Instalación de un IDS/IPS

Suricata

Pedro Egea Ortega

Índice

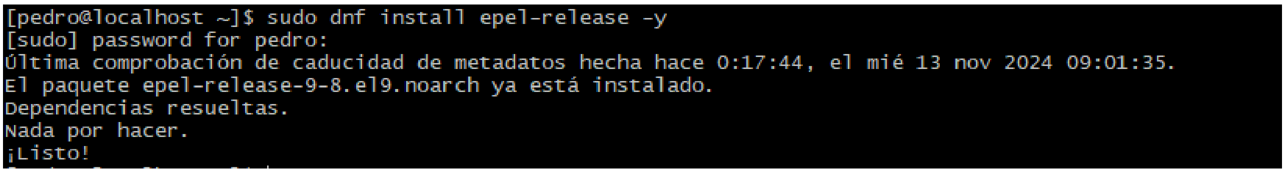
Instalación Suricata

Actualizamos el sistema:

$ sudo dnf update -y

Primero necesitamos añadir una serie de repositorios:

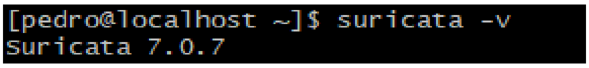
$ sudo dnf install epel-release -y



```
[pedro@localhost ~]$ sudo dnf install epel-release -y
[sudo] password for pedro:
Última comprobación de caducidad de metadatos hecha hace 0:17:44, el mié 13 nov 2024 09:01:35.
El paquete epel-release-9-8.el9.noarch ya está instalado.
Dependencias resueltas.
Nada por hacer.
¡Listo!
```

Ahora ya podemos proceder a instalar Suricata:

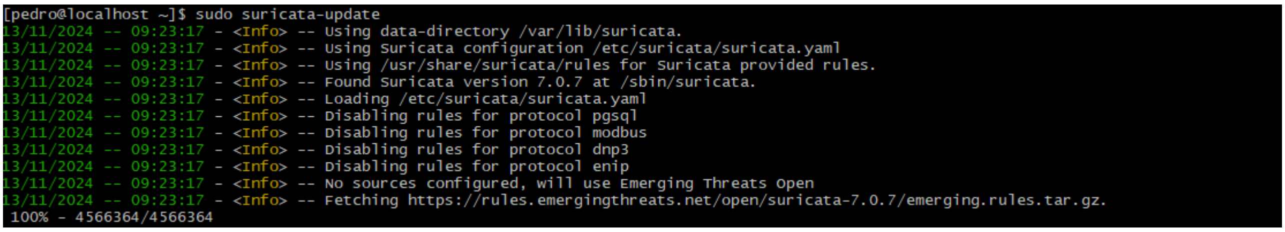
$ sudo dnf install suricata -y



```
[pedro@localhost ~]$ suricata -v
Suricata 7.0.7
```

Descargamos un conjunto de reglas predeterminadas:

$ sudo suricata-update



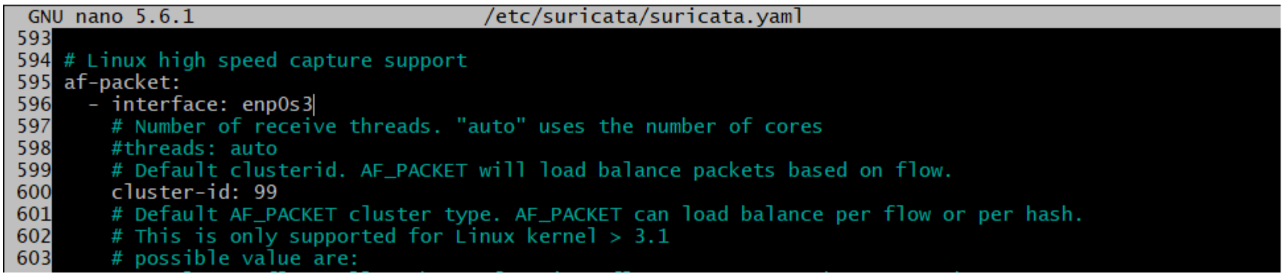
```
[pedro@localhost ~]$ sudo suricata-update
13/11/2024 -- 09:23:17 - <Info> -- Using data-directory /var/lib/suricata.
13/11/2024 -- 09:23:17 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
13/11/2024 -- 09:23:17 - <Info> -- Using /usr/share/suricata/rules for Suricata provided rules.
13/11/2024 -- 09:23:17 - <Info> -- Found Suricata version 7.0.7 at /sbin/suricata.
13/11/2024 -- 09:23:17 - <Info> -- Loading /etc/suricata/suricata.yaml
13/11/2024 -- 09:23:17 - <Info> -- Disabling rules for protocol pgsql
13/11/2024 -- 09:23:17 - <Info> -- Disabling rules for protocol modbus
13/11/2024 -- 09:23:17 - <Info> -- Disabling rules for protocol dnp3
13/11/2024 -- 09:23:17 - <Info> -- Disabling rules for protocol enip
13/11/2024 -- 09:23:17 - <Info> -- No sources configured, will use Emerging Threats Open
13/11/2024 -- 09:23:17 - <Info> -- Fetching https://rules.emergingthreats.net/open/suricata-7.0.7/emerging.rules.tar.gz.
 100% - 4566364/4566364
```

Configuración de Suricata

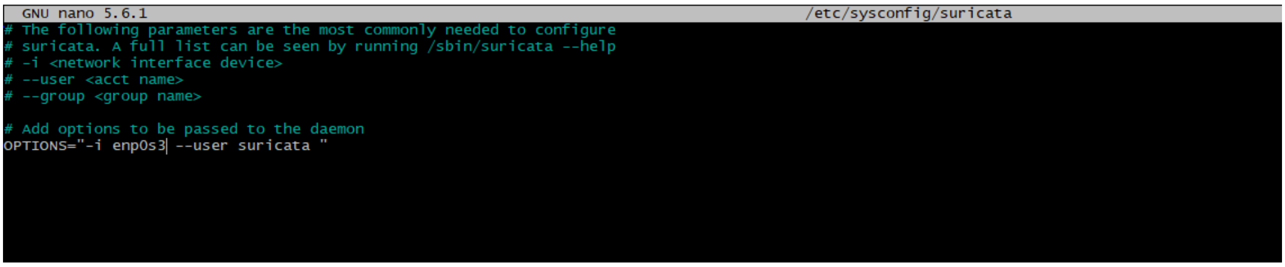
Editamos el archivo de configuración:

$ sudo nano -l /etc/suricata/suricata.yaml

Indicamos la interfaz de red por la que debe capturar el tráfico

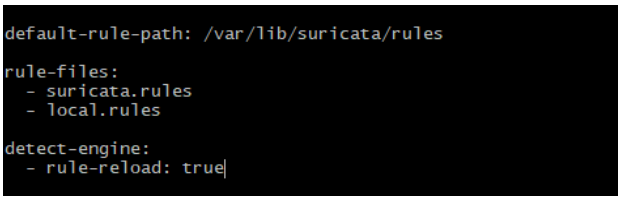


```
  GNU nano 5.6.1                        /etc/suricata/suricata.yaml

# Linux high speed capture support
af-packet:
  - interface: enp0s3
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
    cluster-id: 99
    # Default AF_PACKET cluster type. AF_PACKET can load balance per flow or per hash.
    # This is only supported for Linux kernel > 3.1
    # possible value are:
```



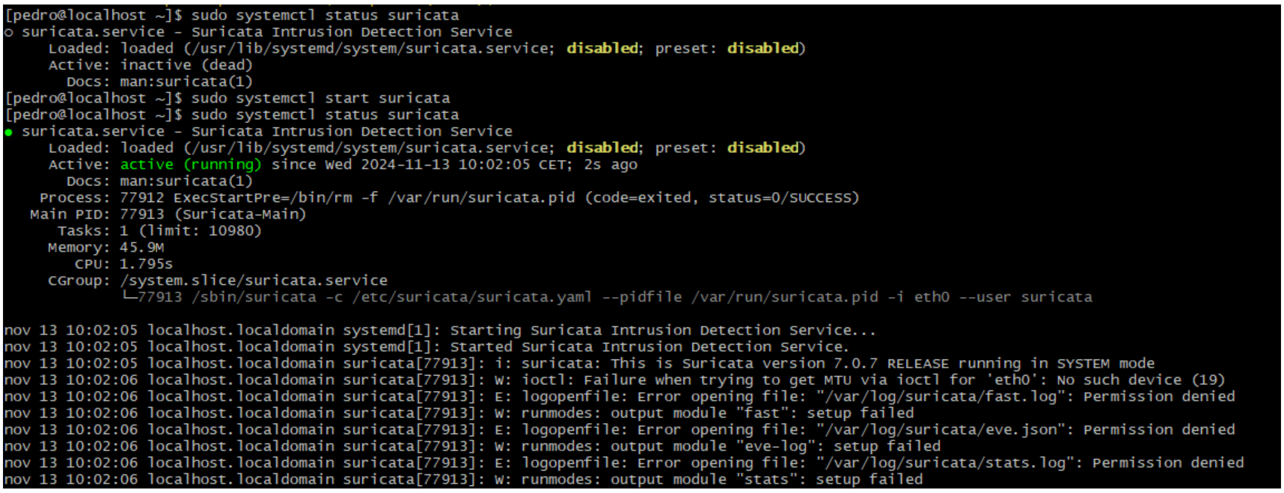
```
  GNU nano 5.6.1                                                        /etc/sysconfig/suricata
# The following parameters are the most commonly needed to configure
# suricata. A full list can be seen by running /sbin/suricata --help
# -i <network interface device>
# --user <acct name>
# --group <group name>

# Add options to be passed to the daemon
OPTIONS="-i enp0s3 --user suricata "
```

Activamos la opción para que se actualice la norma automáticamente



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

rule-files:
  - suricata.rules
  - local.rules

detect-engine:
  - rule-reload: true
```

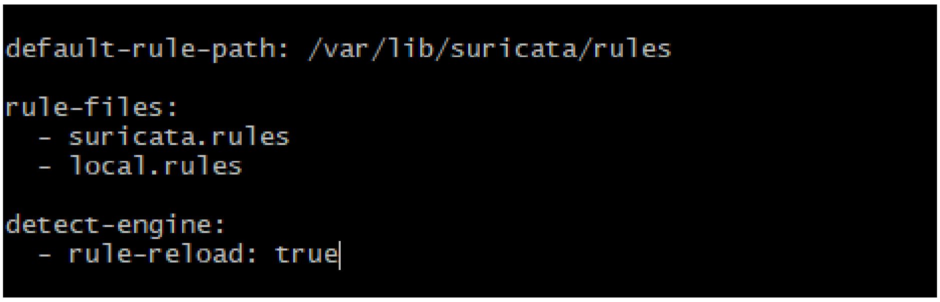
Comprobamos que el servicio esté funcionando



```
[pedro@localhost ~]$ sudo systemctl status suricata
○ suricata.service - Suricata Intrusion Detection Service
     Loaded: loaded (/usr/lib/systemd/system/suricata.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:suricata(1)
[pedro@localhost ~]$ sudo systemctl start suricata
[pedro@localhost ~]$ sudo systemctl status suricata
● suricata.service - Suricata Intrusion Detection Service
     Loaded: loaded (/usr/lib/systemd/system/suricata.service; disabled; preset: disabled)
     Active: active (running) since Wed 2024-11-13 10:02:05 CET; 2s ago
       Docs: man:suricata(1)
    Process: 77912 ExecStartPre=/bin/rm -f /var/run/suricata.pid (code=exited, status=0/SUCCESS)
   Main PID: 77913 (Suricata-Main)
      Tasks: 1 (limit: 10980)
     Memory: 45.9M
        CPU: 1.795s
     CGroup: /system.slice/suricata.service
             └─77913 /sbin/suricata -c /etc/suricata/suricata.yaml --pidfile /var/run/suricata.pid -i eth0 --user suricata

nov 13 10:02:05 localhost.localdomain systemd[1]: Starting Suricata Intrusion Detection Service...
nov 13 10:02:05 localhost.localdomain systemd[1]: Started Suricata Intrusion Detection Service.
nov 13 10:02:05 localhost.localdomain suricata[77913]: i: suricata: This is Suricata version 7.0.7 RELEASE running in SYSTEM mode
nov 13 10:02:06 localhost.localdomain suricata[77913]: W: ioctl: Failure when trying to get MTU via ioctl for 'eth0': No such device (19)
nov 13 10:02:06 localhost.localdomain suricata[77913]: E: logopenfile: Error opening file: "/var/log/suricata/fast.log": Permission denied
nov 13 10:02:06 localhost.localdomain suricata[77913]: W: runmodes: output module "fast": setup failed
nov 13 10:02:06 localhost.localdomain suricata[77913]: E: logopenfile: Error opening file: "/var/log/suricata/eve.json": Permission denied
nov 13 10:02:06 localhost.localdomain suricata[77913]: W: runmodes: output module "eve-log": setup failed
nov 13 10:02:06 localhost.localdomain suricata[77913]: E: logopenfile: Error opening file: "/var/log/suricata/stats.log": Permission denied
nov 13 10:02:06 localhost.localdomain suricata[77913]: W: runmodes: output module "stats": setup failed
```

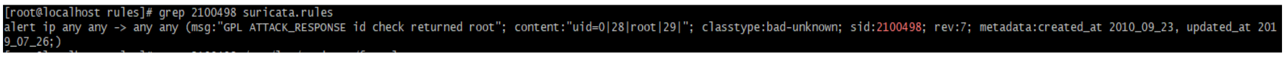
Comprobación regla de prueba

Las reglas se guardan en el archivo definido dentro del archivo de configuración



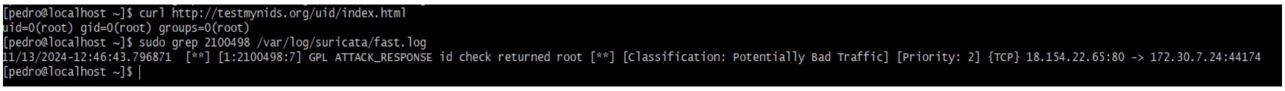
```
default-rule-path: /var/lib/suricata/rules

rule-files:
  - suricata.rules
  - local.rules

detect-engine:
  - rule-reload: true
```

Hacemos un grep de la norma para comprobar que exixte



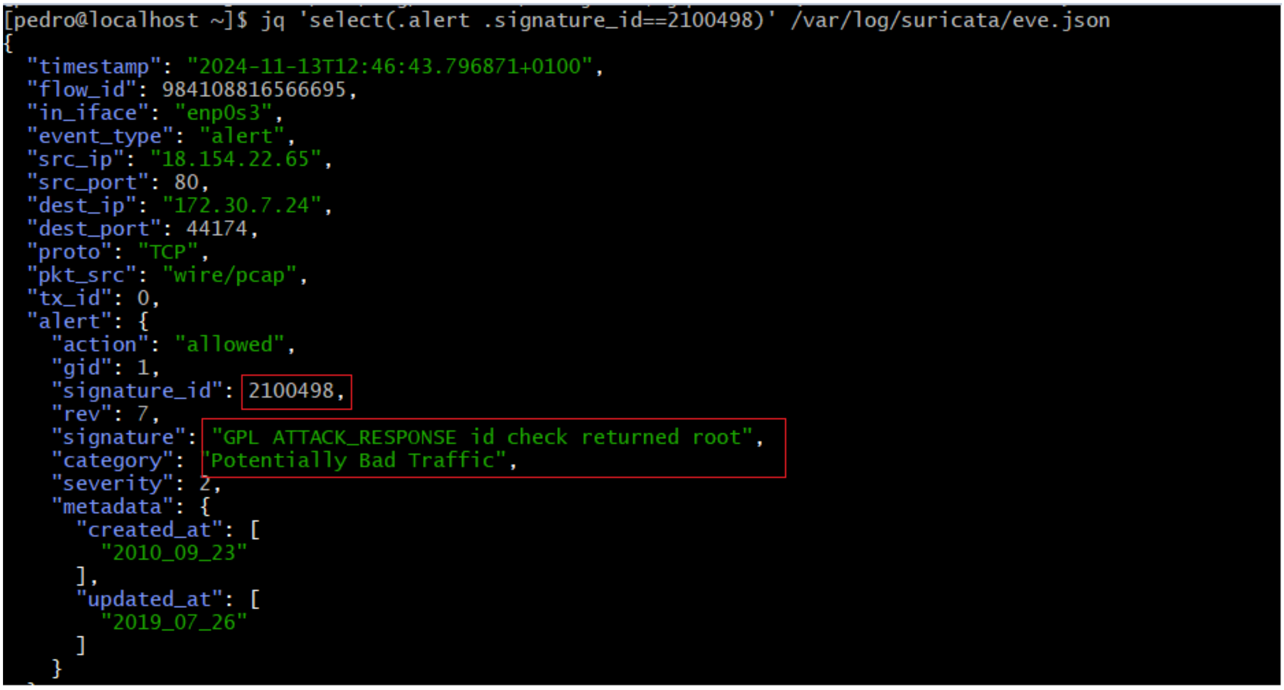
```
[root@localhost rules]# grep 2100498 suricata.rules
alert ip any any -> any any (msg:"GPL ATTACK_RESPONSE id check returned root"; content:"uid=0|28|root|29|"; classtype:bad-unknown; sid:2100498; rev:7; metadata:created_at 2010_09_23, updated_at 201
9_07_26;)
```

Hacemos el curl de ejemplo para fingir un ataque y revisamos el archivo de los logs.



```
[pedro@localhost ~]$ curl http://testmynids.org/uid/index.html
uid=0(root) gid=0(root) groups=0(root)
[pedro@localhost ~]$ sudo grep 2100498 /var/log/suricata/fast.log
11/13/2024-12:46:43.796871  [**] [1:2100498:7] GPL ATTACK_RESPONSE id check returned root [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 18.154.22.65:80 -> 172.30.7.24:44174
[pedro@localhost ~]$ |
```

Suricata también registra eventos en el archivo /var/log/suricata/eve.log, utilizando formato JSON. Para revisar esta información, analizamos el archivo de logs con la herramienta jq, que nos permite leer y filtrar fácilmente las entradas en formato JSON.

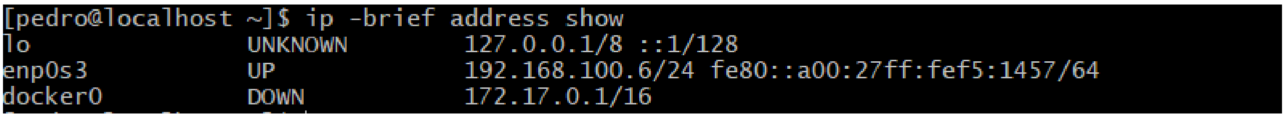


```
[pedro@localhost ~]$ jq 'select(.alert .signature_id==2100498)' /var/log/suricata/eve.json
{
  "timestamp": "2024-11-13T12:46:43.796871+0100",
  "flow_id": 984108816566695,
  "in_iface": "enp0s3",
  "event_type": "alert",
  "src_ip": "18.154.22.65",
  "src_port": 80,
  "dest_ip": "172.30.7.24",
  "dest_port": 44174,
  "proto": "TCP",
  "pkt_src": "wire/pcap",
  "tx_id": 0,
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 2100498,
    "rev": 7,
    "signature": "GPL ATTACK_RESPONSE id check returned root",
    "category": "Potentially Bad Traffic",
    "severity": 2,
    "metadata": {
      "created_at": [
        "2010_09_23"
      ],
      "updated_at": [
        "2019_07_26"
      ]
    }
```

Configuración de Suricata como IPS

Primero necesitamos comprobar nuestra firma

$ ip -brief address show



```
[pedro@localhost ~]$ ip -brief address show
lo               UNKNOWN        127.0.0.1/8 ::1/128
enp0s3           UP             192.168.100.6/24 fe80::a00:27ff:fef5:1457/64
docker0          DOWN           172.17.0.1/16
```

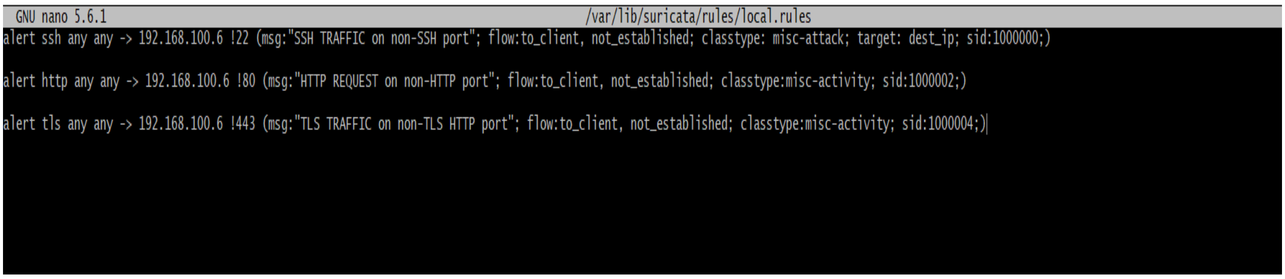
Escribimos la siguiente regla:

alert ssh any any -> 192.168.100.6 !22 (msg:"SSH TRAFFIC on non-SSH port"; flow:to_client, not_established; classtype: misc-attack; target: dest_ip; sid:1000000;)

Y añadimos alguna más personalizada:

alert http any any -> 192.168.100.6 !80 (msg:"HTTP REQUEST on non-HTTP port"; flow:to_client, not_established; classtype:misc-activity; sid:1000002;)

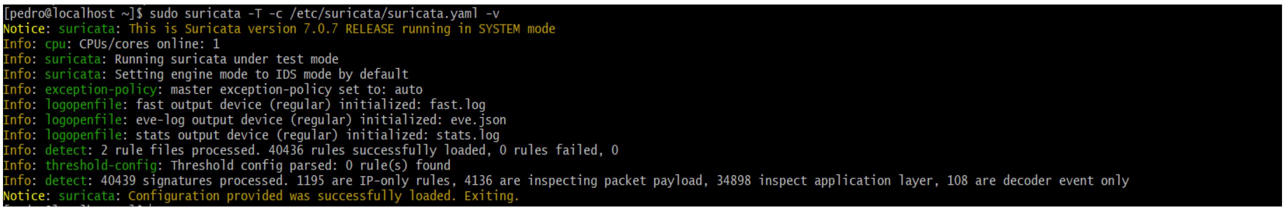
alert tls any any -> 192.168.100.6 !443 (msg:"TLS TRAFFIC on non-TLS HTTP port"; flow:to_client, not_established; classtype:misc-activity; sid:1000004;)



```
  GNU nano 5.6.1                                   /var/lib/suricata/rules/local.rules
alert ssh any any -> 192.168.100.6 !22 (msg:"SSH TRAFFIC on non-SSH port"; flow:to_client, not_established; classtype: misc-attack; target: dest_ip; sid:1000000;)

alert http any any -> 192.168.100.6 !80 (msg:"HTTP REQUEST on non-HTTP port"; flow:to_client, not_established; classtype:misc-activity; sid:1000002;)

alert tls any any -> 192.168.100.6 !443 (msg:"TLS TRAFFIC on non-TLS HTTP port"; flow:to_client, not_established; classtype:misc-activity; sid:1000004;)
```

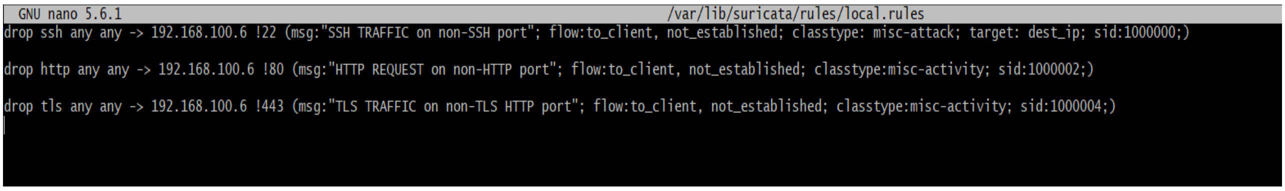
Nos aseguramos de validar la configuración

$ sudo suricata -T -c /etc/suricata/suricata.yaml -v



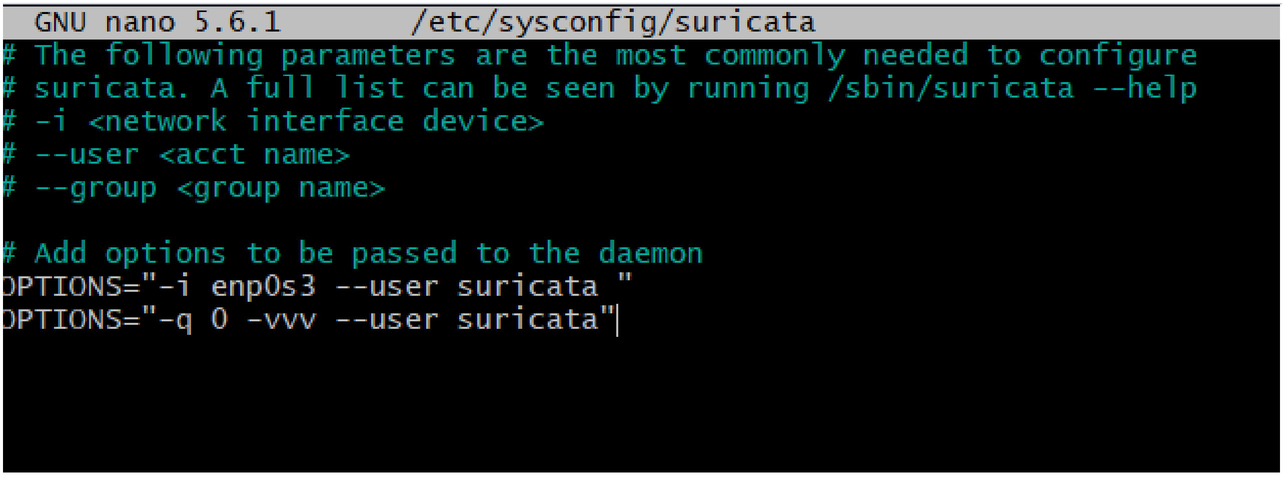
```
[pedro@localhost ~]$ sudo suricata -T -c /etc/suricata/suricata.yaml -v
Notice: suricata: This is Suricata version 7.0.7 RELEASE running in SYSTEM mode
Info: cpu: CPUs/cores online: 1
Info: suricata: Running suricata under test mode
Info: suricata: Setting engine mode to IDS mode by default
Info: exception-policy: master exception-policy set to: auto
Info: logopenfile: fast output device (regular) initialized: fast.log
Info: logopenfile: eve-log output device (regular) initialized: eve.json
Info: logopenfile: stats output device (regular) initialized: stats.log
Info: detect: 2 rule files processed. 40436 rules successfully loaded, 0 rules failed, 0
Info: threshold-config: Threshold config parsed: 0 rule(s) found
Info: detect: 40439 signatures processed. 1195 are IP-only rules, 4136 are inspecting packet payload, 34898 inspect application layer, 108 are decoder event only
Notice: suricata: Configuration provided was successfully loaded. Exiting.
```

Ahora que tenemos las reglas definidas, si queremos que actuen como IPS debemos modificar **alert** por **drop**.



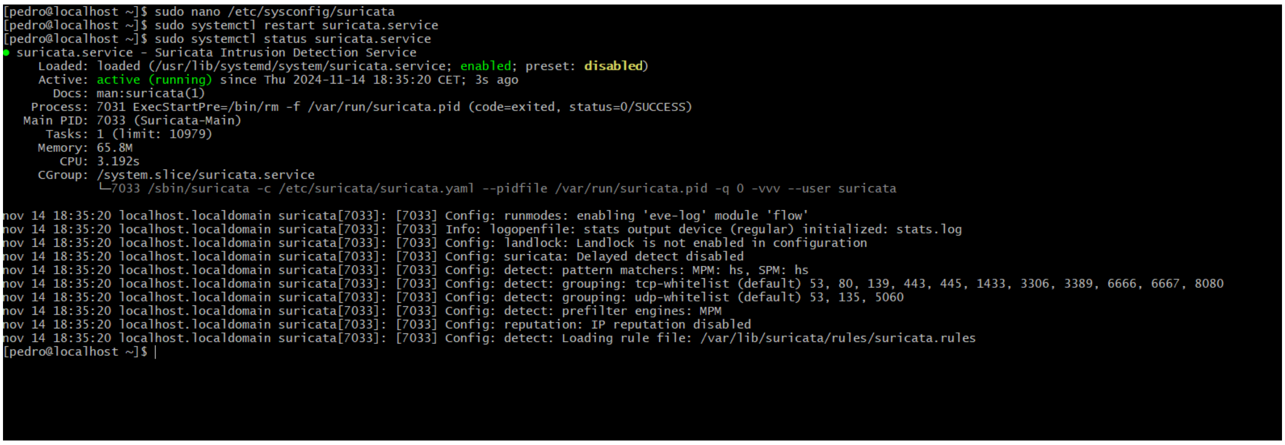
```
  GNU nano 5.6.1                                                              /var/lib/suricata/rules/local.rules
drop ssh any any -> 192.168.100.6 !22 (msg:"SSH TRAFFIC on non-SSH port"; flow:to_client, not_established; classtype: misc-attack; target: dest_ip; sid:1000000;)

drop http any any -> 192.168.100.6 !80 (msg:"HTTP REQUEST on non-HTTP port"; flow:to_client, not_established; classtype:misc-activity; sid:1000002;)

drop tls any any -> 192.168.100.6 !443 (msg:"TLS TRAFFIC on non-TLS HTTP port"; flow:to_client, not_established; classtype:misc-activity; sid:1000004;)
```

Suricata se ejecuta en modo IDS de forma predeterminada, lo que significa que no bloqueará activamente el tráfico de red. Para cambiar al modo IPS, debemos editar /etc/sysconfig/suricata (el archivo de configuración de Suricata).



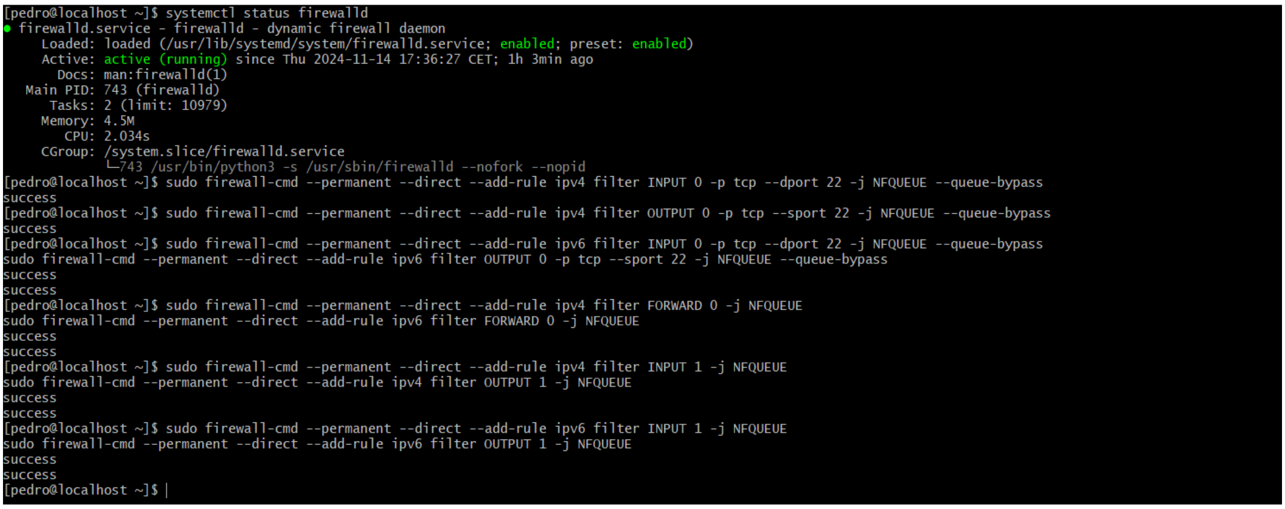
```
  GNU nano 5.6.1         /etc/sysconfig/suricata
# The following parameters are the most commonly needed to configure
# suricata. A full list can be seen by running /sbin/suricata --help
# -i <network interface device>
# --user <acct name>
# --group <group name>

# Add options to be passed to the daemon
OPTIONS="-i enp0s3 --user suricata "
OPTIONS="-q 0 -vvv --user suricata"
```

Reiniciamos el servicio y comprobamos que todo funciona correctamente



```
[pedro@localhost ~]$ sudo nano /etc/sysconfig/suricata
[pedro@localhost ~]$ sudo systemctl restart suricata.service
[pedro@localhost ~]$ sudo systemctl status suricata.service
● suricata.service - Suricata Intrusion Detection Service
     Loaded: loaded (/usr/lib/systemd/system/suricata.service; enabled; preset: disabled)
     Active: active (running) since Thu 2024-11-14 18:35:20 CET; 3s ago
       Docs: man:suricata(1)
    Process: 7031 ExecStartPre=/bin/rm -f /var/run/suricata.pid (code=exited, status=0/SUCCESS)
   Main PID: 7033 (Suricata-Main)
      Tasks: 1 (limit: 10979)
     Memory: 65.8M
        CPU: 3.192s
     CGroup: /system.slice/suricata.service
             └─7033 /sbin/suricata -c /etc/suricata/suricata.yaml --pidfile /var/run/suricata.pid -q 0 -vvv --user suricata

nov 14 18:35:20 localhost.localdomain suricata[7033]: [7033] Config: runmodes: enabling 'eve-log' module 'flow'
nov 14 18:35:20 localhost.localdomain suricata[7033]: [7033] Info: logopenfile: stats output device (regular) initialized: stats.log
nov 14 18:35:20 localhost.localdomain suricata[7033]: [7033] Config: landlock: Landlock is not enabled in configuration
nov 14 18:35:20 localhost.localdomain suricata[7033]: [7033] Config: suricata: Delayed detect disabled
nov 14 18:35:20 localhost.localdomain suricata[7033]: [7033] Config: detect: pattern matchers: MPM: hs, SPM: hs
nov 14 18:35:20 localhost.localdomain suricata[7033]: [7033] Config: detect: grouping: tcp-whitelist (default) 53, 80, 139, 443, 445, 1433, 3306, 3389, 6666, 6667, 8080
nov 14 18:35:20 localhost.localdomain suricata[7033]: [7033] Config: detect: grouping: udp-whitelist (default) 53, 135, 5060
nov 14 18:35:20 localhost.localdomain suricata[7033]: [7033] Config: detect: prefilter engines: MPM
nov 14 18:35:20 localhost.localdomain suricata[7033]: [7033] Config: reputation: IP reputation disabled
nov 14 18:35:20 localhost.localdomain suricata[7033]: [7033] Config: detect: Loading rule file: /var/lib/suricata/rules/suricata.rules
[pedro@localhost ~]$ |
```

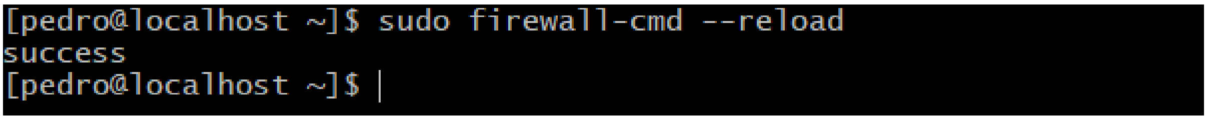
Configuramos el firewall para que mande el trafico a Suricata



```
[pedro@localhost ~]$ systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
     Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
     Active: active (running) since Thu 2024-11-14 17:36:27 CET; 1h 3min ago
       Docs: man:firewalld(1)
   Main PID: 743 (firewalld)
      Tasks: 2 (limit: 10979)
     Memory: 4.5M
        CPU: 2.034s
     CGroup: /system.slice/firewalld.service
             └─743 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid
[pedro@localhost ~]$ sudo firewall-cmd --permanent --direct --add-rule ipv4 filter INPUT 0 -p tcp --dport 22 -j NFQUEUE --queue-bypass
success
[pedro@localhost ~]$ sudo firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0 -p tcp --sport 22 -j NFQUEUE --queue-bypass
success
[pedro@localhost ~]$ sudo firewall-cmd --permanent --direct --add-rule ipv6 filter INPUT 0 -p tcp --dport 22 -j NFQUEUE --queue-bypass
sudo firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0 -p tcp --sport 22 -j NFQUEUE --queue-bypass
success
success
[pedro@localhost ~]$ sudo firewall-cmd --permanent --direct --add-rule ipv4 filter FORWARD 0 -j NFQUEUE
sudo firewall-cmd --permanent --direct --add-rule ipv6 filter FORWARD 0 -j NFQUEUE
success
success
[pedro@localhost ~]$ sudo firewall-cmd --permanent --direct --add-rule ipv4 filter INPUT 1 -j NFQUEUE
sudo firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 1 -j NFQUEUE
success
success
[pedro@localhost ~]$ sudo firewall-cmd --permanent --direct --add-rule ipv6 filter INPUT 1 -j NFQUEUE
sudo firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 1 -j NFQUEUE
success
success
[pedro@localhost ~]$ |
```

Y volvemos a cargar el firewall para que las reglas sean persistentes:

$ sudo firewall-cmd –reload

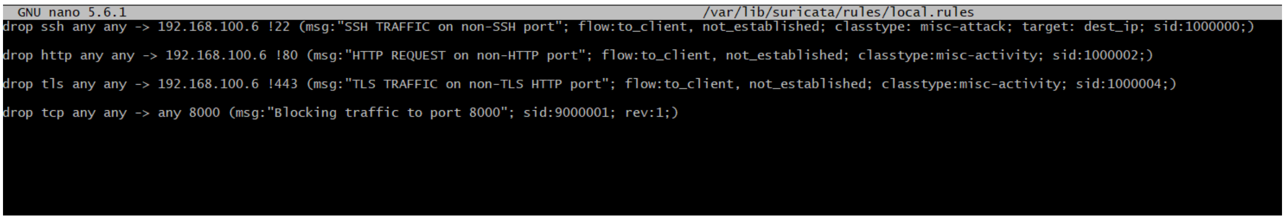


```
[pedro@localhost ~]$ sudo firewall-cmd --reload
success
[pedro@localhost ~]$ |
```

Creando una nueva Regla para Suricata

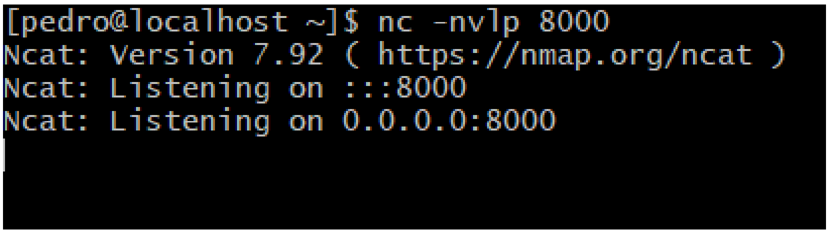
Creamos la regla:

drop tcp any any -> any 8000 (msg:"Blocking traffic to port 8000"; sid:9000001; rev:1;)



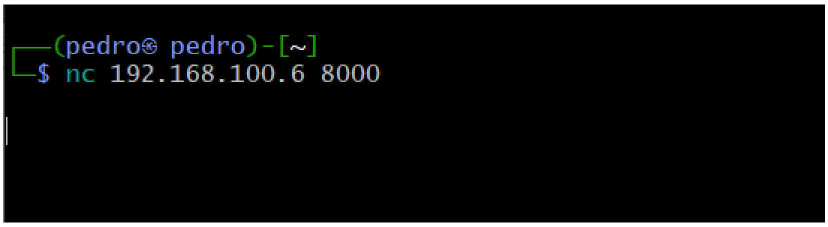
```
  GNU nano 5.6.1                                                                /var/lib/suricata/rules/local.rules
drop ssh any any -> 192.168.100.6 !22 (msg:"SSH TRAFFIC on non-SSH port"; flow:to_client, not_established; classtype: misc-attack; target: dest_ip; sid:1000000;)

drop http any any -> 192.168.100.6 !80 (msg:"HTTP REQUEST on non-HTTP port"; flow:to_client, not_established; classtype:misc-activity; sid:1000002;)

drop tls any any -> 192.168.100.6 !443 (msg:"TLS TRAFFIC on non-TLS HTTP port"; flow:to_client, not_established; classtype:misc-activity; sid:1000004;)

drop tcp any any -> any 8000 (msg:"Blocking traffic to port 8000"; sid:9000001; rev:1;)
```

Ponemos la máquina a escuchar por el puerto 8000



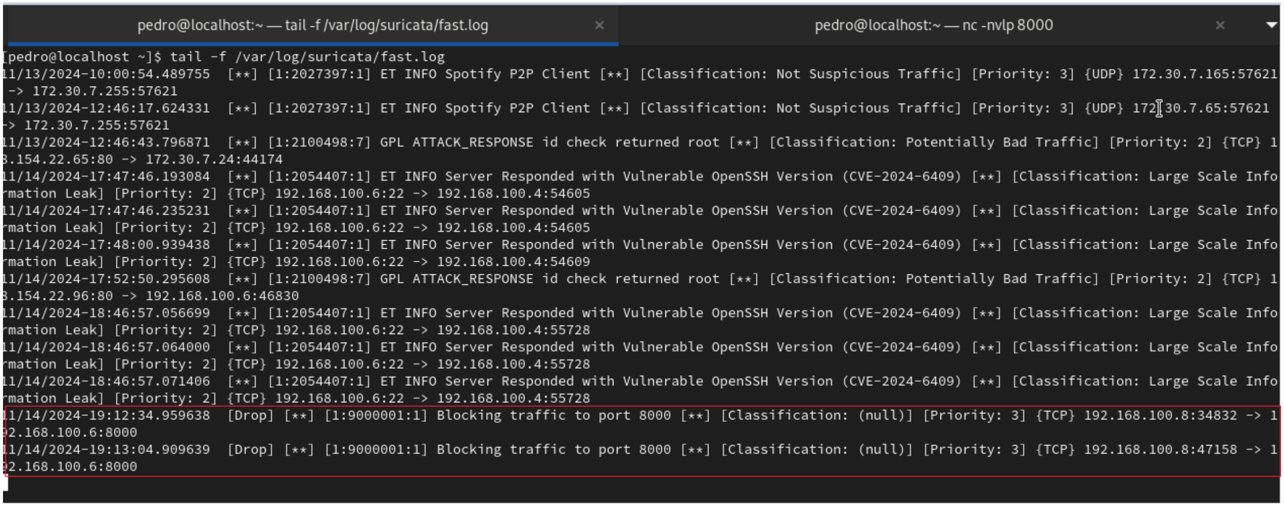
```
[pedro@localhost ~]$ nc -nvlp 8000
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::8000
Ncat: Listening on 0.0.0.0:8000
```

En una máquina atacante simulamos la conexión al puerto 8000



```
┌──(pedro㉿ pedro)-[~]
└─$ nc 192.168.100.6 8000
```

Como podemos observar en las últimas lineas, Suricata está bloqueando el tráfico



pedro@localhost:~ — tail -f /var/log/suricata/fast.log
pedro@localhost:~ — nc -nvlp 8000

```
[pedro@localhost ~]$ tail -f /var/log/suricata/fast.log
11/13/2024-10:00:54.489755  [**] [1:2027397:1] ET INFO Spotify P2P Client [**] [Classification: Not Suspicious Traffic] [Priority: 3] {UDP} 172.30.7.165:57621
 -> 172.30.7.255:57621
11/13/2024-12:46:17.624331  [**] [1:2027397:1] ET INFO Spotify P2P Client [**] [Classification: Not Suspicious Traffic] [Priority: 3] {UDP} 172.30.7.65:57621
-> 172.30.7.255:57621
11/13/2024-12:46:43.796871  [**] [1:2100498:7] GPL ATTACK_RESPONSE id check returned root [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 1
3.154.22.65:80 -> 172.30.7.24:44174
11/14/2024-17:47:46.193084  [**] [1:2054407:1] ET INFO Server Responded with Vulnerable OpenSSH Version (CVE-2024-6409) [**] [Classification: Large Scale Info
rmation Leak] [Priority: 2] {TCP} 192.168.100.6:22 -> 192.168.100.4:54605
11/14/2024-17:47:46.235231  [**] [1:2054407:1] ET INFO Server Responded with Vulnerable OpenSSH Version (CVE-2024-6409) [**] [Classification: Large Scale Info
rmation Leak] [Priority: 2] {TCP} 192.168.100.6:22 -> 192.168.100.4:54605
11/14/2024-17:48:00.939438  [**] [1:2054407:1] ET INFO Server Responded with Vulnerable OpenSSH Version (CVE-2024-6409) [**] [Classification: Large Scale Info
rmation Leak] [Priority: 2] {TCP} 192.168.100.6:22 -> 192.168.100.4:54609
11/14/2024-17:52:50.295608  [**] [1:2100498:7] GPL ATTACK_RESPONSE id check returned root [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 1
3.154.22.96:80 -> 192.168.100.6:46830
11/14/2024-18:46:57.056699  [**] [1:2054407:1] ET INFO Server Responded with Vulnerable OpenSSH Version (CVE-2024-6409) [**] [Classification: Large Scale Info
rmation Leak] [Priority: 2] {TCP} 192.168.100.6:22 -> 192.168.100.4:55728
11/14/2024-18:46:57.064000  [**] [1:2054407:1] ET INFO Server Responded with Vulnerable OpenSSH Version (CVE-2024-6409) [**] [Classification: Large Scale Info
rmation Leak] [Priority: 2] {TCP} 192.168.100.6:22 -> 192.168.100.4:55728
11/14/2024-18:46:57.071406  [**] [1:2054407:1] ET INFO Server Responded with Vulnerable OpenSSH Version (CVE-2024-6409) [**] [Classification: Large Scale Info
rmation Leak] [Priority: 2] {TCP} 192.168.100.6:22 -> 192.168.100.4:55728
11/14/2024-19:12:34.959638  [Drop] [**] [1:9000001:1] Blocking traffic to port 8000 [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.100.8:34832 -> 1
92.168.100.6:8000
11/14/2024-19:13:04.909639  [Drop] [**] [1:9000001:1] Blocking traffic to port 8000 [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.100.8:47158 -> 1
92.168.100.6:8000
```