

SNORT

Explicaré como instalar paso a paso el IDS **Snort** en Kali Linux porque no viene recogido dentro de los repositorios.

El primer paso es hacer un back-up de nuestra carpeta de KALI **source.list** para ello ejecutamos el siguiente comando en el terminal como **root**:

```
mv /etc/apt/sources.list /etc/apt/sources.list.bak
```

```
mv /etc/apt/sources.list /etc/apt/sources.list.bak
```

El segundo paso será remover las updates del sistema. Mediante el siguiente comando lo haremos:

```
find /var/lib/apt/lists -type f -exec rm {} \;
```

```
(root@kali) [/home/kali]  
# find var/lib/apt/lists -type f -exec rm {} \;
```

El tercer paso es abrir el contenido de source.list y añadir los nuevos repositorios para hacer el update.

```
sudo nano /etc/apt/sources.list
```

En el archivo abierto por nano copiamos los siguientes links y guardamos.

```
deb [arch=arm64] http://ports.ubuntu.com/ubuntu-ports focal main  
restricted universe multiverse  
  
deb [arch=arm64] http://ports.ubuntu.com/ubuntu-ports focal-  
updates main restricted universe multiverse  
  
deb [arch=arm64] http://ports.ubuntu.com/ubuntu-ports focal-  
security main restricted universe multiverse
```

```
deb [arch=i386,amd64] http://us.archive.ubuntu.com/ubuntu/ focal
main restricted universe multiverse
```

```
deb [arch=i386,amd64] http://us.archive.ubuntu.com/ubuntu/ focal-
updates main restricted universe multiverse
```

```
deb [arch=i386,amd64] http://security.ubuntu.com/ubuntu focal-
security main restricted universe multiverse
```

```
File Actions Edit View Help
GNU nano 7.2 /etc/apt/sources.list
deb [arch=arm64] http://ports.ubuntu.com/ubuntu-ports focal main restricted universe multiver>
deb [arch=arm64] http://ports.ubuntu.com/ubuntu-ports focal-updates main restricted universe >
deb [arch=arm64] http://ports.ubuntu.com/ubuntu-ports focal-security main restricted universe>
deb [arch=i386,amd64] http://us.archive.ubuntu.com/ubuntu/ focal main restricted universe mul>
deb [arch=i386,amd64] http://us.archive.ubuntu.com/ubuntu/ focal-updates main restricted univ>
deb [arch=i386,amd64] http://security.ubuntu.com/ubuntu focal-security main restricted univer>
```

El cuarto paso es añadir las claves públicas de estos repositorios:

sudo	key	keyserver	recv	keys
apt	adv	keyserver.ubuntu.com		3B4FE6ACC0B21F32
sudo	key	keyserver	recv	keys
apt	adv	keyserver.ubuntu.com		871920D1991BC93C

```
(root@kali)-[/home/kali]
# sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 3B4FE6ACC0B21F32
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
Executing: /tmp/apt-key-gpghome.8ahxZoMJUm/gpg.1.sh --keyserver keyserver.ubuntu.com --recv-keys 3B4FE6ACC0B21F32
gpg: key 3B4FE6ACC0B21F32: "Ubuntu Archive Automatic Signing Key (2012) <ftpmaster@ubuntu.com>" not changed
gpg: Total number processed: 1
gpg: unchanged: 1

(root@kali)-[/home/kali]
# sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 871920D1991BC93C
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
Executing: /tmp/apt-key-gpghome.y3b4jGpsbY/gpg.1.sh --keyserver keyserver.ubuntu.com --recv-keys 871920D1991BC93C
gpg: key 871920D1991BC93C: "Ubuntu Archive Automatic Signing Key (2018) <ftpmaster@ubuntu.com>" not changed
gpg: Total number processed: 1
gpg: unchanged: 1

(root@kali)-[/home/kali]
```

El quinto paso es actualizar los repositorios con el comando

```
sudo apt-get update
```

```
(root@kali)-[/home/kali]  
# sudo apt-get update
```

El último paso es instalar snort para ello usamos

```
sudo apt install snort
```

```
—# sudo apt install snort  
Reading package lists ... Done  
Building dependency tree ... Done  
Reading state information ... Done  
snort is already the newest version (2.9.7.0-5build1).  
0 upgraded, 0 newly installed, 0 to remove and 64 not upgraded.
```

Y ya lo tenemos instalado.

Detección de un ataque con SNORT

Procedemos a crear nuestra propia regla dentro de SNORT para que nos salte la alerta.

Para ello nos vamos a la kali y configuramos las reglas de snort mediante un editor de texto.

```
nano /etc/snort/snort.conf
```

Podemos observar las configuraciones que viene por defecto en el programa SNORT.

Tendremos que hacer un scan a nuestro dispositivo para ver si cumplimos con los requisitos de SNORT para usarlo.

Con el comando:

Joaquim Chagas Neto - SNORT CCIEX

```
sudo snort -T -c /etc/snort/snort.conf -i eth0
```

Hacemos el escaneo.

```
[ Number of patterns truncated to 20 bytes: 1039 ]
pcap DAQ configured to passive.
Acquiring network traffic from "eth0".

--= Initialization Complete ==--

o''~  -*> Snort! <*-
''''  Version 2.9.7.0 GRE (Build 149)
      By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
      Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
      Copyright (C) 1998-2013 Sourcefire, Inc., et al.
      Using libpcap version 1.10.4 (with TPACKET_V3)
      Using PCRE version: 8.39 2016-06-14
      Using ZLIB version: 1.2.13

      Rules Engine: SF_SNORT_DETECTION_ENGINE Version 2.4 <Build 1>
      Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
      Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
      Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
      Preprocessor Object: SF_GTP Version 1.1 <Build 1>
      Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
      Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
      Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
      Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
      Preprocessor Object: SF_DNS Version 1.1 <Build 4>
      Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
      Preprocessor Object: SF_POP Version 1.0 <Build 1>
      Preprocessor Object: SF_SIP Version 1.1 <Build 1>
      Preprocessor Object: SF_SSH Version 1.1 <Build 3>
      Preprocessor Object: SF_SDF Version 1.1 <Build 1>

Snort successfully validated the configuration!
Snort exiting
```

Nos saldrá un mensaje de configuración validada.

En este punto creamos nuestra propia regla

La metemos dentro del snort.conf

```
596 #include $RULE_PATH/exploit-kit.rules
597 include $RULE_PATH/exploit.rules
598 include $RULE_PATH/locals.rules
599 #include $RULE_PATH/file-executable.rules
600 #include $RULE_PATH/file-flash.rules
```

Para ver cómo funciona un poco el IDS SNORT lo ponemos en modo captura con el siguiente comando:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

```

$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
12/18-18:26:40.563756 [**] [1:1384:8] MISC UPnP malformed advertisement [**] [Classification: Misc Attack] [Priority: 2] {UDP} 192.168.1.1:59646 → 239.255.255.250:1900
12/18-18:26:40.563823 [**] [1:1384:8] MISC UPnP malformed advertisement [**] [Classification: Misc Attack] [Priority: 2] {UDP} 192.168.1.1:59646 → 239.255.255.250:1900
12/18-18:26:40.563824 [**] [1:1384:8] MISC UPnP malformed advertisement [**] [Classification: Misc Attack] [Priority: 2] {UDP} 192.168.1.1:59646 → 239.255.255.250:1900
12/18-18:26:40.563899 [**] [1:1384:8] MISC UPnP malformed advertisement [**] [Classification: Misc Attack] [Priority: 2] {UDP} 192.168.1.1:59646 → 239.255.255.250:1900
12/18-18:26:40.563950 [**] [1:1384:8] MISC UPnP malformed advertisement [**] [Classification: Misc Attack] [Priority: 2] {UDP} 192.168.1.1:59646 → 239.255.255.250:1900
12/18-18:26:40.563951 [**] [1:1384:8] MISC UPnP malformed advertisement [**] [Classification: Misc Attack] [Priority: 2] {UDP} 192.168.1.1:59646 → 239.255.255.250:1900
12/18-18:26:40.564205 [**] [1:1384:8] MISC UPnP malformed advertisement [**] [Classification: Misc Attack] [Priority: 2] {UDP} 192.168.1.1:59646 → 239.255.255.250:1900

```

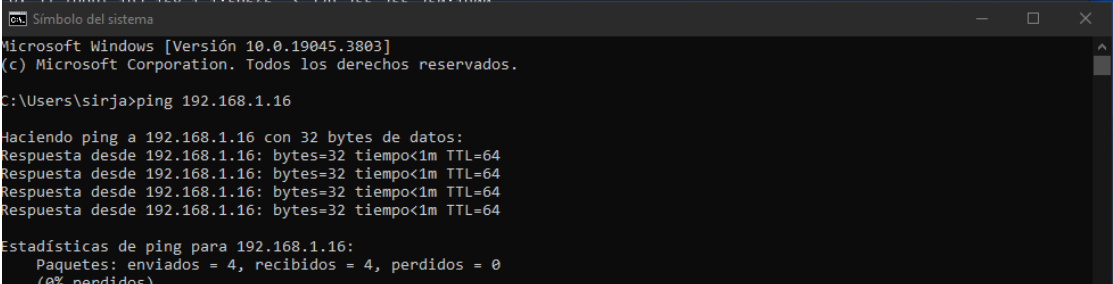
Observamos todos los movimientos que están ocurriendo en nuestra red y que SNORT nos proporciona información de ello; desde la fecha y hora que se realiza, la IP del atacante, puertos, el tipo de ataque, su clasificación y la prioridad y el tipo de protocolo usados.

En la siguiente imagen observamos un ping que he hecho desde mi sistema Windows a la máquina Kali. Nos da detalles sobre la hora ejecutada, el tipo de “ataque” ICMP PING, su clasificación, la prioridad, la IP del atacante...

```

$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
12/18-18:35:25.468883 [**] [1:382:7] ICMP PING Windows [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.15 → 192.168.1.16
12/18-18:35:25.468883 [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.15 → 192.168.1.16
12/18-18:35:25.468979 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.16 → 192.168.1.15
12/18-18:35:26.470606 [**] [1:382:7] ICMP PING Windows [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.15 → 192.168.1.16
12/18-18:35:26.470606 [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.15 → 192.168.1.16
12/18-18:35:26.470621 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.16 → 192.168.1.15
12/18-18:35:26.665284 [**] [1:1384:8] MISC UPnP malformed advertisement [**] [Classification: Misc Attack] [Priority: 2] {UDP} 192.168.1.1:59646 → 239.255.255.250:1900

```



```

C:\Users\sirja>ping 192.168.1.16

Haciendo ping a 192.168.1.16 con 32 bytes de datos:
Respuesta desde 192.168.1.16: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.1.16: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.1.16: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.1.16: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.1.16:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
            (0% perdidos),

```

En esta ocasión he utilizado un programa llamado Sparta para ver que es lo que sucede mientras analizamos la red para tener más información sobre cómo funciona.

Hosts	Services	Tools	Services	Scripts	Information	Notes	
OS	Host						
7	192.168.1.15						
Log							
Progress	Tool	Host	Start time	End time	Status		
<div></div>	nmap (stage 2)	192.168.1.15	18 Dec 2023 18:52:47		Running		
<div></div>	nmap (stage 1)	192.168.1.15	18 Dec 2023 18:52:45	18 Dec 2023 18:52:47	Finished		

Podemos ver que el programa está ejecutando distintos escaneos (ICMP, XMAS) en nuestra red para obtener información a través del protocolo TCP.