

Índice:

Ejercicio 1	2
a) SISTEMA DE DETECCIÓN DE INTRUSOS (IDS): HostIDS	2
b) SISTEMA DE DETECCIÓN DE INTRUSOS (IDS) : NetIDS	15

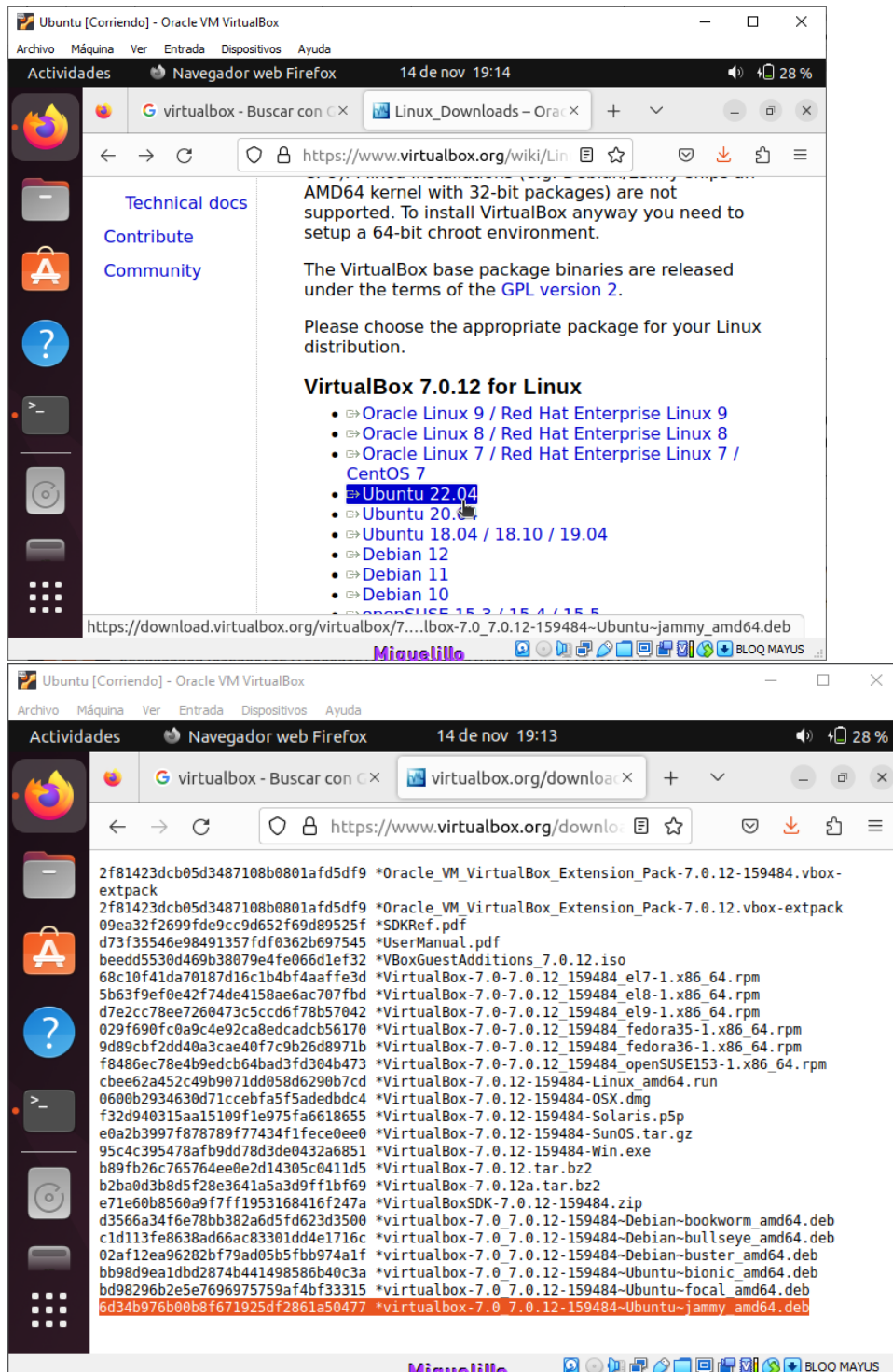
Ejercicio 1

a) SISTEMA DE DETECCIÓN DE INTRUSOS (IDS): HostIDS

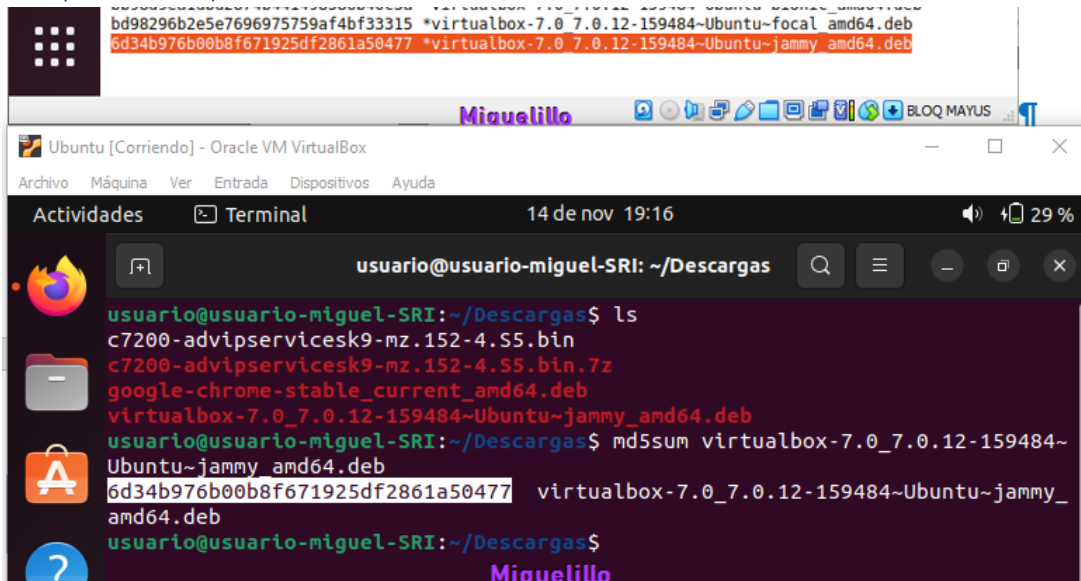
Solución:

Linux

Descargamos un archivo y comprobamos el código hash que nos dan



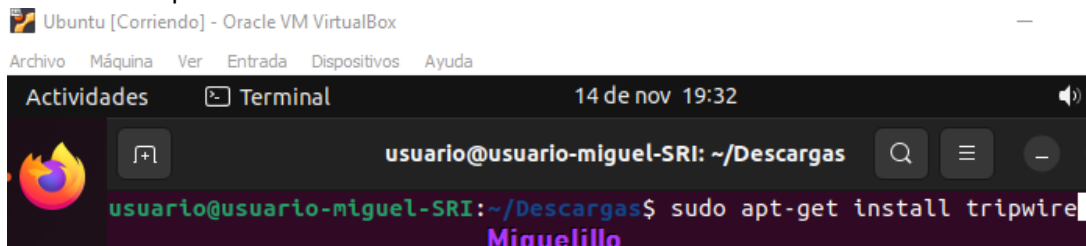
Comprobamos que los números coincidan



```
bd98296b2e5e7696975759af4bf33315 *virtualbox-7.0_7.0.12-159484-Ubuntu~focal_amd64.deb
6d34b976b00b8f671925df2861a50477 *virtualbox-7.0_7.0.12-159484-Ubuntu~jammy_amd64.deb

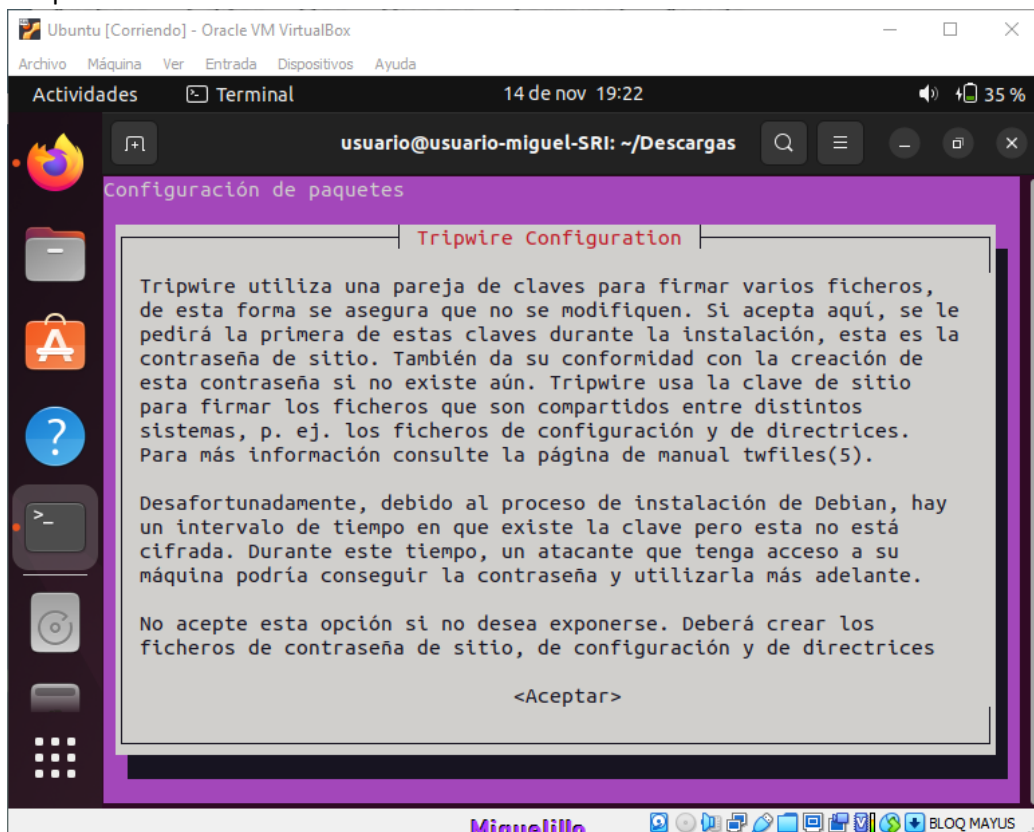
usuario@usuario-miguel-SRI: ~/Descargas
usuario@usuario-miguel-SRI:~/Descargas$ ls
c7200-advipservicesk9-mz.152-4.S5.bin
c7200-advipservicesk9-mz.152-4.S5.bin.7z
google-chrome-stable_current_amd64.deb
virtualbox-7.0_7.0.12-159484-Ubuntu~jammy_amd64.deb
usuario@usuario-miguel-SRI:~/Descargas$ md5sum virtualbox-7.0_7.0.12-159484-
Ubuntu~jammy_amd64.deb
6d34b976b00b8f671925df2861a50477 virtualbox-7.0_7.0.12-159484-Ubuntu~jammy_
amd64.deb
usuario@usuario-miguel-SRI:~/Descargas$
```

Instalamos tripwire



```
usuario@usuario-miguel-SRI:~/Descargas$ sudo apt-get install tripwire
```

Aceptamos hacer las llaves



```
Configuración de paquetes

Tripwire Configuration

Tripwire utiliza una pareja de claves para firmar varios ficheros,
de esta forma se asegura que no se modifiquen. Si acepta aquí, se le
pedirá la primera de estas claves durante la instalación, esta es la
contraseña de sitio. También da su conformidad con la creación de
esta contraseña si no existe aún. Tripwire usa la clave de sitio
para firmar los ficheros que son compartidos entre distintos
sistemas, p. ej. los ficheros de configuración y de directrices.
Para más información consulte la página de manual twfiles(5).

Desafortunadamente, debido al proceso de instalación de Debian, hay
un intervalo de tiempo en que existe la clave pero esta no está
cifrada. Durante este tiempo, un atacante que tenga acceso a su
máquina podría conseguir la contraseña y utilizarla más adelante.

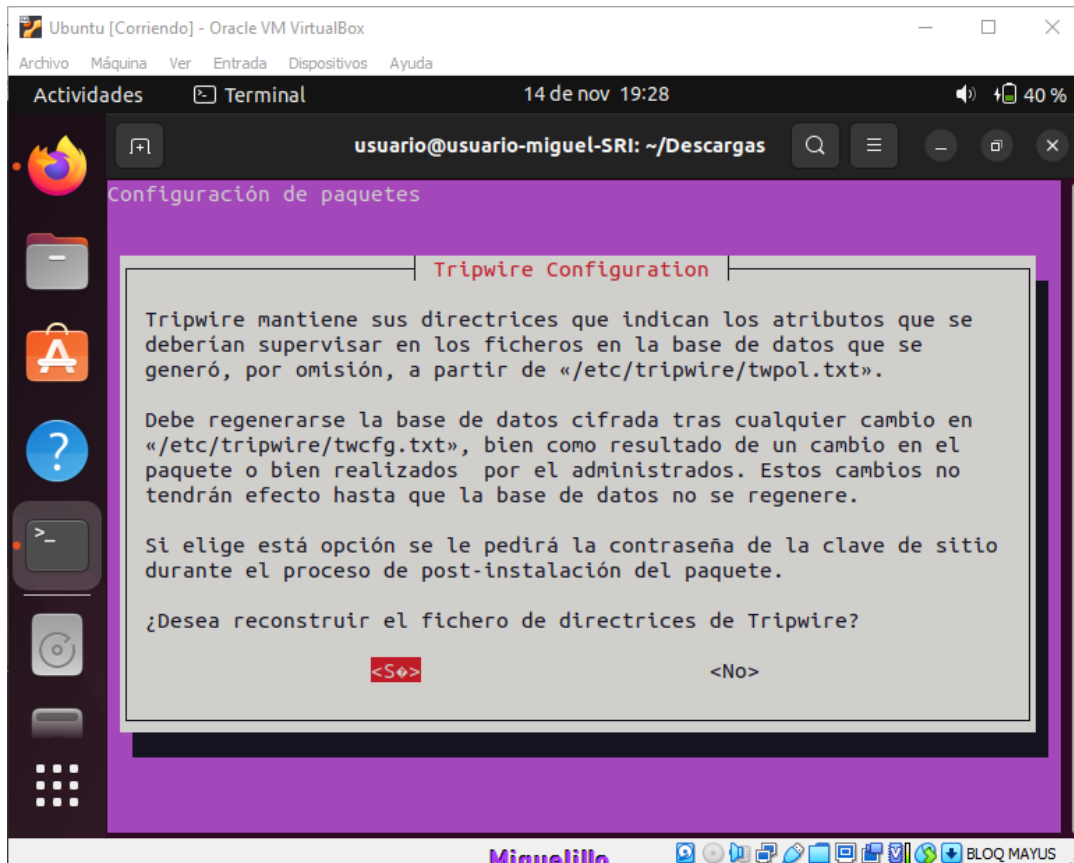
No acepte esta opción si no desea exponerse. Deberá crear los
ficheros de contraseña de sitio, de configuración y de directrices

<Aceptar>
```

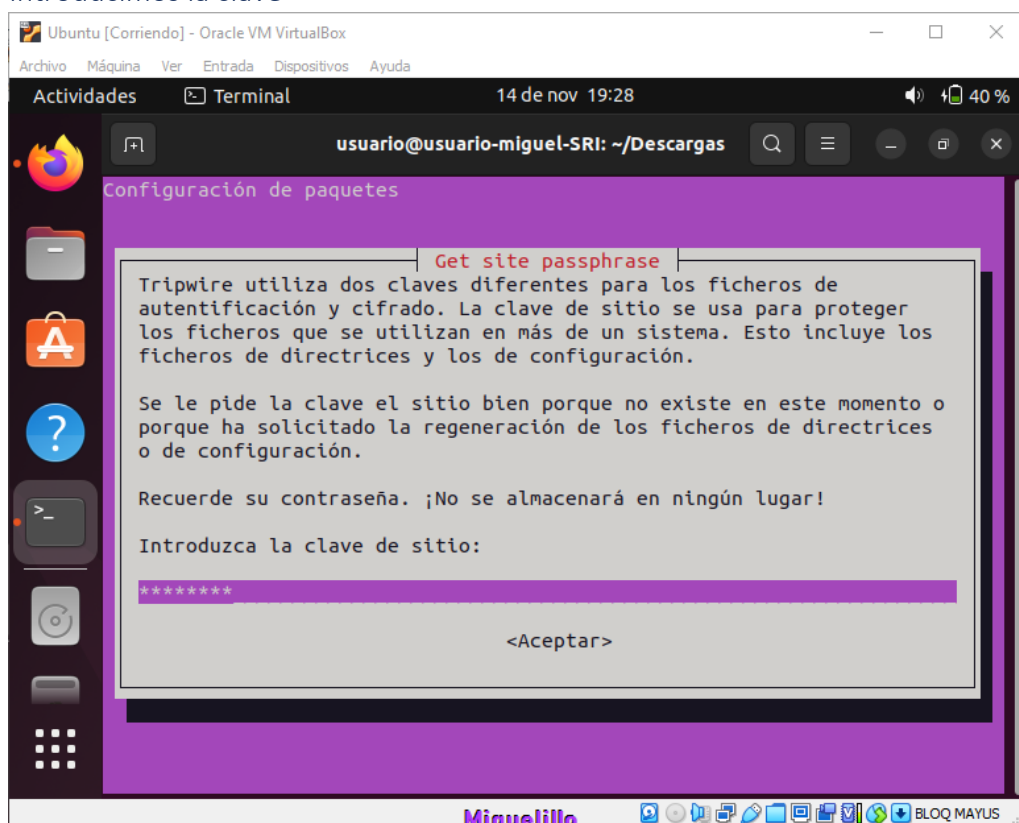
Reconstruimos el fichero de Tripwire



Reconstruimos las directrices de Tripwire



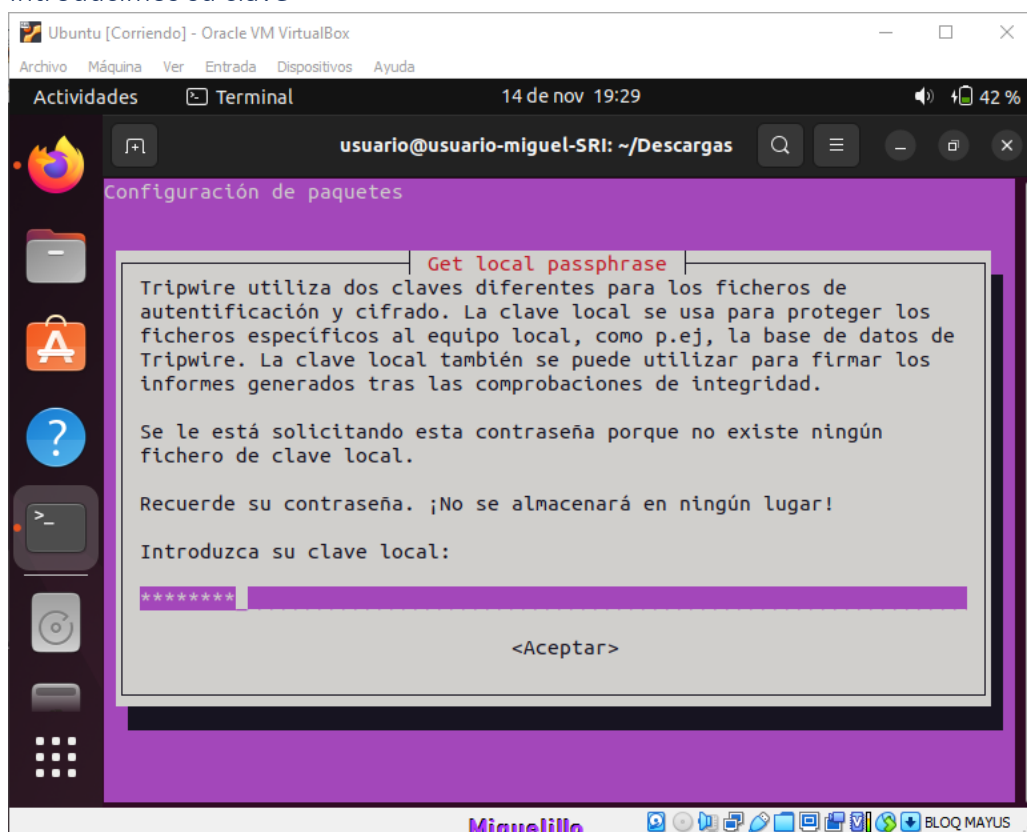
Introducimos la clave



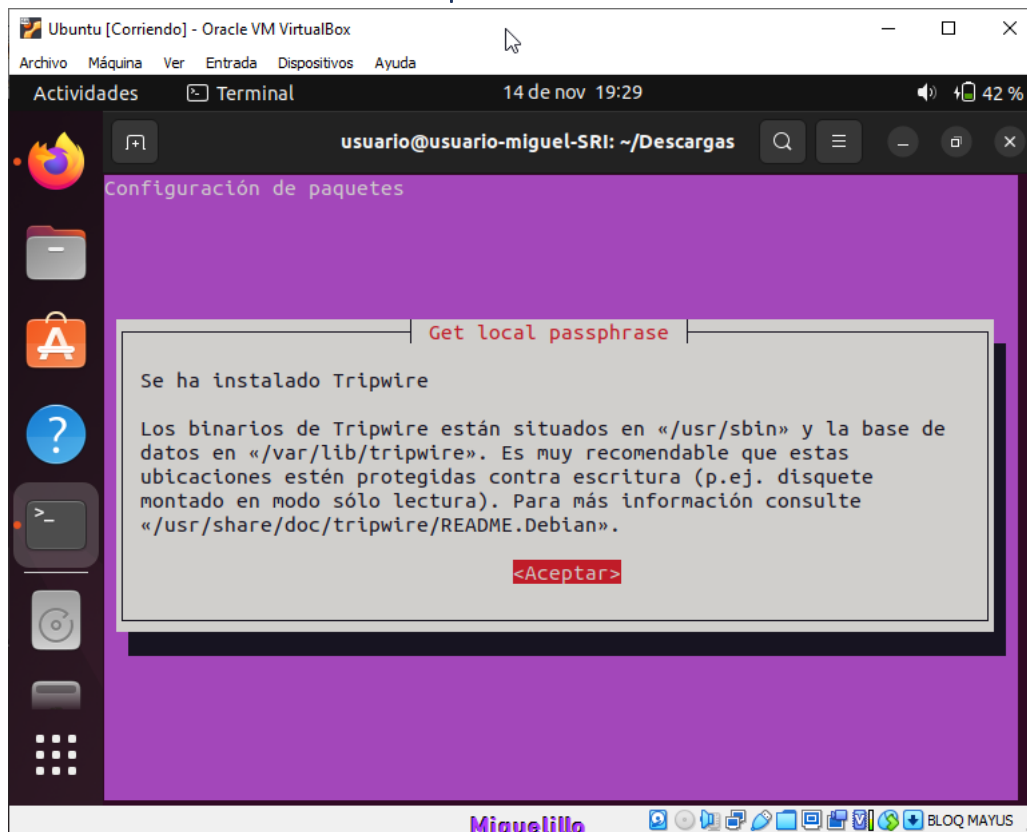
Confirmamos la clave



Introducimos su clave



Una vez instalado clicamos en Aceptar



Con el comando `sudo tripwire -init` podemos ver los registros

```

Ubuntu [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades Terminal 14 de nov 19
usuario@usuario-mi

"/proc/version_signature"
"/proc/vmallocinfo"

-----
Rule Name: Tripwire Data Files (/var/lib/tripwire/report)
Severity Level: 100
-----

Modified:
"/var/lib/tripwire/report"

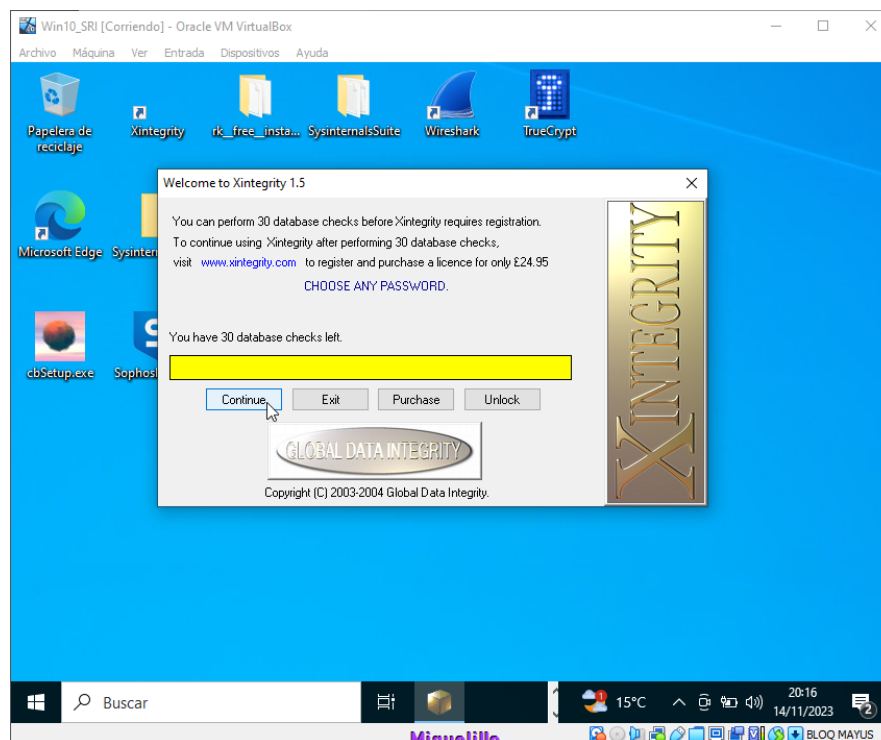
Error Report:
=====

Section: Unix File System
-----

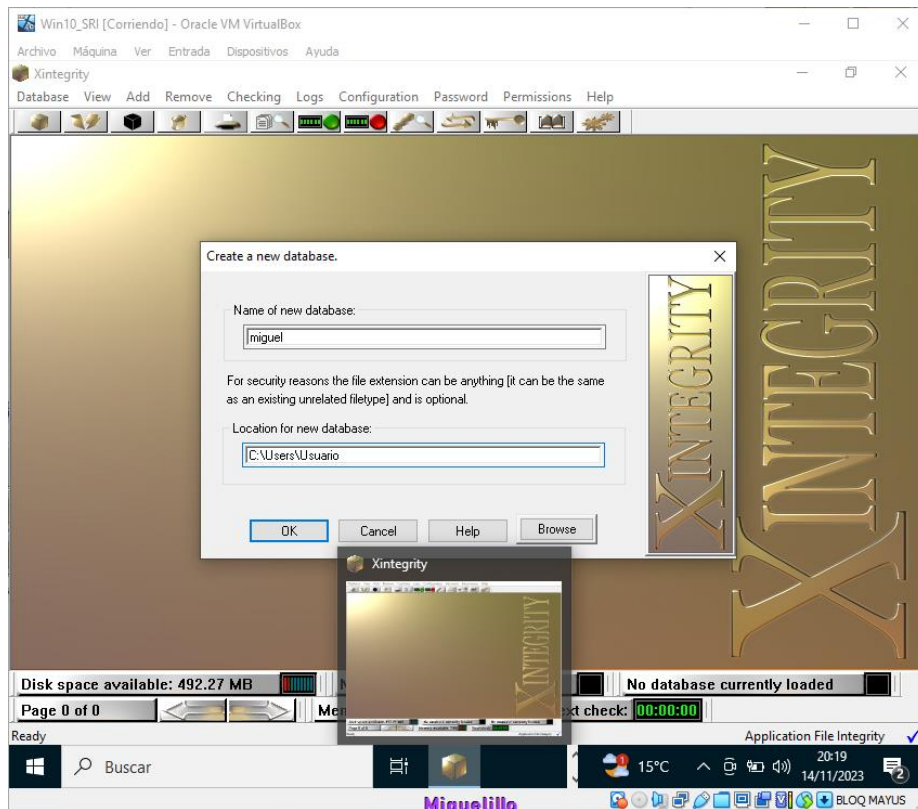
1. Could not access directory contents.
   Filename: /usr/sbin
   Filename: /usr/sbin
   Operación no permitida
2. File could not be opened.
   Filename: /usr/sbin/siggen
   Operación no permitida
3. File could not be opened.
   Filename: /usr/sbin/tripwire
   Operación no permitida
4. File could not be opened.
   Filename: /usr/sbin/twadmin
   Operación no permitida
5. File could not be opened.
   Filename: /usr/sbin/twprint
   Operación no permitida
6. Could not access directory contents.
   Filename: /usr/lib
   Filename: /usr/lib
   Operación no permitida
  
```

Windows

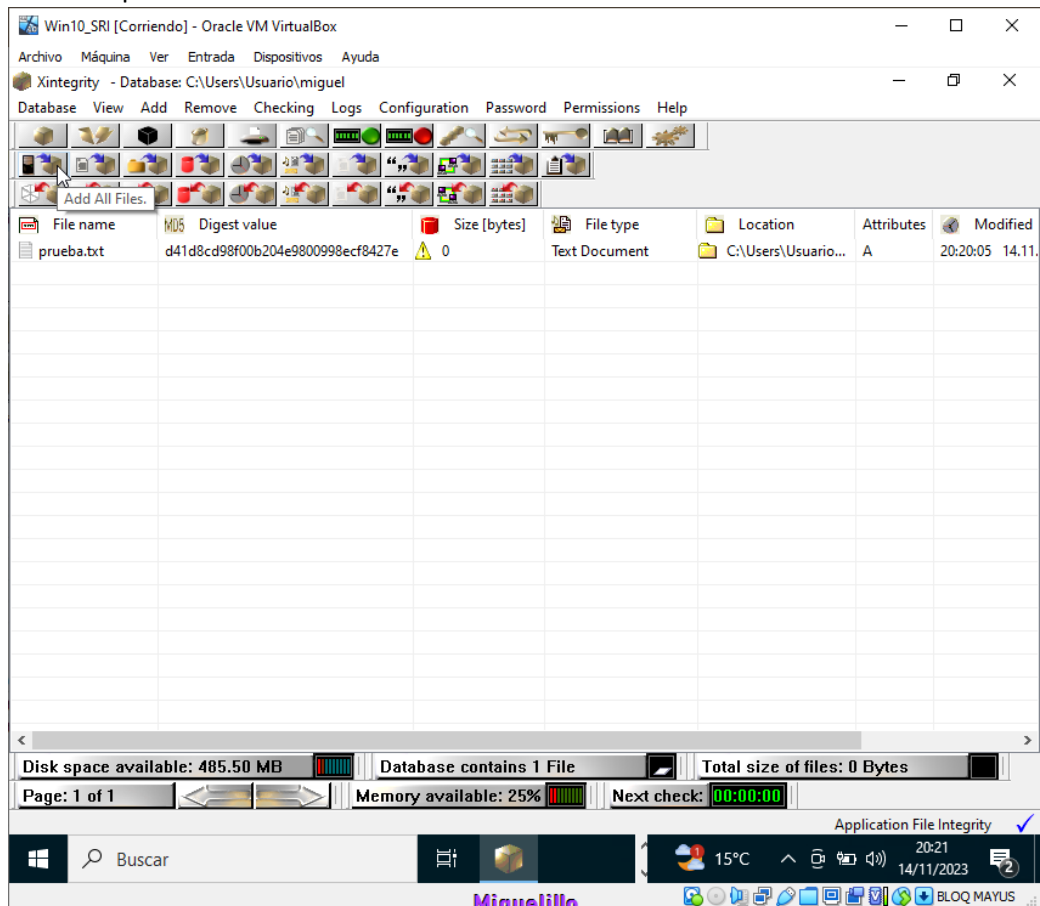
Instalamos Xintegrity le damos a → Continue



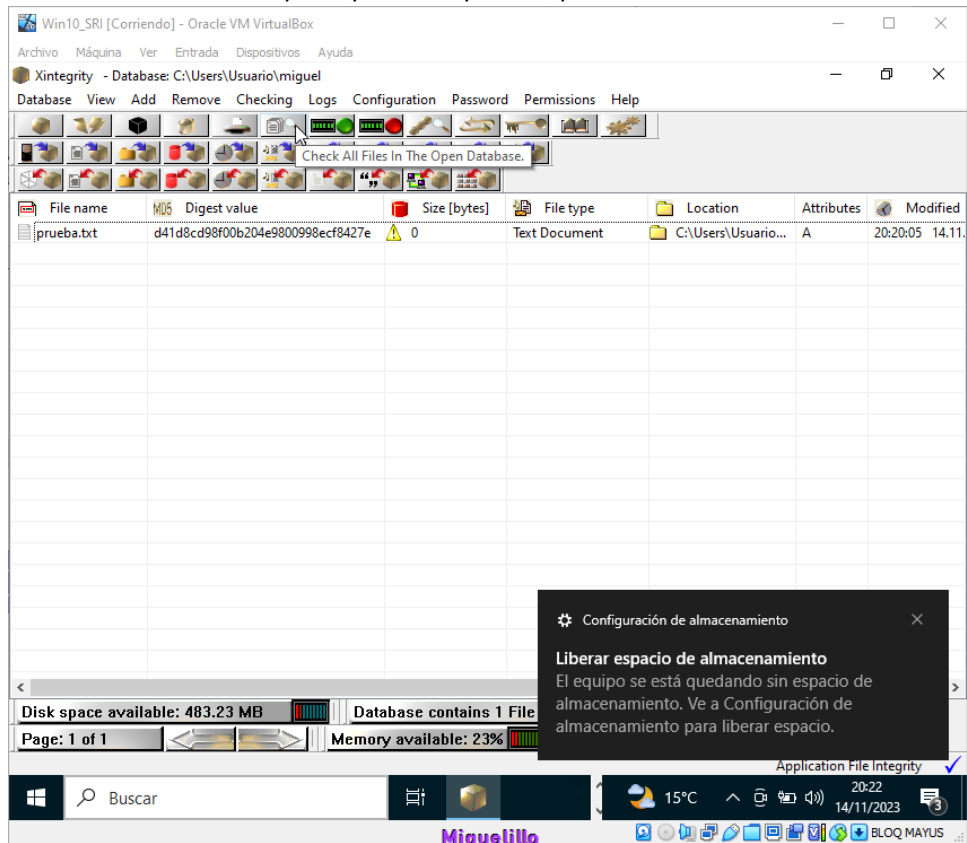
Clico en New database



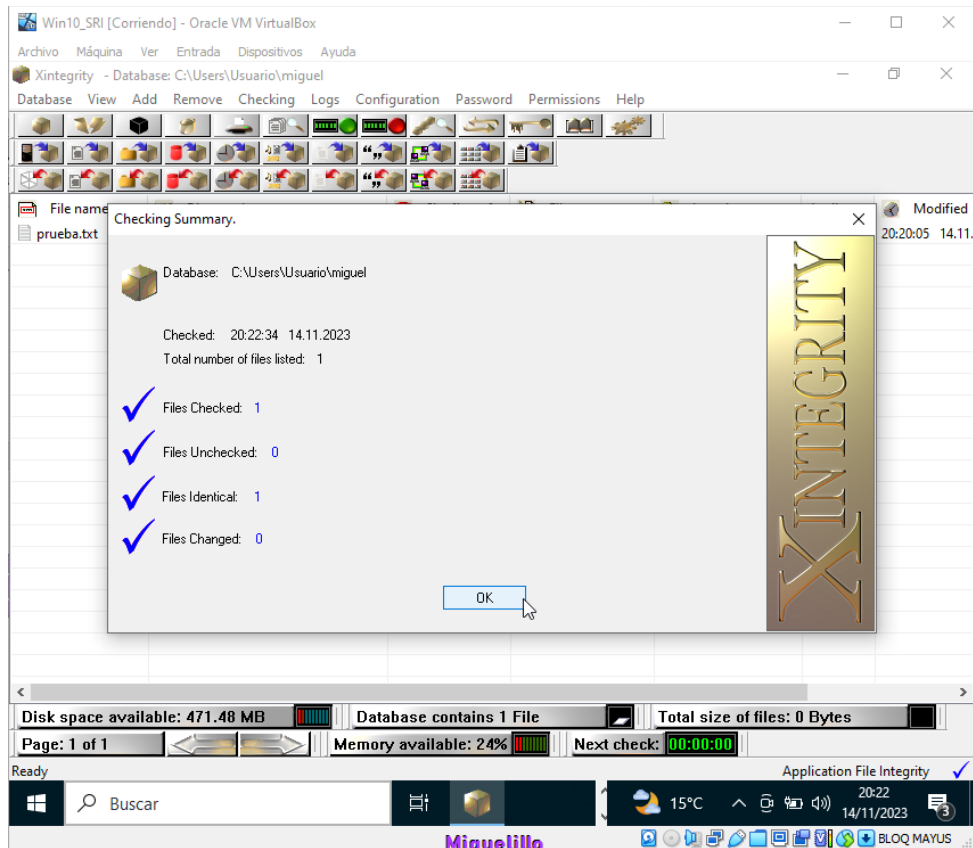
Clicamos para añadir un nuevo archivo



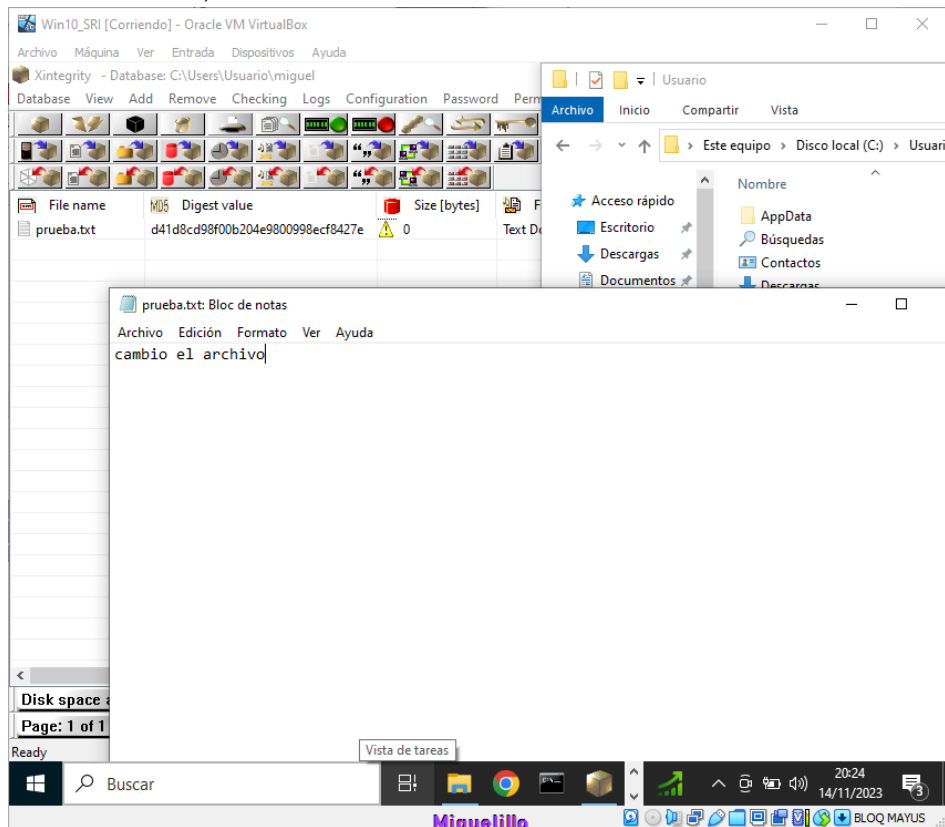
Podemos clicar en chequear para comprobar que no ha sido modificado



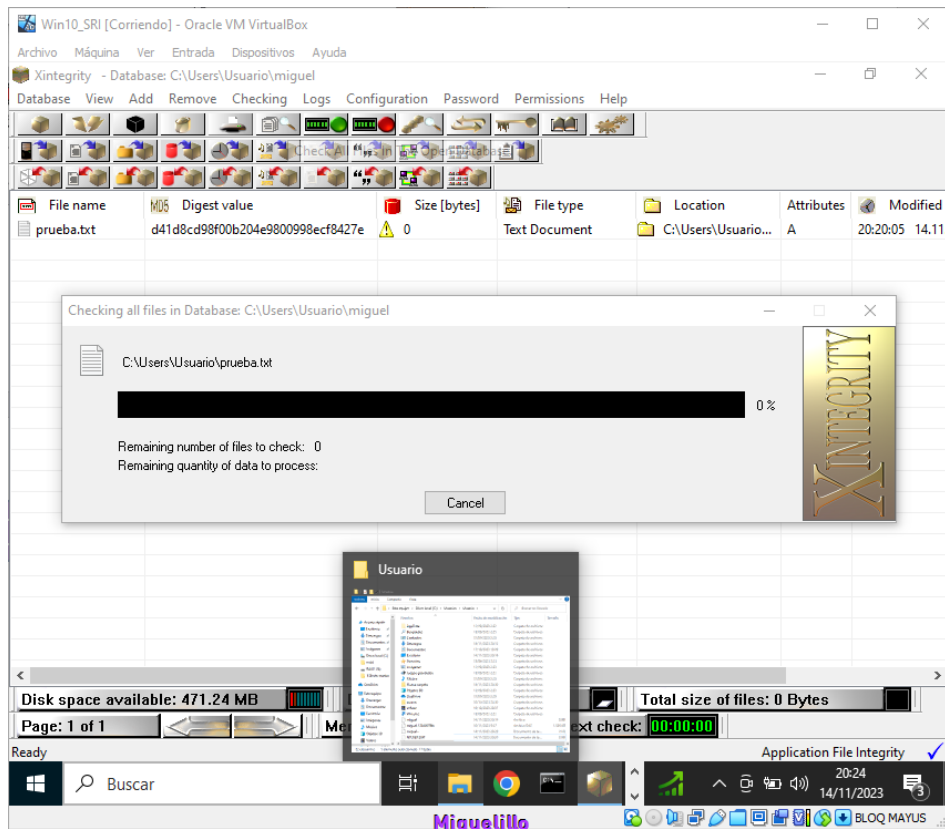
Nos muestra el resultado que no ha sido modificado



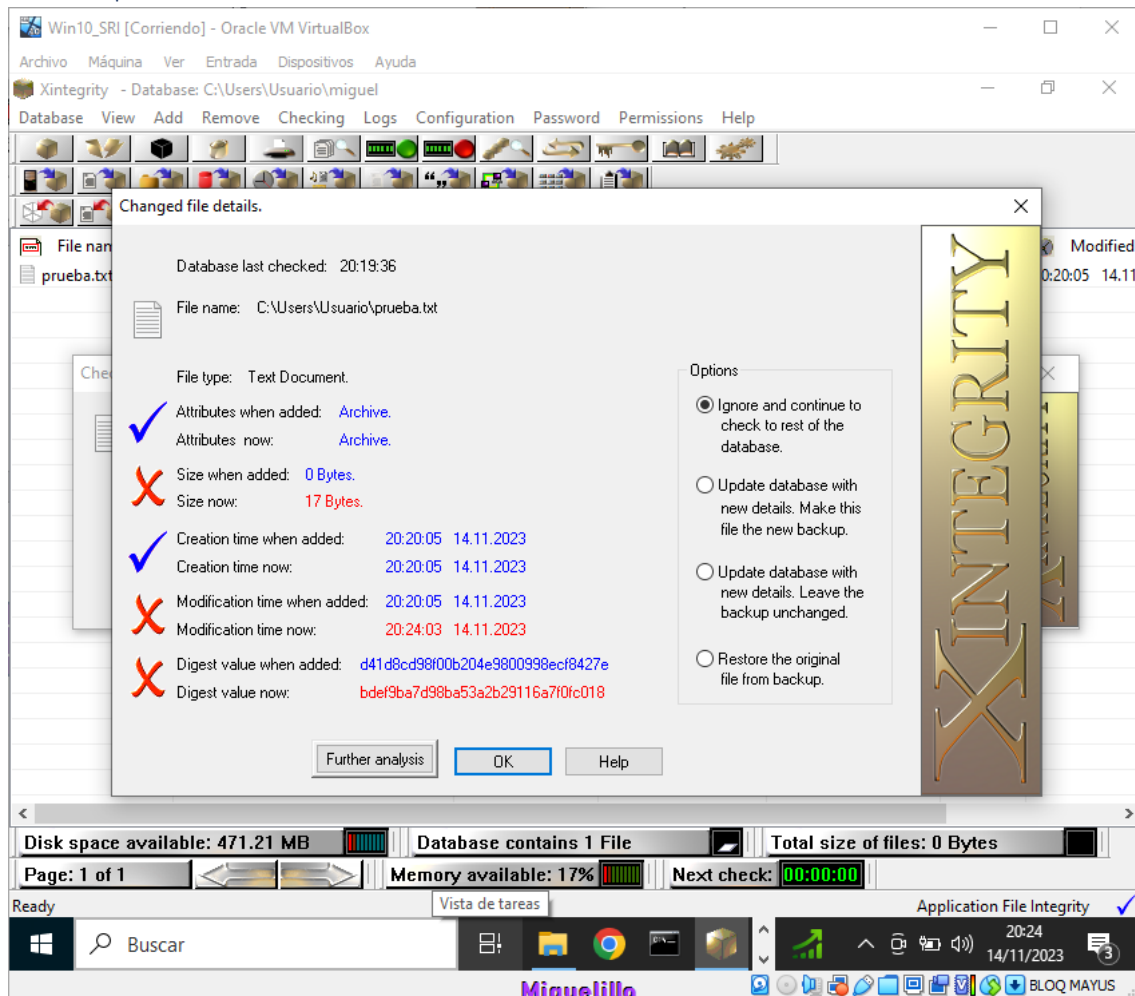
Lo cambiamos y volvemos a analizar



Volvemos a analizar



Vemos que ha sido modificado



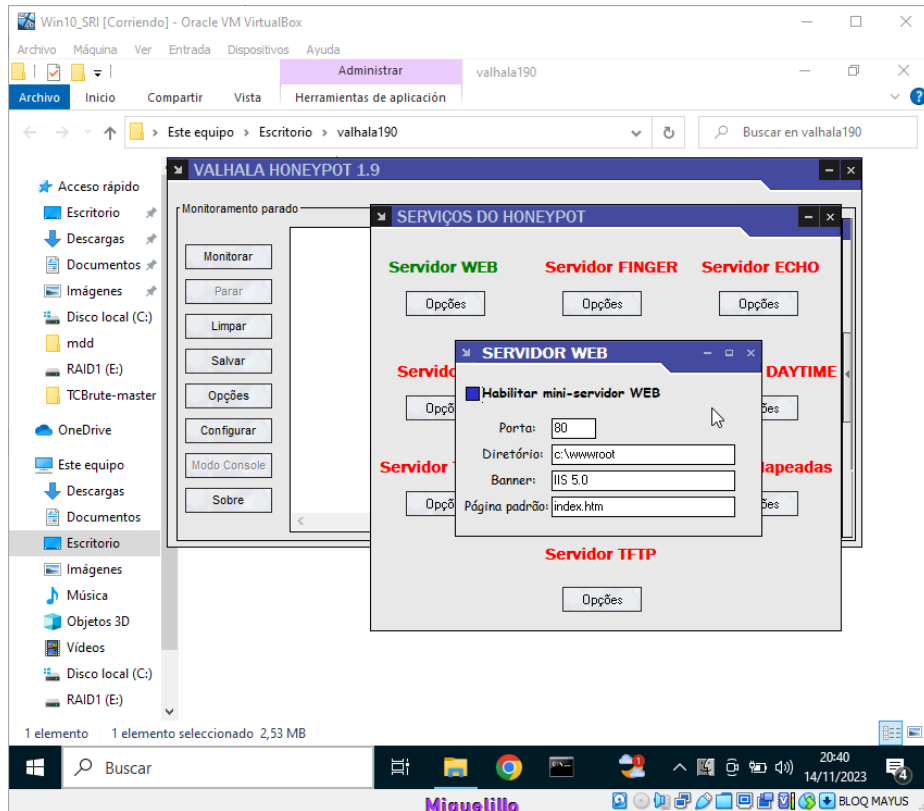
Honey pot

Un honeypot es una trampa de seguridad informática diseñada para atraer y detectar posibles ataques cibernéticos. Se utiliza para estudiar las tácticas, técnicas y procedimientos de los atacantes, así como para recopilar información sobre posibles amenazas y vulnerabilidades.

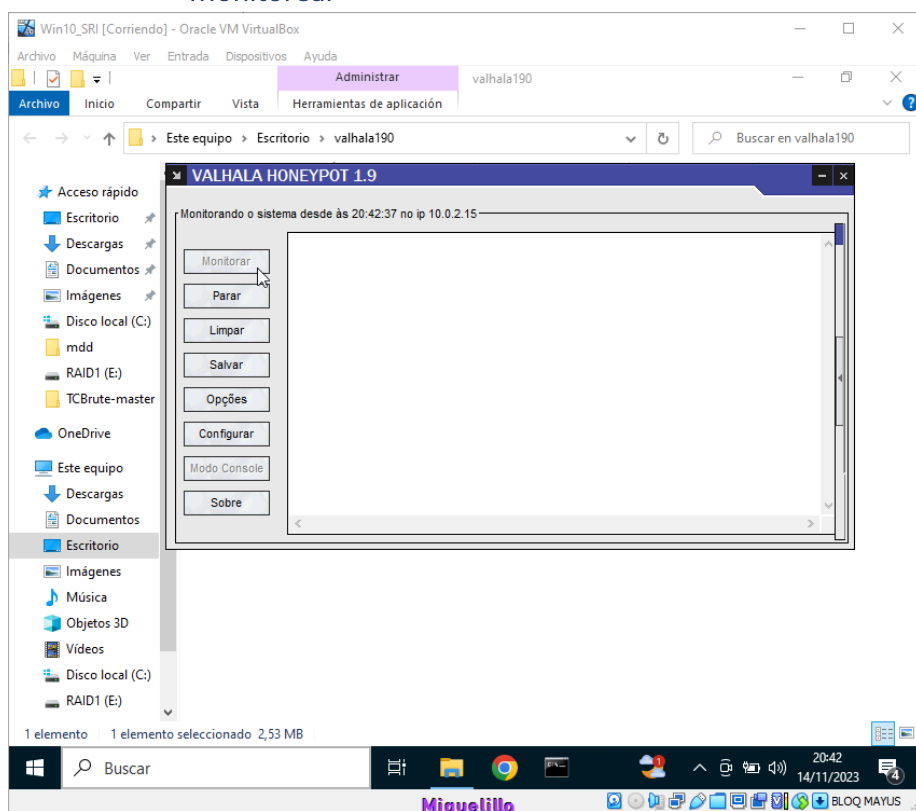
Listado de Software Honey pot:

- Honeyd: Simula múltiples servicios y sistemas operativos.
- Snort: Combina funciones de detección de intrusiones y honeypot.
- Cowrie: Emula un servidor SSH para atraer a atacantes.
- Dionaea: Honeypot de bajo-interacción que emula servicios de red.
- Glustopf: Emula vulnerabilidades web para atraer ataques web.
- Valhala Honey pot es un honeypot fácil de usar para el sistema Windows. El software cuenta con los siguientes servicios: http (web), ftp, tftp, finger, pop3, smtp, echo, daytime, telnet y redireccionamiento de puertos. Algunos servicios son reales, otros son una simulación.

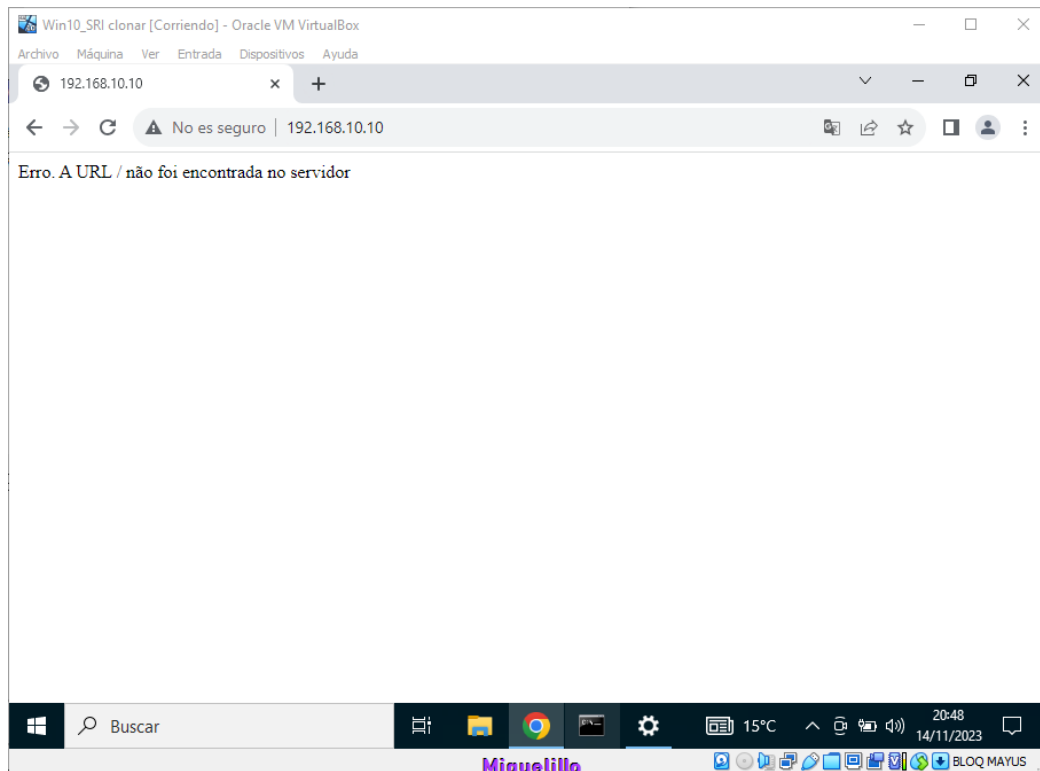
Voy a usar Valhala que es una aplicación portable para activar los servicios del honeypot vamos a **Opciones→Servidor WEB→ Opciones** y marcamos **Habilitar mini-servidor WEB**



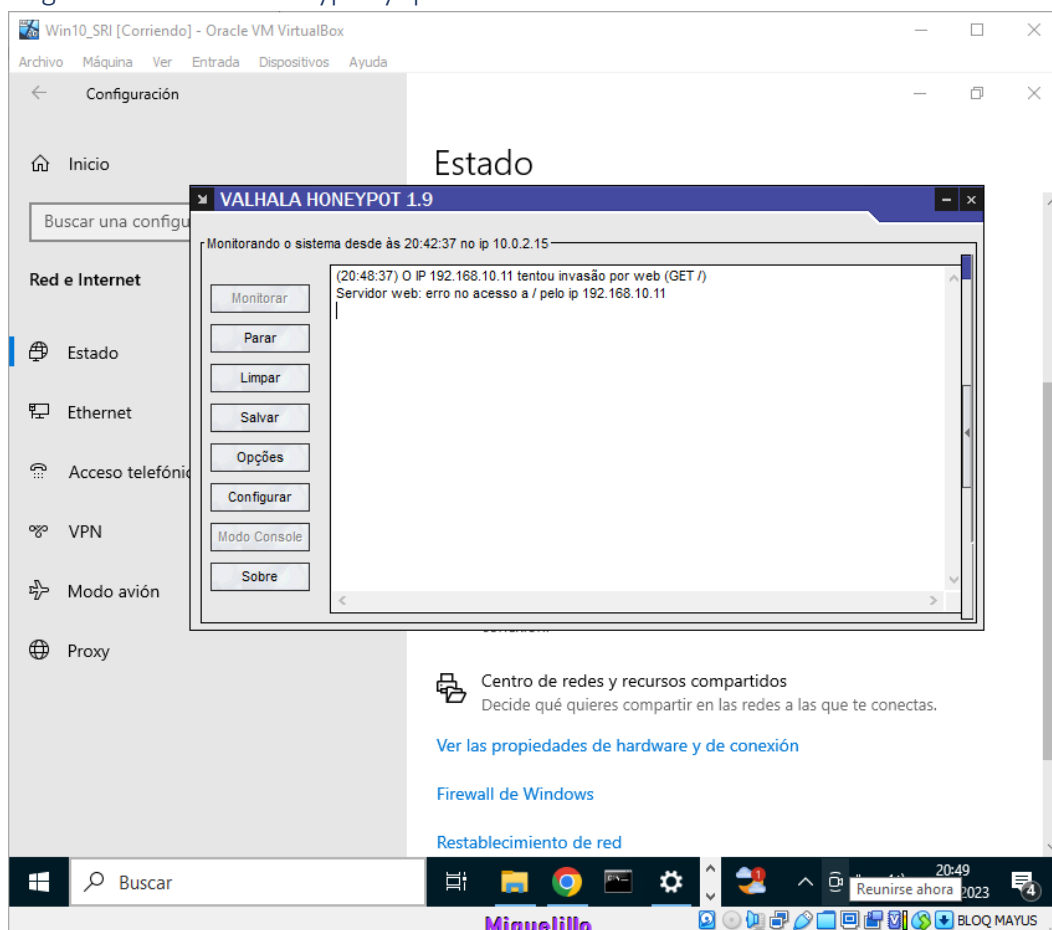
Y clicamos en **Monitorear**



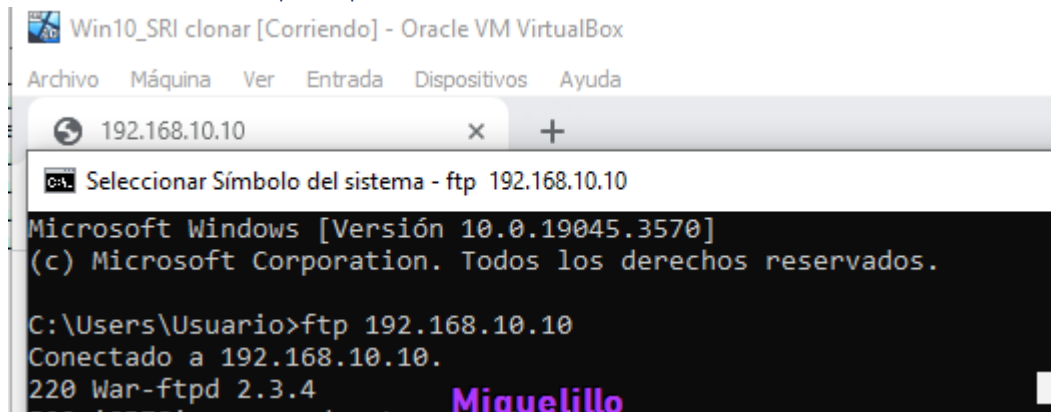
Desde un cliente accedemos al servicio web



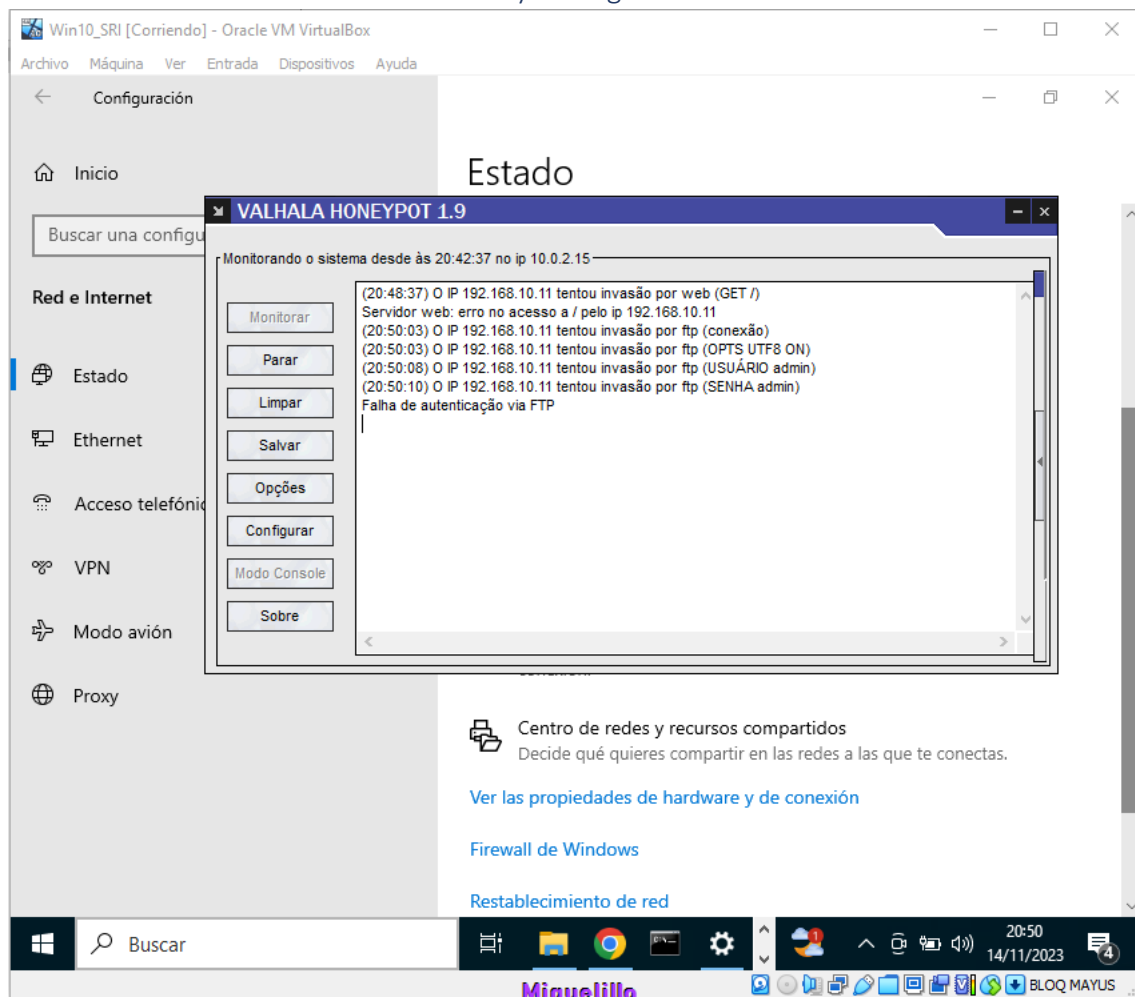
Registra la IP en el honeypot y que han intentado acceder



Probamos a acceder por ftp



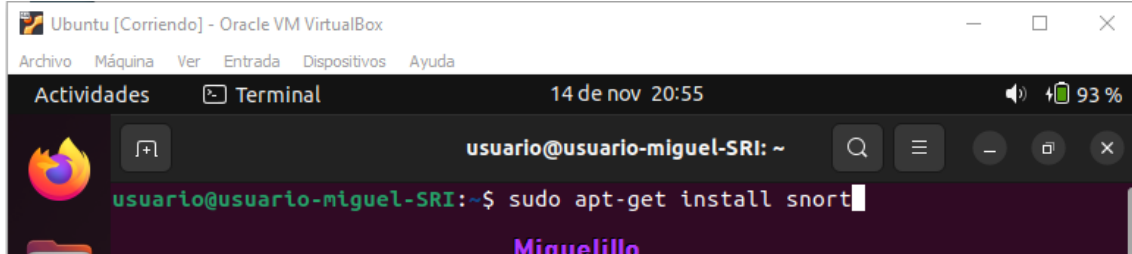
Podemos ver el fallo de autenticación y los registros con la IP



b) SISTEMA DE DETECCIÓN DE INTRUSOS (IDS) : NetIDS

Solución:

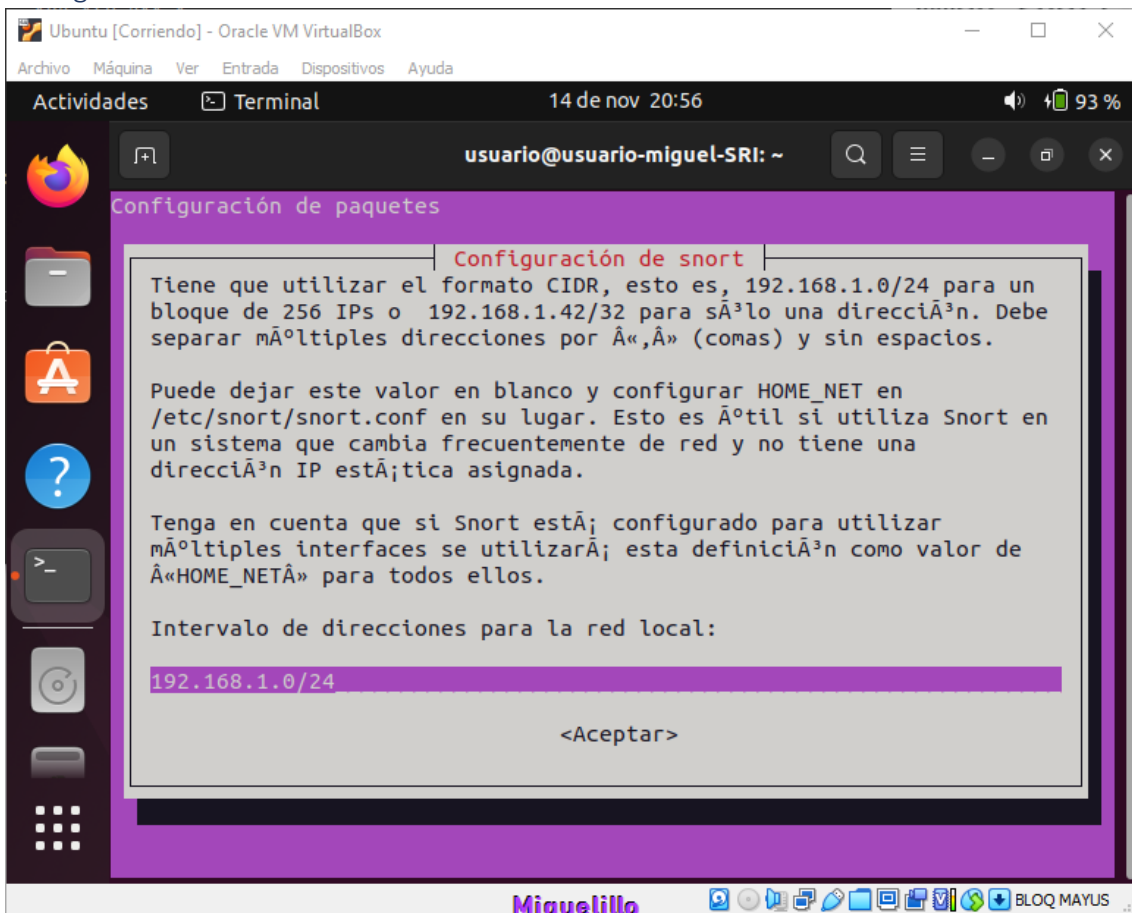
Instalamos Snort



The screenshot shows a terminal window titled 'Ubuntu [Corriendo] - Oracle VM VirtualBox'. The terminal prompt is 'usuario@usuario-miguel-SRI: ~'. The command 'sudo apt-get install snort' has been entered. The output shows the package list being updated and the installation of Snort starting. The name 'Miguelillo' is visible in the bottom right corner of the terminal window.

```
usuario@usuario-miguel-SRI:~$ sudo apt-get install snort
```

Configuramos la red



The screenshot shows a terminal window titled 'Ubuntu [Corriendo] - Oracle VM VirtualBox'. The terminal prompt is 'usuario@usuario-miguel-SRI: ~'. A dialog box titled 'Configuración de paquetes' is open, showing the 'Configuración de snort' section. The dialog box contains the following text:

Configuración de snort

Tiene que utilizar el formato CIDR, esto es, 192.168.1.0/24 para un bloque de 256 IPs o 192.168.1.42/32 para sólo una dirección. Debe separar múltiples direcciones por «,» (comas) y sin espacios.

Puede dejar este valor en blanco y configurar HOME_NET en /etc/snort/snort.conf en su lugar. Esto es útil si utiliza Snort en un sistema que cambia frecuentemente de red y no tiene una dirección IP estática asignada.

Tenga en cuenta que si Snort está configurado para utilizar múltiples interfaces se utilizará esta definición como valor de «HOME_NET» para todos ellos.

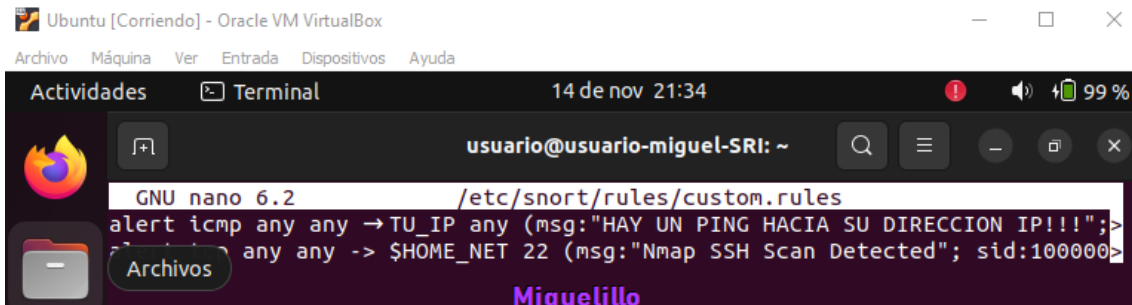
Intervalo de direcciones para la red local:

192.168.1.0/24

<Aceptar>

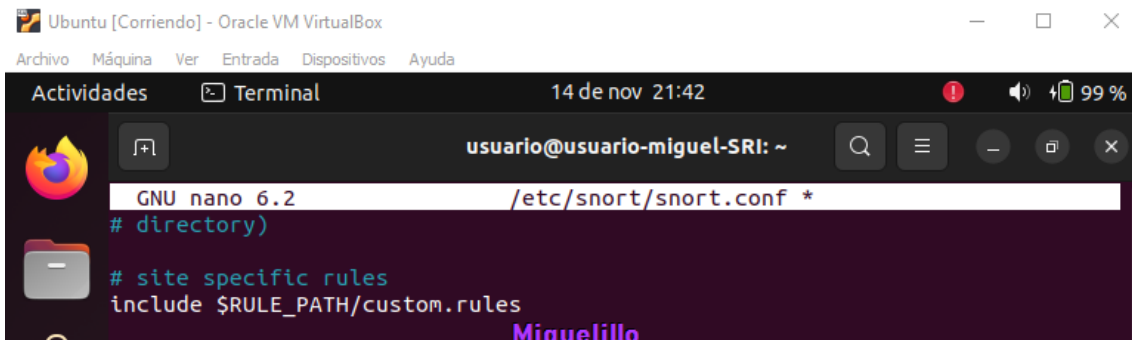
The name 'Miguelillo' is visible in the bottom right corner of the terminal window.

Creamos un archivo con regla para detectar NMAP y pings a el router este archivo estará en **/etc/snort/rules** y se llamará **custom.rules**



```
GNU nano 6.2 /etc/snort/rules/custom.rules
alert icmp any any -> TU_IP any (msg:"HAY UN PING HACIA SU DIRECCION IP!!!";>
any any -> $HOME_NET 22 (msg:"Nmap SSH Scan Detected"; sid:1000000)
```

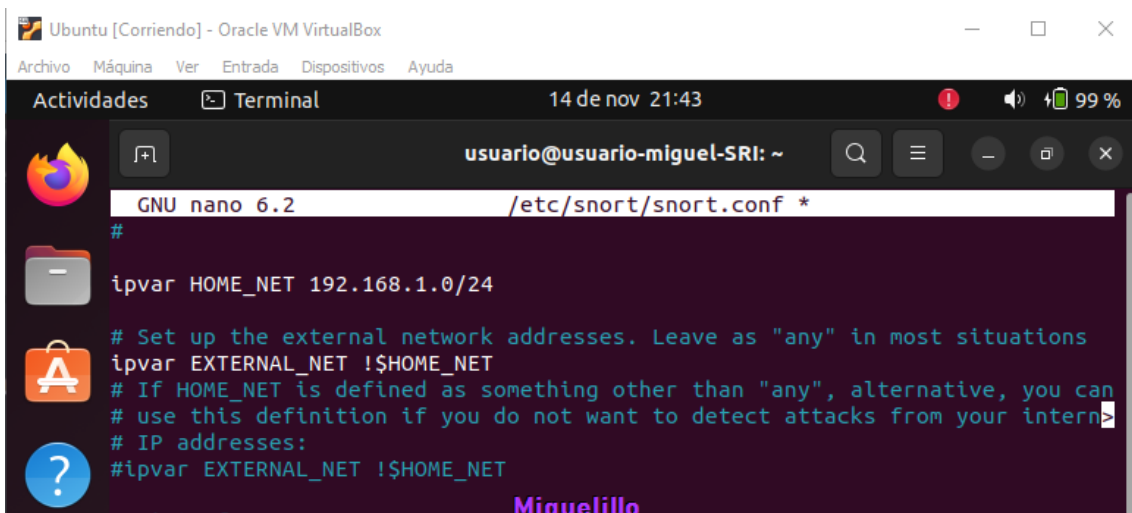
En el archivo **snort.conf** declaramos donde se encuentran las reglas



```
GNU nano 6.2 /etc/snort/snort.conf *
# directory)

# site specific rules
include $RULE_PATH/custom.rules
```

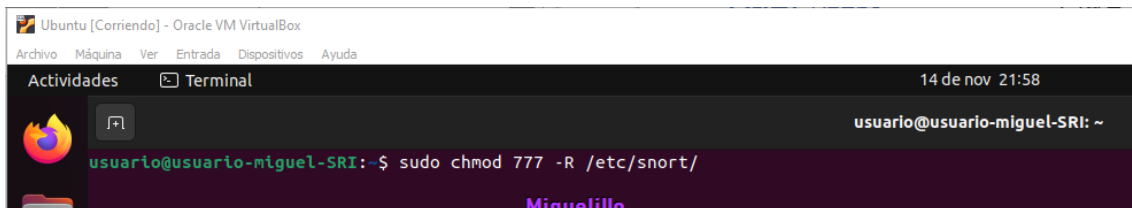
Declaramos la dirección de red y guardamos



```
GNU nano 6.2 /etc/snort/snort.conf *
#
ipvar HOME_NET 192.168.1.0/24

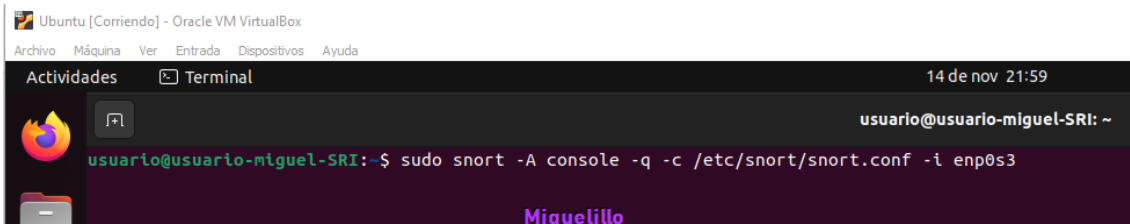
# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET !$HOME_NET
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your intern
# IP addresses:
#ipvar EXTERNAL_NET !$HOME_NET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

Importante darle permisos al archivo que hemos creado



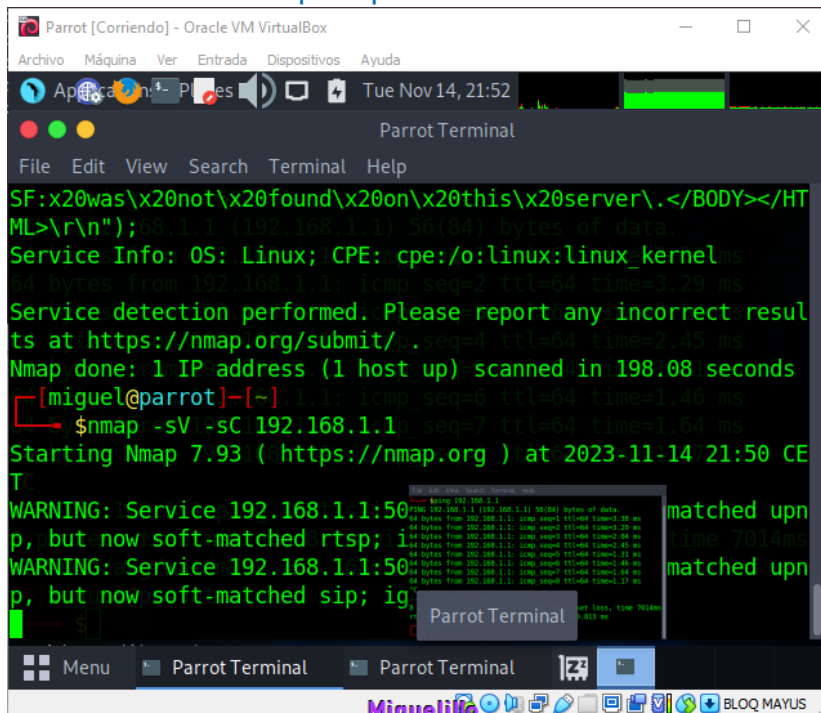
```
usuario@usuario-miguel-SRI:~$ sudo chmod 777 -R /etc/snort/
```

Ejecutamos el snort con nuestra reglas



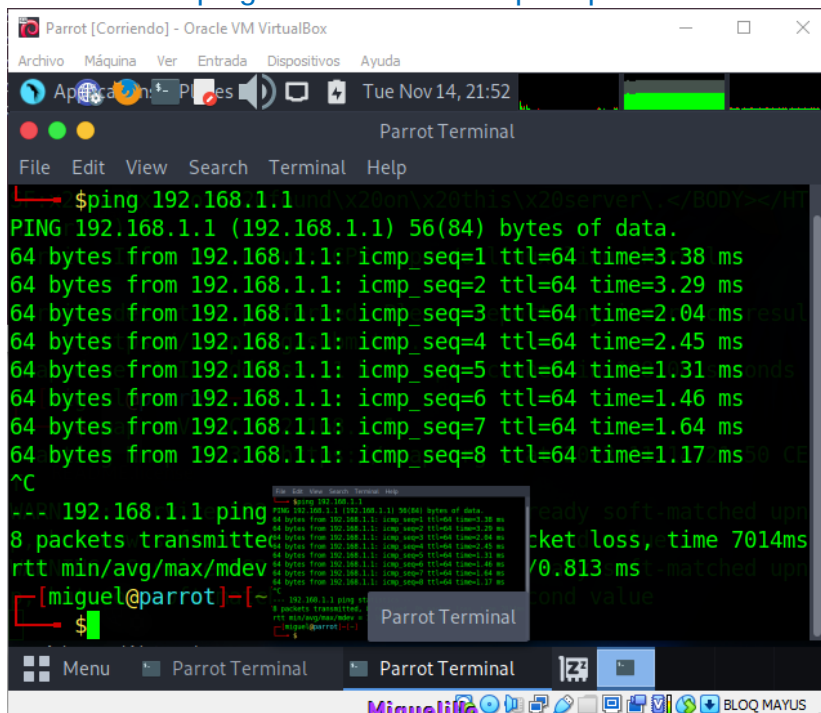
```
Ubuntu [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 14 de nov 21:59
usuario@usuario-miguel-SRI: ~
usuario@usuario-miguel-SRI:~$ sudo snort -A console -q -c /etc/snort/snort.conf -i enp0s3
```

Hacemos un NMAP para probarlo



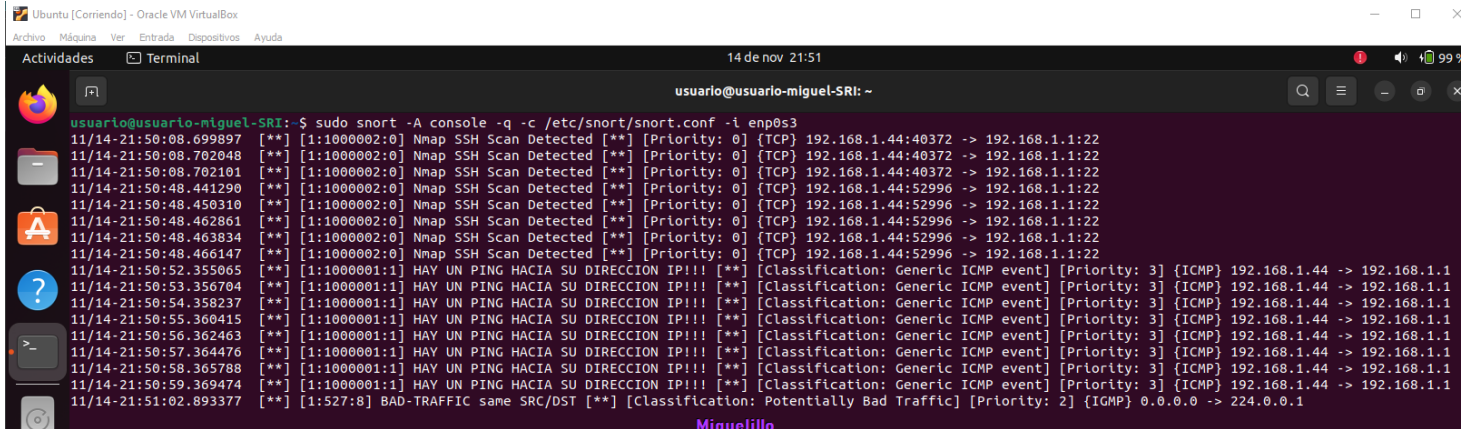
```
Parrot [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Tue Nov 14, 21:52
Parrot Terminal
File Edit View Search Terminal Help
SF:x20was\x20not\x20found\x20on\x20this\x20server\.</BODY></HT
ML>\r\n");68.1.1 (192.168.1.1) 56(84) bytes of data.
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=3.29 ms
Service detection performed. Please report any incorrect results
at https://nmap.org/submit/. seq=4 ttl=64 time=2.45 ms
Nmap done: 1 IP address (1 host up) scanned in 198.08 seconds
[miguel@parrot]~$ nmap -sV -sC 192.168.1.1
seq=7 ttl=64 time=1.64 ms
Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-14 21:50 CE
T
WARNING: Service 192.168.1.1:50 matched upnp, but now soft-matched rtsp; 1
WARNING: Service 192.168.1.1:50 matched upnp, but now soft-matched sip; ig
```

Hacemos un ping al router también para probarlo



```
Parrot [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Tue Nov 14, 21:52
Parrot Terminal
File Edit View Search Terminal Help
$ping 192.168.1.1und\x20not\x20found\x20on\x20this\x20server\.</BODY></HT
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=3.38 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=3.29 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=2.04 ms result
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=2.45 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=1.31 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=64 time=1.46 ms
64 bytes from 192.168.1.1: icmp_seq=7 ttl=64 time=1.64 ms
64 bytes from 192.168.1.1: icmp_seq=8 ttl=64 time=1.17 ms
8 packets transmitted, 8 received, 0% packet loss, time 7014ms
rtt min/avg/max/mdev = 1.17/1.64/3.38/0.813 ms
[miguel@parrot]~$
```

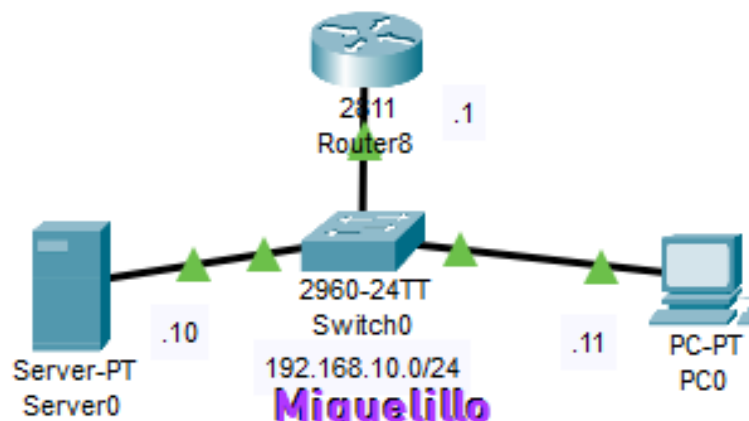
Vemos como lo detecta la regla



```
usuario@usuario-miguel-SRI: ~  
usuario@usuario-miguel-SRI:~$ sudo snort -A console -q -c /etc/snort/snort.conf -i enp0s3  
11/14-21:50:08.699897 ** [1:1000002:0] Nmap SSH Scan Detected ** [Priority: 0] {TCP} 192.168.1.44:40372 -> 192.168.1.1:22  
11/14-21:50:08.702048 ** [1:1000002:0] Nmap SSH Scan Detected ** [Priority: 0] {TCP} 192.168.1.44:40372 -> 192.168.1.1:22  
11/14-21:50:08.702101 ** [1:1000002:0] Nmap SSH Scan Detected ** [Priority: 0] {TCP} 192.168.1.44:40372 -> 192.168.1.1:22  
11/14-21:50:48.441290 ** [1:1000002:0] Nmap SSH Scan Detected ** [Priority: 0] {TCP} 192.168.1.44:52996 -> 192.168.1.1:22  
11/14-21:50:48.450310 ** [1:1000002:0] Nmap SSH Scan Detected ** [Priority: 0] {TCP} 192.168.1.44:52996 -> 192.168.1.1:22  
11/14-21:50:48.462861 ** [1:1000002:0] Nmap SSH Scan Detected ** [Priority: 0] {TCP} 192.168.1.44:52996 -> 192.168.1.1:22  
11/14-21:50:48.463834 ** [1:1000002:0] Nmap SSH Scan Detected ** [Priority: 0] {TCP} 192.168.1.44:52996 -> 192.168.1.1:22  
11/14-21:50:48.466147 ** [1:1000002:0] Nmap SSH Scan Detected ** [Priority: 0] {TCP} 192.168.1.44:52996 -> 192.168.1.1:22  
11/14-21:50:52.355065 ** [1:1000001:1] HAY UN PING HACIA SU DIRECCION IP!!! ** [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.1.44 -> 192.168.1.1  
11/14-21:50:53.356704 ** [1:1000001:1] HAY UN PING HACIA SU DIRECCION IP!!! ** [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.1.44 -> 192.168.1.1  
11/14-21:50:54.358237 ** [1:1000001:1] HAY UN PING HACIA SU DIRECCION IP!!! ** [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.1.44 -> 192.168.1.1  
11/14-21:50:55.360415 ** [1:1000001:1] HAY UN PING HACIA SU DIRECCION IP!!! ** [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.1.44 -> 192.168.1.1  
11/14-21:50:56.362463 ** [1:1000001:1] HAY UN PING HACIA SU DIRECCION IP!!! ** [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.1.44 -> 192.168.1.1  
11/14-21:50:57.364476 ** [1:1000001:1] HAY UN PING HACIA SU DIRECCION IP!!! ** [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.1.44 -> 192.168.1.1  
11/14-21:50:58.365788 ** [1:1000001:1] HAY UN PING HACIA SU DIRECCION IP!!! ** [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.1.44 -> 192.168.1.1  
11/14-21:50:59.369474 ** [1:1000001:1] HAY UN PING HACIA SU DIRECCION IP!!! ** [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.1.44 -> 192.168.1.1  
11/14-21:51:02.893377 ** [1:527:8] BAD-TRAFFIC same SRC/DST ** [Classification: Potentially Bad Traffic] [Priority: 2] {IGMP} 0.0.0.0 -> 224.0.0.1
```

Miguelillo

Escenario



Configuramos los parámetros básicos del router, como la dirección IP

Router4

Physical Config CLI Attributes

IOS Command Line Interface

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Rlmiguelliro
Rlmiguelliro(config)#interface Gi 0/0
Rlmiguelliro(config-if)#ip address 192.168.10.1 255.255.255.0
Rlmiguelliro(config-if)#no shutdown

Rlmiguelliro(config-if)#line vty 0 4
Rlmiguelliro(config-line)#password invess
Rlmiguelliro(config-line)#exec-timeout
% Incomplete command.
Rlmiguelliro(config-line)#exec-timeout 5 0
Rlmiguelliro(config-line)#
```

Miguelillo

Creo la carpeta ipsdir

Router4

Physical Config CLI Attributes

IOS Command Line Interface

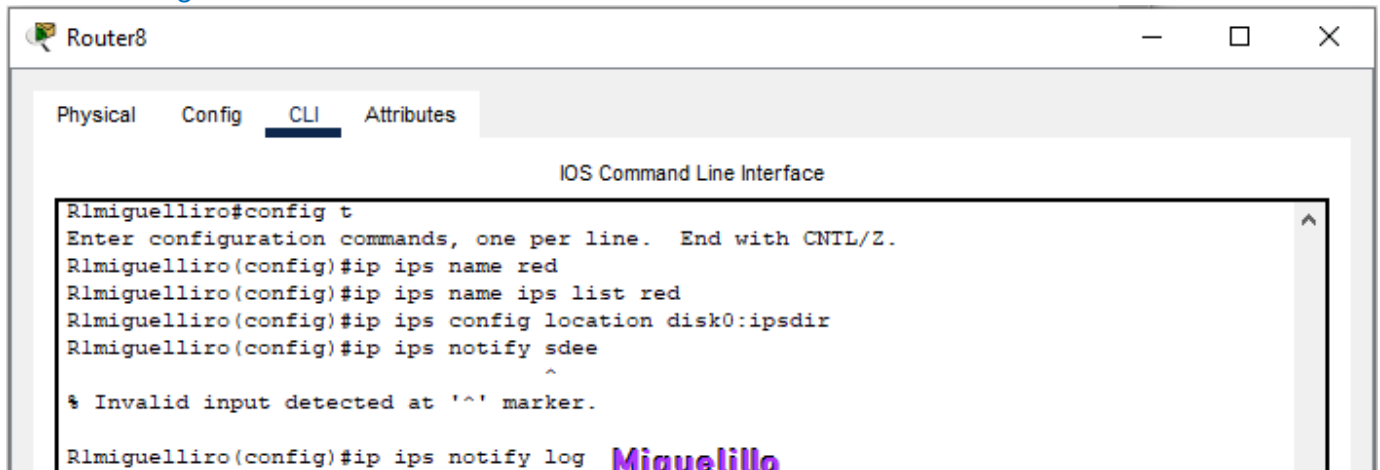
```
Rlmiguelliro#mkdir disk0:ipsdir
Create directory filename [disk0:ipsdir]?config t
Created dir flash:config t
```

Miguelillo

Configuramos

el

IPS



Añadimos

el

syslog

```
Rlmiguelliro(config)#service timestamps log datetime msec
Rlmiguelliro(config)#logging 192.168.10.10
Rlmiguelliro(config)#
```

Miauelillo

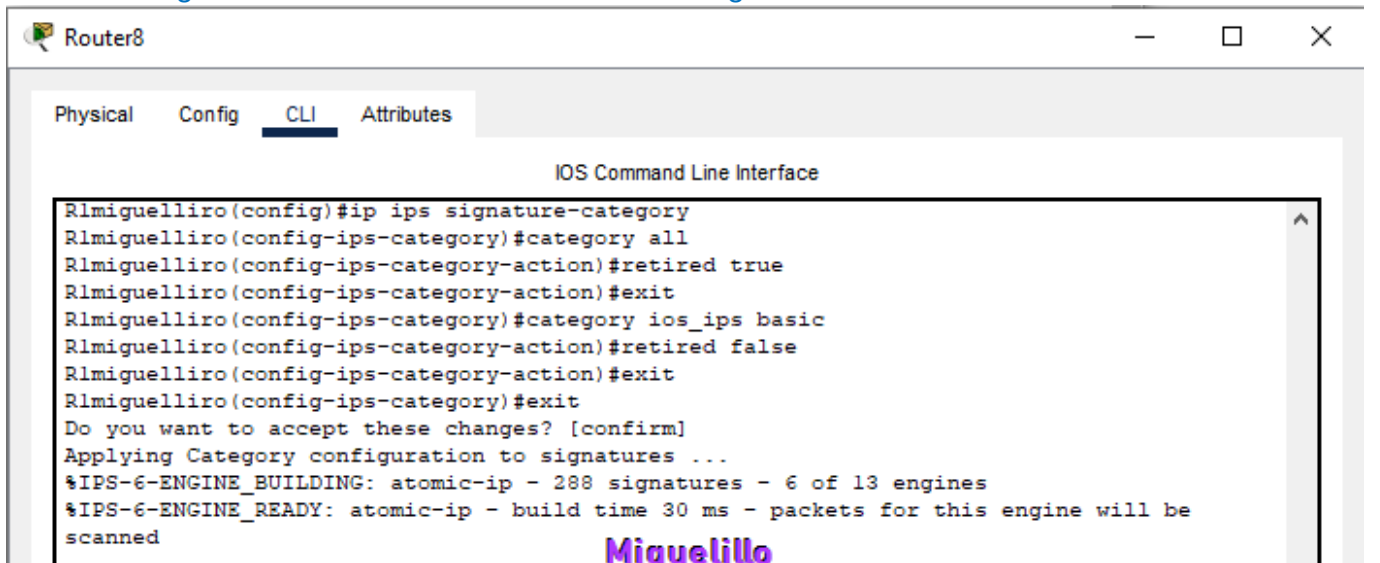
Configuramos

la

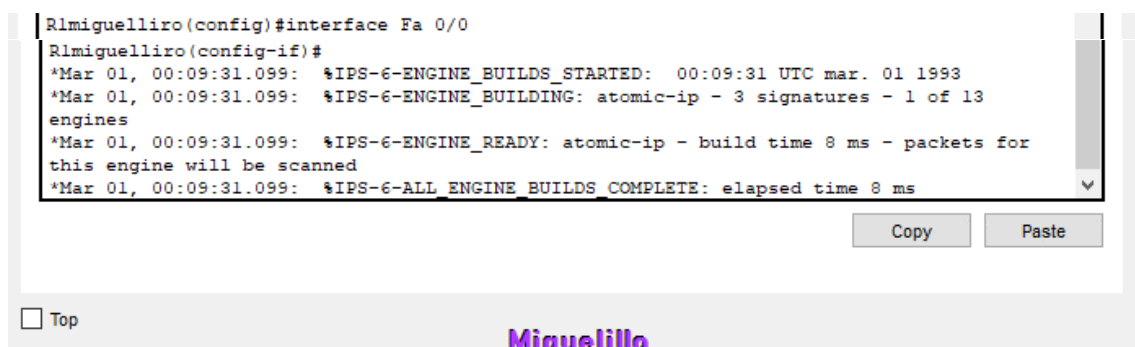
categoría

del

IPS



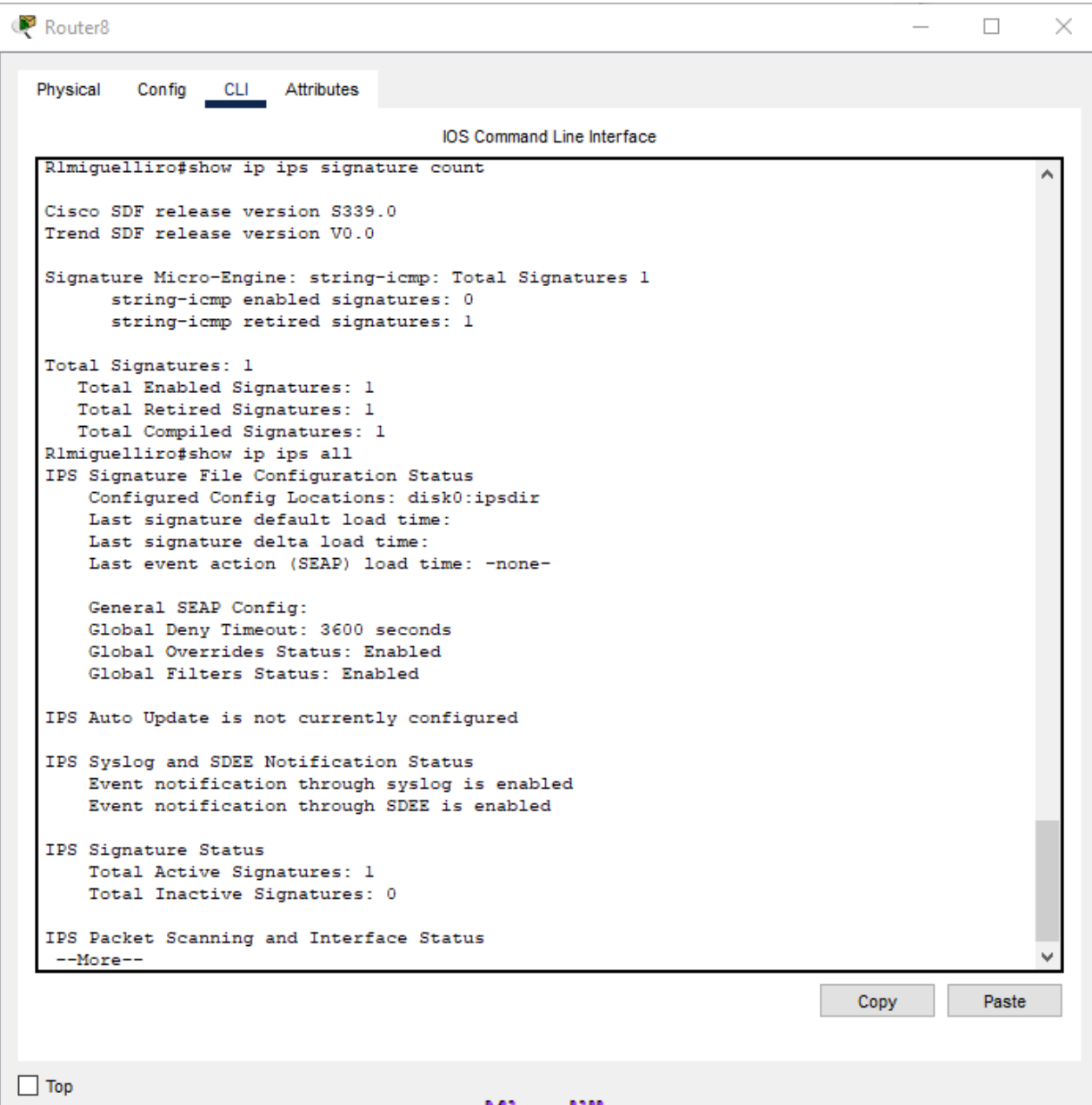
Configuramos la interfaz



Configuramos los eventos del IPS para que se notifique al Syslog

```
Rlmiguelliro(config)#ip ips signature-definition
Rlmiguelliro(config-sigdef)#signature 2004 0
Rlmiguelliro(config-sigdef-sig)#status
Rlmiguelliro(config-sigdef-sig-status)#retired false
Rlmiguelliro(config-sigdef-sig-status)#enable true
Rlmiguelliro(config-sigdef-sig-status)#engine
Rlmiguelliro(config-sigdef-sig-engine)#event-action produce-alert
Rlmiguelliro(config-sigdef-sig-engine)#event-action deny-paket-inline
```

Podemos ver la configuración del IPS



The screenshot shows a terminal window titled "Router8" with tabs for Physical, Config, CLI, and Attributes. The CLI tab is active, displaying the "IOS Command Line Interface". The user has entered the command "show ip ips signature count", which returns the following output:

```
Rlmiguelliro#show ip ips signature count

Cisco SDF release version S339.0
Trend SDF release version V0.0

Signature Micro-Engine: string-icmp: Total Signatures 1
    string-icmp enabled signatures: 0
    string-icmp retired signatures: 1

Total Signatures: 1
    Total Enabled Signatures: 1
    Total Retired Signatures: 1
    Total Compiled Signatures: 1
Rlmiguelliro#show ip ips all
IPS Signature File Configuration Status
    Configured Config Locations: disk0:ipsdir
    Last signature default load time:
    Last signature delta load time:
    Last event action (SEAP) load time: -none-

General SEAP Config:
    Global Deny Timeout: 3600 seconds
    Global Overrides Status: Enabled
    Global Filters Status: Enabled

IPS Auto Update is not currently configured

IPS Syslog and SDEE Notification Status
    Event notification through syslog is enabled
    Event notification through SDEE is enabled

IPS Signature Status
    Total Active Signatures: 1
    Total Inactive Signatures: 0

IPS Packet Scanning and Interface Status
--More--
```

At the bottom of the window, there are "Copy" and "Paste" buttons, and a "Top" link.

Mensajes de syslog

Server0

Physical Config **Services** Desktop Programming Attributes

SERVICES

- HTTP
- DHCP
- DHCPv6
- TFTP
- DNS
- SYSLOG**
- AAA
- NTP
- EMAIL
- FTP
- IoT
- VM Management
- Radius EAP

Syslog

Service ☒ On ☐ Off

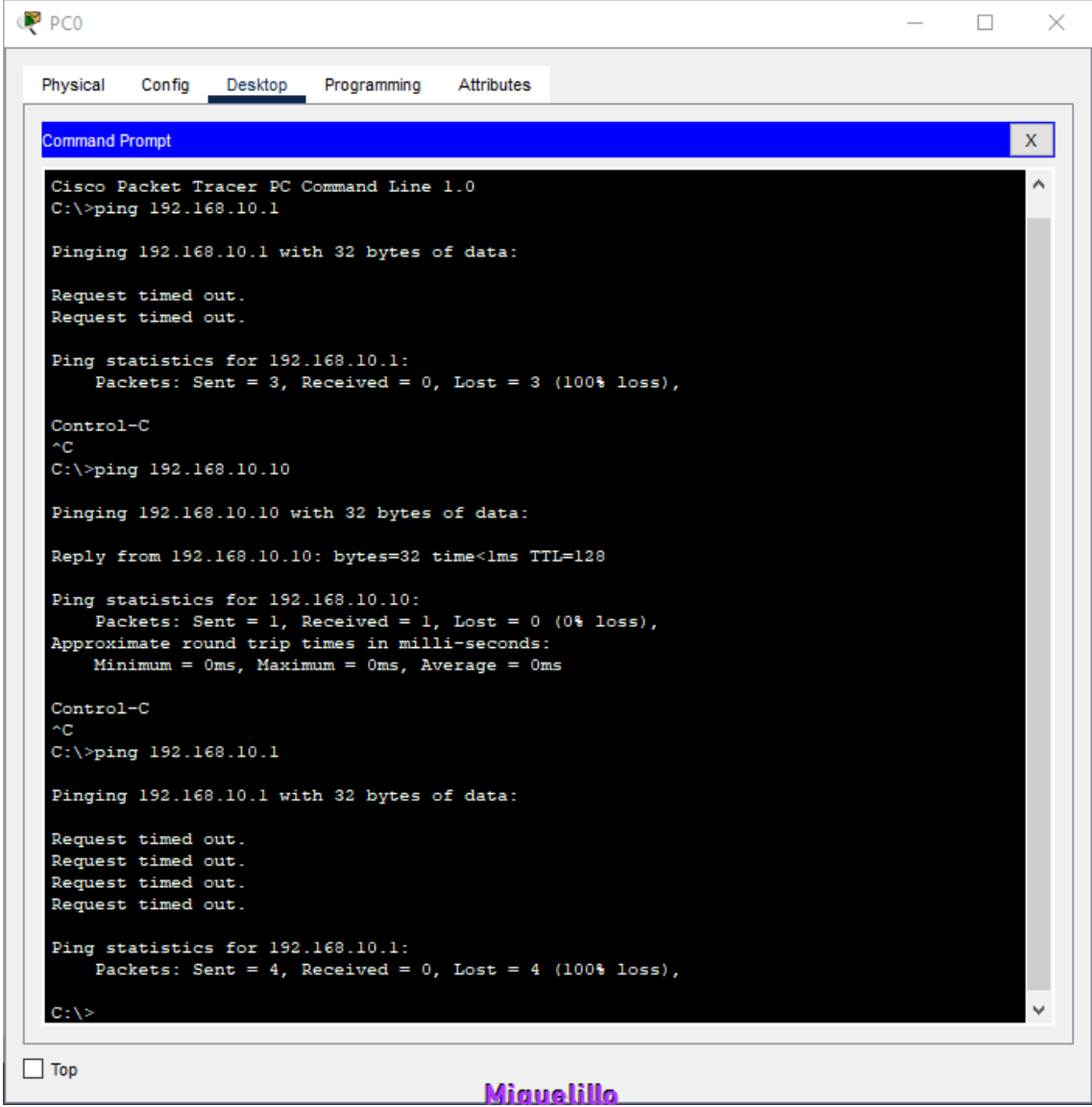
	Time	HostName	Message
1	03.01.1993 12:14:01.088 AM	192.168.10.1	%SYS-5-CONFIG_: Configured from console b...
2	03.01.1993 12:14:01.088 AM	192.168.10.1	: %SYS-6-LOGGINGHOST_STARTST...
3	03.01.1993 12:21:23.886 AM	192.168.10.1	%SYS-5-CONFIG_: Configured from console b...
4	03.01.1993 12:22:51.821 AM	192.168.10.1	%IPS-4-SIGNATURE: Sig:...
5	03.01.1993 12:22:57.847 AM	192.168.10.1	%IPS-4-SIGNATURE: Sig:...
6	03.01.1993 12:23:03.860 AM	192.168.10.1	%IPS-4-SIGNATURE: Sig:...
7	03.01.1993 12:23:11.281 AM	192.168.10.1	%IPS-4-SIGNATURE: Sig:...
8	03.01.1993 12:23:17.285 AM	192.168.10.1	%IPS-4-SIGNATURE: Sig:...
9	03.01.1993 12:23:23.326 AM	192.168.10.1	%IPS-4-SIGNATURE: Sig:...
10	03.01.1993 12:23:29.330 AM	192.168.10.1	%IPS-4-SIGNATURE: Sig:...

Clear Log

☐ Top

Miguelillo

Ping desde cliente



The screenshot shows a Cisco Packet Tracer PC Command Line window for PC0. The window has tabs for Physical, Config, Desktop, Programming, and Attributes. The Desktop tab is active, displaying a black command prompt window with white text. The text shows the execution of two ping commands: one to 192.168.10.1 and another to 192.168.10.10. The first ping to 192.168.10.1 fails with 100% loss. The second ping to 192.168.10.10 succeeds with 0% loss. The third ping to 192.168.10.1 also fails with 100% loss. The window title bar shows 'PC0' and standard window controls. A 'Top' button is visible at the bottom left of the window frame.

```
PC0
Physical Config Desktop Programming Attributes
Command Prompt X
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.10.1

Pinging 192.168.10.1 with 32 bytes of data:

Request timed out.
Request timed out.

Ping statistics for 192.168.10.1:
    Packets: Sent = 3, Received = 0, Lost = 3 (100% loss),

Control-C
^C
C:\>ping 192.168.10.10

Pinging 192.168.10.10 with 32 bytes of data:

Reply from 192.168.10.10: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.10.10:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

Control-C
^C
C:\>ping 192.168.10.1

Pinging 192.168.10.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.10.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```

☐ Top

Miauelillo