UT4 - Ejercicio 2: Suricata

UT4 - Ejercicio 2: Suricata

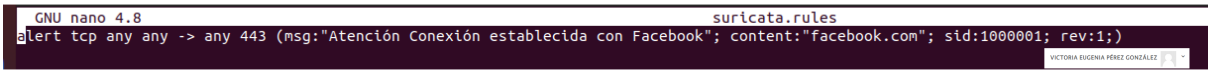
Victoria Eugenia Pérez González

21/02/2023

ÍNDICE

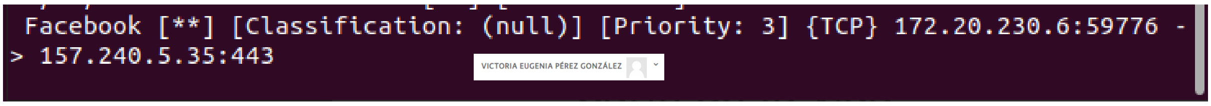
Utilizando Suricata como IDS/IPS, se solicita:

a. Configurar una regla que detecte y alerte las conexiones a Facebook



```
  GNU nano 4.8                                                         suricata.rules
alert tcp any any -> any 443 (msg:"Atención Conexión establecida con Facebook"; content:"facebook.com"; sid:1000001; rev:1;)
```

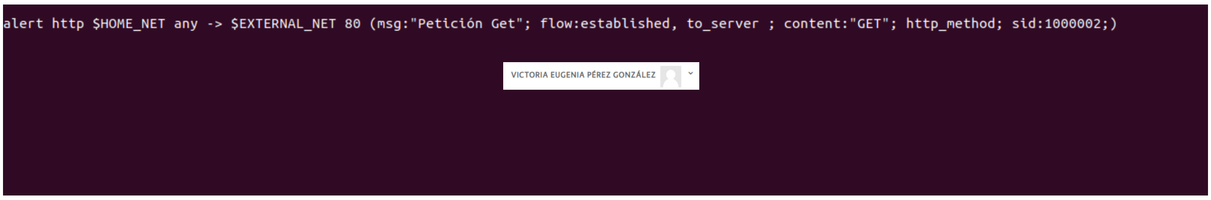
Se prueba que funciona



```
Facebook [**] [Classification: (null)] [Priority: 3] {TCP} 172.20.230.6:59776 -
> 157.240.5.35:443
```

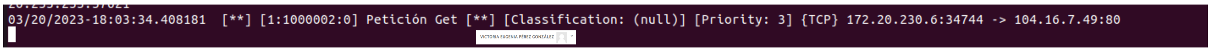
b. Configurar una regla que detecte y alerte cuando, desde nuestra red interna, se haga alguna petición GET al exterior.

Se configura la regla.



```
alert http $HOME_NET any -> $EXTERNAL_NET 80 (msg:"Petición Get"; flow:established, to_server ; content:"GET"; http_method; sid:1000002;)
```

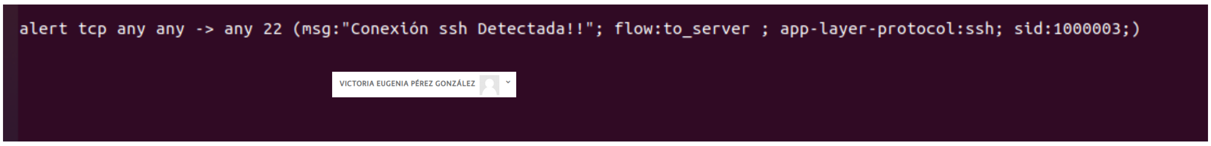
Se prueba la regla.



```
03/20/2023-18:03:34.408181  [**] [1:1000002:0] Petición Get [**] [Classification: (null)] [Priority: 3] {TCP} 172.20.230.6:34744 -> 104.16.7.49:80
```

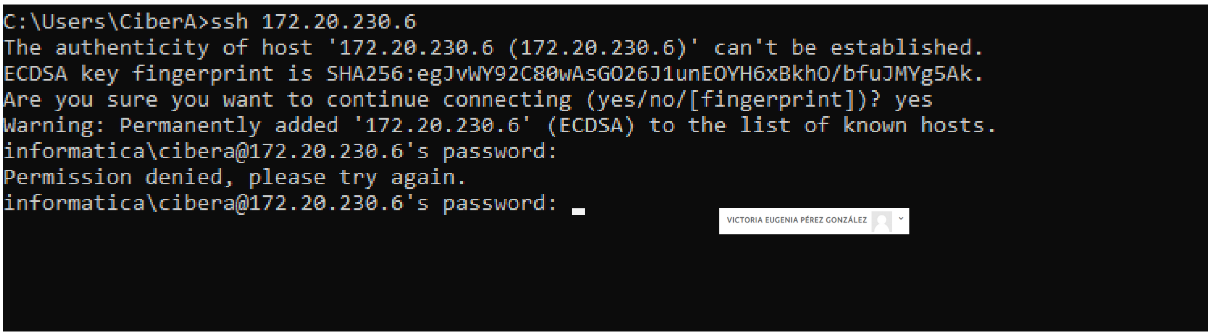
c. Configurar una regla que detecte y alerte cuando se realiza una conexión utilizando ssh

Se configura la regla:



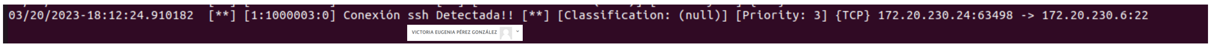
```
alert tcp any any -> any 22 (msg:"Conexión ssh Detectada!!"; flow:to_server ; app-layer-protocol:ssh; sid:1000003;)
```

Se realiza el intento de conexión.



```
C:\Users\CiberA>ssh 172.20.230.6
The authenticity of host '172.20.230.6 (172.20.230.6)' can't be established.
ECDSA key fingerprint is SHA256:egJvWY92C80wAsGO26J1unEOYH6xBkhO/bfuJMYg5Ak.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.20.230.6' (ECDSA) to the list of known hosts.
informatica\cibera@172.20.230.6's password:
Permission denied, please try again.
informatica\cibera@172.20.230.6's password:
```

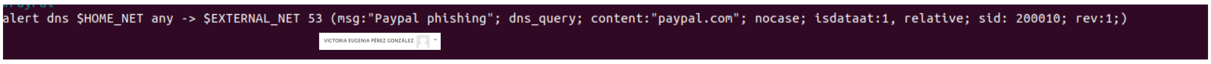
El ids detecta la regla.



```
03/20/2023-18:12:24.910182  [**] [1:1000003:0] Conexión ssh Detectada!! [**] [Classification: (null)] [Priority: 3] {TCP} 172.20.230.24:63498 -> 172.20.230.6:22
```

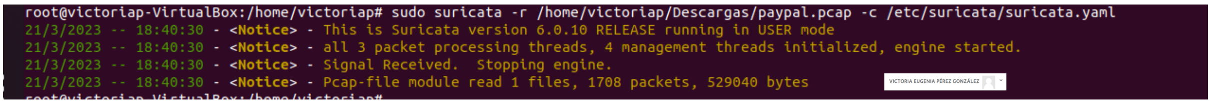
. d. Configurar una regla que detecte y alerte en el caso de que, desde nuestra red interna, se esté intentando acceder a un sitio web que pueda tratarse de una simulación de la web de Paypal (ataque phishing)

Se crea la alerta



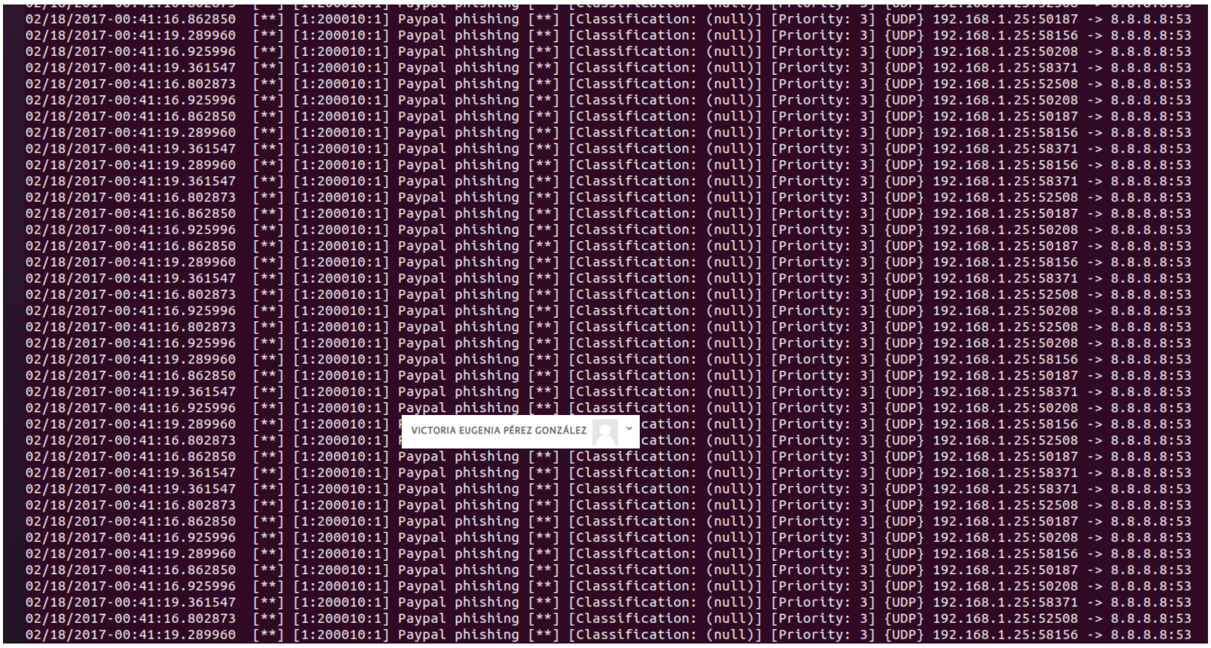
```
alert dns $HOME_NET any -> $EXTERNAL_NET 53 (msg:"Paypal phishing"; dns_query; content:"paypal.com"; nocase; isdataat:1, relative; sid: 200010; rev:1;)
```

Se prueba usando el pcap



```
root@victoriap-VirtualBox:/home/victoriap# sudo suricata -r /home/victoriap/Descargas/paypal.pcap -c /etc/suricata/suricata.yaml
21/3/2023 -- 18:40:30 - <Notice> - This is Suricata version 6.0.10 RELEASE running in USER mode
21/3/2023 -- 18:40:30 - <Notice> - all 3 packet processing threads, 4 management threads initialized, engine started.
21/3/2023 -- 18:40:30 - <Notice> - Signal Received.  Stopping engine.
21/3/2023 -- 18:40:30 - <Notice> - Pcap-file module read 1 files, 1708 packets, 529040 bytes
root@victoriap-VirtualBox:/home/victoriap#
```

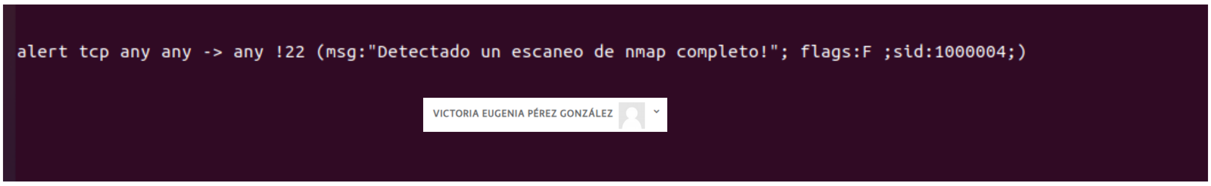
Se comprueba que detecta el phishing en el log del suricata que vuelca en el destino que se le indica.



```
02/18/2017-00:41:16.862850  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:50187 -> 8.8.8.8:53
02/18/2017-00:41:19.289960  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:58156 -> 8.8.8.8:53
02/18/2017-00:41:16.925996  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:50208 -> 8.8.8.8:53
02/18/2017-00:41:19.361547  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:58371 -> 8.8.8.8:53
02/18/2017-00:41:16.802873  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:52508 -> 8.8.8.8:53
02/18/2017-00:41:16.925996  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:50208 -> 8.8.8.8:53
02/18/2017-00:41:16.862850  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:50187 -> 8.8.8.8:53
02/18/2017-00:41:19.289960  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:58156 -> 8.8.8.8:53
02/18/2017-00:41:19.361547  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:58371 -> 8.8.8.8:53
02/18/2017-00:41:19.289960  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:58156 -> 8.8.8.8:53
02/18/2017-00:41:19.361547  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:58371 -> 8.8.8.8:53
02/18/2017-00:41:16.802873  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:52508 -> 8.8.8.8:53
02/18/2017-00:41:16.862850  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:50187 -> 8.8.8.8:53
02/18/2017-00:41:16.925996  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:50208 -> 8.8.8.8:53
02/18/2017-00:41:16.862850  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:50187 -> 8.8.8.8:53
02/18/2017-00:41:19.289960  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:58156 -> 8.8.8.8:53
02/18/2017-00:41:19.361547  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:58371 -> 8.8.8.8:53
02/18/2017-00:41:16.802873  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:52508 -> 8.8.8.8:53
02/18/2017-00:41:16.925996  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:50208 -> 8.8.8.8:53
02/18/2017-00:41:16.802873  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:52508 -> 8.8.8.8:53
02/18/2017-00:41:16.925996  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:50208 -> 8.8.8.8:53
02/18/2017-00:41:19.289960  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:58156 -> 8.8.8.8:53
02/18/2017-00:41:16.862850  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:50187 -> 8.8.8.8:53
02/18/2017-00:41:19.361547  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:58371 -> 8.8.8.8:53
02/18/2017-00:41:16.925996  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:50208 -> 8.8.8.8:53
02/18/2017-00:41:19.289960  [**] [1:200010:1] P                     cation: (null)] [Priority: 3] {UDP} 192.168.1.25:58156 -> 8.8.8.8:53
02/18/2017-00:41:16.802873  [**] [1:200010:1] P                     cation: (null)] [Priority: 3] {UDP} 192.168.1.25:52508 -> 8.8.8.8:53
02/18/2017-00:41:16.862850  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:50187 -> 8.8.8.8:53
02/18/2017-00:41:19.361547  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:58371 -> 8.8.8.8:53
02/18/2017-00:41:19.361547  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:58371 -> 8.8.8.8:53
02/18/2017-00:41:16.802873  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:52508 -> 8.8.8.8:53
02/18/2017-00:41:16.862850  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:50187 -> 8.8.8.8:53
02/18/2017-00:41:16.925996  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:50208 -> 8.8.8.8:53
02/18/2017-00:41:19.289960  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:58156 -> 8.8.8.8:53
02/18/2017-00:41:16.862850  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:50187 -> 8.8.8.8:53
02/18/2017-00:41:16.925996  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:50208 -> 8.8.8.8:53
02/18/2017-00:41:19.361547  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:58371 -> 8.8.8.8:53
02/18/2017-00:41:16.802873  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:52508 -> 8.8.8.8:53
02/18/2017-00:41:19.289960  [**] [1:200010:1] Paypal phishing [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.1.25:58156 -> 8.8.8.8:53
```

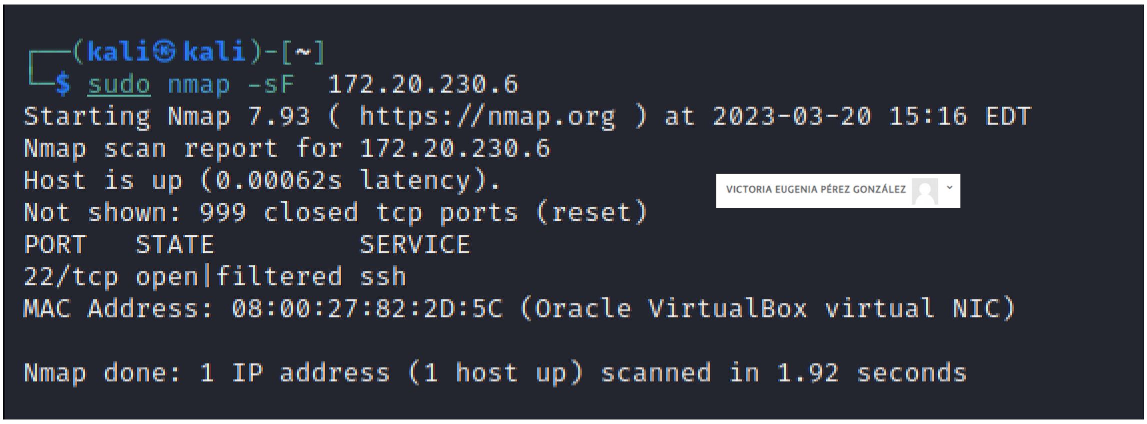
e. Configurar un conjunto de reglas capaces de detectar y alertar cuando nuestra máquina está recibiendo un escaneo de puertos.

Regla 1:se configuran las regla:



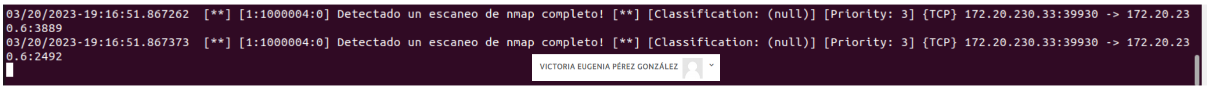
```
alert tcp any any -> any !22 (msg:"Detectado un escaneo de nmap completo!"; flags:F ;sid:1000004;)
```

Se hace un nmap desde kali



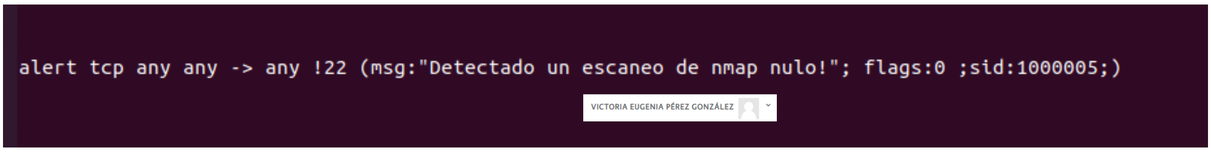
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sF  172.20.230.6
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-20 15:16 EDT
Nmap scan report for 172.20.230.6
Host is up (0.00062s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE         SERVICE
22/tcp open|filtered ssh
MAC Address: 08:00:27:82:2D:5C (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.92 seconds
```

El ids detecta el ataque.



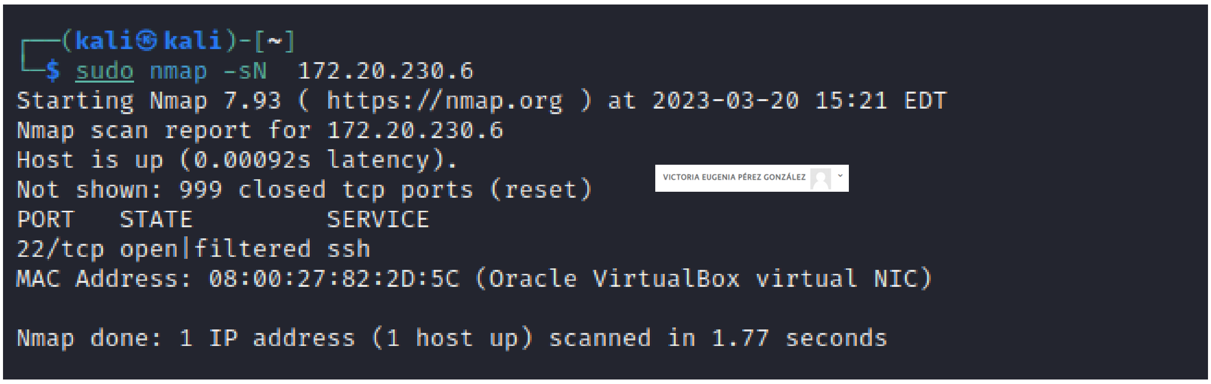
```
03/20/2023-19:16:51.867262  [**] [1:1000004:0] Detectado un escaneo de nmap completo! [**] [Classification: (null)] [Priority: 3] {TCP} 172.20.230.33:39930 -> 172.20.23
0.6:3889
03/20/2023-19:16:51.867373  [**] [1:1000004:0] Detectado un escaneo de nmap completo! [**] [Classification: (null)] [Priority: 3] {TCP} 172.20.230.33:39930 -> 172.20.23
0.6:2492
```

Regla 2:



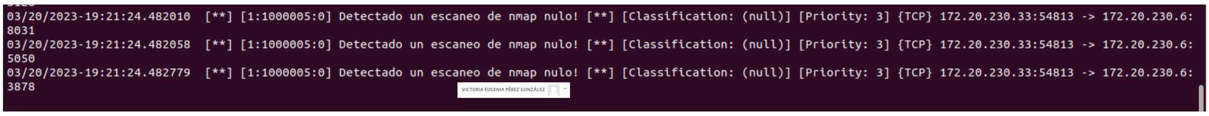
```
alert tcp any any -> any !22 (msg:"Detectado un escaneo de nmap nulo!"; flags:0 ;sid:1000005;)
```

Se prueba



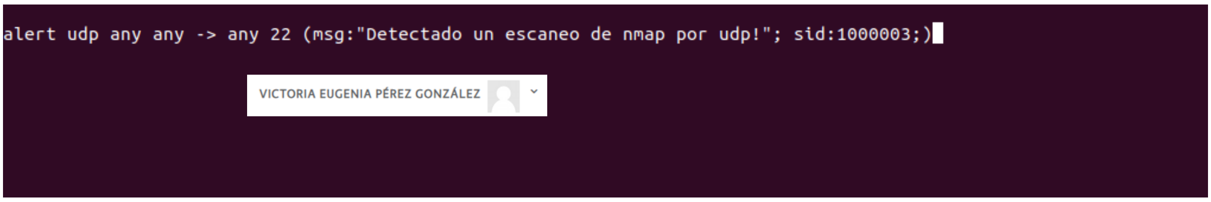
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sN  172.20.230.6
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-20 15:21 EDT
Nmap scan report for 172.20.230.6
Host is up (0.00092s latency).
Not shown: 999 closed tcp ports (reset)
PORT    STATE         SERVICE
22/tcp open|filtered ssh
MAC Address: 08:00:27:82:2D:5C (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.77 seconds
```

El ids detecta las condiciones.



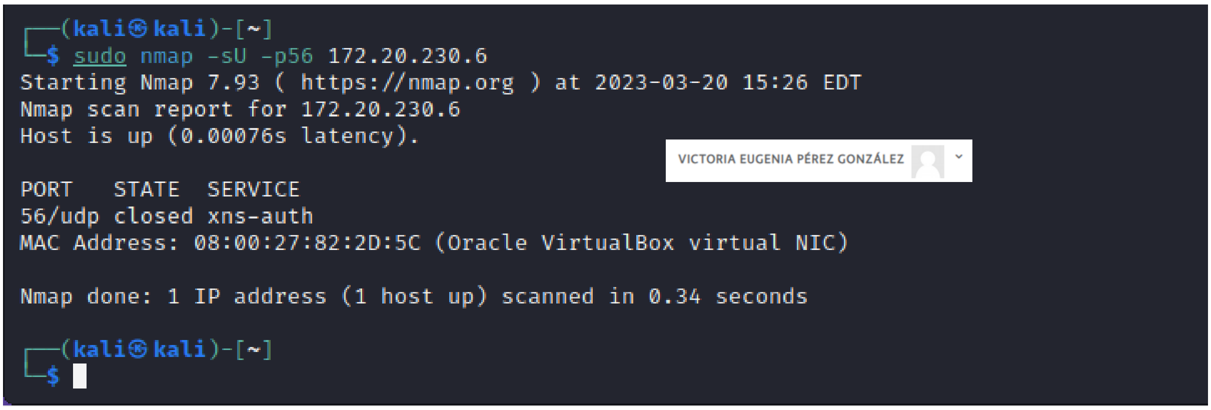
```
03/20/2023-19:21:24.482010  [**] [1:1000005:0] Detectado un escaneo de nmap nulo! [**] [Classification: (null)] [Priority: 3] {TCP} 172.20.230.33:54813 -> 172.20.230.6:
8031
03/20/2023-19:21:24.482058  [**] [1:1000005:0] Detectado un escaneo de nmap nulo! [**] [Classification: (null)] [Priority: 3] {TCP} 172.20.230.33:54813 -> 172.20.230.6:
5050
03/20/2023-19:21:24.482779  [**] [1:1000005:0] Detectado un escaneo de nmap nulo! [**] [Classification: (null)] [Priority: 3] {TCP} 172.20.230.33:54813 -> 172.20.230.6:
3878
```

Regla 3:



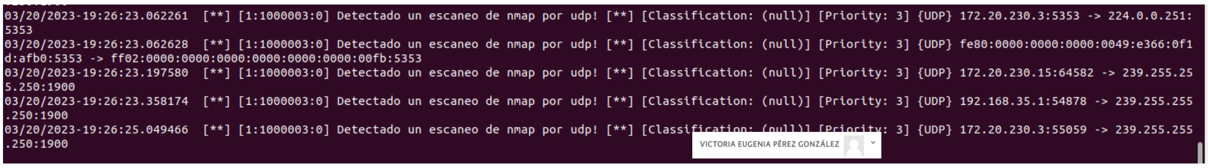
```
alert udp any any -> any 22 (msg:"Detectado un escaneo de nmap por udp!"; sid:1000003;)
```

Se prueba:



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sU -p56 172.20.230.6
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-20 15:26 EDT
Nmap scan report for 172.20.230.6
Host is up (0.00076s latency).

PORT   STATE  SERVICE
56/udp closed xns-auth
MAC Address: 08:00:27:82:2D:5C (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.34 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

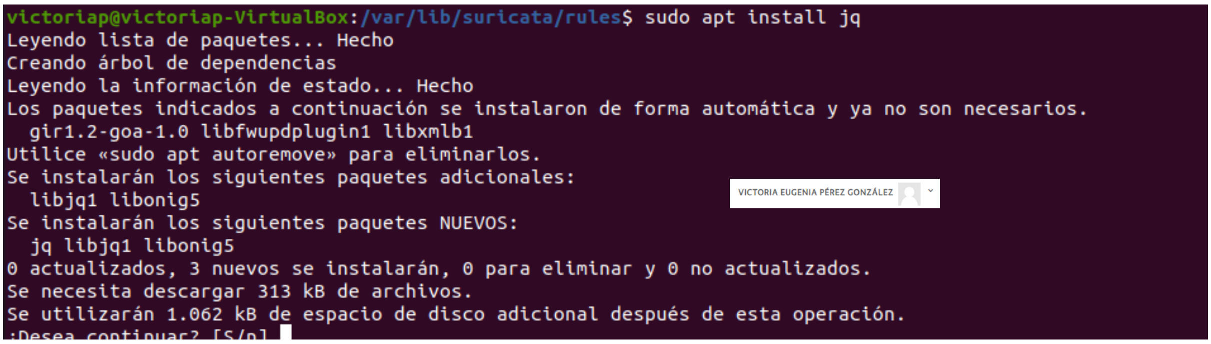
El ids detecta la acción:



```
03/20/2023-19:26:23.062261  [**] [1:1000003:0] Detectado un escaneo de nmap por udp! [**] [Classification: (null)] [Priority: 3] {UDP} 172.20.230.3:5353 -> 224.0.0.251:
5353
03/20/2023-19:26:23.062628  [**] [1:1000003:0] Detectado un escaneo de nmap por udp! [**] [Classification: (null)] [Priority: 3] {UDP} fe80:0000:0000:0000:0049:e366:0f1
d:afb0:5353 -> ff02:0000:0000:0000:0000:0000:0000:00fb:5353
03/20/2023-19:26:23.197580  [**] [1:1000003:0] Detectado un escaneo de nmap por udp! [**] [Classification: (null)] [Priority: 3] {UDP} 172.20.230.15:64582 -> 239.255.25
5.250:1900
03/20/2023-19:26:23.358174  [**] [1:1000003:0] Detectado un escaneo de nmap por udp! [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.35.1:54878 -> 239.255.255
.250:1900
03/20/2023-19:26:25.049466  [**] [1:1000003:0] Detectado un escaneo de nmap por udp! [**] [Classification: (null)] [Priority: 3] {UDP} 172.20.230.3:55059 -> 239.255.255
.250:1900
```

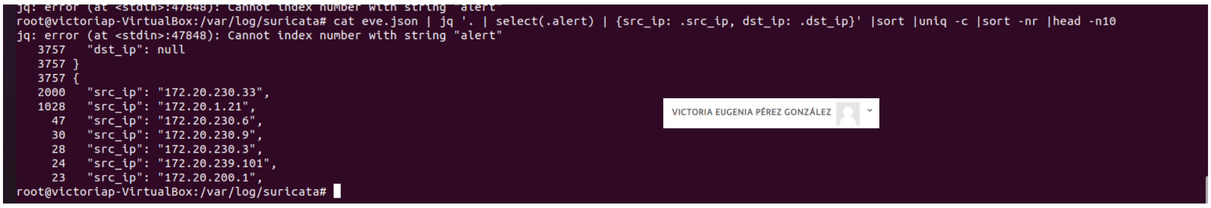
f. Utilizar JQ para realizar una búsqueda concreta (la que tú elijas), sobre el archivo json que se ha generado durante nuestras pruebas

se instala el jq



```
victoriap@victoriap-VirtualBox:/var/lib/suricata/rules$ sudo apt install jq
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  gir1.2-goa-1.0 libfwupdplugin1 libxmlb1
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  libjq1 libonig5
Se instalarán los siguientes paquetes NUEVOS:
  jq libjq1 libonig5
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 313 kB de archivos.
Se utilizarán 1.062 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] 
```

Se ejecuta el comando

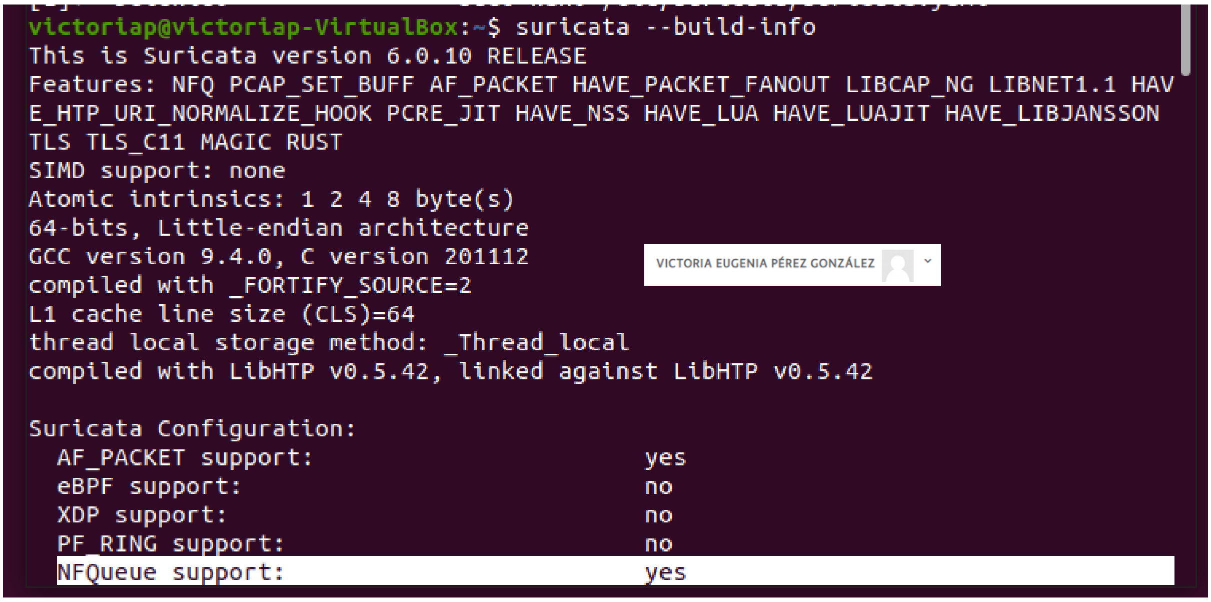


```
jq: error (at <stdin>:47848): Cannot index number with string "alert"
root@victoriap-VirtualBox:/var/log/suricata# cat eve.json | jq '. | select(.alert) | {src_ip: .src_ip, dst_ip: .dst_ip}' |sort |uniq -c |sort -nr |head -n10
jq: error (at <stdin>:47848): Cannot index number with string "alert"
   3757   "dst_ip": null
   3757 }
   3757 {
   2000   "src_ip": "172.20.230.33",
   1028   "src_ip": "172.20.1.21",
     47   "src_ip": "172.20.230.6",
     30   "src_ip": "172.20.230.9",
     28   "src_ip": "172.20.230.3",
     24   "src_ip": "172.20.239.101",
     23   "src_ip": "172.20.200.1",
root@victoriap-VirtualBox:/var/log/suricata# 
```

Este comando cuenta cuántas veces aparece cada combinación de direcciones IP de origen y destino en los eventos de red en "eve.json" que contienen una alerta de Suricata, ordena los resultados por frecuencia y muestra solo las 10 combinaciones más frecuentes.

g. Configurar Suricata como IPS para que, a partir de las reglas que hemos creado en los apartados a, b, c y d, además de alertar, también bloquee esas conexiones.

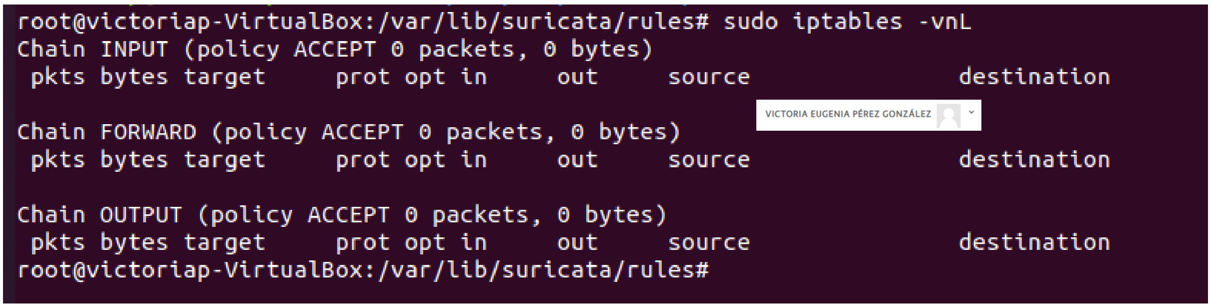
Para configurar suricata como ips hay que mirar primero si NFQueue support:
yes ,para lo que se ejecuta:



```
victoriap@victoriap-VirtualBox:~$ suricata --build-info
This is Suricata version 6.0.10 RELEASE
Features: NFQ PCAP_SET_BUFF AF_PACKET HAVE_PACKET_FANOUT LIBCAP_NG LIBNET1.1 HAV
E_HTP_URI_NORMALIZE_HOOK PCRE_JIT HAVE_NSS HAVE_LUA HAVE_LUAJIT HAVE_LIBJANSSON
TLS TLS_C11 MAGIC RUST
SIMD support: none
Atomic intrinsics: 1 2 4 8 byte(s)
64-bits, Little-endian architecture
GCC version 9.4.0, C version 201112
compiled with _FORTIFY_SOURCE=2
L1 cache line size (CLS)=64
thread local storage method: _Thread_local
compiled with LibHTP v0.5.42, linked against LibHTP v0.5.42

Suricata Configuration:
  AF_PACKET support:                        yes
  eBPF support:                             no
  XDP support:                              no
  PF_RING support:                          no
  NFQueue support:                          yes
```

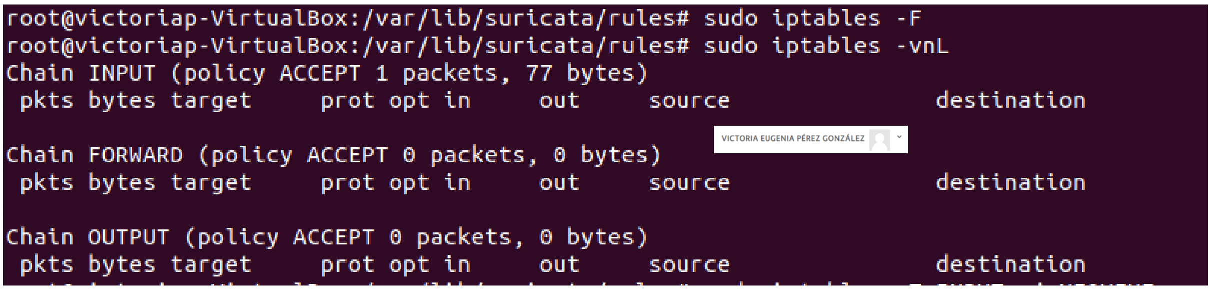
hay que hacer un bypass en iptables para que el suricata sea ips para que derive todo el trafico a suricata.

Ejecutamos



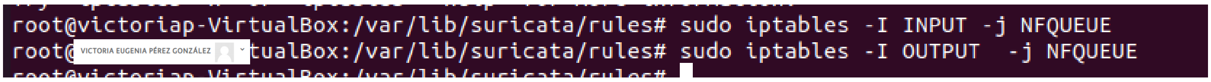
```
root@victoriap-VirtualBox:/var/lib/suricata/rules# sudo iptables -vnL
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@victoriap-VirtualBox:/var/lib/suricata/rules#
```

vemos que hay reglas configuradas y se quitan con sudo iptables -F

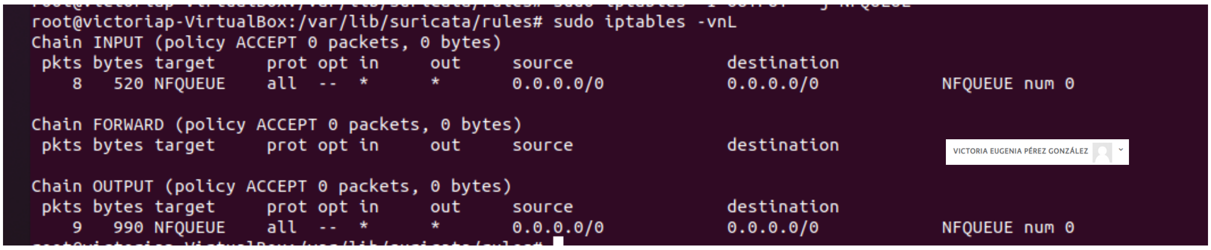


```
root@victoriap-VirtualBox:/var/lib/suricata/rules# sudo iptables -F
root@victoriap-VirtualBox:/var/lib/suricata/rules# sudo iptables -vnL
Chain INPUT (policy ACCEPT 1 packets, 77 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Se configura iptables para el trafico saliente y entrante



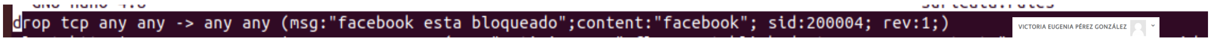
```
root@victoriap-VirtualBox:/var/lib/suricata/rules# sudo iptables -I INPUT -j NFQUEUE
root@victoriap-VirtualBox:/var/lib/suricata/rules# sudo iptables -I OUTPUT  -j NFQUEUE
```



```
root@victoriap-VirtualBox:/var/lib/suricata/rules# sudo iptables -vnL
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    8   520 NFQUEUE    all  --  *      *       0.0.0.0/0            0.0.0.0/0            NFQUEUE num 0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    9   990 NFQUEUE    all  --  *      *       0.0.0.0/0            0.0.0.0/0            NFQUEUE num 0
```

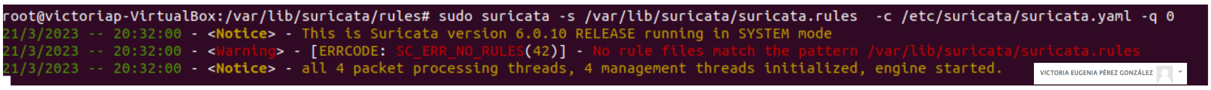
De esta forma todo el trafico pasa por el suricata.

Se cambian los encabezados de las reglas en vez de alert se pone drop y se dejan las anteriores.



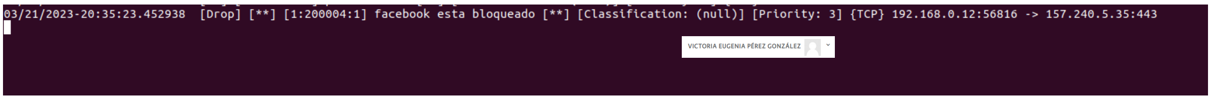
```
drop tcp any any -> any any (msg:"facebook esta bloqueado";content:"facebook"; sid:200004; rev:1;)
```

Se lanza el suricata.

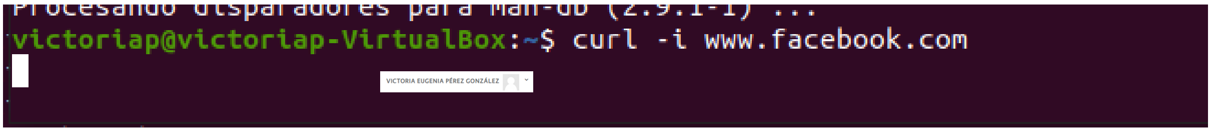


```
root@victoriap-VirtualBox:/var/lib/suricata/rules# sudo suricata -s /var/lib/suricata/suricata.rules  -c /etc/suricata/suricata.yaml -q 0
21/3/2023 -- 20:32:00 - <Notice> - This is Suricata version 6.0.10 RELEASE running in SYSTEM mode
21/3/2023 -- 20:32:00 - <Warning> - [ERRCODE: SC_ERR_NO_RULES(42)] - No rule files match the pattern /var/lib/suricata/suricata.rules
21/3/2023 -- 20:32:00 - <Notice> - all 4 packet processing threads, 4 management threads initialized, engine started.
```

se accede a facebook y se comprueba que lo ha bloqueado



```
03/21/2023-20:35:23.452938  [Drop] [**] [1:200004:1] facebook esta bloqueado [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.12:56816 -> 157.240.5.35:443
```

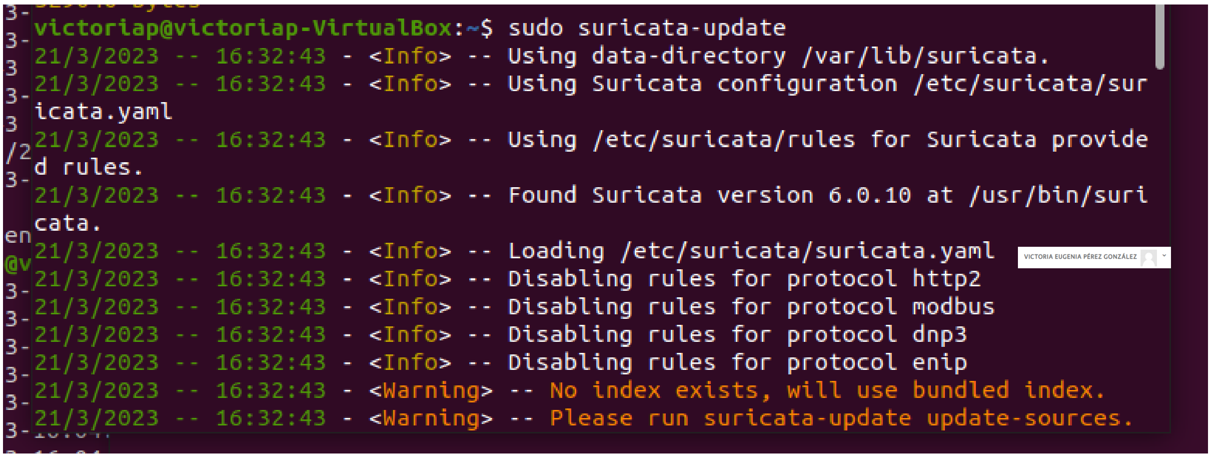


```
Procesando disparadores para man-db (2.9.1-1) ...
victoriap@victoriap-VirtualBox:~$ curl -i www.facebook.com
```

El resto de casos sería igual al cambiar alert por drop,se bloquean las acciones.

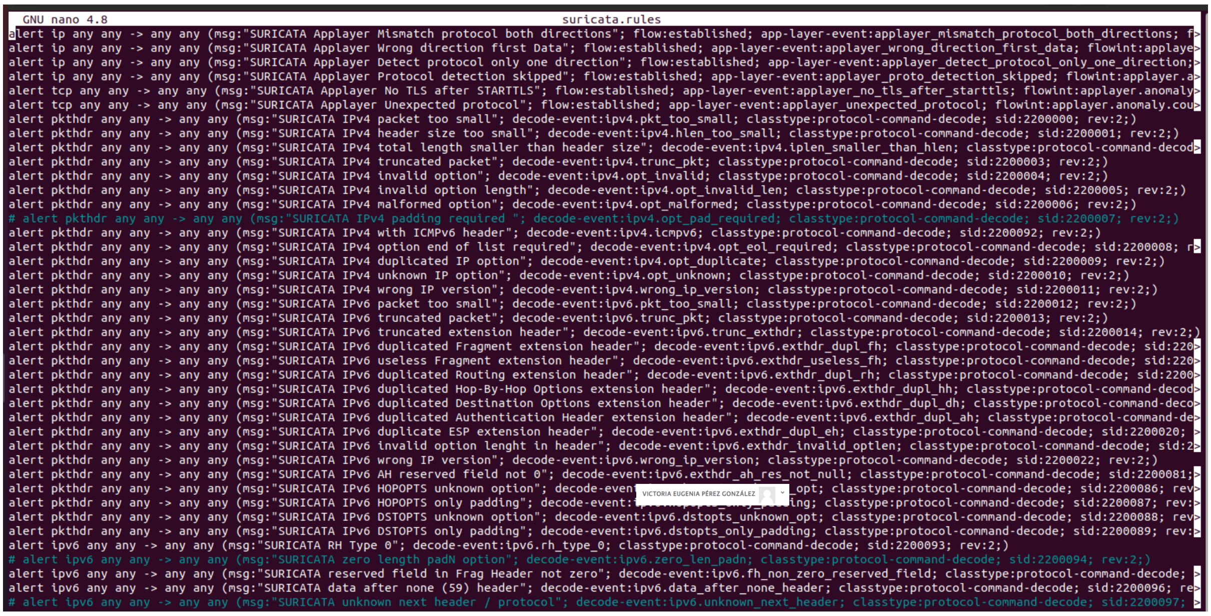
h. Descarga y añade las reglas de la comunidad ‘Emerging Threats’.

Se descargan las reglas.



```
victoriap@victoriap-VirtualBox:~$ sudo suricata-update
21/3/2023 -- 16:32:43 - <Info> -- Using data-directory /var/lib/suricata.
21/3/2023 -- 16:32:43 - <Info> -- Using Suricata configuration /etc/suricata/sur
icata.yaml
21/3/2023 -- 16:32:43 - <Info> -- Using /etc/suricata/rules for Suricata provide
d rules.
21/3/2023 -- 16:32:43 - <Info> -- Found Suricata version 6.0.10 at /usr/bin/suri
cata.
21/3/2023 -- 16:32:43 - <Info> -- Loading /etc/suricata/suricata.yaml
21/3/2023 -- 16:32:43 - <Info> -- Disabling rules for protocol http2
21/3/2023 -- 16:32:43 - <Info> -- Disabling rules for protocol modbus
21/3/2023 -- 16:32:43 - <Info> -- Disabling rules for protocol dnp3
21/3/2023 -- 16:32:43 - <Info> -- Disabling rules for protocol enip
21/3/2023 -- 16:32:43 - <Warning> -- No index exists, will use bundled index.
21/3/2023 -- 16:32:43 - <Warning> -- Please run suricata-update update-sources.
```

se instalan en /var/lib/suricata/suricata.rules.



```
  GNU nano 4.8                                          suricata.rules
alert ip any any -> any any (msg:"SURICATA Applayer Mismatch protocol both directions"; flow:established; app-layer-event:applayer_mismatch_protocol_both_directions; f>
alert ip any any -> any any (msg:"SURICATA Applayer Wrong direction first Data"; flow:established; app-layer-event:applayer_wrong_direction_first_data; flowint:applaye>
alert ip any any -> any any (msg:"SURICATA Applayer Detect protocol only one direction"; flow:established; app-layer-event:applayer_detect_protocol_only_one_direction;>
alert ip any any -> any any (msg:"SURICATA Applayer Protocol detection skipped"; flow:established; app-layer-event:applayer_proto_detection_skipped; flowint:applayer.a>
alert tcp any any -> any any (msg:"SURICATA Applayer No TLS after STARTTLS"; flow:established; app-layer-event:applayer_no_tls_after_starttls; flowint:applayer.anomaly>
alert tcp any any -> any any (msg:"SURICATA Applayer Unexpected protocol"; flow:established; app-layer-event:applayer_unexpected_protocol; flowint:applayer.anomaly.cou>
alert pkthdr any any -> any any (msg:"SURICATA IPv4 packet too small"; decode-event:ipv4.pkt_too_small; classtype:protocol-command-decode; sid:2200000; rev:2;)
alert pkthdr any any -> any any (msg:"SURICATA IPv4 header size too small"; decode-event:ipv4.hlen_too_small; classtype:protocol-command-decode; sid:2200001; rev:2;)
alert pkthdr any any -> any any (msg:"SURICATA IPv4 total length smaller than header size"; decode-event:ipv4.iplen_smaller_than_hlen; classtype:protocol-command-decod>
alert pkthdr any any -> any any (msg:"SURICATA IPv4 truncated packet"; decode-event:ipv4.trunc_pkt; classtype:protocol-command-decode; sid:2200003; rev:2;)
alert pkthdr any any -> any any (msg:"SURICATA IPv4 invalid option"; decode-event:ipv4.opt_invalid; classtype:protocol-command-decode; sid:2200004; rev:2;)
alert pkthdr any any -> any any (msg:"SURICATA IPv4 invalid option length"; decode-event:ipv4.opt_invalid_len; classtype:protocol-command-decode; sid:2200005; rev:2;)
alert pkthdr any any -> any any (msg:"SURICATA IPv4 malformed option"; decode-event:ipv4.opt_malformed; classtype:protocol-command-decode; sid:2200006; rev:2;)
# alert pkthdr any any -> any any (msg:"SURICATA IPv4 padding required "; decode-event:ipv4.opt_pad_required; classtype:protocol-command-decode; sid:2200007; rev:2;)
alert pkthdr any any -> any any (msg:"SURICATA IPv4 with ICMPv6 header"; decode-event:ipv4.icmpv6; classtype:protocol-command-decode; sid:2200092; rev:2;)
alert pkthdr any any -> any any (msg:"SURICATA IPv4 option end of list required"; decode-event:ipv4.opt_eol_required; classtype:protocol-command-decode; sid:2200008; r>
alert pkthdr any any -> any any (msg:"SURICATA IPv4 duplicated IP option"; decode-event:ipv4.opt_duplicate; classtype:protocol-command-decode; sid:2200009; rev:2;)
alert pkthdr any any -> any any (msg:"SURICATA IPv4 unknown IP option"; decode-event:ipv4.opt_unknown; classtype:protocol-command-decode; sid:2200010; rev:2;)
alert pkthdr any any -> any any (msg:"SURICATA IPv4 wrong IP version"; decode-event:ipv4.wrong_ip_version; classtype:protocol-command-decode; sid:2200011; rev:2;)
alert pkthdr any any -> any any (msg:"SURICATA IPv6 packet too small"; decode-event:ipv6.pkt_too_small; classtype:protocol-command-decode; sid:2200012; rev:2;)
alert pkthdr any any -> any any (msg:"SURICATA IPv6 truncated packet"; decode-event:ipv6.trunc_pkt; classtype:protocol-command-decode; sid:2200013; rev:2;)
alert pkthdr any any -> any any (msg:"SURICATA IPv6 truncated extension header"; decode-event:ipv6.trunc_exthdr; classtype:protocol-command-decode; sid:2200014; rev:2;)
alert pkthdr any any -> any any (msg:"SURICATA IPv6 duplicated Fragment extension header"; decode-event:ipv6.exthdr_dupl_fh; classtype:protocol-command-decode; sid:220>
alert pkthdr any any -> any any (msg:"SURICATA IPv6 useless Fragment extension header"; decode-event:ipv6.exthdr_useless_fh; classtype:protocol-command-decode; sid:220>
alert pkthdr any any -> any any (msg:"SURICATA IPv6 duplicated Routing extension header"; decode-event:ipv6.exthdr_dupl_rh; classtype:protocol-command-decode; sid:2200>
alert pkthdr any any -> any any (msg:"SURICATA IPv6 duplicated Hop-By-Hop Options extension header"; decode-event:ipv6.exthdr_dupl_hh; classtype:protocol-command-decod>
alert pkthdr any any -> any any (msg:"SURICATA IPv6 duplicated Destination Options extension header"; decode-event:ipv6.exthdr_dupl_dh; classtype:protocol-command-deco>
alert pkthdr any any -> any any (msg:"SURICATA IPv6 duplicated Authentication Header extension header"; decode-event:ipv6.exthdr_dupl_ah; classtype:protocol-command-de>
alert pkthdr any any -> any any (msg:"SURICATA IPv6 duplicate ESP extension header"; decode-event:ipv6.exthdr_dupl_eh; classtype:protocol-command-decode; sid:2200020; >
alert pkthdr any any -> any any (msg:"SURICATA IPv6 invalid option lenght in header"; decode-event:ipv6.exthdr_invalid_optlen; classtype:protocol-command-decode; sid:2>
alert pkthdr any any -> any any (msg:"SURICATA IPv6 wrong IP version"; decode-event:ipv6.wrong_ip_version; classtype:protocol-command-decode; sid:2200022; rev:2;)
alert pkthdr any any -> any any (msg:"SURICATA IPv6 AH reserved field not 0"; decode-event:ipv6.exthdr_ah_res_not_null; classtype:protocol-command-decode; sid:2200081;>
alert pkthdr any any -> any any (msg:"SURICATA IPv6 HOPOPTS unknown option"; decode-even[illegible]_opt; classtype:protocol-command-decode; sid:2200086; rev>
alert pkthdr any any -> any any (msg:"SURICATA IPv6 HOPOPTS only padding"; decode-event:[illegible]ing; classtype:protocol-command-decode; sid:2200087; rev:>
alert pkthdr any any -> any any (msg:"SURICATA IPv6 DSTOPTS unknown option"; decode-event:ipv6.dstopts_unknown_opt; classtype:protocol-command-decode; sid:2200088; rev>
alert pkthdr any any -> any any (msg:"SURICATA IPv6 DSTOPTS only padding"; decode-event:ipv6.dstopts_only_padding; classtype:protocol-command-decode; sid:2200089; rev:>
alert ipv6 any any -> any any (msg:"SURICATA RH Type 0"; decode-event:ipv6.rh_type_0; classtype:protocol-command-decode; sid:2200093; rev:2;)
# alert ipv6 any any -> any any (msg:"SURICATA zero length padN option"; decode-event:ipv6.zero_len_padn; classtype:protocol-command-decode; sid:2200094; rev:2;)
alert ipv6 any any -> any any (msg:"SURICATA reserved field in Frag Header not zero"; decode-event:ipv6.fh_non_zero_reserved_field; classtype:protocol-command-decode; >
alert ipv6 any any -> any any (msg:"SURICATA data after none (59) header"; decode-event:ipv6.data_after_none_header; classtype:protocol-command-decode; sid:2200096; re>
# alert ipv6 any any -> any any (msg:"SURICATA unknown next header / protocol"; decode-event:ipv6.unknown_next_header; classtype:protocol-command-decode; sid:2200097; >
```

Para el desarrollo de esta práctica, se recomienda utilizar: Máquina Virtual con Ubuntu 20.04 MV con Ubuntu 20.04 cómo máquina cliente en las pruebas. Suricata (IDS/IPS) Nmap (Escaneo de puertos) Openssh (SSH) JQ (lector de registros)