

Práctica de blue team

Realizado por Yesurún González Román

Contacto: yesurn_g@hotmail.com

Administración y securización de redes

Indice

1)Objetivos

2) Configuración Pfsense

2.1) Interfaz web

2.2) Aplicando reglas

3) Creación Honeypots

4)Creación ELK

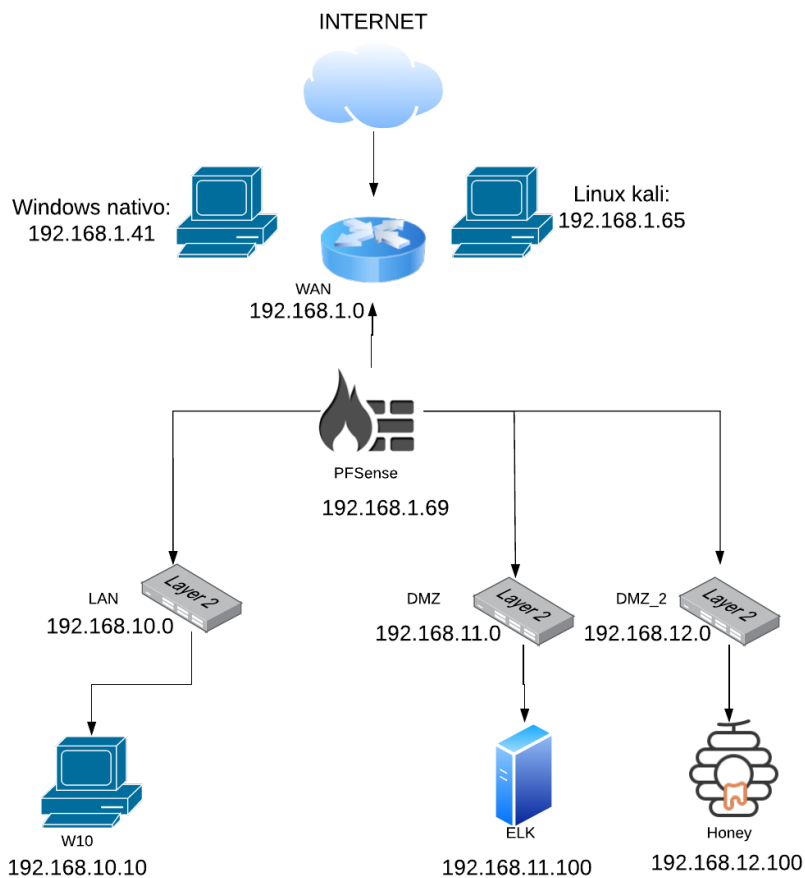
5) Realización de pruebas

1)Objetivos

En esta práctica debo cumplir los siguientes objetivos:

- Creación de un PfSense en bridge que conecte 3 redes, LAN, DMZ y DMZ_2 estas como red interna.
- Un equipo W10 en LAN, un stack ELK en DMZ y un grupo de honeypots en DMZ_2.
- Transmitir los logs de los honeypots al ELK stack, pero los honeypots no deben tener acceso a las otras redes(solo para transmitir logs), aunque los honeypots deben ser accesible desde la red WAN.
- El servidor ELK debe almacenar y poder visualizar los diferentes logs de los honeypots.
- El W10 debe poder conectarse a ELK vía Kibana.

2) Configuración Pfsense



Comenzamos creando una máquina virtual en VirtualBox que nombramos UTM (en esta máquina creamos el Pfsense).

La máquina debe crearse con sistema operativo BSD con versión Free BSD

Posteriormente añadimos la iso de Pfsense como secundario y CD en vivo.

Creamos 4 adaptadores de red con las siguientes características:

- Adaptador 1: modo puente
- Adaptador 2: red local interna con nombre LAN
- Adaptador 3: red local interna con nombre DMZ
- Adaptador 4: red local interna con nombre DMZ_2

Arrancamos la máquina para iniciar instalación. Seleccionamos el tipo de idioma de teclado deseado y reiniciamos el sistema con la iso de Pfsense desactivada.

Pfsense detecta los cuatro adaptadores que hemos creado y nos pregunta a que interfaces tuyas predeterminadas queremos asignar. También nos pregunta si queremos crear Vlans y respondemos que no.

- Si la mac de em0 concuerda con la mac del adaptador 1, asignamos éste a la red wan de pfsense.
- Si la mac de em1 concuerda con la mac del adaptador 2, asignamos éste a la red lan de pfsense.
- Si la mac de em2 concuerda con la mac del adaptador 3, asignamos éste a la red opt1 de pfsense.
- Si la mac de em3 concuerda con la mac del adaptador 4, asignamos éste a la red opt2 de pfsense.

2.1) Interfaz Web

Como en mi caso particular, el servidor DHCP de la red wan me genera un rango IP 192.168.1.69/24, me puede ocasionar colisiones con el rango de red lan 192.168.1.1/24

Por lo tanto vamos a acceder a la aplicación web de Pfsense en 192.168.1.69 en vez de 192.168.1.1, y así cambiar la red lan posteriormente sin problemas.

Para poder hacer esto tenemos que crear otra máquina virtual con la OVA de VirtualBos de Windows 10 en red interna LAN.

Una vez dentro de la dirección 192.168.1.69, Pfsense nos pide que realicemos diversos pasos, y los realizaremos de la siguiente manera:

General Information

On this screen the general pfSense parameters will be set.

Hostname

Firewall

EXAMPLE: myserver

Domain

blueteam.local

EXAMPLE: mydomain.com

The default behavior of the DNS Resolver will ignore manually configured DNS servers for client queries and query root DNS servers directly. To use the manually configured DNS servers below for client queries, visit Services > DNS Resolver and enable DNS Query Forwarding after completing the wizard.

Primary DNS Server

1.1.1.1

Secondary DNS Server

8.8.8.8

Override DNS

☒

Allow DNS servers to be overridden by DHCP/PPP on WAN

Time Server Information

Please enter the time, date and time zone.

Time server hostname

2.pfsense.pool.ntp.org

Enter the hostname (FQDN) of the time server.

Timezone

Europe/Madrid

RFC1918 Networks

Block RFC1918 Private Networks

☐ Block private networks from entering via WAN

When set, this option blocks traffic from IP addresses that are reserved for private networks as per RFC 1918 (10/8, 172.16/12, 192.168/16) as well as loopback addresses (127/8). This option should generally be left turned on, unless the WAN network lies in such a private address space, too.

Block bogon networks

Block bogon networks

☐ Block non-Internet routed networks from entering via WAN

When set, this option blocks traffic from IP addresses that are reserved (but not RFC 1918) or not yet assigned by IANA. Bogons are prefixes that should never appear in the Internet routing table, and obviously should not appear as the source address in any packets received.

Ahora en la sección de rango ip lan ponemos 192.168.10.1/24.

Una vez hechos estos pasos tenemos acceso a todos los tipos de configuración de Pfsense.

Nos dirigimos a la pestaña de asignación de interfaces y habilitamos las interfaces OPT1 y OPT2 y les cambiamos los nombres a DMZ y DMZ_2 respectivamente.

Interfaces / Interface Assignments Link ?

Interface Assignments Interface Groups Wireless VLANs QinQs PPPs GREs GIFs Bridges LAGGs

Interface	Network port	
WAN	em0 (08:00:27:f4:7e:58)	
LAN	em1 (08:00:27:49:46:b8)	Delete
DMZ	em2 (08:00:27:37:34:76)	Delete
DMZ_2	em3 (08:00:27:03:38:67)	Delete

Save

Entramos en las interfaces LAN, DMZ y DMZ_2 y les asignamos static ipv4. En el apartado IPV6 de DMZ y DMZ_2 seleccionamos "NONE".

Ahora debemos dirigirnos a la pestaña servicios -> servidor DHCP. Una vez aquí, habilitamos los servicios en DMZ y DMZ_2 (en LAN ya está activado por defecto).

LAN DMZ DMZ_2

General Options

Enable ☒ Enable DHCP server on LAN interface

BOOTP ☐ Ignore BOOTP queries

Deny 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 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 below (i.e. for this interface) will get an IP address within this scope/range.

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

Ignore client identifiers ☐ If a client includes a unique identifier in its DHCP request, that UID will not be recorded in its lease.
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.

Subnet 192.168.10.0

Subnet mask 255.255.255.0

Available range 192.168.10.1 - 192.168.10.254








Range 192.168.10.10 192.168.10.245

From To

Adjudicamos un rango, por ejemplo, 192.168.11.100 - 192.168.11.200 en DMZ y 192.168.12.100 - 192.168.12.200 en DMZ_2.

Más abajo ponemos servidor DNS y gateway 192.168.x.1 en LAN, DMZ y DMZ_2 (la "x" corresponde al número correspondiente de cada interfaz). Podemos añadir DNS secundarios con 1.1.1.1 y 8.8.8.8.

Ya tenemos las interfaces creadas, dando como resultado la siguiente imagen:

Interfaces   			
 WAN	↑	1000baseT <full-duplex>	192.168.1.69
 LAN	↑	1000baseT <full-duplex>	192.168.10.1
 DMZ	↑	1000baseT <full-duplex>	192.168.11.1
 DMZ_2	↑	1000baseT <full-duplex>	192.168.12.1

2.2) Aplicando reglas

Por último, nos adentramos en la pestaña firewall -> NAT -> Port Forward y aplicamos las reglas que redirijan tráfico hacia los honeypots en DMZ_2, dando como resultado la siguiente imagen:

Firewall / NAT / Port Forward













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"/>	<input checked="" type="checkbox"/>	 WAN	TCP	*	*	WAN address	10000	192.168.12.100	10001	Gaspot honeypot	  
<input type="checkbox"/>	<input checked="" type="checkbox"/>	 WAN	TCP	*	*	WAN address	22 (SSH)	192.168.12.100	2222	cowrie honeypot	  
<input type="checkbox"/>	<input checked="" type="checkbox"/>	 WAN	TCP	*	*	WAN address	3389 (MS RDP)	192.168.12.100	333	rdp honeypot	  

Ahora nos dirigimos a Firewall -> Rules y aplicamos las reglas en cada interfaz, dando como resultado las siguientes imágenes:

Firewall / Rules / WAN

Floating WAN LAN DMZ DMZ_2

Rules (Drag to Change Order)

<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0 / 24 KiB	IPv4 TCP	*	*	192.168.12.100	2222	*	none		NAT cowrie honeypot	
<input type="checkbox"/>	✓ 0 / 61 KiB	IPv4 TCP	*	*	192.168.12.100	333	*	none		NAT rdp honeypot	
<input type="checkbox"/>	✓ 0 / 1 KiB	IPv4 TCP	*	*	192.168.12.100	10001	*	none		NAT Gaspot honeypot	

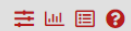
Firewall / Rules / LAN

Floating WAN LAN DMZ DMZ_2

Rules (Drag to Change Order)

<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	✓ 6 / 522 KiB	*	*	*	LAN Address	443 80	*	*		Anti-Lockout Rule	
<input type="checkbox"/>	✗ 0 / 0 B	IPv4 *	*	*	DMZ_2 net	*	*	none		Not To DMZ_2	
<input type="checkbox"/>	✓ 0 / 780 B	IPv4 TCP	LAN net	*	192.168.11.100	5601	*	none		LAN To Kibana	
<input type="checkbox"/>	✗ 0 / 0 B	IPv4 *	LAN net	*	192.168.11.100	*	*	none		LAN NOT TO DMZ	
<input type="checkbox"/>	✓ 0 / 4.28 MiB	IPv4 *	LAN net	*	*	*	*	none		Default allow LAN to any rule	
<input type="checkbox"/>	✓ 0 / 0 B	IPv6 *	LAN net	*	*	*	*	none		Default allow LAN IPv6 to any rule	

Firewall / Rules / DMZ

Floating WAN LAN DMZ DMZ_2

Rules (Drag to Change Order)

<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
No rules are currently defined for this interface All incoming connections on this interface will be blocked until pass rules are added. Click the button to add a new rule.											
Add Add Delete Save Separator											

Firewall / Rules / DMZ_2

Floating WAN LAN DMZ DMZ_2

Rules (Drag to Change Order)

<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0 / 41.56 MiB	IPv4 TCP/UDP	DMZ_2 net	*	192.168.11.100	9200	*	none		honeypot to elk	

3) Creación honeypots

Para crear los honeypots he creado otra máquina virtual con sistema operativo Linux con Kali.

En este Kali decido crear tres honeypots con los siguientes comandos en terminal (en modo puente para poder descargar de internet).

Para dos honeypots es necesario tener instalado docker:

- `curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -`
- `sudo apt-get update`
- `sudo apt-get install docker.io`

GASPOT:

- `git clone https://github.com/sjhilt/GasPot.git`
- `cd gaspot`
- `cp config.ini.dist config.ini`
- `python GasPot.py`

COWRIE:

- `Git clone https://github.com/cowrie/docker-cowrie.git`
- `docker run -p 2222:2222 cowrie/cowrie`

RDP:

- `docker pull amazedostrich/rdpy`
- `docker run -p 333:3389 amazedostrich/rdpy`

Una vez levantados los honeypots, instalamos el NIDS Suricata para tener un control a nivel de red de qué paquetes se llegan a recibir en los puertos abiertos de los honeypots. Realizamos los siguientes comandos:

- `Apt install suricata`
- `Cd /etc/suricata/rules`
- `Touch logs.rules`
- `Nano logs.rules` (añadimos las siguientes reglas con la ip de los honeypots)

```
alert tcp any any → 192.168.12.100 2222 (msg: "COWRIE"; sid:1; priority:1;)
alert tcp any any → 192.168.12.100 333 (msg: "RDP"; sid:2; priority:1;)
alert tcp any any → 192.168.12.100 10001 (msg: "GASPOT"; sid:3; priority:1;)
```

- `CD ..`
- `Sudo nano suricata.yaml` (Añadimos el fichero de reglas logs.rules en la sección default-rule-path)

```
667 wc -l host_scan
##668 nikto -h 146.20.26.85
## Configure Suricata to load Suricata-Update managed rules.
##670 docker ps
671 docker container 84dbacbb2b60 stop
default-rule-path: /etc/suricata/rules660
673 sudo service filebeat start
rule-files: ./elastic-agent install -f --fleet-server-es=htt
- logs.rules
676 ls
##677 cd etc
## Auxiliary configuration files.
##679 cd filebeat
```

- `sudo suricata -c /etc/suricata/suricata.yaml -i eth0`
- `Service suricata start`

Ahora debemos cambiar la configuración red a interna DMZ_2.

Con esto, ya tenemos los honeypots activos y accesibles a través de la red wan

4) Creación ELK

Para desplegar ELK debemos tener otra máquina virtual con docker. Para ello he vuelto a crearla con Kali Linux.

Se tiene que realizar los siguientes comandos:

- Git clone <https://github.com/deviantony/docker-elk.git>
- Sudo chcon -R system_u:object_r:admin_home_t:s0 docker-elk/
- Cd docker-elk
- docker-compose up
- Cd elasticsearch
- Cd config
- Nano elasticsearch.yml (añadimos la línea "xpack.security.authc.api_key.enabled: true")

```
#
xpack.license.self_generated.type: trial
xpack.security.enabled: true
xpack.security.authc.api_key.enabled: true
xpack.monitoring.collection.enabled: true
```

Con esto ya tenemos acceso a kibana yendo a <http://localhost:5601>

Nos vamos a la ventana añadir agente. En la pestaña policies creamos una política añadiendo las integraciones linux y suricata.

[View all agent policies](#)

honeypot

Revision 3 | Integrations 3 | Used by 0 | Last updated on Jul 18, 2021 | [Actions](#)

[Integrations](#) Settings

Search... Namespace [Add integration](#)

Name ↑	Description	Integration	Namespace	Actions
linux-1		Linux v0.3.8	default	...
suricata-1		Suricata v0.6.0	default	...
system-1		System v0.12.7	default	...

Tenemos que crear un servidor Fleet rellenando los siguientes campos en Fleet settings:

Fleet settings

These settings are applied globally to the **outputs** section of all agent policies and affect all enrolled agents.

Fleet Server hosts

http://192.168.11.101:80 ×

Specify the URLs that your agents will use to connect to a Fleet Server. If multiple URLs exist, Fleet shows the first provided URL for enrollment purposes. Refer to the [Fleet User Guide](#).

Elasticsearch hosts

http://192.168.11.100:9200 ×

Specify the Elasticsearch URLs where agents send data.

Elasticsearch output configuration (YAML)

```
# YAML settings here will be added to the Elasticsearch output section of each policy
```

Ahora nos disponemos a crear un agente y desplegarlo en los honeypots.

Le aplicamos la acción add agent y en modo enroll in fleet cogemos el enlace de descarga de agente debian.

Más abajo cogemos el comando que nos ofrece ELK al escoger el modo DEB.

Abrimos la máquina kali de los honeypots 192.168.12.100 y descargamos el agente debian del enlace que obtuvimos anteriormente.

Descomprimos el paquete y en /elastic-agent/etc/init.d realizamos el comando que nos ofreció Elasticsearch en la máquina ELK 192.168.11.100.

Comprobamos el estado de elastic-agent para ver que está activo. Comando service elastic-agent status.

```

$ service elastic-agent status
● elastic-agent.service - Agent manages other beats based on configuration provided.
   Loaded: loaded (/lib/systemd/system/elastic-agent.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2021-07-18 16:23:49 EDT; 20h ago
     Docs: https://www.elastic.co/beats/elastic-agent
   Main PID: 10957 (elastic-agent)
      Tasks: 59 (limit: 2299)
     Memory: 389.0M
        CPU: 3min 56.235s
    CGroup: /system.slice/elastic-agent.service
            └─10957 elastic-agent --path.home /var/lib/elastic-agent --path.config /etc/elastic-agent --path.logs /var/log/elastic-agent run --environment systemd -c /etc/elastic-agent/elastic-agent.yml
            └─11412 /var/lib/elastic-agent/data/elastic-agent-3ddad4/install/filebeat-7.13.3-linux-x86_64/filebeat -E setup.ilm.enabled=false -E setup.template.enabled=false -E management.mode=x-pack-fleet -E management.en
            └─11428 /var/lib/elastic-agent/data/elastic-agent-3ddad4/install/metricbeat-7.13.3-linux-x86_64/metricbeat -E setup.ilm.enabled=false -E setup.template.enabled=false -E management.mode=x-pack-fleet -E management.en
            └─11449 /var/lib/elastic-agent/data/elastic-agent-3ddad4/install/filebeat-7.13.3-linux-x86_64/filebeat -E setup.ilm.enabled=false -E setup.template.enabled=false -E management.mode=x-pack-fleet -E management.en
            └─11468 /var/lib/elastic-agent/data/elastic-agent-3ddad4/install/metricbeat-7.13.3-linux-x86_64/metricbeat -E setup.ilm.enabled=false -E setup.template.enabled=false -E management.mode=x-pack-fleet -E management.en
            └─13016 /var/lib/elastic-agent/data/elastic-agent-3ddad4/install/fleet-server-7.13.3-linux-x86_64/fleet-server --agent-mode -E logging.level=info -E http.enabled=true -E http.host-unix:///var/lib/elastic-agent/
lines 1-15/15 (END)

```

Después de todos estos pasos podemos cambiamos el modo de red ELK a red local interna “DMZ”.

5) Realización de pruebas

Para realizar las conexiones a través de la red wan tengo una máquina Kali 192.168.1.65 en adaptador puente y el sistema Windows 10 nativo con su ip 192.168.1.41.

Con Windows trato de realizar conexiones a los honeypots RDP y Cowrie.

Abro la consola CMD y ejecuto el comando `ssh -p22 root@192.168.1.69:`

```

C:\> Símbolo del sistema - ssh root@192.168.1.69 -p 22

C:\Users\yesur>ssh root@192.168.1.69 -p 22
The authenticity of host '192.168.1.69 (192.168.1.69)' can't be established.
RSA key fingerprint is SHA256:ILgqXRqo9Ie0utCM9Fdx2hEmLwPbgMIz8mopUd0XoEA.
Are you sure you want to continue connecting (yes/no/[fingerprint])?

```

Observo el mismo fingerprint si realizo el comando en localhost de kali_honeypots:

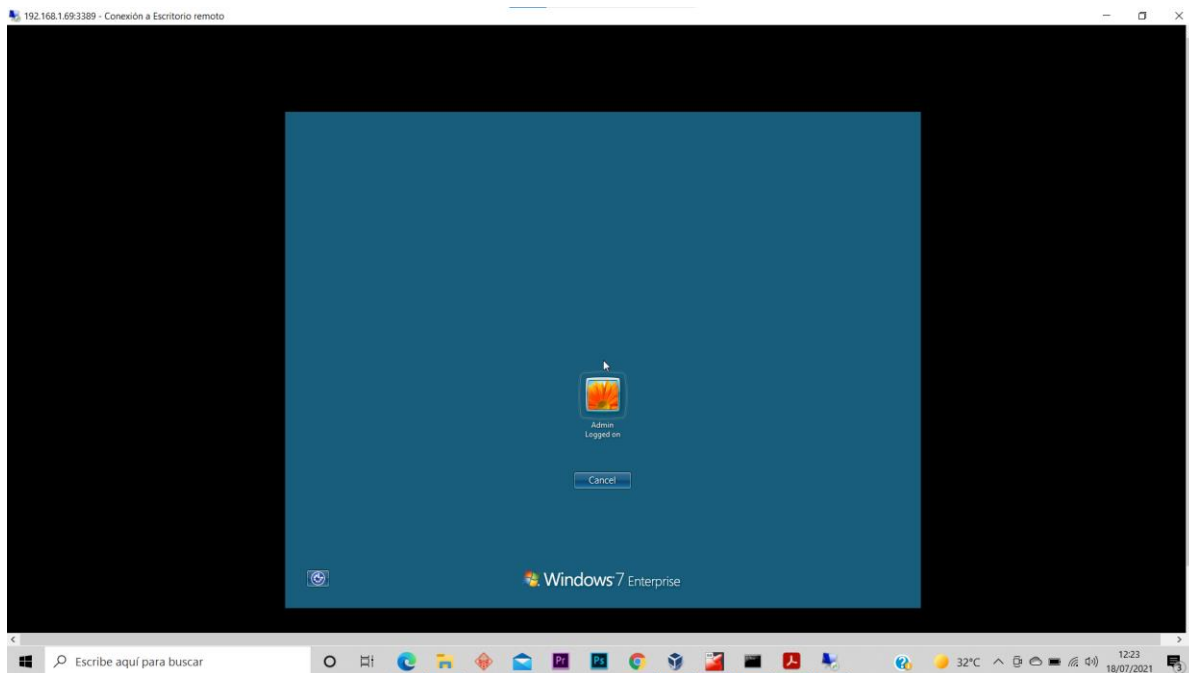
```

(kali@kali)-[~]
$ ssh root@192.168.12.100 -p 2222
The authenticity of host '[192.168.12.100]:2222 ([192.168.12.100]:2222)' can't be established.
RSA key fingerprint is SHA256:ILgqXRqo9Ie0utCM9Fdx2hEmLwPbgMIz8mopUd0XoEA.
Are you sure you want to continue connecting (yes/no/[fingerprint])?

```

Ejecuto otro comando para acceder a RDP:

- `MSTSTC.exe /V:192.168.1.69:3389 /w:1920 /h:1080`

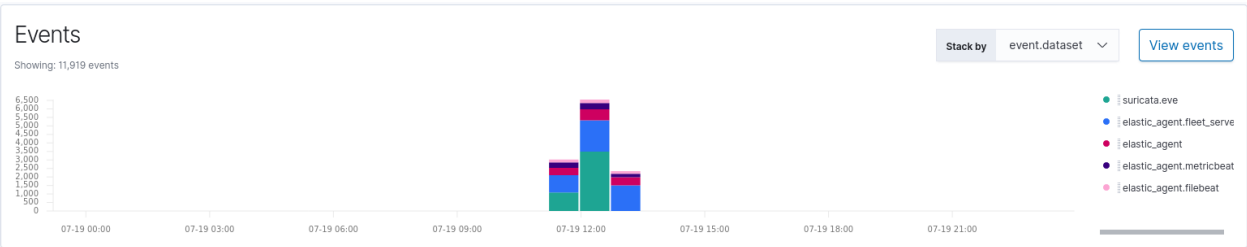


Para acceder al honeypot Gaspot utilizo la máquina Kali 192.168.1.65:

```
$ telnet 192.168.1.69 10000
Trying 192.168.1.69 ...
Connected to 192.168.1.69.
Escape character is '^]'.
ls
Connection closed by foreign host.
```

Con estas pruebas realizadas, abrimos la máquina windows10 192.168.10.10 en red LAN y monitorizamos kibana de ELK en <http://192.168.11.100:5601>

Nos metemos en dashboard de analytics o overview de security y comprobamos que tenemos datos sobre las anteriores intrusiones:



Dashboard / [Logs Suricata] Alert Overview ✓

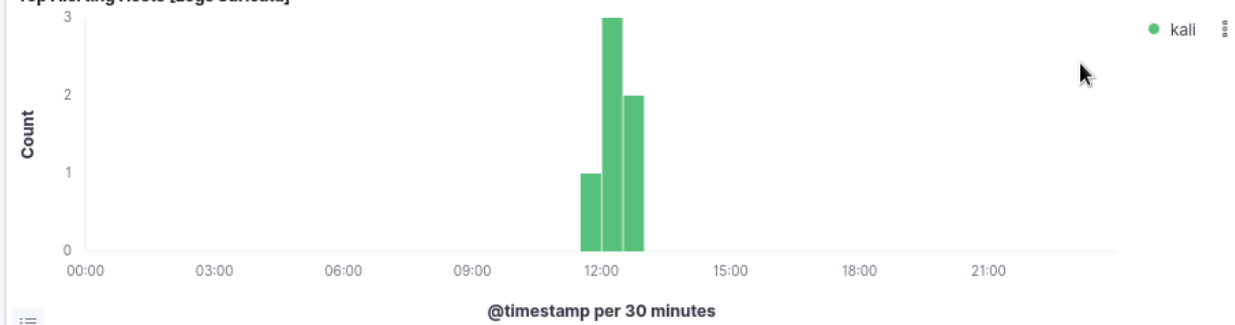
Search

+ Add filter

Navigation [Logs Suricata]

SURICATA Events | Alerts

Top Alerting Hosts [Logs Suricata]



Alerts [Logs Suricata]

Time	host.name	suricata.eve.flow_id	source.ip	source.port	destination.ip	destination.port
> Jul 19, 2021 @ 12:43:12.777	kali	1536991286058404	192.168.1.65	47362	-	10001
> Jul 19, 2021 @ 12:41:21.617	kali	1073100451048858	192.168.1.65	47358	-	10001
> Jul 19, 2021 @ 12:03:45.626	kali	147126829026838	192.168.1.41	64153	-	333
> Jul 19, 2021 @ 12:03:45.484	kali	935940522600061	192.168.1.41	64152	-	333
> Jul 19, 2021 @ 12:00:12.269	kali	984679797496645	192.168.1.41	51134	-	2222
> Jul 19, 2021 @ 11:59:50.235	kali	1486675573643659	192.168.1.41	51133	-	2222