



Práctica 3.6 Uso y reglas de Suricata

INCIDENTES DE CIBERSEGURIDAD
ERIC SERRANO MARÍN

Contenido

Práctica 3.6 Utilización y reglas de Suricata.	2
1. Ya sabes cómo activar reglas, bien pues ahora vamos a activar las siguientes: 2	
2. Muestra las reglas que tienes activadas.....	3
3. Vamos a probar como lanza Suricata una alerta cuando detecta que se cumple una regla, para ello vamos a usar la herramienta curl y las siguientes URL:.....	3
4. Regla 0: Ahora vamos a trabajar con las reglas, vamos a empezar por una sencilla:	4
5. Prueba regla 0: Desde otra máquina que pueda ver la máquina donde tienes instalado Suricata haz ping a la IP de tu máquina. Después revisa el fichero log, busca el SID de la regla y se deberá haber registrado algo como:.....	7
6. Regla 1: Ahora vamos a modificar la regla que hemos creado para usar las variables de entorno que tiene Suricata.....	8
7. Prueba regla 1:.....	9
8. Añadir clasificación a nuestras reglas:	10
9. Crea las siguientes reglas Snort, después deberás incluirlas en Suricata y probarlas:	11

Práctica 3.6 Utilización y reglas de Suricata.

En esta parte de la práctica vamos a utilizar distintos elementos de la herramienta Suricata:

- Reglas predefinidas
- Reglas personalizadas

ACTIVANDO Y PROBANDO REGLAS PREDEFINIDAS

1. Ya sabes cómo activar reglas, bien pues ahora vamos a activar las siguientes:

a. oisf/trafficid

```
(root@kali)-[/etc/suricata]
# suricata-update list-sources | grep oisf
Name: oisf/trafficid

(root@kali)-[/etc/suricata]
# suricata-update enable-source oisf/trafficid

29/2/2024 -- 16:20:50 - <Info> -- Using data-directory /var/lib/suricata.
29/2/2024 -- 16:20:50 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
└
29/2/2024 -- 16:20:50 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
29/2/2024 -- 16:20:50 - <Info> -- Found Suricata version 7.0.3 at /usr/bin/suricata.
29/2/2024 -- 16:20:50 - <Info> -- Source oisf/trafficid enabled
```

b. sslbl/ssl-fp-blacklist

```
(root@kali)-[/etc/suricata]
# suricata-update list-sources | grep sslbl
Name: sslbl/ssl-fp-blacklist
Name: sslbl/ja3-fingerprints

(root@kali)-[/etc/suricata]
# suricata-update enable-source sslbl/ssl-fp-blacklist

29/2/2024 -- 16:22:04 - <Info> -- Using data-directory /var/lib/suricata.
29/2/2024 -- 16:22:04 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
└
29/2/2024 -- 16:22:04 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
29/2/2024 -- 16:22:04 - <Info> -- Found Suricata version 7.0.3 at /usr/bin/suricata.
29/2/2024 -- 16:22:04 - <Info> -- Source sslbl/ssl-fp-blacklist enabled
```

2. Muestra las reglas que tienes activadas.

```
(root@kali)-[/etc/suricata]
# suricata-update list-enabled-sources
29/2/2024 -- 16:25:04 - <Info> -- Using data-directory /var/lib/suricata.
29/2/2024 -- 16:25:04 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
29/2/2024 -- 16:25:04 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
29/2/2024 -- 16:25:04 - <Info> -- Found Suricata version 7.0.3 at /usr/bin/suricata.
Enabled sources:
- et/open
- oisf/trafficid
- sslbl/ssl-fp-blacklist
```

3. Vamos a probar como lanza Suricata una alerta cuando detecta que se cumple una regla, para ello vamos a usar la herramienta curl y las siguientes URL:

a) **`curl -s http://testmynids.org/uid/index.html > /dev/null`**

Después, visiona los resultados en el log del fichero `/var/log/suricata/fast.log` buscando el id de la firma de la regla: 2100498. Deberá haber registrado una alerta de “Potentially Bad Traffic”.

```
(kali@kali)-[~]
$ cat /var/log/suricata/fast.log | grep 2100498
02/29/2024-16:26:50.102534  [**] [1:2100498:7] GPL ATTACK_RESPONSE id check returned root [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 18.154.22.58:80 → 172.22.235.128:49700
```

b) **`curl -s https://superfish.badssl.com/ > /dev/null`**

Después, visiona los resultados en el log del fichero `/var/log/suricata/fast.log` buscando el id de la firma de la regla: 2020493. Deberá haber registrado una alerta de “SuperFish”.

```
(kali@kali)-[~]
$ cat /var/log/suricata/fast.log | grep 2020493
02/29/2024-16:31:38.985186  [**] [1:2020493:3] ET MALWARE SuperFish Possible SSL Cert Signed By Compromised Root CA [**] [Classification: SuperFish] [Priority: 1] {TCP} 104.154.89.105:443 → 172.22.235.128:59736
```

c) `curl -s https://edellroot.badssl.com/ > /dev/null`

Después, visiona los resultados en el log del fichero `/var/log/suricata/fast.log` buscando el id de la firma de la regla: 2022134. Deberá haber registrado una alerta de “eDellRoot”.

```
(root@kali)-[/etc/suricata]
# cat /var/log/suricata/fast.log | grep eDellRoot
02/29/2024-16:38:21.761507  [**] [1:2022134:4] ET WEB_CLIENT Possible eDellRoot Rogue Root CA [**] [Classification: A Network Trojan was detected] [Priority: 1] {TCP} 104.154.89.105:443 → 172.22.235.128:57494
```

El sitio testmynids.org es un sitio diseñado para realizar pruebas de detección. (<https://github.com/3CORESec/testmynids.org>)

CREANDO REGLAS ESPECÍFICAS

4. Regla 0: Ahora vamos a trabajar con las reglas, vamos a empezar por una sencilla:

Para ello, crearemos y editaremos un fichero `/var/lib/suricata/rules/custom.rules` ó `/etc/suricata/rules/custom.rules` (dependiendo del sistema las reglas se almacenan en un directorio u otro, compruébalo en el fichero `suricata.yaml` en el parámetro `default-rule-path`), y añadimos la siguiente línea:

```
alert icmp any any -> 172.22.244.0/24 any (msg: "Paquete ICMP detectado";sid:1000001;rev:1;)
```

```
(root@kali)-[/var/lib/suricata/rules]
# nano custom.rules

(root@kali)-[/var/lib/suricata/rules]
# cat custom.rules
alert icmp any any -> 172.22.0.0/16 any (msg: "Paquete ICMP detectado";sid:1000001;rev:1;)
```

Con esta regla lo que haremos básicamente, será dar una alerta cada vez que detecte un paquete ICMP con nuestra IP de destino. (Recuerda poner como IP la estructura de tu red)

Agregar el fichero de reglas personalizadas *custom.rules* a la configuración de Suricata para que pueda aplicarlas sobre el tráfico escuchado. Debemos agregarla al fichero */etc/suricata/suricata.yaml* en el apartado de *rule-files* de la siguiente manera:

```
GNU nano 2.9.3 /etc/suricata/suricata.yaml Modified
#
hashmode: hash5tuplesorted

##
## Configure Suricata to load Suricata-Update managed rules.
##

default-rule-path: /var/lib/suricata/rules

rule-files:
- suricata.rules
- custom.rules
```

```
File Actions Edit View Help
GNU nano 7.2 /etc/suricata/suricata.yaml

##
## Configure Suricata to load Suricata-Update managed rules.
##

default-rule-path: /var/lib/suricata/rules

rule-files:
- suricata.rules
- custom.rules
```

Después es necesario reiniciar el servicio para que recoja las nuevas reglas agregadas. Revisar después el estado del servicio para asegurarte que está activo.

Ejecuta el comando **suricata-update**.

```
(root@kali)-[/var/lib/suricata/rules]
# suricata-update
29/2/2024 -- 16:52:37 - <Info> -- Using data-directory /var/lib/suricata.
29/2/2024 -- 16:52:37 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
29/2/2024 -- 16:52:37 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
29/2/2024 -- 16:52:37 - <Info> -- Found Suricata version 7.0.3 at /usr/bin/suricata

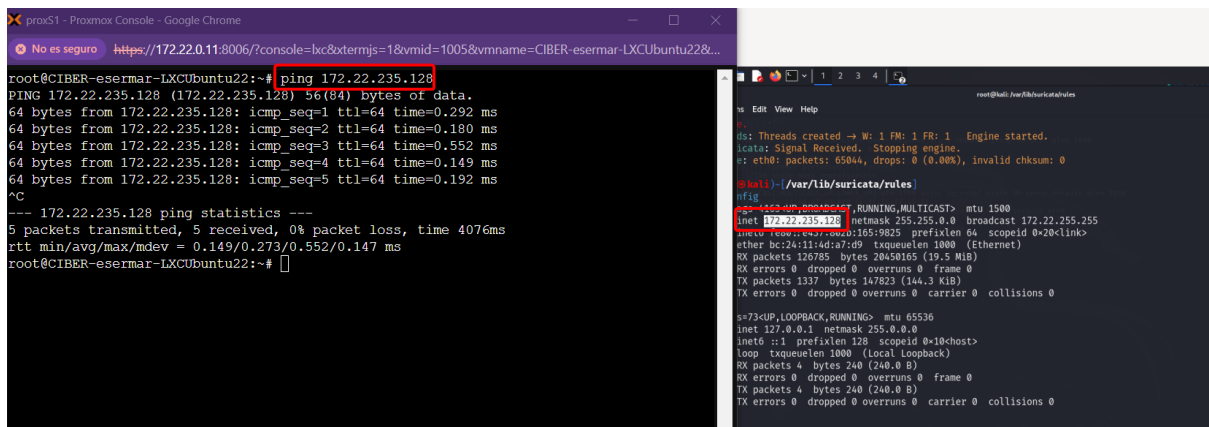
29/2/2024 -- 16:52:45 - <Info> -- Loaded 53384 rules.
29/2/2024 -- 16:52:45 - <Info> -- Disabled 14 rules.
29/2/2024 -- 16:52:45 - <Info> -- Enabled 0 rules.
29/2/2024 -- 16:52:45 - <Info> -- Modified 0 rules.
29/2/2024 -- 16:52:45 - <Info> -- Dropped 0 rules.
29/2/2024 -- 16:52:46 - <Info> -- Enabled 134 rules for flowbit dependencies.
29/2/2024 -- 16:52:46 - <Info> -- Backing up current rules.
29/2/2024 -- 16:52:47 - <Info> -- Writing rules to /var/lib/suricata/rules/suricata.rules: total: 53384; enabled: 42400; added: 6134; removed 83; modified: 9895
29/2/2024 -- 16:52:47 - <Info> -- Writing /var/lib/suricata/rules/classification.config
29/2/2024 -- 16:52:47 - <Info> -- Testing with suricata -T.
29/2/2024 -- 16:52:58 - <Info> -- Done.
```

Para saber si hemos colocado bien una regla, podemos verificarlo con el comando **suricata -c /etc/suricata/suricata.yaml -i (interface)**.

```
(root@kali)-[/var/lib/suricata/rules]
# suricata -c /etc/suricata/suricata.yaml -i eth0
i: suricata: This is Suricata version 7.0.3 RELEASE running in SYSTEM mode
E: af-packet: fanout not supported by kernel: Kernel too old or cluster-id 99 already in use.
i: threads: Threads created → W: 1 FM: 1 FR: 1 Engine started.
^Ci: suricata: Signal Received. Stopping engine.
i: device: eth0: packets: 65044, drops: 0 (0.00%), invalid checksum: 0
```

5. Prueba regla 0: Desde otra máquina que pueda ver la máquina donde tienes instalado Suricata haz ping a la IP de tu máquina. Después revisa el fichero log, busca el SID de la regla y se deberá haber registrado algo como:

```
02/13/2024-19:01:49.236519  [**] [1:1000001:1] Paquete ICMP detectado [**] [Classification: (null)] [Priority: 3] {ICMP} 172.22.255.184:8 -> 172.22.244.253:0
```



The screenshot shows a Proxmox console window with two terminal sessions. The left terminal is a root shell on a CIBER-esermary-LXC Ubuntu 22.04 system, where a ping test is performed to 172.22.235.128. The right terminal is a root shell on a Kali Linux system, showing the Suricata engine status and the contents of the /var/lib/suricata/rules directory. The rule file 1000001.rules is highlighted, showing its configuration for detecting ICMP traffic to the target IP.

```
root@CIBER-esermary-LXCubuntu22:~# ping 172.22.235.128
PING 172.22.235.128 (172.22.235.128) 56(84) bytes of data.
64 bytes from 172.22.235.128: icmp_seq=1 ttl=64 time=0.292 ms
64 bytes from 172.22.235.128: icmp_seq=2 ttl=64 time=0.180 ms
64 bytes from 172.22.235.128: icmp_seq=3 ttl=64 time=0.552 ms
64 bytes from 172.22.235.128: icmp_seq=4 ttl=64 time=0.149 ms
64 bytes from 172.22.235.128: icmp_seq=5 ttl=64 time=0.192 ms
^C
--- 172.22.235.128 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4076ms
rtt min/avg/max/mdev = 0.149/0.273/0.552/0.147 ms
root@CIBER-esermary-LXCubuntu22:~#
```

```
root@kali:~# cat /var/lib/suricata/rules/1000001.rules
# Suricata rule for detecting ICMP traffic to 172.22.235.128
# Rule ID: 1000001
# Rule Name: ICMP to 172.22.235.128
# Rule Author: CIBER-esermary
# Rule Version: 1.0
# Rule Description: Detects ICMP traffic to 172.22.235.128
# Rule Signature: 1000001
# Rule Category: ICMP
# Rule Action: alert
# Rule Options:
#   - meta:signature 1000001
#   - meta:category ICMP
#   - meta:action alert
#   - meta:options
#   - meta:signature 1000001
#   - meta:category ICMP
#   - meta:action alert
#   - meta:options
# Rule Content:
#   - meta:signature 1000001
#   - meta:category ICMP
#   - meta:action alert
#   - meta:options
# Rule End
```

```
02/29/2024-18:26:09.488130  [**] [1:1000001:1] Paquete ICMP detectado [**] [Classification: (null)] [Priority: 3] {ICMP} 172.22.227.1:8 -> 172.22.235.128:0
02/29/2024-18:26:09.488142  [**] [1:1000001:1] Paquete ICMP detectado [**] [Classification: (null)] [Priority: 3] {ICMP} 172.22.235.128:0 -> 172.22.227.1:0
02/29/2024-18:26:33.417163  [**] [1:2027397:1] ET POLICY Spotify P2P Client [**] [Classification: Not Suspicious Traffic] [Priority: 3] {UDP} 172.22.150.1:57621 -> 172.22.255.255:57621
```


6. Regla 1: Ahora vamos a modificar la regla que hemos creado para usar las variables de entorno que tiene Suricata.

Para ello tendremos que definir nuestra red local dentro de Suricata, esto nos servirá para saber si un paquete viene de dentro de nuestra red, o de alguna externa (u otra subred). Debemos editar el fichero `/etc/suricata/suricata.yaml` de forma que definamos la variable `HOME_NET` de la siguiente forma.

```
GNU nano 2.9.3 /etc/suricata/suricata.yaml
##
## Step 1: Inform Suricata about your network
##
vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    #HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    HOME_NET: "[172.22.244.0/24]"
    #HOME_NET: "any"

    EXTERNAL_NET: "!$HOME_NET"
    #EXTERNAL_NET: "any"
```

```
vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    #HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    HOME_NET: "[172.22.235.0/24]"
    #HOME_NET: "any"

    EXTERNAL_NET: "!$HOME_NET"
```

Se puede observar, que la variable `EXTERNAL_NET` la podemos dejar como está, ya que considera que todo lo que no es `HOME_NET` es red externa, pero podríamos concretar más si quisiéramos.

Ahora modificaremos nuestra regla de *custom.rules* y añadiremos una nueva:

```
alert icmp HOME_NET any -> any any (msg: "Paquete ICMP detectado";sid:1000001;rev:1;)
```

```
alert icmp any any -> EXTERNAL_NET any (msg: "Paquete ICMP de entrada detectado";sid:1000002;rev:1;)
```

```
root@kali: /var/lib/suricata/rules
File Actions Edit View Help
GNU nano 7.2 custom.rules *
alert icmp $HOME_NET any -> any any (msg: "Paquete ICMP detectado";sid:1000001;rev:1;)
alert icmp any any -> $EXTERNAL_NET any (msg: "Paquete ICMP de entrada detectado";sid:1000002;rev:1;)
```

Recuerda reiniciar el servicio y actualizar suricata, y no está de más comprobar que la regla está correctamente configurada.

7. Prueba regla 1:

Fuerza a que se lancen las alertas y compruébalo en el log.

```
usuario@Ubuntu-MV:~$ ping 172.22.235.128
PING 172.22.235.128 (172.22.235.128) 56(84) bytes of data.
64 bytes from 172.22.235.128: icmp_seq=1 ttl=64 time=0.148 ms
64 bytes from 172.22.235.128: icmp_seq=2 ttl=64 time=0.123 ms
64 bytes from 172.22.235.128: icmp_seq=3 ttl=64 time=0.135 ms
64 bytes from 172.22.235.128: icmp_seq=4 ttl=64 time=0.107 ms
64 bytes from 172.22.235.128: icmp_seq=5 ttl=64 time=0.165 ms
64 bytes from 172.22.235.128: icmp_seq=6 ttl=64 time=0.154 ms
^C
--- 172.22.235.128 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5113ms
rtt min/avg/max/mdev = 0.107/0.138/0.165/0.019 ms
```

```
(kali@kali)-[/etc/suricata]
$ tail -f /var/log/suricata/fast.log
03/01/2024-00:44:12.783177  [**] [1:2100366:8] GPL ICMP_INFO PING *NIX [**] [Classification: Misc activity] [Priority: 3] {IC
.235.128:0
03/01/2024-00:44:12.783191  [**] [1:1000002:1] Paquete ICMP de entrada detectado [**] [Classification: (null)] [Priority: 3]
.22.229.215:0
03/01/2024-00:44:13.807194  [**] [1:2100366:8] GPL ICMP_INFO PING *NIX [**] [Classification: Misc activity] [Priority: 3] {IC
.235.128:0
03/01/2024-00:44:13.807209  [**] [1:1000002:1] Paquete ICMP de entrada detectado [**] [Classification: (null)] [Priority: 3]
.22.229.215:0
03/01/2024-00:44:14.831170  [**] [1:2100366:8] GPL ICMP_INFO PING *NIX [**] [Classification: Misc activity] [Priority: 3] {IC
.235.128:0
03/01/2024-00:44:14.831182  [**] [1:1000002:1] Paquete ICMP de entrada detectado [**] [Classification: (null)] [Priority: 3]
.22.229.215:0
03/01/2024-00:44:15.855198  [**] [1:2100366:8] GPL ICMP_INFO PING *NIX [**] [Classification: Misc activity] [Priority: 3] {IC
.235.128:0
03/01/2024-00:44:15.855213  [**] [1:1000002:1] Paquete ICMP de entrada detectado [**] [Classification: (null)] [Priority: 3]
.22.229.215:0
03/01/2024-00:44:16.879174  [**] [1:2100366:8] GPL ICMP_INFO PING *NIX [**] [Classification: Misc activity] [Priority: 3] {IC
.235.128:0
03/01/2024-00:44:16.879193  [**] [1:1000002:1] Paquete ICMP de entrada detectado [**] [Classification: (null)] [Priority: 3]
.22.229.215:0
```

8. Añadir clasificación a nuestras reglas:

Existe otra opción de las reglas Snort que es *classtype* que nos indica, a partir de una definición en el fichero de configuración `/etc/suricata/classification.yaml`, la clasificación de reglas y su peligrosidad.

Vamos a definir una clasificación en el fichero `/etc/suricata/classification.config`, añadiendo al final la siguiente línea:

```
config classification: icmp-custom-event,ICMP event,2
```

Son tres campos a rellenar:

- Identificador de la clasificación: icmp-evento.
- Mensaje: "ICMP evento". Aparece en los logs como *category*.
- Prioridad: 4. Podemos ir variando el valor de menor a mayor en función de su peligrosidad o lo importante que consideremos que es la alerta, de 1 a 255.

```
# 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
config classification: icmp-evento,ICMP evento,8
# NEW CLASSIFICATIONS
```

Después modificaremos nuestras reglas para añadirle la opción *classtype*:

```
alert icmp HOME_NET any -> any any (msg: "Paquete de salida ICMP detectado";sid:1000001;rev:1; classtype:icmp-evento;)
```

```
alert icmp any any -> EXTERNAL_NET any (msg: "Paquete ICMP de entrada detectado";sid:1000002;rev:1;classtype:icmp-evento;)
```

```
root@kali: /etc/suricata
File Actions Edit View Help
GNU nano 7.2 /var/lib/suricata/rules/custom.rules *
alert icmp $HOME_NET any -> any any (msg: "Paquete ICMP detectado";sid:1000001;rev:1; classtype:icmp-evento;)
```

Repite la prueba del punto 7 y comprueba como se registran ahora las alertas en el log.

Ahora se registran con Classification: ICMP evento de Prioridad 8.

```
(root@kali)~[/etc/suricata]
# tail -f /var/log/suricata/fast.log
03/01/2024-13:44:21.555288 [**] [1:2100366:8] GPL ICMP_INFO PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 172.22.229.215:8 -> 172.22.235.128:0
03/01/2024-13:44:21.555393 [**] [1:1000002:1] Paquete ICMP de entrada detectado [**] [Classification: ICMP evento] [Priority: 8] {ICMP} 172.22.235.128:0 -> 172.22.229.215:0
03/01/2024-13:44:22.579290 [**] [1:2100366:8] GPL ICMP_INFO PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 172.22.229.215:8 -> 172.22.235.128:0
03/01/2024-13:44:22.579385 [**] [1:1000002:1] Paquete ICMP de entrada detectado [**] [Classification: ICMP evento] [Priority: 8] {ICMP} 172.22.235.128:0 -> 172.22.229.215:0
03/01/2024-13:44:23.603276 [**] [1:2100366:8] GPL ICMP_INFO PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 172.22.229.215:8 -> 172.22.235.128:0
03/01/2024-13:44:23.603291 [**] [1:1000002:1] Paquete ICMP de entrada detectado [**] [Classification: ICMP evento] [Priority: 8] {ICMP} 172.22.235.128:0 -> 172.22.229.215:0
03/01/2024-13:44:24.627263 [**] [1:2100366:8] GPL ICMP_INFO PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 172.22.229.215:8 -> 172.22.235.128:0
03/01/2024-13:44:24.627276 [**] [1:1000002:1] Paquete ICMP de entrada detectado [**] [Classification: ICMP evento] [Priority: 8] {ICMP} 172.22.235.128:0 -> 172.22.229.215:0
03/01/2024-13:44:25.651250 [**] [1:2100366:8] GPL ICMP_INFO PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 172.22.229.215:8 -> 172.22.235.128:0
03/01/2024-13:44:25.651262 [**] [1:1000002:1] Paquete ICMP de entrada detectado [**] [Classification: ICMP evento] [Priority: 8] {ICMP} 172.22.235.128:0 -> 172.22.229.215:0
```

9. Crea las siguientes reglas Snort, después deberás incluirlas en Suricata y probarlas:

a) Regla que registre un mensaje en el log sobre cualquier conexión establecida con un dominio o subdominios de TikTok.

```
alert tcp any any -> any any (msg:"Connection to TikTok domain"; app-layer-protocol:tls; tls.sni; content:"tiktok"; sid:100004;classtype:TikTok-evento;)
```

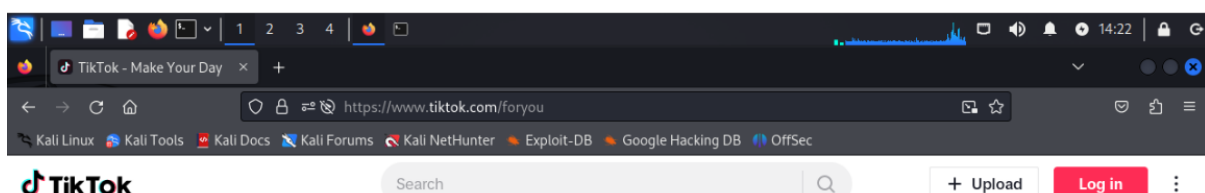
- alert tcp any any -> any any: Esta parte define la acción de la regla. En este caso, estamos generando una alerta para cualquier tráfico TCP desde cualquier dirección y hacia cualquier dirección.
- msg:"Connection to TikTok domain": El mensaje de alerta que se mostrará cuando se detecte una coincidencia con esta regla. En este caso, indica que se ha establecido una conexión con un dominio de TikTok.
- app-layer-protocol:tls: Esto especifica que la regla se aplica a conexiones TLS (Transport Layer Security).

- `tls.sni`: Se enfoca en el Server Name Indication (SNI), que es una extensión del protocolo TLS utilizada para indicar el nombre de host al que se está conectando el cliente.
- `content:"tiktok"`: Busca la cadena "tiktok" en el campo SNI. Si se encuentra, se activará la alerta.
- `sid:100004`: Número de identificación único para la regla.
- `classtype:TikTok-evento`: Clasificación personalizada para este evento relacionado con TikTok.

```
alert tcp any any -> any any (msg:"Connection to TikTok domain"; app-layer-protocol:tls; tls.sni; content:"tiktok"; sid:100004; classtype:TikTok-evento;)
```

```
config classification: tiktok-evento, TIKTOK evento, 9
```

Entramos a TikTok.



Antes de probar tendríamos que hacer un restart a suricata y un update.

```
(root@kali)~# tail -f /var/log/suricata/fast.log
03/01/2024-14:20:11.643037 [**] [1:2033078:4] ET INFO Session Traversal Utilities for NAT (STUN Binding Request On Non-Standard High Port) [**] [Classification: Misc activity] [Priority: 3] {UDP} 172.22.235.128:42798 -> 142.251.17.127:19302
03/01/2024-14:20:11.896664 [**] [1:2033078:4] ET INFO Session Traversal Utilities for NAT (STUN Binding Request On Non-Standard High Port) [**] [Classification: Misc activity] [Priority: 3] {UDP} 172.22.235.128:33017 -> 142.251.25.127:19302
03/01/2024-14:20:12.443174 [**] [1:2033078:4] ET INFO Session Traversal Utilities for NAT (STUN Binding Request On Non-Standard High Port) [**] [Classification: Misc activity] [Priority: 3] {UDP} 172.22.235.128:42798 -> 142.251.17.127:19302
03/01/2024-14:20:12.696826 [**] [1:2033078:4] ET INFO Session Traversal Utilities for NAT (STUN Binding Request On Non-Standard High Port) [**] [Classification: Misc activity] [Priority: 3] {UDP} 172.22.235.128:33017 -> 142.251.25.127:19302
03/01/2024-14:20:14.043331 [**] [1:2033078:4] ET INFO Session Traversal Utilities for NAT (STUN Binding Request On Non-Standard High Port) [**] [Classification: Misc activity] [Priority: 3] {UDP} 172.22.235.128:42798 -> 142.251.17.127:19302
03/01/2024-14:20:14.296987 [**] [1:2033078:4] ET INFO Session Traversal Utilities for NAT (STUN Binding Request On Non-Standard High Port) [**] [Classification: Misc activity] [Priority: 3] {UDP} 172.22.235.128:33017 -> 142.251.25.127:19302
03/01/2024-14:20:15.022439 [**] [1:100004:0] Connection to TikTok domain [**] [Classification: TIKTOK evento] [Priority: 9] {TCP} 172.22.235.128:45140 -> 185.43.182.122:443
03/01/2024-14:20:15.022999 [**] [1:100004:0] Connection to TikTok domain [**] [Classification: TIKTOK evento] [Priority: 9] {TCP} 172.22.235.128:45126 -> 185.43.182.122:443
03/01/2024-14:20:17.243485 [**] [1:2033078:4] ET INFO Session Traversal Utilities for NAT (STUN Binding Request On Non-Standard High Port) [**] [Classification: Misc activity] [Priority: 3] {UDP} 172.22.235.128:42798 -> 142.251.17.127:19302
03/01/2024-14:20:17.409136 [**] [1:2033078:4] ET INFO Session Traversal Utilities for NAT (STUN Binding Request On Non-Standard High Port) [**] [Classification: Misc activity] [Priority: 3] {UDP} 172.22.235.128:42798 -> 142.251.17.127:19302
```

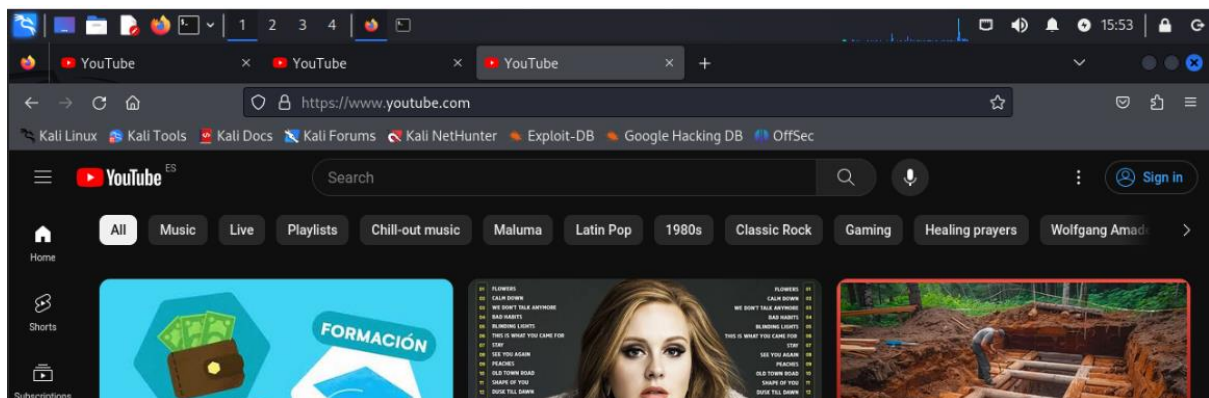
Podemos observar como nos la clasifica como TIKTOK evento con prioridad 9.

b) Regla dns que **bloquee** el paquete cuando la consulta DNS se refiera a un dominio concreto que tu elijas.

- alert dns any any -> any any: Detecta cualquier consulta DNS.
- msg:"Bloqueo de consulta DNS para dominio malicioso.com": Mensaje de alerta.
- dns.query: Se enfoca en las consultas DNS.
- content:"youtube.com": Busca el dominio específico que deseas bloquear.
- sid:100005: Número de identificación único para la regla.

```
alert tcp any any -> any any (msg:"Connection to tiktok domain"; app-layer-protocol:tls; tls.sni, content:"tiktok"; sid:100004; classtype:tiktok-evento;)  
alert dns any any -> any any (msg:"Bloqueo de consulta DNS para dominio youtube.com"; dns.query; content:"youtube.com"; sid:100005; classtype:DNS-evento;)
```

```
config classification: DNS-evento,DNS evento,10
```



```
[root@kali:~/etc/suricata]# tail -f /var/log/suricata/fast.log  
03/01/2024-15:20:10.376289 1:1000004:0 Connection to TikTok domain 1:1000004:0 [Classification: TIKTOK evento] [Priority: 9] {TCP} 172.22.235.128:60322 -> 96.16.88.167:443  
03/01/2024-15:36:14.549427 1:1000004:0 Connection to TikTok domain 1:1000004:0 [Classification: TIKTOK evento] [Priority: 9] {TCP} 172.22.235.128:48024 -> 71.18.51.225:443  
03/01/2024-15:36:22.194669 1:1000004:0 Connection to TikTok domain 1:1000004:0 [Classification: TIKTOK evento] [Priority: 9] {TCP} 172.22.235.128:49168 -> 96.16.88.144:443  
03/01/2024-15:36:35.096162 1:2022973:1 ET POLICY Possible Kali Linux hostname in DHCP Request Packet 1:1000005:0 [Classification: Potential Corporate Privacy Violation] [Priority: 1] {UDP} 172.22.235.128:68 -> 172.22.1.1:67  
03/01/2024-15:40:38.006069 1:1000005:0 Blocked DNS query for example.com 1:1000005:0 [Classification: DNS evento] [Priority: 10] {UDP} 172.22.235.128:33337 -> 80.58.61.250:53  
03/01/2024-15:40:38.007634 1:1000005:0 Blocked DNS query for example.com 1:1000005:0 [Classification: DNS evento] [Priority: 10] {UDP} 172.22.235.128:48351 -> 80.58.61.250:53  
03/01/2024-15:50:29.548686 1:1000005:0 Bloqueo de consulta DNS para dominio youtube.com 1:1000005:0 [Classification: DNS evento] [Priority: 10] {UDP} 172.22.235.128:33130 -> 80.58.61.250:53  
03/01/2024-15:51:20.172016 1:1000005:0 Bloqueo de consulta DNS para dominio youtube.com 1:1000005:0 [Classification: DNS evento] [Priority: 10] {UDP} 172.22.235.128:48089 -> 80.58.61.250:53  
03/01/2024-15:51:22.809370 1:1000005:0 Bloqueo de consulta DNS para dominio youtube.com 1:1000005:0 [Classification: DNS evento] [Priority: 10] {UDP} 172.22.235.128:37929 -> 80.58.61.250:53  
03/01/2024-15:51:24.049354 1:1000005:0 Bloqueo de consulta DNS para dominio youtube.com 1:1000005:0 [Classification: DNS evento] [Priority: 10] {UDP} 172.22.235.128:55681 -> 80.58.61.250:53
```

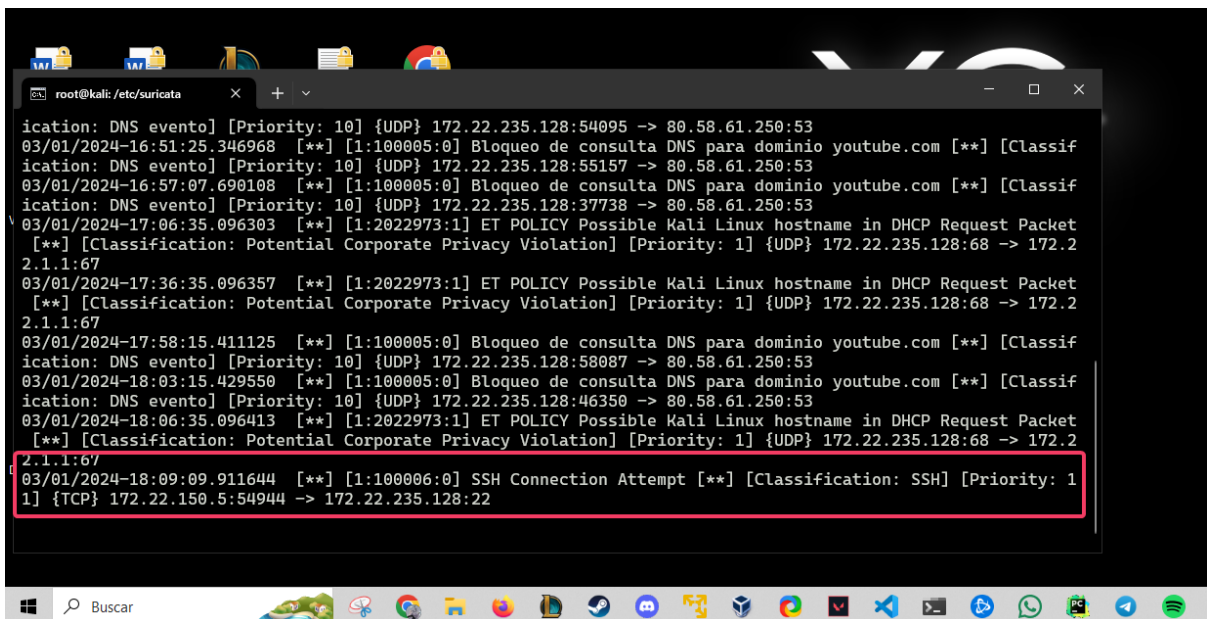

c) Regla de alerta tcp en la que utilices una variable de entorno de Suricata de puertos (port-groups)

Viendo los port-groups que hay, he decidido usar el SSH_PORTS, ya que me viene de perlas porque estoy conectado la Kali de proxmox por SSH.

```
port-groups:
  HTTP_PORTS: "80, 8080, 8000"
  SHELLCODE_PORTS: "!80"
  ORACLE_PORTS: 1521
  SSH_PORTS: 22
  DNP3_PORTS: 20000
  MODBUS_PORTS: 502
  FILE_DATA_PORTS: "[$HTTP_PORTS,110,143]"
  FTP_PORTS: 21
  GENEVE_PORTS: 6081
  VXLAN_PORTS: 4789
  TEREDO_PORTS: 3544
```

```
alert dns any any -> any any (msg: Bloqueo de consulta DNS para dominio youtube.com , dns.query, Com
alert tcp any any -> any $SSH_PORTS (msg:"SSH Connection Attempt"; sid:100006;classtype:SSH-evento;)
```

```
config classification: SSH-evento,SSH,11
```



```
root@kali: /etc/suricata
[Classification: DNS evento] [Priority: 10] {UDP} 172.22.235.128:54095 -> 80.58.61.250:53
03/01/2024-16:51:25.346968  [**] [1:100005:0] Bloqueo de consulta DNS para dominio youtube.com [**] [Classif
[Classification: DNS evento] [Priority: 10] {UDP} 172.22.235.128:55157 -> 80.58.61.250:53
03/01/2024-16:57:07.690108  [**] [1:100005:0] Bloqueo de consulta DNS para dominio youtube.com [**] [Classif
[Classification: DNS evento] [Priority: 10] {UDP} 172.22.235.128:37738 -> 80.58.61.250:53
03/01/2024-17:06:35.096303  [**] [1:2022973:1] ET POLICY Possible Kali Linux hostname in DHCP Request Packet
[**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {UDP} 172.22.235.128:68 -> 172.2
2.1.1:67
03/01/2024-17:36:35.096357  [**] [1:2022973:1] ET POLICY Possible Kali Linux hostname in DHCP Request Packet
[**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {UDP} 172.22.235.128:68 -> 172.2
2.1.1:67
03/01/2024-17:58:15.411125  [**] [1:100005:0] Bloqueo de consulta DNS para dominio youtube.com [**] [Classif
[Classification: DNS evento] [Priority: 10] {UDP} 172.22.235.128:58087 -> 80.58.61.250:53
03/01/2024-18:03:15.429550  [**] [1:100005:0] Bloqueo de consulta DNS para dominio youtube.com [**] [Classif
[Classification: DNS evento] [Priority: 10] {UDP} 172.22.235.128:46350 -> 80.58.61.250:53
03/01/2024-18:06:35.096413  [**] [1:2022973:1] ET POLICY Possible Kali Linux hostname in DHCP Request Packet
[**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {UDP} 172.22.235.128:68 -> 172.2
2.1.1:67
03/01/2024-18:09:09.911644  [**] [1:100006:0] SSH Connection Attempt [**] [Classification: SSH] [Priority: 1
1] {TCP} 172.22.150.5:54944 -> 172.22.235.128:22
```

- Alertará sobre tráfico TCP.
- any any -> any \$SSH_PORTS: define la dirección origen y destino, en este caso estoy buscando tráfico que vaya desde cualquier dirección a cualquier dirección en los puertos SSH.
- msg: será el meansje.
- sid será su ID único.
- classtype: para clasificarlos y categorizar las reglas.

d) Crea una clasificación acorde con la regla anterior con una prioridad distinta a la que aparece actualmente en el log cuando se lanza la regla. Después modifica la regla anterior para añadirle la clasificación que ha creado. Prueba la regla para ver que en el log ahora aparecen los datos de la nueva categoría.

La regla anterior de SSH_PORTS tiene prioridad 11 y Clasificación SSH, ahora vamos a cambiarlo a prioridad: 1 y clasificación SSH22.

```
alert tcp any any -> any $SSH_PORTS (msg:"SSH Connection Attempt"; sid:100006;classtype:SSH22-evento;)
```

```
config classification: SSH22-evento,SSH,1
```

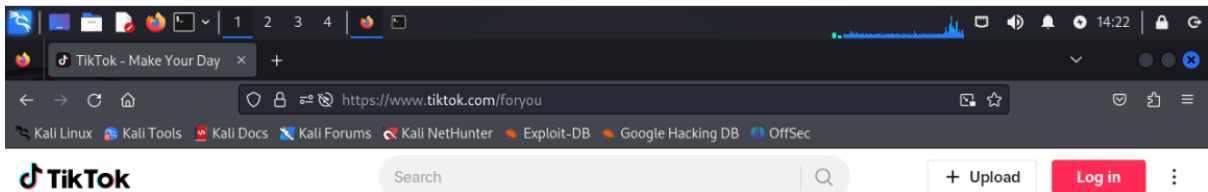
Podemos observar como el cambio se ha hecho correctamente.

```
(root@kali)~# tail -f /etc/suricata/fast.log
03/01/2024-16:51:25.346968 3 1:100005:0 Bloqueo de consulta DNS para dominio youtube.com 1:100005:0 [Classification: DNS evento] [Priority: 10] {UDP} 172.22.235.12
03/01/2024-16:57:07.690108 3 1:100005:0 Bloqueo de consulta DNS para dominio youtube.com 1:100005:0 [Classification: DNS evento] [Priority: 10] {UDP} 172.22.235.12
03/01/2024-17:06:35.096303 3 1:2022973:1 ET POLICY Possible Kali Linux hostname in DHCP Request Packet 1:2022973:1 [Classification: Potential Corporate Privacy Viol
} 172.22.235.128:68 -> 172.22.1.1:67
03/01/2024-17:36:35.096357 3 1:2022973:1 ET POLICY Possible Kali Linux hostname in DHCP Request Packet 1:2022973:1 [Classification: Potential Corporate Privacy Viol
} 172.22.235.128:68 -> 172.22.1.1:67
03/01/2024-17:58:15.411125 3 1:100005:0 Bloqueo de consulta DNS para dominio youtube.com 1:100005:0 [Classification: DNS evento] [Priority: 10] {UDP} 172.22.235.12
03/01/2024-18:03:15.429550 3 1:100005:0 Bloqueo de consulta DNS para dominio youtube.com 1:100005:0 [Classification: DNS evento] [Priority: 10] {UDP} 172.22.235.12
03/01/2024-18:06:35.096413 3 1:2022973:1 ET POLICY Possible Kali Linux hostname in DHCP Request Packet 1:2022973:1 [Classification: Potential Corporate Privacy Viol
} 172.22.235.128:68 -> 172.22.1.1:67
03/01/2024-18:09:09.911644 3 1:100006:0 SSH Connection Attempt 1:100006:0 [Classification: SSH] [Priority: 11] {TCP} 172.22.150.5:54944 -> 172.22.235.128:22
03/01/2024-18:13:36.041294 3 1:100006:0 SSH Connection Attempt 1:100006:0 [Classification: SSH] [Priority: 11] {TCP} 172.22.150.5:54944 -> 172.22.235.128:22
03/01/2024-18:15:09.260500 3 1:100006:0 SSH Connection Attempt 1:100006:0 [Classification: SSH] [Priority: 1] {TCP} 172.22.150.5:54944 -> 172.22.235.128:22
```


e) Regla en la que uses la opción *app-layer-protocol*. La acción y resto de opciones pueden ser como quieras, pero que sea coherente.

```
alert tcp any any -> any any (msg:"Connection to TikTok domain"; app-layer-protocol:tls; tls.sni; content:"tiktok"; sid:100004; classtype:TikTok-evento;)
```

```
config classification: tiktok-evento, TIKTOK evento, 9
```



```
(root@kali)~# tail -f /etc/suricata/fast.log
03/01/2024-14:20:11.643037 [**] [1:2033078:4] ET INFO Session Traversal Utilities for NAT (STUN Binding Request On Non-Standard High Port) [**] [Classification: Misc activity] [Priority: 3] {UDP} 172.22.235.128:42798 -> 142.251.17.127:19302
03/01/2024-14:20:11.896664 [**] [1:2033078:4] ET INFO Session Traversal Utilities for NAT (STUN Binding Request On Non-Standard High Port) [**] [Classification: Misc activity] [Priority: 3] {UDP} 172.22.235.128:33017 -> 142.251.25.127:19302
03/01/2024-14:20:12.443174 [**] [1:2033078:4] ET INFO Session Traversal Utilities for NAT (STUN Binding Request On Non-Standard High Port) [**] [Classification: Misc activity] [Priority: 3] {UDP} 172.22.235.128:42798 -> 142.251.17.127:19302
03/01/2024-14:20:12.696826 [**] [1:2033078:4] ET INFO Session Traversal Utilities for NAT (STUN Binding Request On Non-Standard High Port) [**] [Classification: Misc activity] [Priority: 3] {UDP} 172.22.235.128:33017 -> 142.251.25.127:19302
03/01/2024-14:20:14.043331 [**] [1:2033078:4] ET INFO Session Traversal Utilities for NAT (STUN Binding Request On Non-Standard High Port) [**] [Classification: Misc activity] [Priority: 3] {UDP} 172.22.235.128:42798 -> 142.251.17.127:19302
03/01/2024-14:20:14.296987 [**] [1:2033078:4] ET INFO Session Traversal Utilities for NAT (STUN Binding Request On Non-Standard High Port) [**] [Classification: Misc activity] [Priority: 3] {UDP} 172.22.235.128:33017 -> 142.251.25.127:19302
03/01/2024-14:20:15.022439 [**] [1:100004:0] Connection to TikTok domain [**] [Classification: TIKTOK evento] [Priority: 9] {TCP} 172.22.235.128:45140 -> 185.43.182.122:443
03/01/2024-14:20:15.022999 [**] [1:100004:0] Connection to TikTok domain [**] [Classification: TIKTOK evento] [Priority: 9] {TCP} 172.22.235.128:45126 -> 185.43.182.122:443
03/01/2024-14:20:17.243485 [**] [1:2033078:4] ET INFO Session Traversal Utilities for NAT (STUN Binding Request On Non-Standard High Port) [**] [Classification: Misc activity] [Priority: 3] {UDP} 172.22.235.128:42798 -> 142.251.17.127:19302
03/01/2024-14:20:17.489126 [**] [1:2033078:4] ET INFO Session Traversal Utilities for NAT (STUN Binding Request On Non-Standard High Port) [**] [Classification: Misc activity] [Priority: 3] {UDP} 172.22.235.128:42798 -> 142.251.17.127:19302
```

Esta regla está ya explicada un poco más atrás del documento, he usado esta, ya que me ha servido como ejemplo.