

Incidentes de ciberseguridad

Tarea 6: IDS/IPS Snort

Índice

Estudio de Rsyslog.....	2
Investiga en Internet el funcionamiento de Rsyslog.....	2
Prepara una guía de uso resumida de Rsyslog (instalación, configuración y uso).....	4
Incluye la instalación de Snort en dicha guía.....	8
Configuración teórica de clientes y servidor Rsyslog.....	10
Describe las labores de configuración necesarias para comunicar los clientes con el servidor Rsyslog y documenta los datos a consignar en los ficheros de configuración.....	10
Implementación de la maqueta de detección multipunto.....	13
Prepara una maqueta (estructura con la ubicación de las máquinas y su IP en la organización) que tenga al menos un cliente (puede tener más) y un servidor Rsyslog.....	13
Configura Rsyslog en los clientes y el servidor utilizando los datos del apartado anterior (incluye capturas de pantalla).....	14
Configura, al menos, 4 reglas en total de los agentes Snort (incluye capturas de pantalla).....	18
Comprueba el funcionamiento de la configuración efectuada, revisando el envío de información de logging desde los clientes hasta el servidor (incluye capturas de pantalla).....	20
Bibliografía.....	22

Estudio de Rsyslog

Investiga en Internet el funcionamiento de Rsyslog.

Rsyslog es un sistema de registro de logs muy utilizado en sistemas operativos Linux. Su nombre viene de "Rocket-fast System for Log Processing". Es una evolución del sistema tradicional `syslog`, con muchas más capacidades y un rendimiento mejorado.

Dentro de sus características principales destaca el filtrado avanzado, el formato flexible de los logs recogidos, el transporte seguro con distintos protocolos, la salida a múltiples destinos (email, bases de datos, sistemas remotos...) y su alto rendimiento.

En sistemas Linux con Rsyslog, los archivos de log se almacenan en directorios específicos, normalmente dentro de `/var/log/`. La ubicación exacta de cada tipo de log y su destino se define en los archivos de configuración de Rsyslog.

Estas son algunas ubicaciones por defecto de los archivos de log:

ARCHIVO	DESCRIPCIÓN
<code>/var/log/syslog</code>	Kernel del sistema operativo
<code>/var/log/messages</code>	Aplicaciones del usuario
<code>/var/log/auth.log</code>	Servicios de correo electrónico
<code>/var/log/secure</code>	Daemon del sistema (servicios en segundo plano)
<code>/var/log/cron</code>	Autenticación del sistema (login, su, etc.)
<code>/var/log/mail.log</code>	Mensajes del propio daemon <code>syslog</code>
<code>/var/log/kern.log</code>	Subsistemas de impresión
<code>/var/log/dmesg</code>	Buffer de mensajes de arranque del kernel
<code>/var/log/boot.log</code>	Proceso de arranque

Los mensajes de log se clasifican según dos ejes principales:

- **Subsistemas o facilidades (facilities):** Indica el tipo de servicio o subsistema que generó el mensaje.
- **Niveles de severidad (severities):** Indica la criticidad del log.

Dentro de los subsistemas, podemos identificar los siguientes como los más comunes:

CÓDIGO	NOMBRE	DESCRIPCIÓN
0	kern	Kernel del sistema operativo
1	user	Aplicaciones del usuario
2	mail	Servicios de correo electrónico
3	daemon	Daemon del sistema (servicios en segundo plano)
4	auth	Autenticación del sistema (login, su, etc.)
5	syslog	Mensajes del propio daemon syslog
6	lpr	Subsistemas de impresión
7	news	Noticias (NNTP, servicios antiguos de noticias)
8	uucp	Unix-to-Unix Copy
9	cron	Planificador de tareas
10	authpriv	Autenticación con privacidad
11	ftp	Servidores FTP

Por otra parte, también distinguimos los niveles de prioridad que indican la importancia o urgencia del mensaje de log. Van desde el más crítico hasta el más informativo.

He aquí una tabla detallada con cada nivel con su respectivo nombre y descripción:

CÓDIGO	NOMBRE	DESCRIPCIÓN
1	alert	Se necesita acción inmediata.
2	crit	Error grave, como fallo de hardware.
3	err	Errores en ejecución, pero el sistema sigue funcionando.
4	warning	Algo inesperado ocurrió, pero no es grave.
5	notice	Eventos importantes pero normales.
6	info	Mensajes informativos, como el estado de los servicios.
7	debug	Mensajes detallados para desarrolladores o diagnóstico.

Para ver su uso en Rsyslog, podemos usar esta línea de un archivo de configuración como ejemplo, la cual registra solo mensajes de error o más graves del subsistema authpriv.:

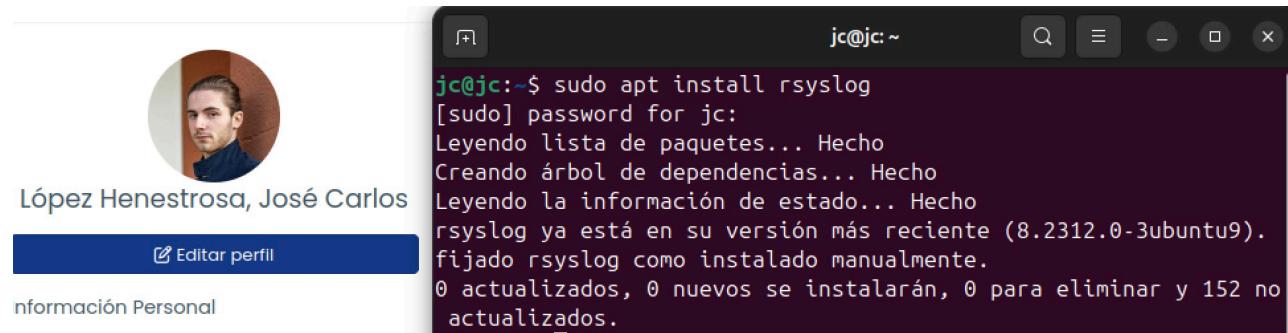
```
authpriv.err      /var/log/auth_errors.log
```

Prepara una guía de uso resumida de Rsyslog (instalación, configuración y uso).

INSTALACIÓN

Para instalar Rsyslog en Ubuntu/Debian, ejecutamos el siguiente comando:

```
sudo apt install rsyslog
```



Instalación de Rsyslog en Ubuntu

CONFIGURACIÓN

El archivo principal para la configuración de Rsyslog se encuentra en `/etc/rsyslog.conf`, aunque también existen archivos de configuración en la ruta `/etc/rsyslog.d/*.conf`, donde `*.conf` se refiere a todos los archivos de configuración localizados en el directorio `/etc/rsyslog.d`. Es usada para almacenar configuraciones adicionales o personalizadas de Rsyslog.

El orden de carga de los ficheros es el siguiente:

1. `/etc/rsyslog.conf`
2. Si está habilitado en el archivo `/etc/rsyslog.conf`, se incluyen todas las configuraciones que estén en `/etc/rsyslog.d/*.conf`.

Esto permite organizar configuraciones en archivos separados, crear reglas de log específicas por servicio o aplicación y añadir o quitar configuraciones sin modificar el archivo principal.

En los archivos de configuración se pueden realizar los siguientes ajustes:

- **Cargar módulos**

Cada módulo proporciona una funcionalidad (recepción local, remota, por TCP/UDP, etc.).

Ejemplo:

```
module(load="imuxsock")          # Recibe logs del sistema
module(load="imklog")            # Recibe logs del kernel
input(type="imudp" port="514")   # Recibe logs por UDP en el puerto 514
module(load="imtcp")             # Recibe logs por TCP (servidor)
```

- **Definición de plantillas**

Crea una ruta dinámica para guardar logs de cada host remoto y aplicación.

```
template(name="RemoteLogs" type="string"
        string="/var/log/remote/%HOSTNAME%/%PROGRAMNAME%.log")
```

- **Reglas de filtrado y destino**

```
authpriv.*      /var/log/secure
mail.*          -/var/log/mail.log
cron.*          /var/log/cron.log
*.emerg         *
```

- **Formato:** `FACILIDAD.NIVEL DESTINO`
- El `-` antes del nombre de archivo reduce el uso de disco, ya que no fuerza la sincronización inmediata.
- El `*` significa "todos".

- **Incluir archivos adicionales**

Permite dividir la configuración en archivos separados, para facilitar la gestión.

```
$IncludeConfig /etc/rsyslog.d/*.conf
```

USO

Se pueden realizar multitud de acciones con Rsyslog. Estas son algunas de las más utilizadas, las cuales también se ven en el [siguiente apartado](#):

- **Enviar logs a un servidor remoto**

Para ello, añadimos la siguiente línea en un archivo de configuración de Rsyslog:

```
*.* @@192.168.1.100:514
```

Donde:

- *.*: Aplica a todos los logs, sin importar la facilidad (como auth, mail, cron, etc.) ni el nivel de gravedad (como info, error, debug, etc.)
- @@192.168.1.100:514: Envía dichos logs a la dirección IP 192.168.1.100 por el puerto 514 usando TCP (@@).

Para que los cambios surtan efecto, reiniciamos el servicio:

```
sudo systemctl restart rsyslog
```

- **Recibir logs en un servidor remoto**

Habilitamos la recepción de logs editando el archivo `/etc/rsyslog.conf` o usando módulos:

```
# Para UDP
module(load="imudp")
input(type="imudp" port="514")

# Para TCP
module(load="imtcp")
input(type="imtcp" port="514")
```

Permitir el puerto 514 en el firewall:

```
sudo ufw allow 514/tcp
sudo ufw allow 514/udp
```

Para que los cambios surtan efecto, reiniciamos el servicio:

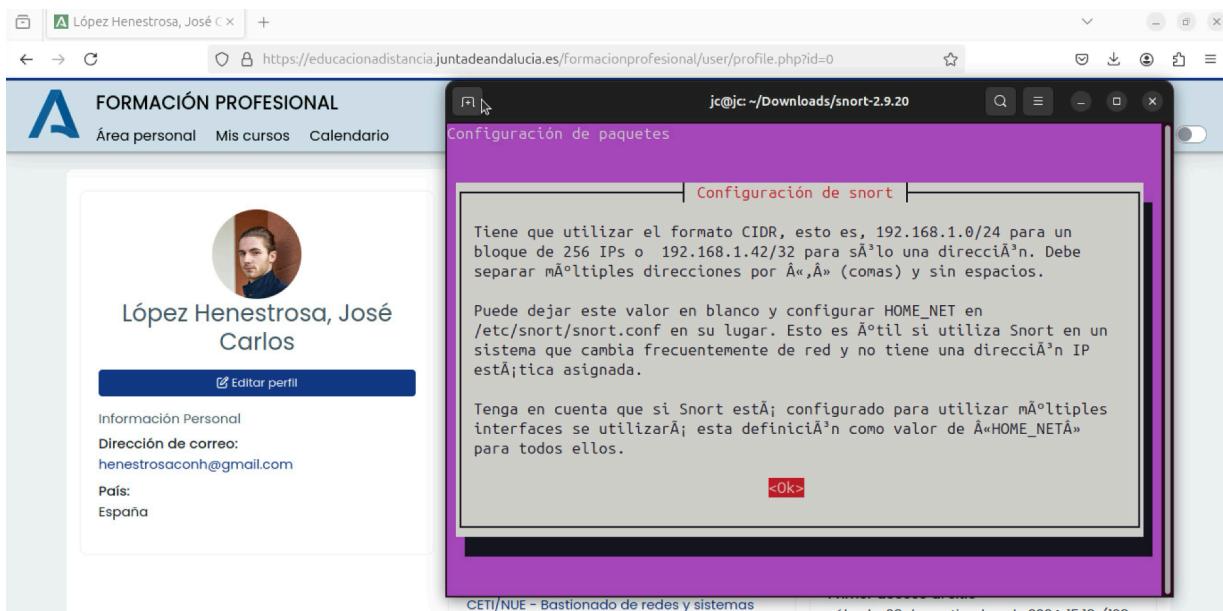
```
sudo systemctl restart rsyslog
```

Incluye la instalación de Snort en dicha guía.

Vamos a instalar Snort en Ubuntu con este comando:

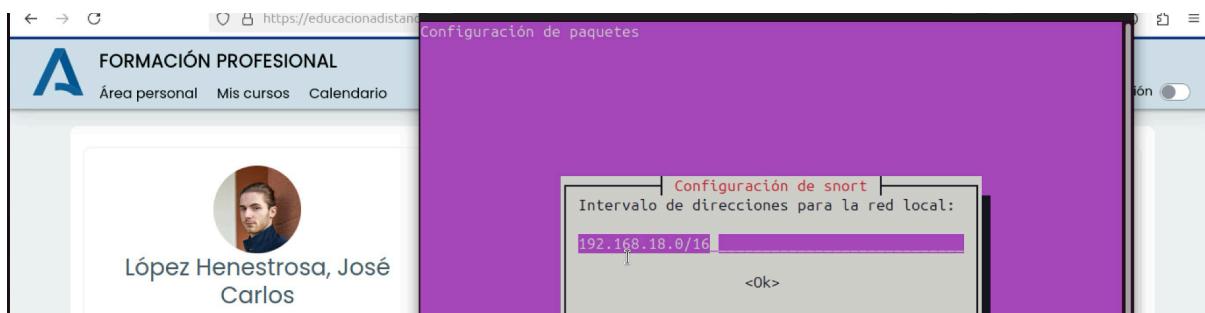
```
sudo apt-get install snort
```

Tras ejecutarlo, veremos la siguiente pantalla para configurar la herramienta:



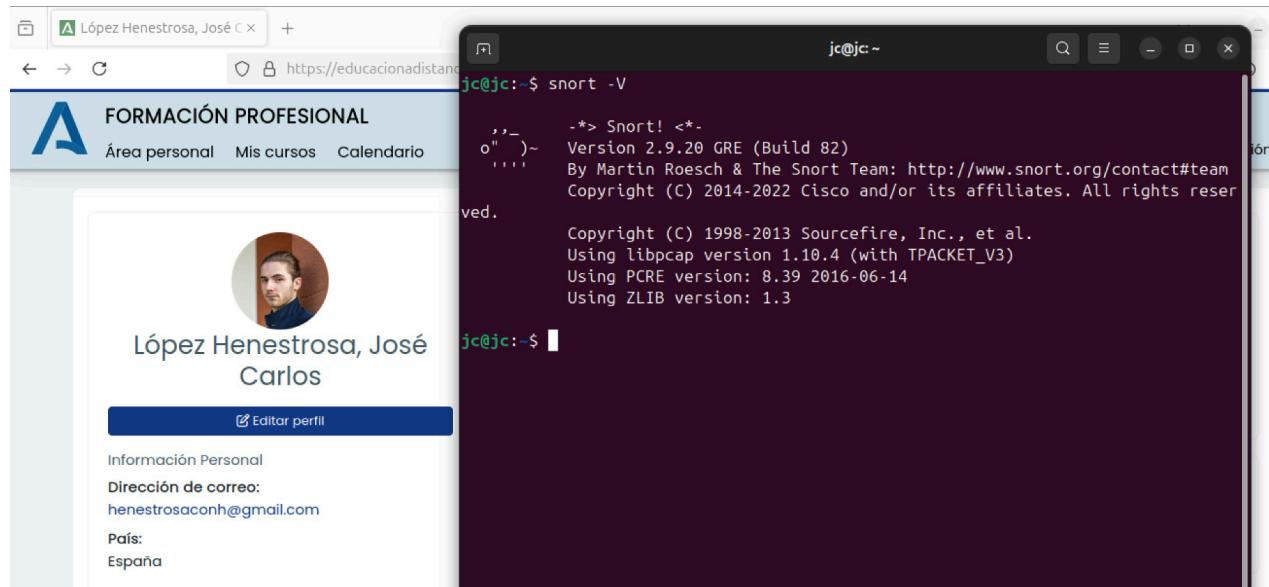
Primera pantalla de configuración de Snort

Al continuar, nos pedirá introducir el intervalo de direcciones IP para la red local. La IP de mi router es 192.168.18.1, lo cual significa que mi red local usa el rango 192.168.18.0/24, lo que abarca las direcciones 192.168.18.1 - 192.168.18.254.



Configuración del intervalo de direcciones IP para la red local

Al aceptar, la instalación se completará correctamente. Para comprobar que, efectivamente, Snort está disponible en el sistema, ejecutamos `snort -V`, lo cual debe mostrar el número de versión instalada.



La versión de Snort aparece, lo que indica que ha sido instalado con éxito

Configuración teórica de clientes y servidor Rsyslog

Describe las labores de configuración necesarias para comunicar los clientes con el servidor Rsyslog y documenta los datos a consignar en los ficheros de configuración.

Para comunicar clientes con un servidor Rsyslog, se requiere una configuración en ambos extremos: en el servidor (que recibe logs) y en los clientes (que los envían).

CONFIGURACIÓN EN EL SERVIDOR

En primer lugar, configuraremos el **servidor de Rsyslog**, el cual será el **receptor** de los logs. Para ello, seguimos estos pasos.

1. Crear o editar el archivo /etc/rsyslog.d/30-snort.conf

```
sudo vim /etc/rsyslog.d/30-snort.conf
```

2. Habilitar recepción por UDP y/o TCP

```
# Para UDP
module(load="imudp")
input(type="imudp" port="514")

# Para TCP
module(load="imtcp")
input(type="imtcp" port="514")
```

3. Añadir regla de filtrado para logs de Snort

Esta línea sirve para filtrar y redirigir los logs generados por Snort a un archivo específico.

```
# Filtrado de logs de Snort
if $programname == 'snort' then /var/log/snort/snort.log
& stop
```

4. Guardar y cerrar el archivo

5. Crear o editar el archivo /etc/rsyslog.d/remote.conf

```
sudo vim /etc/rsyslog.d/remote.conf
```

6. Crear una regla de guardado

```
$template RemoteLogs,"/var/log/remote/%HOSTNAME%/%PROGRAMNAME%.log"  
*.* ?RemoteLogs  
& stop
```

Esta regla hace que los logs de los clientes se guarden de forma ordenada en carpetas por host y programa. & stop asegura que los logs solo se guarden en el archivo especificado y no se redirijan a otros destinos.

7. Guardar y cerrar el archivo

8. Crear el directorio /var/log/remote y darle los permisos pertinentes

```
sudo mkdir -p /var/log/remote  
sudo chmod 755 /var/log/remote
```

9. Abrir el puerto 514 en el firewall

```
sudo ufw allow 514/udp # Para UDP  
sudo ufw allow 514/tcp # Para TCP
```

10. Reiniciar Rsyslog

```
sudo systemctl restart rsyslog
```

CONFIGURACIÓN EN EL CLIENTE

Una vez hecho esto, **configuramos los clientes Rsyslog**, los cuales serán los **emisores** de los logs. Con tal de conseguirlo, seguimos estos pasos:

1. Comprobar IP privada del servidor

Desde la máquina que aloja el servidor, tenemos que ejecutar el siguiente comando y apuntar la dirección IP:

```
hostname -I
```

2. Editar o crear el archivo 90-forwarding.conf en el directorio /etc/rsyslog.d/

```
sudo vim /etc/rsyslog.d/90-forwarding.conf
```

3. Añadir configuración

Para ello, añadimos la siguiente línea en un archivo de configuración de Rsyslog:

```
*.* @@IP_SERVIDOR:514
```

Donde:

1. `*.*`: Aplica a todos los logs, sin importar la facilidad (como auth, mail, cron, etc.) ni el nivel de gravedad (como info, error, debug, etc.).
2. `@@IP_SERVIDOR:514`: Envía dichos logs a la dirección IP del servidor por el puerto 514 usando TCP (`@@`).

4. Reiniciar Rsyslog

```
sudo systemctl restart rsyslog
```

Para verificar que todo está correcto, miramos los logs entrantes en el servidor con este comando:

```
tail -f /var/log/remote/*/*.log
```

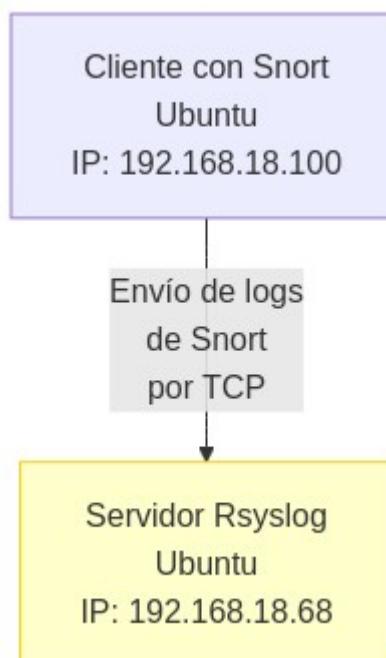
En el cliente, generamos un mensaje de prueba:

```
logger "Mensaje de prueba desde cliente"
```

Al ejecutar este comando, el servidor debería recibir el mensaje de prueba enviado por el cliente.

Implementación de la maqueta de detección multipunto (Raspberry Pi o máquinas virtuales)

Prepara una maqueta (estructura con la ubicación de las máquinas y su IP en la organización) que tenga al menos un cliente (puede tener más) y un servidor Rsyslog.



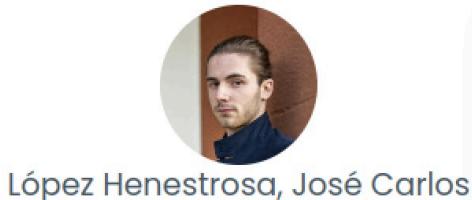
Estructura con la ubicación de las máquinas y su IP en una organización ficticia

Configura Rsyslog en los clientes y el servidor utilizando los datos del apartado anterior (incluye capturas de pantalla).

CONFIGURACIÓN EN EL SERVIDOR

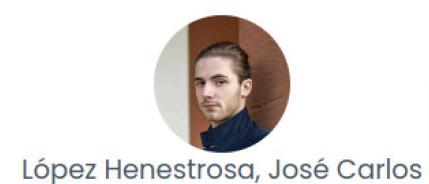
Vamos a aplicar los pasos detallados en el [apartado anterior](#) para la parte del **servidor**:

1. Crear el archivo /etc/rsyslog.d/30-snort.conf



```
jc@jc:~$ sudo vim /etc/rsyslog.d/30-snort.conf
[sudo] password for jc:
jc@jc:~$
```

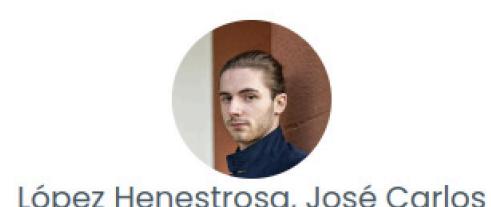
2. Añadir la configuración especificada al archivo



```
module(load="imtcp")
input(type="imtcp" port="514")
if $programname == 'snort' then /var/log/snort/snort.log
```

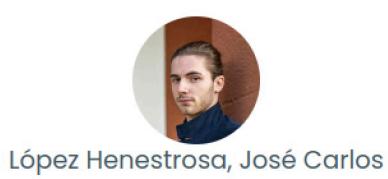
Tras esto, cerramos y guardamos el archivo

3. Crear el archivo /etc/rsyslog.d/remote.conf



```
jc@jc:~$ sudo vim /etc/rsyslog.d/remote.conf
[sudo] password for jc:
jc@jc:~$
```

4. Añadir la configuración especificada al archivo



```
$template RemoteLogs, "/var/log/remote/%HOSTNAME%/%PROGRAMNAME%.log" *.* ?RemoteLogs
~
```

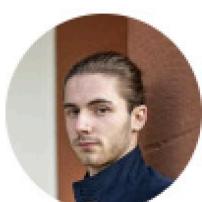
5. Crear el directorio /var/log/remote y darle los permisos pertinentes



López Henestrosa, José Carlos

```
jc@jc:~$ sudo mkdir -p /var/log/remote  
jc@jc:~$ sudo chmod 755 /var/log/remote  
jc@jc:~$
```

6. Abrir el puerto 514 en el firewall



López Henestrosa, José Carlos

```
jc@jc:~$ sudo ufw allow 514/tcp  
Rules updated  
Rules updated (v6)
```

7. Reiniciar Rsyslog



López Henestrosa, José Carlos

```
jc@jc:~$ sudo systemctl restart rsyslog  
jc@jc:~$
```

Una vez completada esta fase, se habrá finalizado la configuración de Rsyslog en el **servidor**.

CONFIGURACIÓN EN EL CLIENTE

Vamos a aplicar los pasos detallados en el [apartado anterior](#) para la parte del **cliente** en otra máquina virtual diferente a la del servidor:

1. Comprobar IP privada del servidor

Es importante matizar que si clonamos una máquina virtual de otra, probablemente ambas comparten la misma dirección MAC. Esto puede causar que reciban la misma dirección IP del servidor DHCP, lo que genera conflictos en la red.

Para solucionar este problema, es importante cambiar la dirección MAC de una de las dos. La forma de hacerlo depende del programa que se esté utilizando. En el caso de UTM (Mac), tenemos que seleccionar la máquina clonada (en este caso, la del cliente), e ir a *Configuración > Red*. Una vez ahí, generamos una dirección MAC aleatoria y guardamos los cambios para continuar.

Una vez hecho esto, ejecutamos el siguiente comando en la **máquina servidor**:

```
hostname -I
```

López Henestrosa, José Carlos

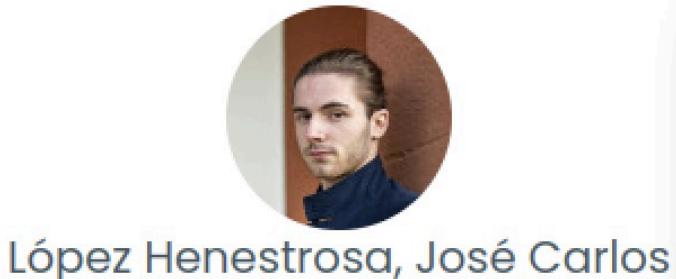
```
jc@jc:~$ hostname -I  
192.168.18.68  
jc@jc:~$
```

2. Crear el archivo 90-forwarding.conf en el directorio /etc/rsyslog.d/

López Henestrosa, José Carlos

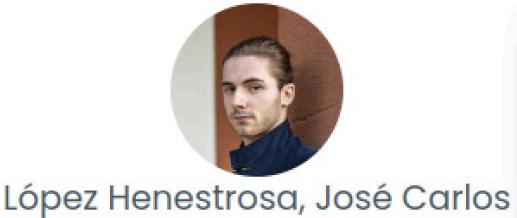
```
jc@jc:~$ sudo vim /etc/rsyslog.d/90-forwarding.conf
```

3. Añadir la configuración especificada al archivo



```
*.* @@192.168.18.68:514
```

4. Reiniciar Rsyslog



```
jc@jc:~$ sudo systemctl restart rsyslog
jc@jc:~$
```

Una vez hecho esto, la configuración de Rsyslog en el **cliente** ya está concluida.

Configura, al menos, 4 reglas en total de los agentes Snort (incluye capturas de pantalla).

Para configurar dichas reglas de los agentes Snort, debemos incluirlas en un archivo `.rules` (por ejemplo, `local.rules`) y referenciarlas desde el archivo principal de configuración de Snort (`snort.conf`).

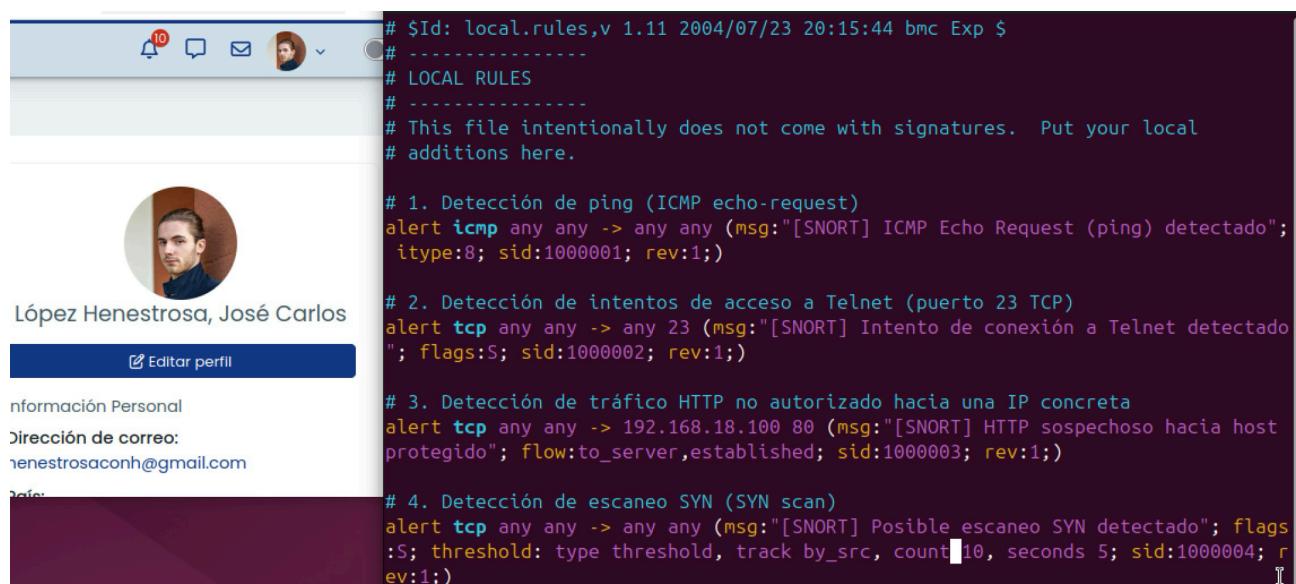
Dicho esto, procedemos a crear el archivo `/etc/snort/rules/local.rules` y le añadimos las siguientes reglas, donde `192.168.18.100` es la IP privada del cliente:

```
# 1. Detección de ping (ICMP echo-request)
alert icmp any any -> any any (msg:"[SNORT] ICMP Echo Request (ping) detectado"; itype:8; sid:1000001; rev:1;)

# 2. Detección de intentos de acceso a Telnet (puerto 23 TCP)
alert tcp any any -> any 23 (msg:"[SNORT] Intento de conexión a Telnet detectado"; flags:S; sid:1000002; rev:1;)

# 3. Detección de tráfico HTTP no autorizado hacia una IP concreta
alert tcp any any -> 192.168.18.100 80 (msg:"[SNORT] HTTP sospechoso hacia host protegido"; flow:to_server,established; sid:1000003; rev:1;)

# 4. Detección de escaneo SYN (SYN scan)
alert tcp any any -> any any (msg:"[SNORT] Posible escaneo SYN detectado"; flags:S; threshold:type threshold, track by_src, count 10, seconds 5; sid:1000004; rev:1;)
```



```
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.

# 1. Detección de ping (ICMP echo-request)
alert icmp any any -> any any (msg:"[SNORT] ICMP Echo Request (ping) detectado";
itype:8; sid:1000001; rev:1;)

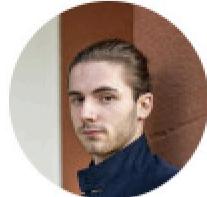
# 2. Detección de intentos de acceso a Telnet (puerto 23 TCP)
alert tcp any any -> any 23 (msg:"[SNORT] Intento de conexión a Telnet detectado";
flags:S; sid:1000002; rev:1;)

# 3. Detección de tráfico HTTP no autorizado hacia una IP concreta
alert tcp any any -> 192.168.18.100 80 (msg:"[SNORT] HTTP sospechoso hacia host
protegido"; flow:to_server,established; sid:1000003; rev:1;)

# 4. Detección de escaneo SYN (SYN scan)
alert tcp any any -> any any (msg:"[SNORT] Posible escaneo SYN detectado"; flags
:S; threshold:type threshold, track by_src, count 10, seconds 5; sid:1000004; rev:1;)
```

Para habilitar estas reglas, tenemos que asegurarnos de que Snort esté leyendo el archivo recién creado `local.rules`. Para ello, abrimos el archivo de configuración global de Snort, `/etc/snort/snort.conf` e incluimos la siguiente línea:

```
include $RULE_PATH/local.rules
```

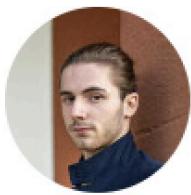


López Henestrosa, José Carlos

```
# rules files that are available  
# directory)  
  
# site specific rules  
include $RULE_PATH/local.rules
```

Una vez modificado, reiniciamos Snort:

```
sudo systemctl restart snort
```



López Henestrosa, José Carlos

```
[+]  
jc@jc:~$ sudo systemctl restart snort  
jc@jc:~$ █
```

Comprueba el funcionamiento de la configuración efectuada, revisando el envío de información de logging desde los clientes hasta el servidor (incluye capturas de pantalla).

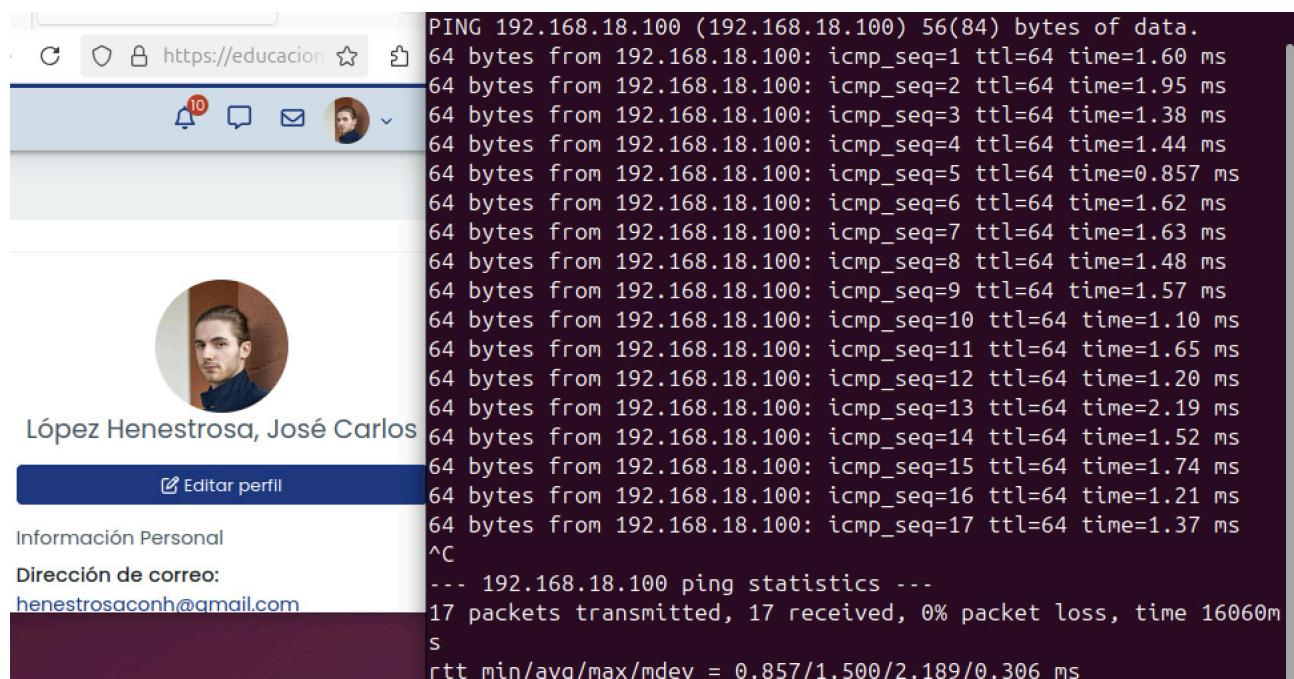
Ahora que tenemos las reglas de Snort configuradas junto a las configuraciones pertinentes de Rsyslog tanto en el cliente como en el servidor, procedemos a ejecutar un tráfico que dispare una regla Snort establecida.

Para ello, probamos a hacer un ping al cliente Snort (IP local 192.168.1.50) desde otra máquina distinta:

```
ping 192.168.18.100
```

Lo cual dispara esta regla:

```
# 1. Detección de ping (ICMP echo-request)
alert icmp any any -> any any (msg:"[SNORT] ICMP Echo Request (ping)
detectado"; itype:8; sid:1000001; rev:1;)
```



Si comprobamos el directorio `/var/log/remote/*/snort.log` en la máquina **servidor**, donde el * es el nombre del cliente, encontramos lo siguiente:

Como podemos apreciar, el servidor recibe correctamente los logs enviados por el cliente.

Bibliografía

- Rsyslog. *Wikipedia*. <https://en.wikipedia.org/wiki/Rsyslog>
- Registros centralizados en Linux con Rsyslog. *ochobitshacenunbyte*.
<https://www.ochobitshacenunbyte.com/2018/10/29/registros-centralizados-en-linux-con-rsyslog>
- Documentación oficial de Rsyslog. <https://www.rsyslog.com/doc/index.html>
- Chapter 14. Configuring a remote logging solution. *Red Hat*.
https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/security_hardening/assembly_configuring-a-remote-logging-solution_security-hardening
- How to Set Up Centralized Logging on Linux with Rsyslog. *Better Stack*.
<https://betterstack.com/community/guides/logging/how-to-configure-centralised-rsyslog-server/>
- Configuración de Open Source SNORT (con Rsyslog). *IBM*.
<https://www.ibm.com/docs/es/dsm?topic=snort-configuring-open-source>
- Alert Logging. *Documentación de Snort*. https://docs.snort.org/start/alert_logging
- How to Set Up Remote Logging on Linux Using rsyslog. *Make Use Of*.
<https://www.makeuseof.com/set-up-linux-remote-logging-using-rsyslog/>