: suspicious-filename-detect,A suspicious filename was detected,2
config classification: suspicious-login,An attempted login using a suspicious username was detected,2
config classification: system-call-detect,A system call was detected,2
config classification: tcp-connection,A TCP connection was detected,4
config classification: trojan-activity,A Network Trojan was detected, 1
config classification: unusual-client-port-connection,A client was using an unusual port,2
config classification: network-scan,Detection of a Network Scan,3
config classification: denial-of-service,Detection of a Denial of Service Attack,2
config classification: non-standard-protocol,Detection of a non-standard protocol or event,2
config classification: protocol-command-decode,Generic Protocol Command Decode,3
config classification: web-application-activity,access to a potentially vulnerable web application,2
config classification: web-application-attack,Web Application Attack,1
config classification: misc-activity,Misc activity,3
config classification: misc-attack,Misc Attack,2
config classification: icmp-event,Generic ICMP event,3
config classification: inappropriate-content,Inappropriate Content was Detected,1
config classification: policy-violation,Potential Corporate Privacy Violation,1
config classification: default-login-attempt,Attempt to login by a default username and password,2

# Update
config classification: targeted-activity,Targeted Malicious Activity was Detected,1
config classification: exploit-kit,Exploit Kit Activity Detected,1
config classification: external-ip-check,Device Retrieving External IP Address Detected,2
config classification: domain-c2,Domain Observed Used for C2 Detected,1
config classification: pup-activity,Possibly Unwanted Program Detected,2
config classification: credential-theft,Successful Credential Theft Detected,1
config classification: social-engineering,Possible Social Engineering Attempted,2
config classification: coin-mining,Crypto Currency Mining Activity Detected,2
config classification: command-and-control,Malware Command and Control Activity Detected,1

#Clasificaciones propias
config classification: file-download,Descarga de archivo detectado,2
```



```

GNU nano 8.3 suricata.rules *
alert tcp any any -> any any (msg:"trafico detectado"; sid:1;)
alert tcp any any -> any any (msg:"PDF Archivo descargado"; flow:established,to_client; fileext:"pdf"; sid:3; classtype:file-download;)

```

Esto permite generar logs que facilitan el estudio y la monitorización del tráfico web, así como su tipo o direcciones IP de origen y destino:

```

(root@kali)-[/var/log/suricata]
# tail -f fast.log
01/18/2025-13:56:14.622589  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.250.99:54476 -> 34.155.97.253
:443
01/18/2025-13:56:14.647839  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 34.155.97.253:443 -> 192.168.250.99:5
4476
01/18/2025-13:56:24.624753  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.250.99:59300 -> 34.155.97.253
:443
01/18/2025-13:56:24.649899  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 34.155.97.253:443 -> 192.168.250.99:5
9300
01/18/2025-13:56:34.644933  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.250.99:34426 -> 34.155.97.253
:443
01/18/2025-13:56:34.671191  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 34.155.97.253:443 -> 192.168.250.99:3
4426
01/18/2025-13:56:56.340191  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.250.99:59618 -> 34.155.97.253
:443
01/18/2025-13:56:56.365877  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 34.155.97.253:443 -> 192.168.250.99:5
9618
01/18/2025-13:56:59.978919  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.250.99:43218 -> 34.155.97.253
:443
01/18/2025-13:56:59.995301  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 34.155.97.253:443 -> 192.168.250.99:4
3218
01/18/2025-13:57:06.317354  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.250.99:53710 -> 34.155.97.253
:443
01/18/2025-13:57:06.342744  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 34.155.97.253:443 -> 192.168.250.99:5
3710
01/18/2025-13:57:15.997248  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.250.99:59848 -> 18.67.240.77:
443
01/18/2025-13:57:15.999185  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.250.99:33706 -> 151.101.133.5
0:443
01/18/2025-13:57:15.999690  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.250.99:53274 -> 18.67.240.20:
443
01/18/2025-13:57:16.001924  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.250.99:49540 -> 216.58.209.78
:443
01/18/2025-13:57:16.006840  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 18.67.240.77:443 -> 192.168.250.99:59
848
01/18/2025-13:57:16.007873  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 151.101.133.50:443 -> 192.168.250.99:
33706
01/18/2025-13:57:16.008978  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 18.67.240.20:443 -> 192.168.250.99:53
274
01/18/2025-13:57:16.011927  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 216.58.209.78:443 -> 192.168.250.99:4
9540
01/18/2025-13:57:16.052862  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.250.99:38902 -> 108.157.109.5
3:443
01/18/2025-13:57:16.061757  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 108.157.109.53:443 -> 192.168.250.99:
38902
01/18/2025-13:57:16.177002  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.250.99:55880 -> 172.217.18.20
2:443
01/18/2025-13:57:16.201045  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 172.217.18.202:443 -> 192.168.250.99:
55880
01/18/2025-13:57:16.286452  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.250.99:36988 -> 216.239.34.36
:443
01/18/2025-13:57:16.287260  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.250.99:37760 -> 142.250.185.3
:443
01/18/2025-13:57:16.292029  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.250.99:52008 -> 216.239.32.3:
443
01/18/2025-13:57:16.294308  [**] [1:1:0] trafico detectado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.250.99:58450 -> 142.250.200.9

```

Una vez instalado y configurado el Honeypot en la DMZ y Suricata en la DMZ2, **se procede a crear una máquina virtual con Windows en la subred LAN**, la cual estará aislada de ambas subredes, pero podrá acceder a ellas. (La DMZ y DMZ2 no tienen acceso a LAN, pero LAN puede acceder a ellas)

Floating WAN <u>LAN</u> DMZ DMZ2											
Rules (Drag to Change Order)											
<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	0/0 B	*	*	*	LAN Address	443 80	*	*		Anti-Lockout Rule	
<input type="checkbox"/>	0/0 B	IPv4 TCP	DMZ2 subnets	*	LAN subnets	*	*	none			
<input type="checkbox"/>	0/0 B	IPv4 TCP	DMZ subnets	*	LAN subnets	*	*	none			
<input type="checkbox"/>	49/177.39 MiB	IPv4 *	LAN subnets	*	*	*	*	none		Default allow LAN to any rule	
<input type="checkbox"/>	0/0 B	IPv6 *	LAN subnets	*	*	*	*	none		Default allow LAN IPv6 to any rule	

File Actions Edit View Help

```

(kali@kali)~$ ping 192.168.1.102
PING 192.168.1.102 (192.168.1.102) 56(84) bytes of data:

```

Firewall / Rules / LAN

Símbolo del sistema

```

C:\Users\vboxuser>ping 192.168.200.99

Haciendo ping a 192.168.200.99 con 32 bytes de datos:
Respuesta desde 192.168.200.99: bytes=32 tiempo<1m TTL=63
Respuesta desde 192.168.200.99: bytes=32 tiempo<1m TTL=63
Respuesta desde 192.168.200.99: bytes=32 tiempo<1m TTL=63
Respuesta desde 192.168.200.99: bytes=32 tiempo<1m TTL=63

Estadísticas de ping para 192.168.200.99:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms

```

Como se muestra en la imagen, la máquina alojada en la DMZ no puede realizar un ping a la máquina en LAN, pero sí ocurre a la inversa.

Con la infraestructura de red montada y todos los activos preparados, se procede a crear una cuenta en **Elastic Cloud** que se utilizará como **SIEM**. A continuación, se agregan las **políticas de agente** (*Agent Policies*) a todos los equipos configurados, lo que permite **recopilar los logs generados** por las distintas máquinas (sistema operativo, aplicaciones instaladas, etc.).

En este caso, se comienza con Suricata, instalando su agente en la máquina virtual que aloja la aplicación:

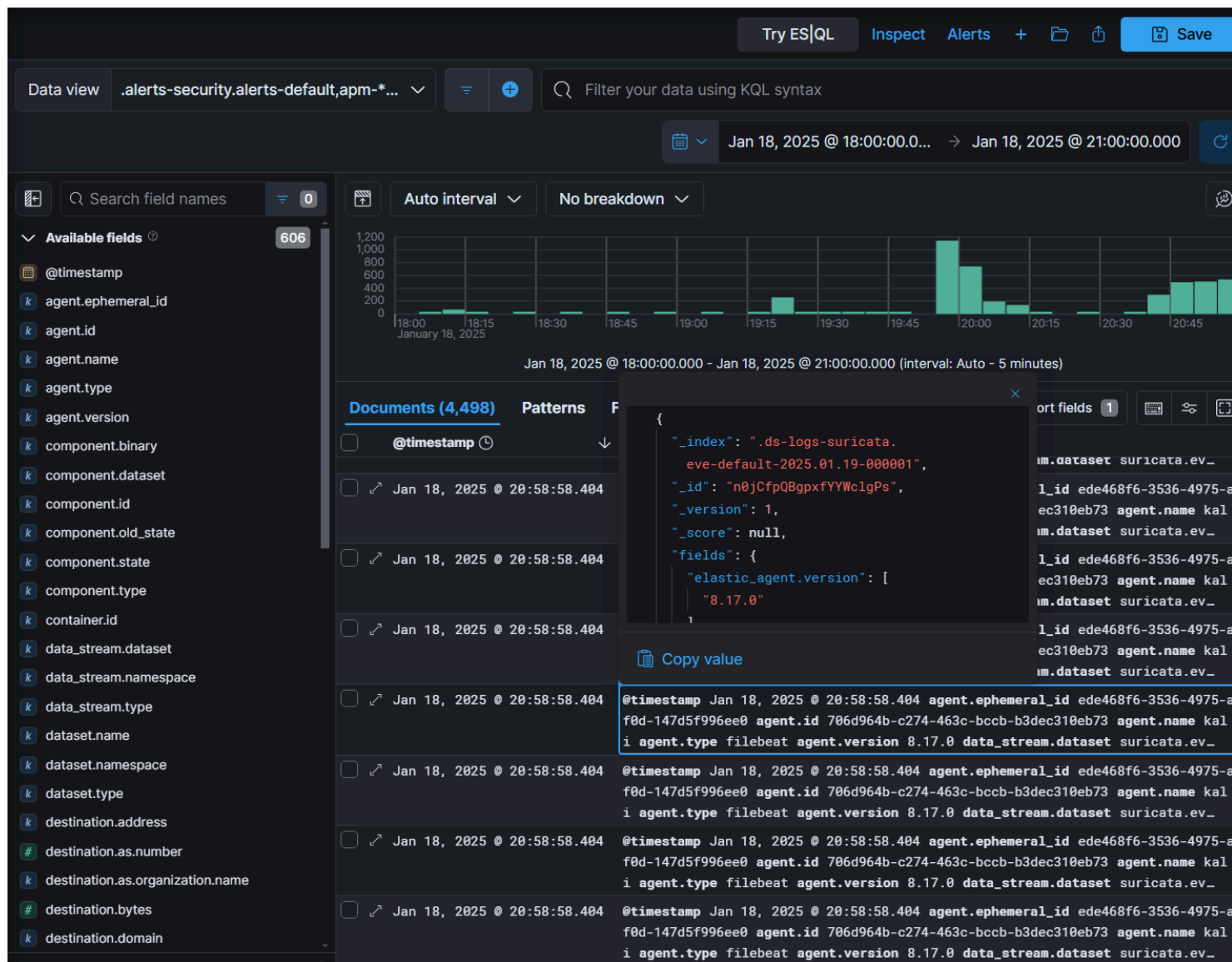
The screenshot shows the Elastic Security console interface. On the left is a sidebar with navigation options like Discover, Dashboards, Rules, Alerts, Attack discovery, Findings, Cases, Investigations, Intelligence, Explore, Assets, and Machine learning. The main content area is titled 'Suricata' and includes a 'Back to integrations' link. It displays the Suricata logo, version 2.21.4, and 0 agent policies. Below this, there are tabs for Overview, Integration policies, Assets, Settings, Configs, and API reference. The 'Overview' tab is active, showing 'Suricata Integration' and 'Compatibility' sections. A 'Screenshots' section on the right shows a preview of the Suricata interface. At the bottom, there is a 'Details' section.

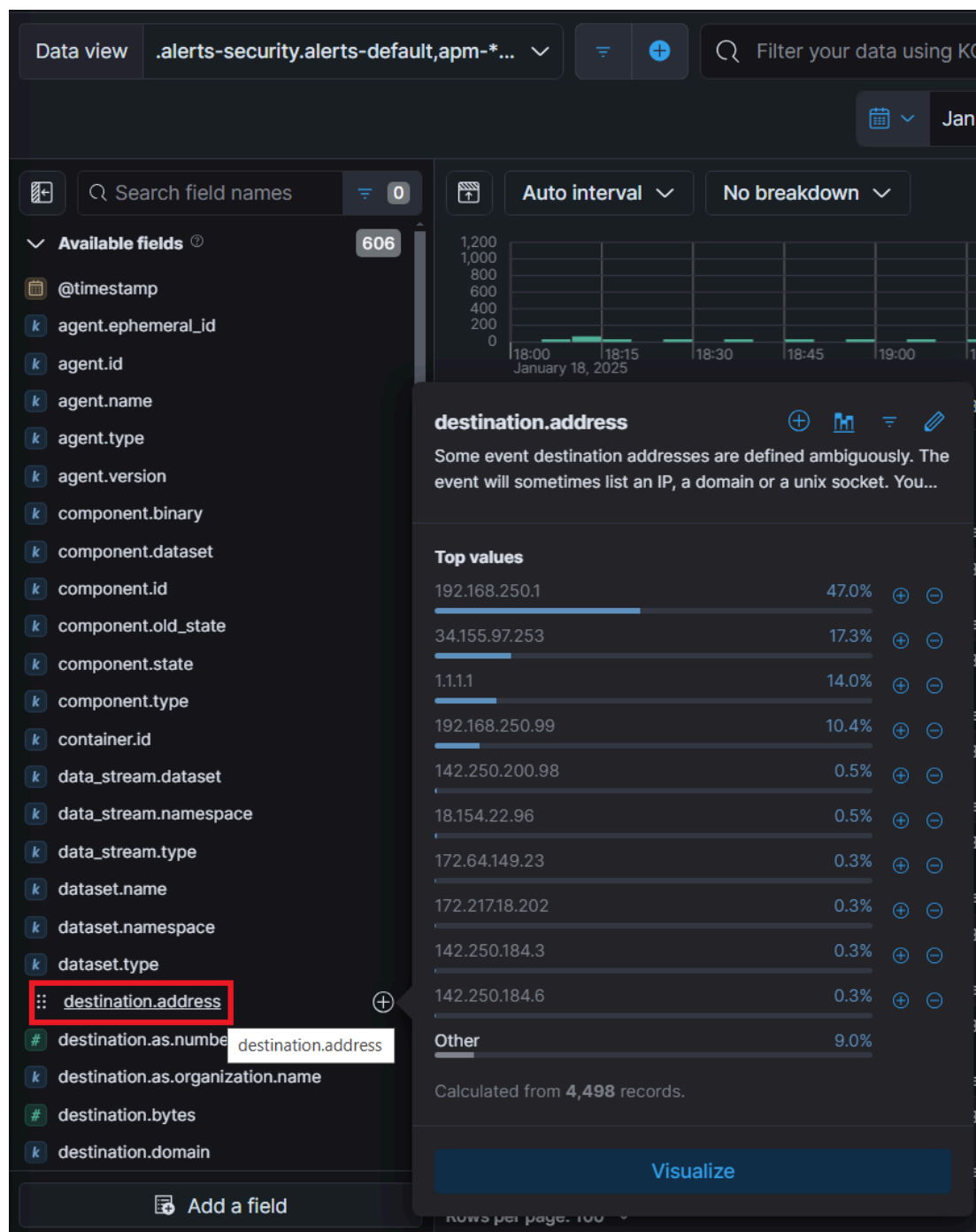
This screenshot shows a confirmation message in the Elastic Security console. It features a green checkmark icon and the text 'Agent enrollment confirmed'. Below this, it states '✓ 1 agent has been enrolled.' and provides a button labeled 'View enrolled agents'.

The screenshot displays the Elastic Fleet console, which is used for centralized management of Elastic Agents. The top navigation bar includes links for Agents, Agent policies, Enrollment tokens, Uninstall tokens, Data streams, and Settings. The 'Agents' tab is selected. Below the navigation bar, there are filters for 'fleet-agents.policy_id' and 'Status' (5 agents). A table lists the agents, showing columns for Status, Host, Agent policy, CPU, Memory, Last activity, and Version. One agent is listed: 'kali' with status 'Healthy', agent policy 'Suricata - Linux rev. 6', CPU usage of 1.07%, memory usage of 250 MB, last activity 36 seconds ago, and version 8.17.0. The bottom of the page shows 'Rows per page: 20' and pagination controls.

De esta manera, los **logs estarían disponibles** combinando los de Suricata, los del sistema, y los de Elastic.

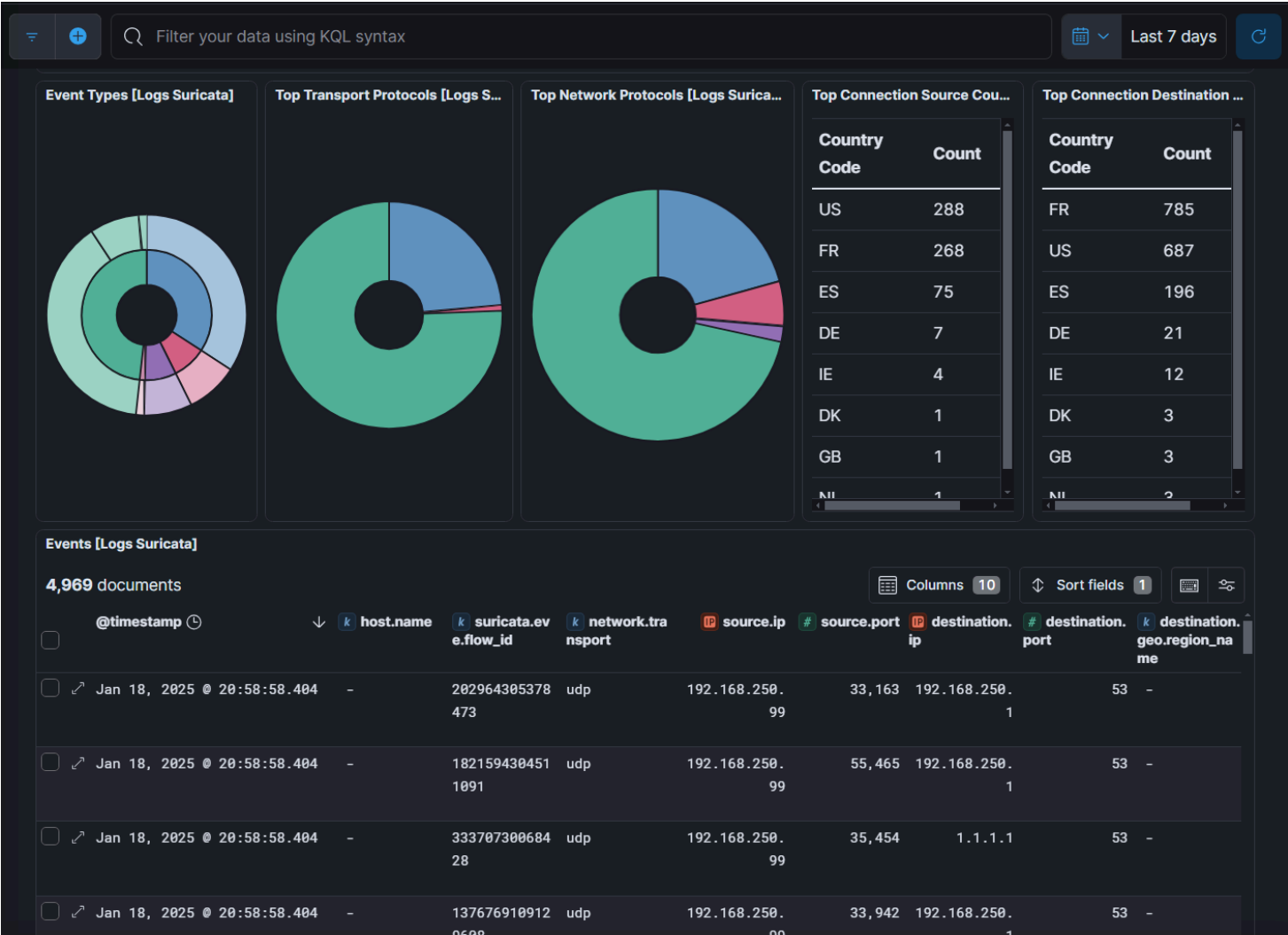
Mediante la búsqueda por franjas de tiempo y observando los picos de tráfico, se puede **filtrar** para localizar los logs específicos de Suricata. Además, utilizando las reglas en la columna izquierda, es posible aplicar filtros adicionales por distintos campos y buscar **valores concretos**, como la dirección de destino (*destination address*):





Con **queries de búsqueda más largas** se podría acotar la búsqueda todavía más y buscar **entradas más específicas**.


Asimismo, es posible visualizar los eventos de Suricata mediante **gráficos y diagramas** que facilitan su monitorización, y se pueden añadir gráficos **personalizados** además de los preconfigurados por defecto.



Una vez configurado Suricata, se procede a preparar el agente del Honeypot con una **Agent Policy de Custom Logs**. Al no tratarse de una política predeterminada, es necesario **especificar manualmente la ruta al archivo** que almacena los logs del Honeypot.

En la máquina Kali que aloja Cowrie, se ha creado un archivo específico para los logs, desde el cual Elastic podrá extraer la información. Para ello, se ha utilizado el siguiente comando:

```
docker logs -f a0ae3311d999 >> /home/kali/cowrie-logs/logs.log
```



Edit Custom Logs integration

Modify integration settings and deploy changes to the selected agent policy.

1

Configure integration

Integration settings

Choose a name and description to help identify how this integration will be used.

Integration name

Honeypot Cowrie

Description Optional

> Advanced options

☒ Custom log file

Change defaults ^

Log file path

/home/kali/cowrie-logs/logs.log

+ Add row

Path to log files to be collected

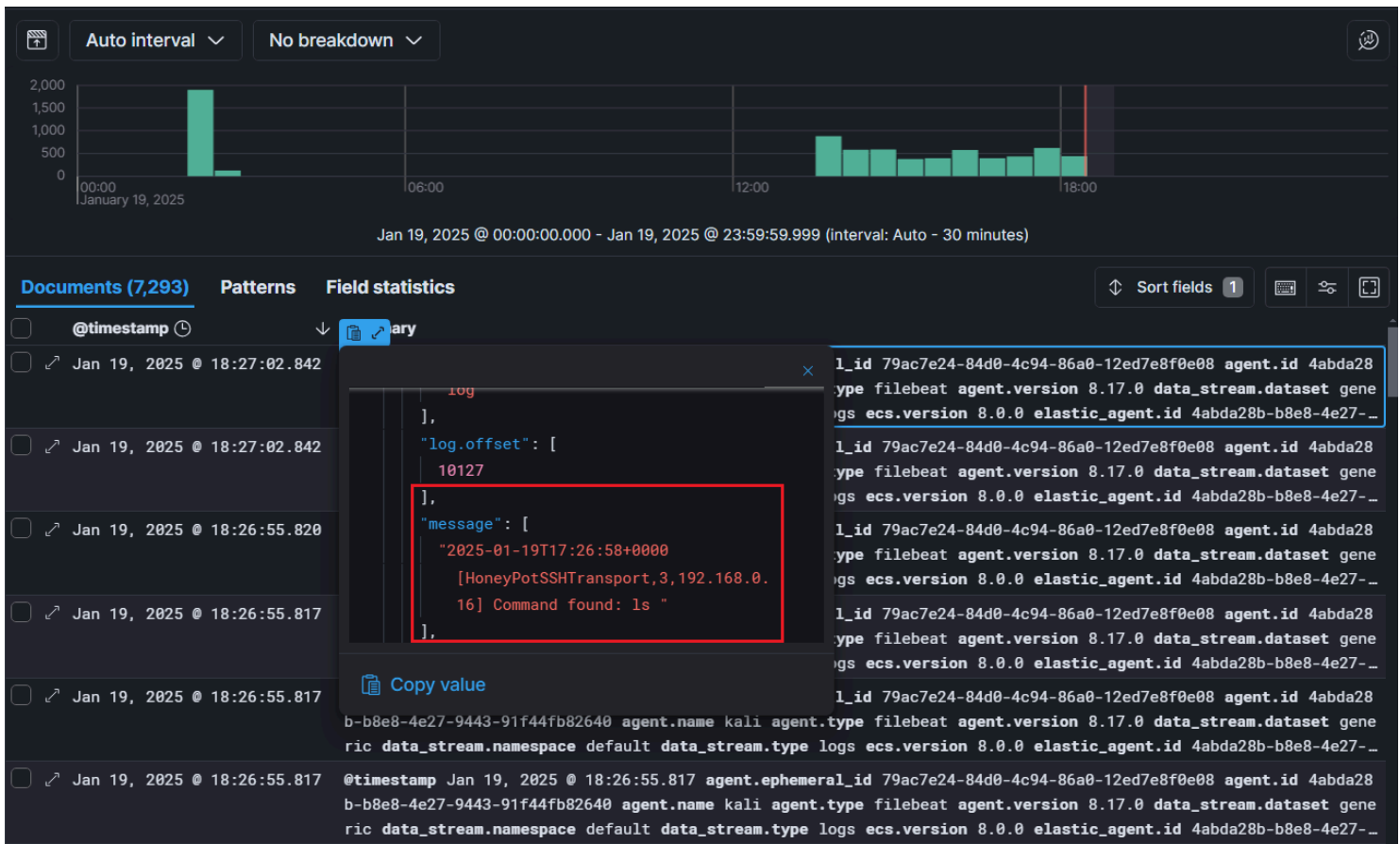
Dataset name

generic

Set the name for your dataset. Changing the dataset will send the data to a different index. You can't use - in the name of a dataset and only valid characters for Elasticsearch index names.

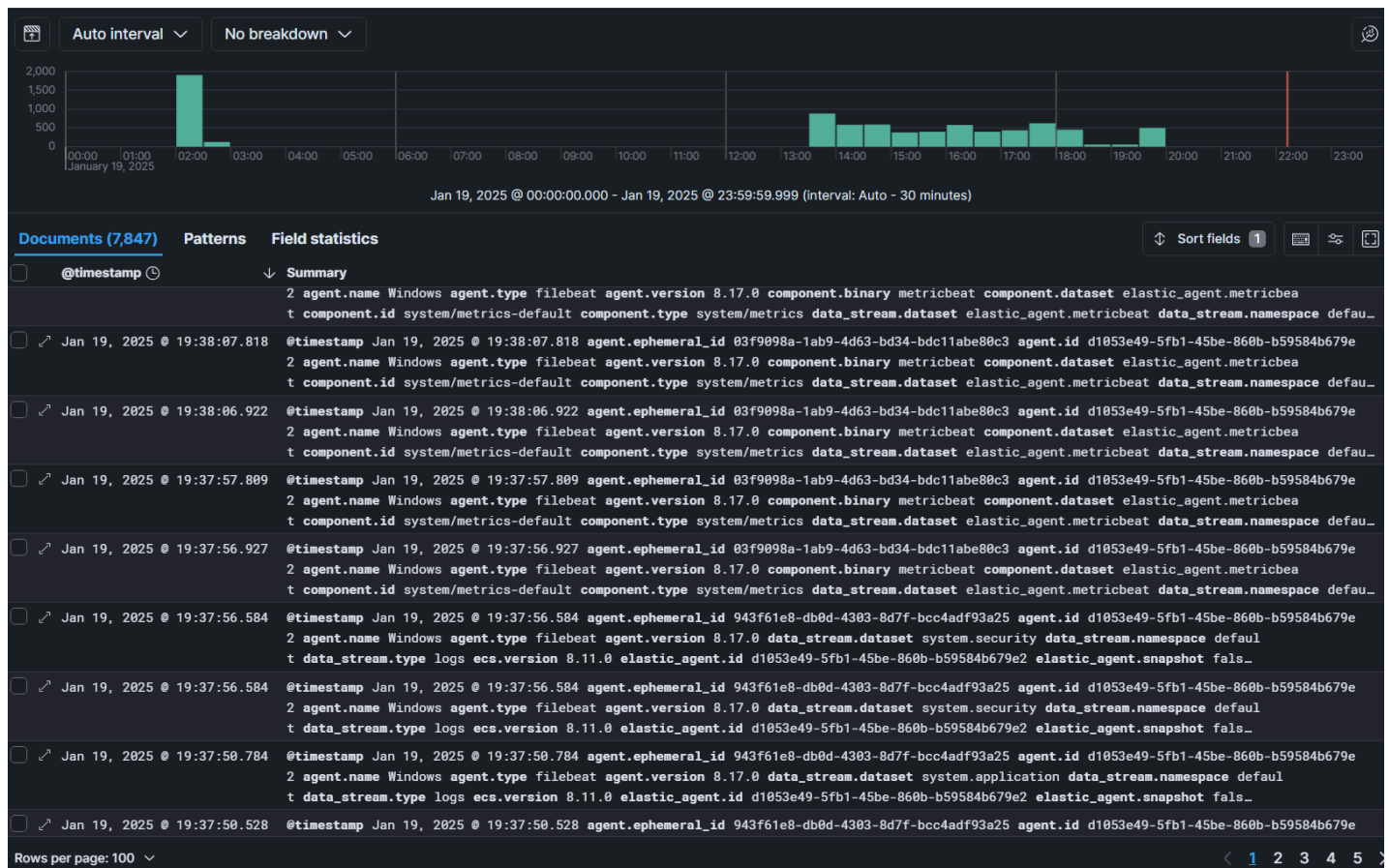
> Advanced options

Los logs del Honeypot **no tienen ninguna expresión regular asociada**, por lo que no cuentan con campos estructurados y se presentan en formato crudo. Sería recomendable realizar un **parsing** para facilitar la búsqueda de datos.



```
2025-01-19T17:26:55+0000 [cowrie.ssh.connection.CowrieSSHConnection#debug] got global b'no-more-sessions@openssh.com' request
2025-01-19T17:26:55+0000 [twisted.conch.ssh.session#info] Handling pty request: b'xterm-256color' (30, 120, 640, 480)
2025-01-19T17:26:55+0000 [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotSSHTransport,3,192.168.0.16] Terminal Size: 120 30
2025-01-19T17:26:55+0000 [twisted.conch.ssh.session#info] Getting shell
2025-01-19T17:26:58+0000 [HoneyPotSSHTransport,3,192.168.0.16] CMD: ls
2025-01-19T17:26:58+0000 [HoneyPotSSHTransport,3,192.168.0.16] Command found: ls
```


Finalmente, se ha añadido el agente de Elastic en **Windows** para registrar logs del sistema, aplicaciones, eventos de seguridad y otros datos relevantes:



Y de esta forma es como se ha configurado la infraestructura de red para ser controlada por el SIEM, quedando la flota resultante de esta manera:

Fleet

Centralized management for Elastic Agents.

[Agents](#) [Agent policies](#) [Enrollment tokens](#) [Uninstall tokens](#) [Data streams](#) [Settings](#)

[Ingest Overview Metrics](#) [Agent Info Metrics](#) [Agent activity](#) [Add Fleet Server](#) [Add agent](#)

Filter your data using KQL syntax Status 4 Tags 0 Agent policy 4 Upgrade available

Showing 4 agents [Clear filters](#) Healthy 4 Unhealthy 0 Updating 0 Offline 0 Inactive 0 Unenrolled 0

Status	Host	Agent policy	CPU	Memory	Last activity	Version	Actions
Healthy	kali	Honeybot SSH - Linux rev. 3	1.20 %	217 MB	14 seconds ago	8.17.0	...
Healthy	kali	Suricata - Linux rev. 6	1.13 %	253 MB	31 seconds ago	8.17.0	...
Healthy	Windows	Windows rev. 2	N/A	143 MB	17 seconds ago	8.17.0	...
Healthy	301f124b867b	Elastic Cloud agent policy rev. 5	N/A	N/A	25 seconds ago	8.17.0	...

Junto a este PDF se adjuntan los archivos .txt pertenecientes a un log de Suricata, Honeypot y Windows respectivamente, extraídos del SIEM.