P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Licenciatura em Segurança informática em redes de computadores

Segurança de Redes

Análise de Vulnerabilidades

Elementos:

8220190 -Tânia Morais

8220202- Gonçalo Ferraz

Índice

Introdução

Este trabalho prático tem, como objetivo, garantir o conhecimento sobre algumas aplicações e/ou temas anteriormente analisados. Temos alguns exemplos como: *gateway*, diferenciação *telnet*/*SSH* e *HTTP*/ *HTTPS*, *nmap*, *OpenVas*, *Metasploit*.

Para a realização deste trabalho foi necessário montar um cenário constituído por três máquinas virtuais (*kioptrix*, *ubuntu* e *Windows XP* conectadas por uma interface comum a um *gateway*, *kali*. Este gateway, encontra-se ligado a um *host* através de uma interface e à internet através de outra.

Construção de laboratório de testes

Para esta etapa começou-se por proceder à instalação das máquinas virtuais, com o objetivo, de montar o cenário mencionado anteriormente.

Após a instalação, estabeleceu-se algumas fases tais como: conexões de rede, configuração das interfaces, configuração DHCP, encaminhamento IP, configuração da rota, configuração da firewall e, por fim, testar a conectividade das máquinas.

1. Conexões de rede

Neste tópico, definiu-se as conexões de rede nas máquinas virtuais, de forma a facilitar a conectividade entre elas. Para cada máquina foi definido um adaptador, exceto no *kali,* que foram definidos três.

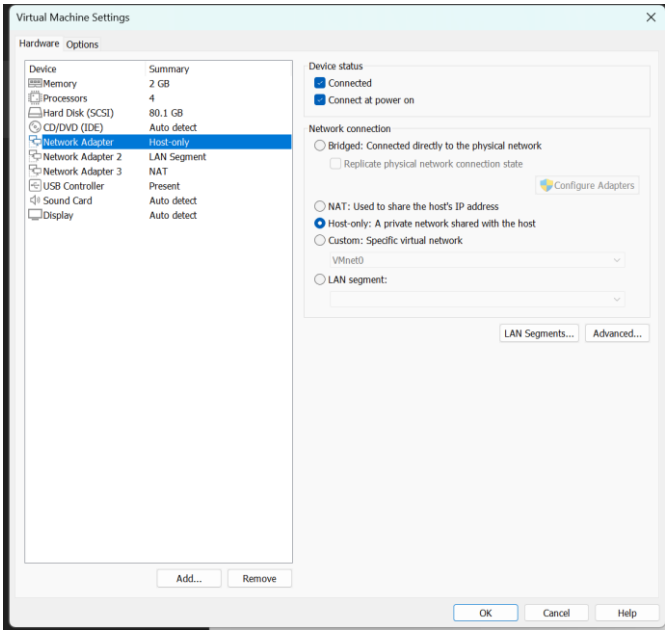


Figura 1-adaptador1 (kali)

Este adaptador serve para estabelecer comunicação entre a máquina virtual e o *host*, computador, possibilitando uma rede isolada.

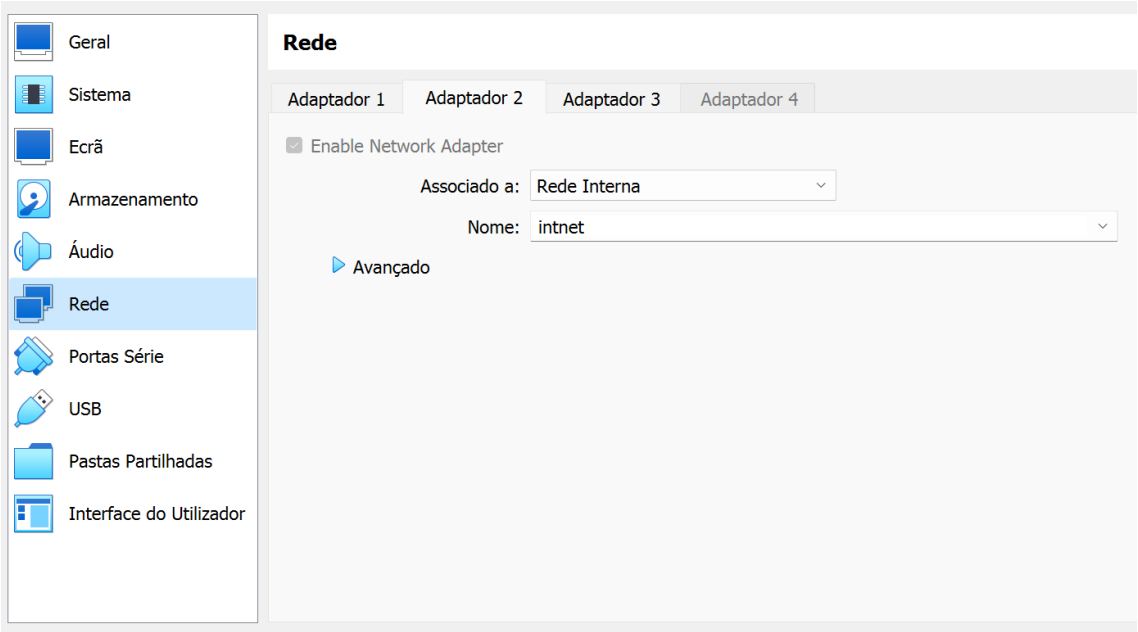


Figura 2- adaptador2 (kali)

Este adaptador permite a comunicação entre as máquinas virtuais, neste caso, *kali, kioptrix, Windows XP e Ubuntu*. Ou seja, cria uma rede virtual isolada da rede externa, ou seja, as máquinas virtuais não têm acesso à internet nem a outras redes físicas.

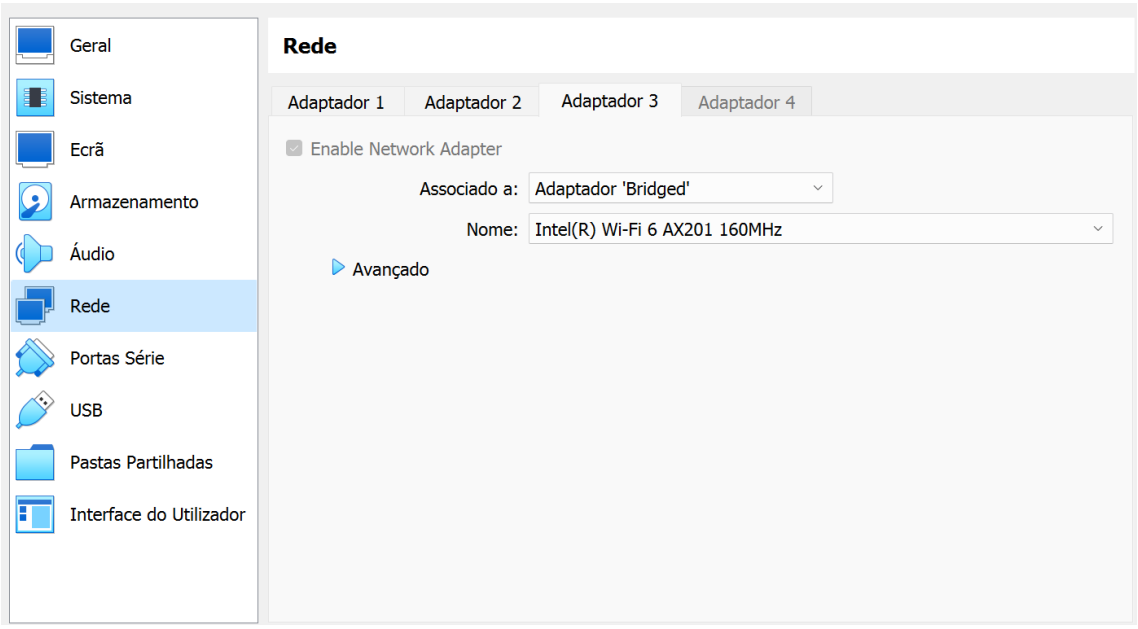


Figura 3- adaptador3 (kali)

Este modo permite que uma máquina virtual se conecte diretamente à rede física do *host*.

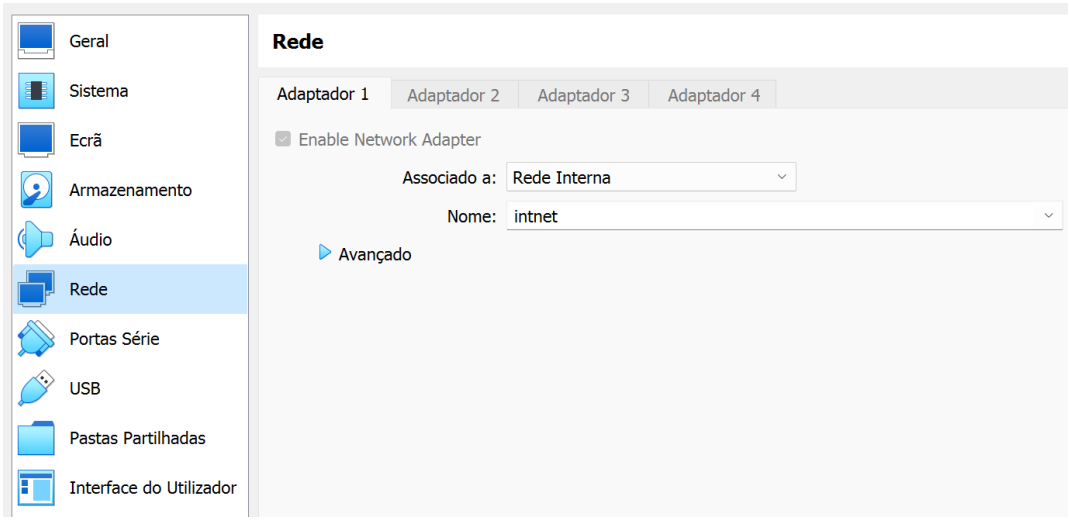


Figura 4- Adaptador (ubuntu)

Figura 5- Adaptador (kioptrix)

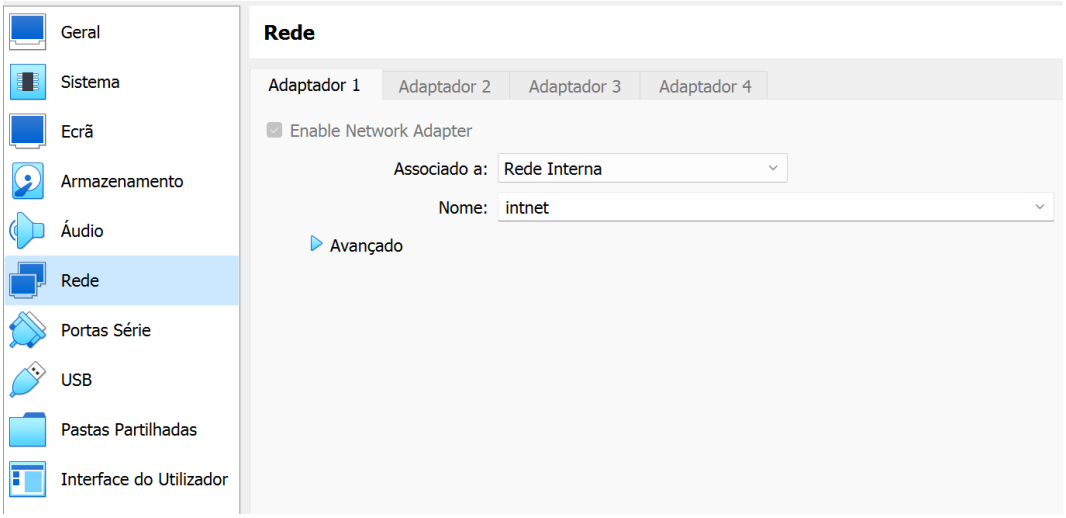


Figura 6- Adaptador (Windows XP)

2. Configuração das interfaces

Nesta fase, procedeu-se à configuração dos IP'S no *gateway*, *Kali*, das interfaces eth0 e eth1.

Para tal, utilizou-se o comando abaixo indicado.

```
sudo ifconfig <interface> <ip> netmask <netmask> up
```

Este comando permite a configuração de uma determinada interface com um determinado IP e uma determinada máscara de rede.

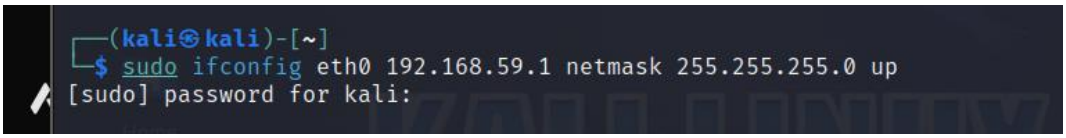


Figura 7- Interface eth0

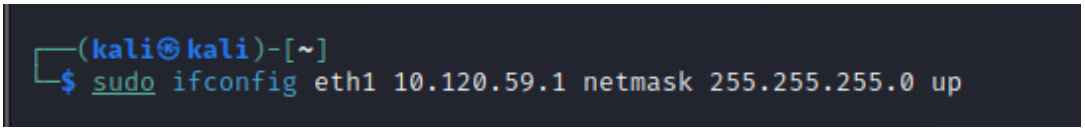


Figura 8- Interface eth1

3. Configuração DHCP

O DHCP possibilita que as máquinas virtuais, ligadas através da Rede interna, obtenham automaticamente endereços IP únicos, máscaras de rede e configuração do gateway.

Começou-se por instalar o servidor DHCP através do seguinte comando:

```
sudo apt install isc-dhcp-server
```

Após a instalação, atualizou-se a lista de pacotes disponíveis.

```
sudo apt update
```

De seguida, abriu-se o arquivo de configuração DHCP, através do editor *nano,* para posteriormente ser possível configurá-lo.

```
sudo nano /etc/default/isc-dhcp-server
```

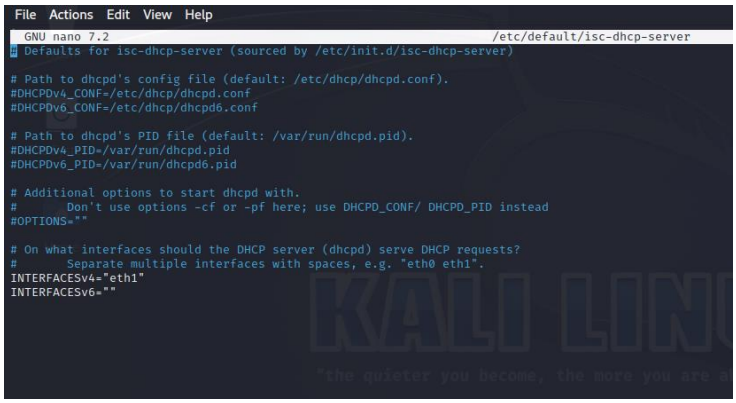


Figura 9- Interface

Na imagem anterior, adicionou-se a interface que seria utilizada para configurar o servidor DHCP.

Com as interfaces já configuradas procede-se agora à configuração do serviço DHCP, começando por entrar no ficheiro de configuração, para tal usamos o seguinte comando.

```
sudo nano /etc/dhcp/dhcpd.conf
```

Figura 10- Ficheiro configuração DHCP

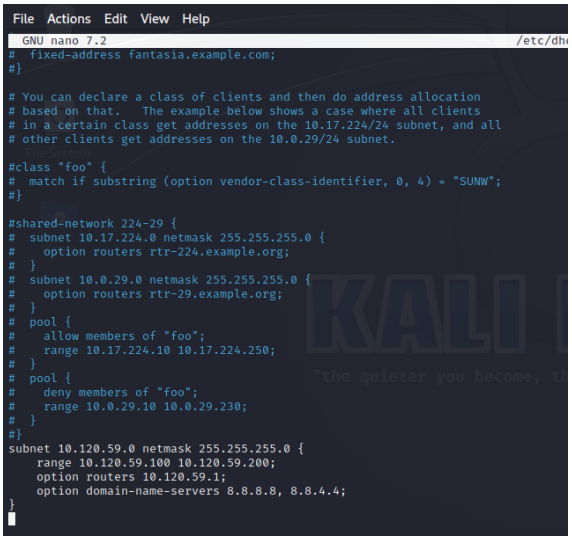


Figura 11- Configuração DHCP

Neste trecho, *subnet <ip> netmask <netmask>* define uma determinada *subrede* com um determinado ip e máscara de rede, respetivamente; "*range*" específica o intervalo de endereços IP que o servidor DHCP irá atribuir.; "*option routers*" define o *gateway* para a subrede em questão; "*option domain-name-servers*" especifica os servidores *DNS* que os dispositivos da subrede podem utilizar.

De seguida reiniciamos o servidor DHCP já configurado.

```
sudo service isc-dhcp-server restart
```

4. Encaminhamento do IP

Neste passo, procedeu-se à ativação do encaminhamento do IP através do *gateway, Kali,* começando por descomentar a linha abaixo indicada do ficheiro */etc/sysctl.conf*.

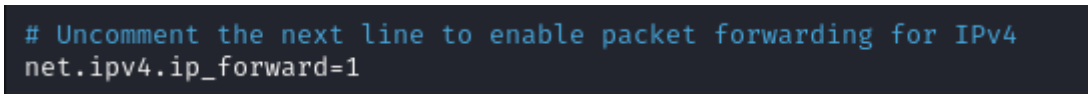


Figura 12- Descomentação

De seguida, aplicou-se o comando a seguir, de forma a guardar as alterações estabelecidas.

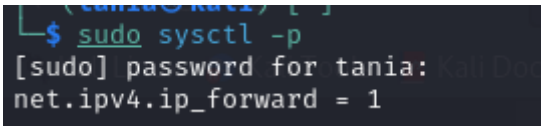


Figura 13- Comando sysctl -p

5. Configuração das rotas

Este passo é importante para permitir o encaminhamento do tráfego de rede, bem como, da conectividade com a Internet das restantes máquinas.

Primeiramente, configurou-se a rota através do gateway:

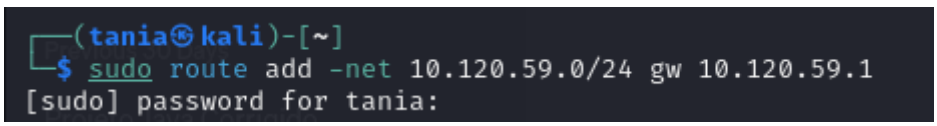


Figura 14- Configuração de rota

De seguida, configurou-se a rota no host, que no caso é Windows.

```
C:\Windows\System32>route add 192.168.59.0/24 192.168.59.1
```

Figura 15- Configuração de rota

6. Configuração da firewall

Neste caso, usamos a ferramenta *iptables*, para filtragem de pacotes, redireccionamento de tráfego e tradução de endereços de rede.

Inicialmente, efetuou-se este comando:

```
┌──(tania㉿kali)-[~]
└─$ sudo iptables -A FORWARD -i eth1 -o eth2 -j ACCEPT
```

Figura 16- Comando Iptables

Este comando permite o encaminhamento dos pacotes da interface th1 para a interface eth2.

```
┌──(tania㉿kali)-[~]
└─$ sudo iptables -t nat -A POSTROUTING -o eth2 -j MASQUERADE
```

Figura 17-Comando iptables

No entanto. Este serve para configurar o NAT, ou seja, permite que várias máquinas na mesma rede interna possam aceder à internet.

7. Testagem da conectividade entre máquinas

Este passo é extremamente importante para ser possível compreender se as máquinas estão devidamente conectadas à internet e entre si.

Abaixo são indicadas as testagens entre o host e o gateway.

```
C:\Windows\System32>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=64
Reply from 192.168.59.1: bytes=32 time<1ms TTL=64
Reply from 192.168.59.1: bytes=32 time<1ms TTL=64
Reply from 192.168.59.1: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.59.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Figura 18- Testagem com o kali

```
┌──(tania㉿kali)-[~]
└─$ ping 192.168.59.2
PING 192.168.59.2 (192.168.59.2) 56(84) bytes of data.
64 bytes from 192.168.59.2: icmp_seq=1 ttl=128 time=0.603 ms
64 bytes from 192.168.59.2: icmp_seq=2 ttl=128 time=0.649 ms
64 bytes from 192.168.59.2: icmp_seq=3 ttl=128 time=0.448 ms
64 bytes from 192.168.59.2: icmp_seq=4 ttl=128 time=0.416 ms
64 bytes from 192.168.59.2: icmp_seq=5 ttl=128 time=0.453 ms
64 bytes from 192.168.59.2: icmp_seq=6 ttl=128 time=0.613 ms
64 bytes from 192.168.59.2: icmp_seq=7 ttl=128 time=0.554 ms
64 bytes from 192.168.59.2: icmp_seq=8 ttl=128 time=0.593 ms
64 bytes from 192.168.59.2: icmp_seq=9 ttl=128 time=0.502 ms
^S64 bytes from 192.168.59.2: icmp_seq=10 ttl=128 time=0.411 ms
64 bytes from 192.168.59.2: icmp_seq=11 ttl=128 time=0.476 ms
^C
── 192.168.59.2 ping statistics ──
11 packets transmitted, 11 received, 0% packet loss, time 10198ms
rtt min/avg/max/mdev = 0.411/0.519/0.649/0.081 ms
```

Figura 19- Testagem com o host

```
C:\Documents and Settings\Forense>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=3ms TTL=64
Reply from 10.120.59.1: bytes=32 time=4ms TTL=64
Reply from 10.120.59.1: bytes=32 time=3ms TTL=64

Ping statistics for 10.120.59.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 4ms, Average = 2ms
```

Figura 20- Windows XP ao Kali

```
^Ctania@tania:~$ ping 10.120.59.1
PING 10.120.59.1 (10.120.59.1) 56(84) bytes of data.
64 bytes from 10.120.59.1: icmp_seq=1 ttl=64 time=0.965 ms
64 bytes from 10.120.59.1: icmp_seq=2 ttl=64 time=1.59 ms
64 bytes from 10.120.59.1: icmp_seq=3 ttl=64 time=2.85 ms
64 bytes from 10.120.59.1: icmp_seq=4 ttl=64 time=1.87 ms
64 bytes from 10.120.59.1: icmp_seq=5 ttl=64 time=3.11 ms
64 bytes from 10.120.59.1: icmp_seq=6 ttl=64 time=1.82 ms
^C
--- 10.120.59.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 0.965/2.033/3.110/0.734 ms
```

Figura 21- Ubuntu ao Kali

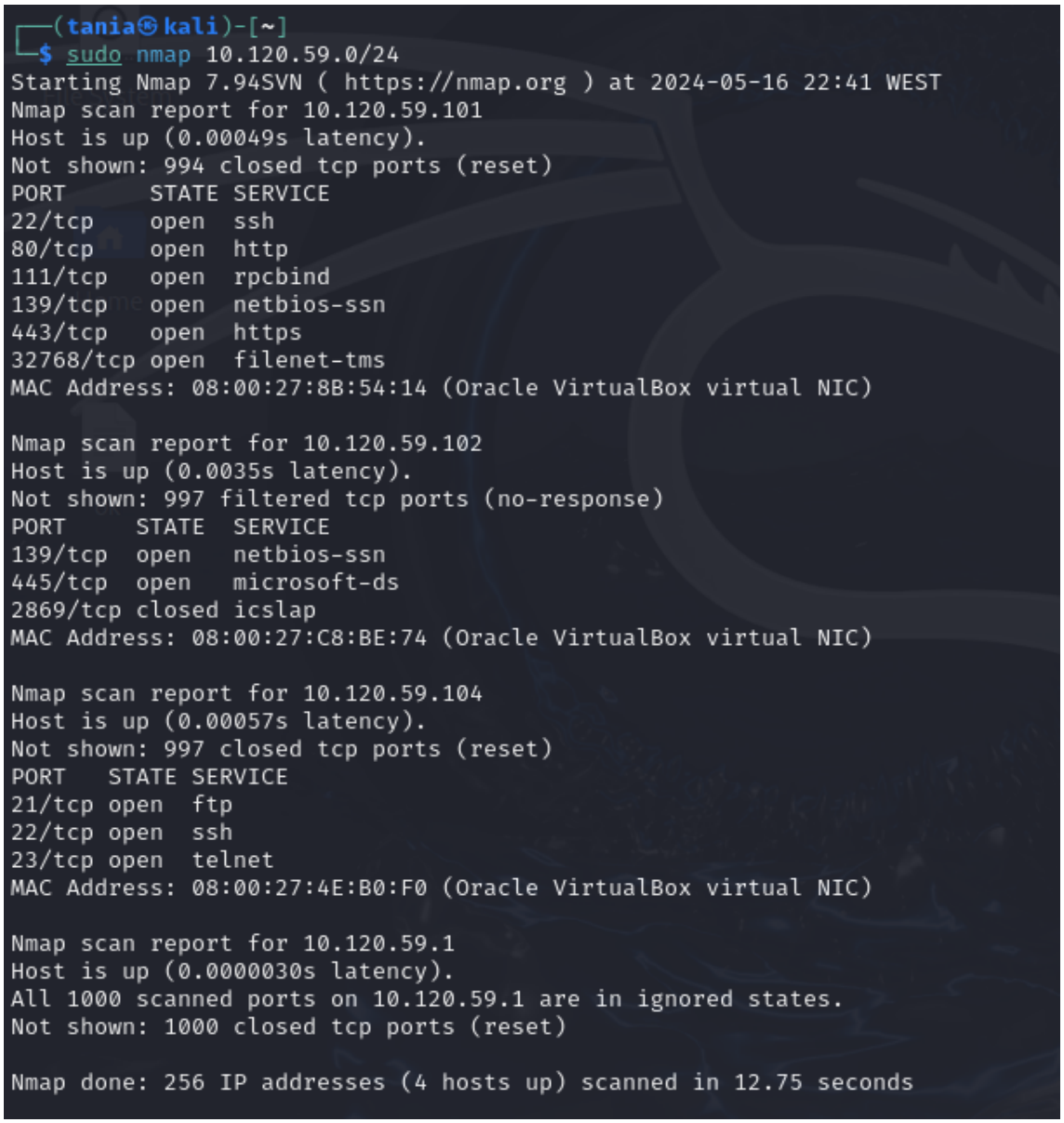


Figura 22- Mapeamento da rede

```
┌──(tania㉿kali)-[~]
└─$ ping 10.120.59.101
PING 10.120.59.101 (10.120.59.101) 56(84) bytes of data.
64 bytes from 10.120.59.101: icmp_seq=1 ttl=255 time=0.948 ms
64 bytes from 10.120.59.101: icmp_seq=2 ttl=255 time=1.56 ms
64 bytes from 10.120.59.101: icmp_seq=3 ttl=255 time=2.15 ms
^C
── 10.120.59.101 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.948/1.550/2.146/0.489 ms

┌──(tania㉿kali)-[~]
└─$ ping 10.120.59.102
PING 10.120.59.102 (10.120.59.102) 56(84) bytes of data.
64 bytes from 10.120.59.102: icmp_seq=1 ttl=128 time=1.83 ms
64 bytes from 10.120.59.102: icmp_seq=2 ttl=128 time=4.70 ms
64 bytes from 10.120.59.102: icmp_seq=3 ttl=128 time=3.81 ms
64 bytes from 10.120.59.102: icmp_seq=4 ttl=128 time=4.34 ms
^C
── 10.120.59.102 ping statistics ──
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 1.834/3.670/4.695/1.106 ms

┌──(tania㉿kali)-[~]
└─$ ping 10.120.59.104
PING 10.120.59.104 (10.120.59.104) 56(84) bytes of data.
64 bytes from 10.120.59.104: icmp_seq=1 ttl=64 time=0.879 ms
64 bytes from 10.120.59.104: icmp_seq=2 ttl=64 time=1.75 ms
64 bytes from 10.120.59.104: icmp_seq=3 ttl=64 time=2.25 ms
64 bytes from 10.120.59.104: icmp_seq=4 ttl=64 time=1.74 ms
64 bytes from 10.120.59.104: icmp_seq=5 ttl=64 time=1.63 ms
^C
── 10.120.59.104 ping statistics ──
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
rtt min/avg/max/mdev = 0.879/1.648/2.248/0.440 ms
```

Figura 23- Ping do kali às restantes máquinas

Por fim, testamos a conectividade das máquinas à internet.

```
┌──(tania㉿kali)-[~]
└─$ ping google.com
PING google.com (142.250.184.14) 56(84) bytes of data.
64 bytes from mad41s10-in-f14.1e100.net (142.250.184.14): icmp_seq=1 ttl=116 time=39.9 ms
64 bytes from mad41s10-in-f14.1e100.net (142.250.184.14): icmp_seq=2 ttl=116 time=40.6 ms
64 bytes from mad41s10-in-f14.1e100.net (142.250.184.14): icmp_seq=3 ttl=116 time=29.6 ms
64 bytes from mad41s10-in-f14.1e100.net (142.250.184.14): icmp_seq=4 ttl=116 time=31.8 ms
^C
── google.com ping statistics ──
5 packets transmitted, 4 received, 20% packet loss, time 4005ms
rtt min/avg/max/mdev = 29.590/35.471/40.595/4.842 ms
```

Figura 24- Acesso à internet do Kali

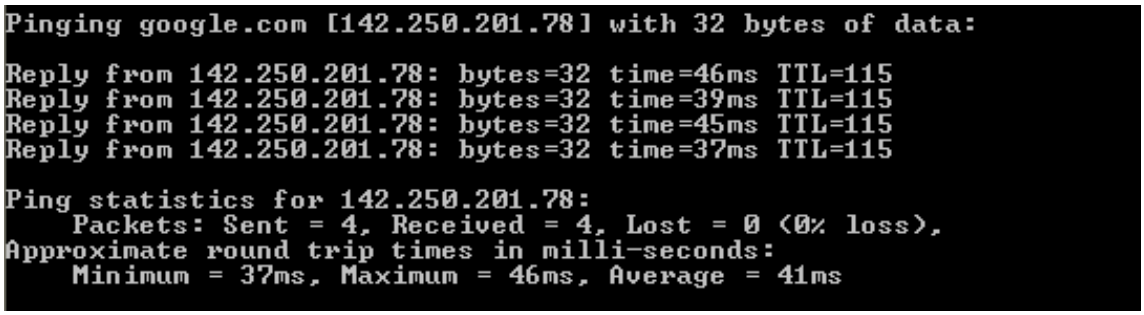


Figura 25- Conexão à internet (Windows XP)

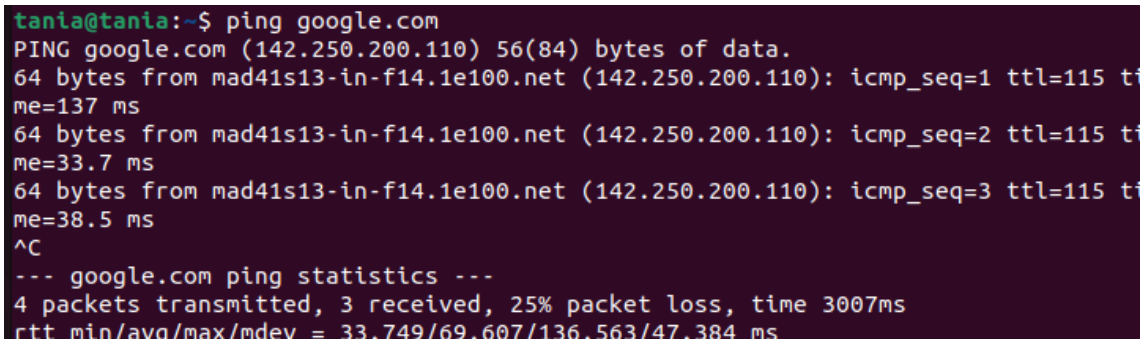


Figura 26- Conexão à internet (Ubuntu)

Análise de tráfego

Nesta próxima etapa, analisa-se o tráfego de todas as máquinas através do *gateway, Kali*. Neste caso, são seis tópicos que iremos abordar, a conectividade entre as máquinas, serviço *ftp,* serviço *ssh* e *telnet*, fluxo de dados de uma ligação *telnet/ssh,* as diferenças dos dois serviços e diferença entre um acesso HTTP e HTTPS.

1. Conectividade entre as máquinas

Neste tópico comprava-se a conectividade entre as máquinas usando, neste caso, o *wireshark*. O *wireshark* é um analisador de protocolos utilizado para monitorar o tráfego da rede.

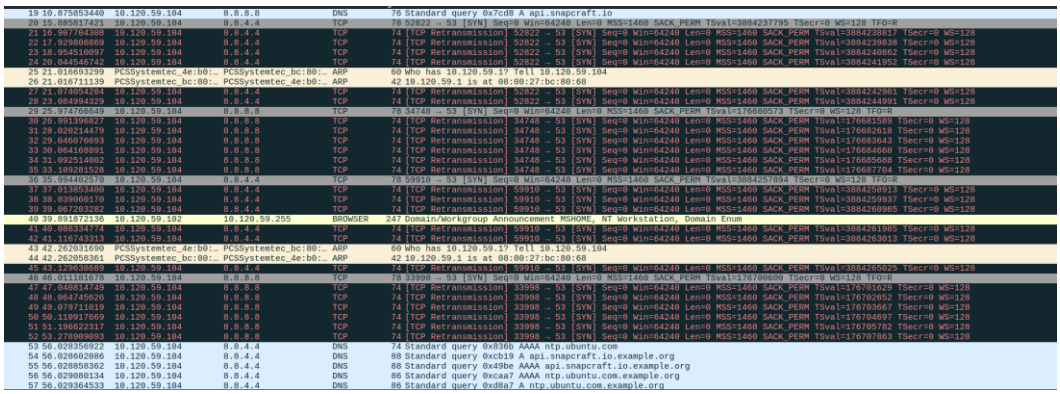
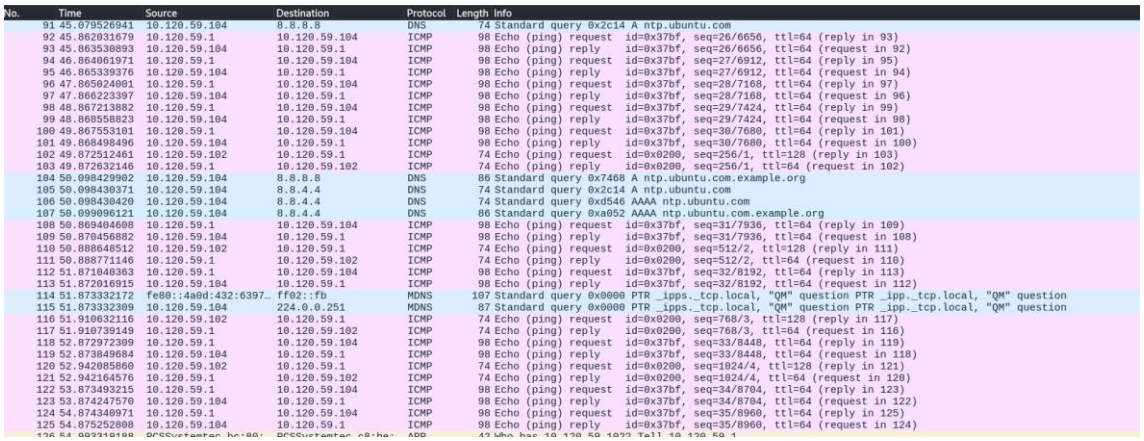


Figura 27- Análise 1



No.	Time	Source	Destination	Protocol	Length	Info
91	45.079526941	10.120.59.104	8.8.8.8	DNS	74	Standard query 0x2c14 A ntp.ubuntu.com
92	45.862031679	10.120.59.1	10.120.59.104	ICMP	98	Echo (ping) request id=0x37bf, seq=26/6656, ttl=64 (reply in 93)
93	45.863530893	10.120.59.104	10.120.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=26/6656, ttl=64 (request in 92)
94	46.864061971	10.120.59.1	10.120.59.104	ICMP	98	Echo (ping) request id=0x37bf, seq=27/6912, ttl=64 (reply in 95)
95	46.865339376	10.120.59.104	10.120.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=27/6912, ttl=64 (request in 94)
96	47.865024001	10.120.59.1	10.120.59.104	ICMP	98	Echo (ping) request id=0x37bf, seq=28/7168, ttl=64 (reply in 97)
97	47.866223397	10.120.59.104	10.120.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=28/7168, ttl=64 (request in 96)
98	48.867213882	10.120.59.1	10.120.59.104	ICMP	98	Echo (ping) request id=0x37bf, seq=29/7424, ttl=64 (reply in 99)
99	48.868558823	10.120.59.104	10.120.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=29/7424, ttl=64 (request in 98)
100	49.867553101	10.120.59.1	10.120.59.104	ICMP	98	Echo (ping) request id=0x37bf, seq=30/7680, ttl=64 (reply in 101)
101	49.868498496	10.120.59.104	10.120.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=30/7680, ttl=64 (request in 100)
102	49.872512461	10.120.59.102	10.120.59.1	ICMP	74	Echo (ping) request id=0x0200, seq=256/1, ttl=128 (reply in 103)
103	49.872632146	10.120.59.1	10.120.59.102	ICMP	74	Echo (ping) reply id=0x0200, seq=256/1, ttl=64 (request in 102)
104	50.098429902	10.120.59.104	8.8.8.8	DNS	86	Standard query 0x7468 A ntp.ubuntu.com.example.org
105	50.098430371	10.120.59.104	8.8.4.4	DNS	74	Standard query 0x2c14 A ntp.ubuntu.com
106	50.098430420	10.120.59.104	8.8.4.4	DNS	74	Standard query 0xd546 AAAA ntp.ubuntu.com
107	50.099096121	10.120.59.104	8.8.4.4	DNS	86	Standard query 0xa052 AAAA ntp.ubuntu.com.example.org
108	50.869404608	10.120.59.1	10.120.59.104	ICMP	98	Echo (ping) request id=0x37bf, seq=31/7936, ttl=64 (reply in 109)
109	50.870456882	10.120.59.104	10.120.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=31/7936, ttl=64 (request in 108)
110	50.888648512	10.120.59.102	10.120.59.1	ICMP	74	Echo (ping) request id=0x0200, seq=512/2, ttl=128 (reply in 111)
111	50.888771146	10.120.59.1	10.120.59.102	ICMP	74	Echo (ping) reply id=0x0200, seq=512/2, ttl=64 (request in 110)
112	51.871040363	10.120.59.1	10.120.59.104	ICMP	98	Echo (ping) request id=0x37bf, seq=32/8192, ttl=64 (reply in 113)
113	51.872016915	10.120.59.104	10.120.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=32/8192, ttl=64 (request in 112)
114	51.873332172	fe80::4a0d:432:6397..	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR _ipp._tcp.local, "QM" question
115	51.873332309	10.120.59.104	224.0.0.251	MDNS	87	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR _ipp._tcp.local, "QM" question
116	51.910632116	10.120.59.102	10.120.59.1	ICMP	74	Echo (ping) request id=0x0200, seq=768/3, ttl=128 (reply in 117)
117	51.910739149	10.120.59.1	10.120.59.102	ICMP	74	Echo (ping) reply id=0x0200, seq=768/3, ttl=64 (request in 116)
118	52.872972309	10.120.59.1	10.120.59.104	ICMP	98	Echo (ping) request id=0x37bf, seq=33/8448, ttl=64 (reply in 119)
119	52.873849684	10.120.59.104	10.120.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=33/8448, ttl=64 (request in 118)
120	52.942085860	10.120.59.102	10.120.59.1	ICMP	74	Echo (ping) request id=0x0200, seq=1024/4, ttl=128 (reply in 121)
121	52.942164576	10.120.59.1	10.120.59.102	ICMP	74	Echo (ping) reply id=0x0200, seq=1024/4, ttl=64 (request in 120)
122	53.873493215	10.120.59.1	10.120.59.104	ICMP	98	Echo (ping) request id=0x37bf, seq=34/8704, ttl=64 (reply in 123)
123	53.874247570	10.120.59.104	10.120.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=34/8704, ttl=64 (request in 122)
124	54.874340971	10.120.59.1	10.120.59.104	ICMP	98	Echo (ping) request id=0x37bf, seq=35/8960, ttl=64 (reply in 125)
125	54.875252808	10.120.59.104	10.120.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=35/8960, ttl=64 (request in 124)
126	54.993319188	PCSSystemtec_bc:80:	PCSSystemtec_c8:be:	ARP	42	Who has 10.120.59.102? Tell 10.120.59.1

Figura 28- Análise 2

Nas figuras 27 e 28, é possível perceber que há conectividade entre as máquinas dado que, na coluna *source* e na coluna *destination*, existe comunicação entre duas máquinas diferentes, inseridas na mesma rede.

2. Serviço *ftp*

O serviço *ftp* é usado para transferir arquivos entre computadores em redes locais ou na internet. Neste caso, o uso do serviço ftp irá ajudar a detetar o *user* e a *password* da máquina 3, ou seja, no *Ubuntu*.

Começou-se por atualizar os serviços, usando o comando abaixo.

```
sudo apt update
```

De seguida, procedeu-se à instalação do serviço *ftp*.

```
sudo apt install vsftpd
```

No host, instalou-se a aplicação *FileZilla*, ou seja, é um *software* capaz de conectar máquinas e transferir arquivos usando o serviço *ftp*.

Nesta aplicação, no servidor colocamos o IP da máquina que desejamos, o nome de utilizador dessa máquina e a respetiva palavra-passe.

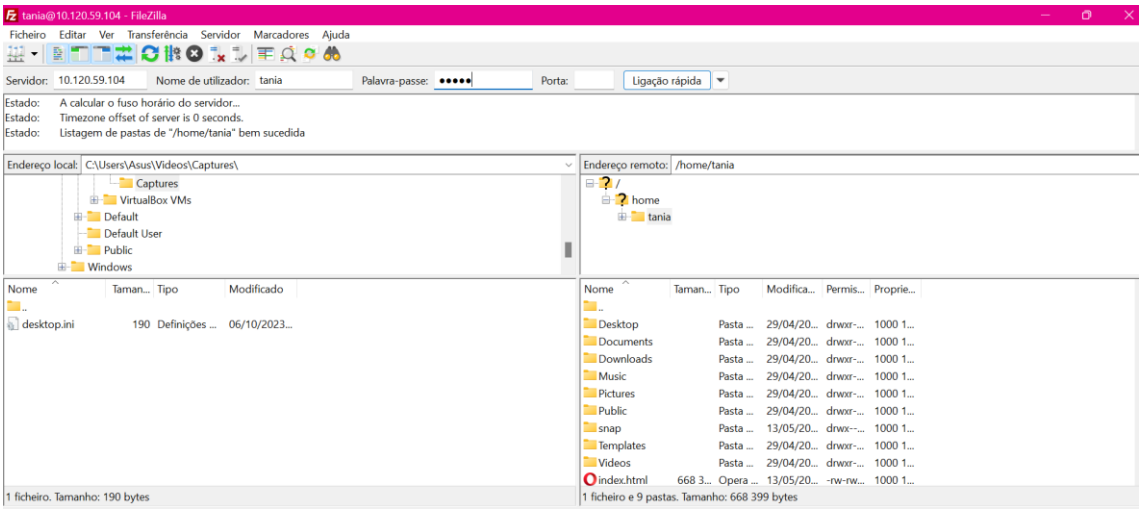


Figura 29- Serviço ftp (host)

Na máquina que se deseja descobrir a *password*, inicia-se o serviço *ftp*.

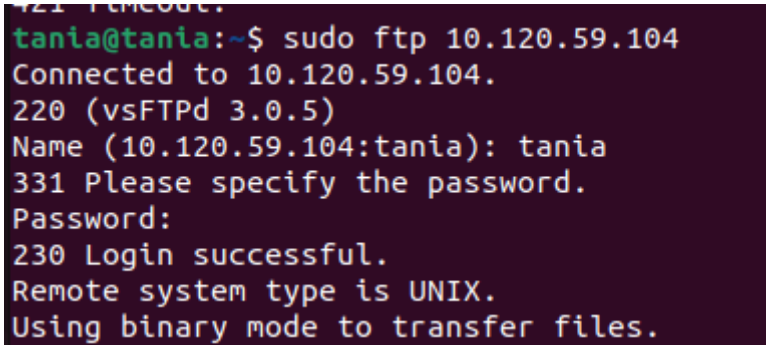


Figura 30- serviço ftp (Ubuntu)

No *gateway, Kali,* com a ajuda do *wireshark*, vamos analisar o tráfego e, assim, descobrir a palavra-passe e o utilizador dado que, o serviço *ftp*, não é seguro.

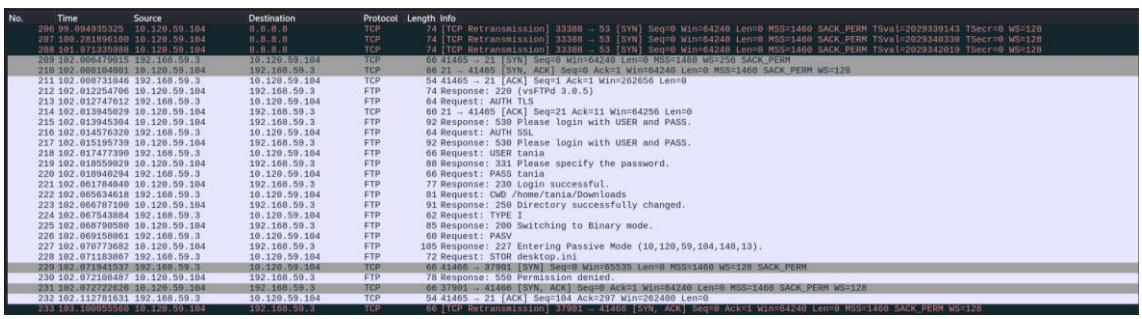


Figura 31- Wireshark

É possível perceber-se que a palavra-passe é tania e o utilizador é tania.

3. Serviço *ssh* e *telnet*

Estes dois serviços permitem a conexão remota aos servidores físicos, em cloud ou híbridos. A principal diferença é que o *ssh* é encriptado enquanto, o telnet não.

Começou-se por instalar os serviços, executando:

```
sudo apt install openssh-server
```

Após instalados iniciam-se os serviços.

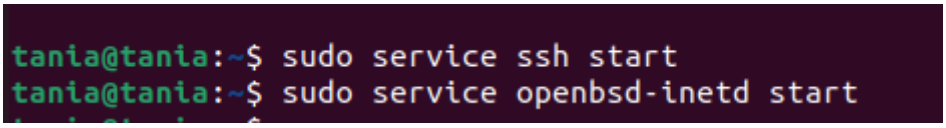


Figura 32- serviços

Através do *Kali,* executamos o *ssh* e o *telnet* para ser possível verificar a existência de ambos no *Ubuntu*.

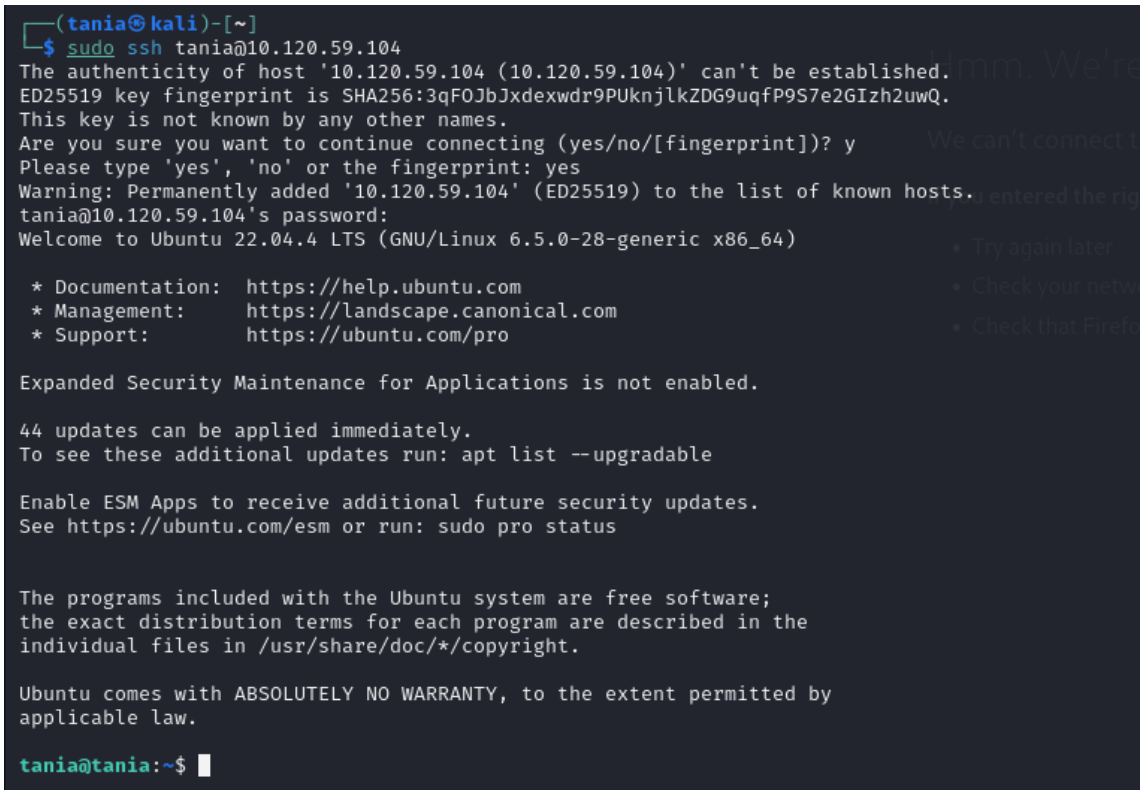


Figura 33- Serviço ssh

```
┌──(tania㉿kali)-[~]
└─$ sudo telnet 10.120.59.104
Trying 10.120.59.104 ...
Connected to 10.120.59.104.
Escape character is '^]'.
Ubuntu 22.04.4 LTS
tania login: tania
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

Expanded Security Maintenance for Applications is not enabled.

44 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue May 14 11:36:02 WEST 2024 from 10.120.59.1 on pts/1
tania@tania:~$
```

Figura 34- Serviço telnet

4. Fluxo de dados de uma ligação *telnet/ssh*

Para a análise do fluxo de dados, recorreu-se ao *wireshark*, para permitir uma análise mais aprofundado por protocolo.

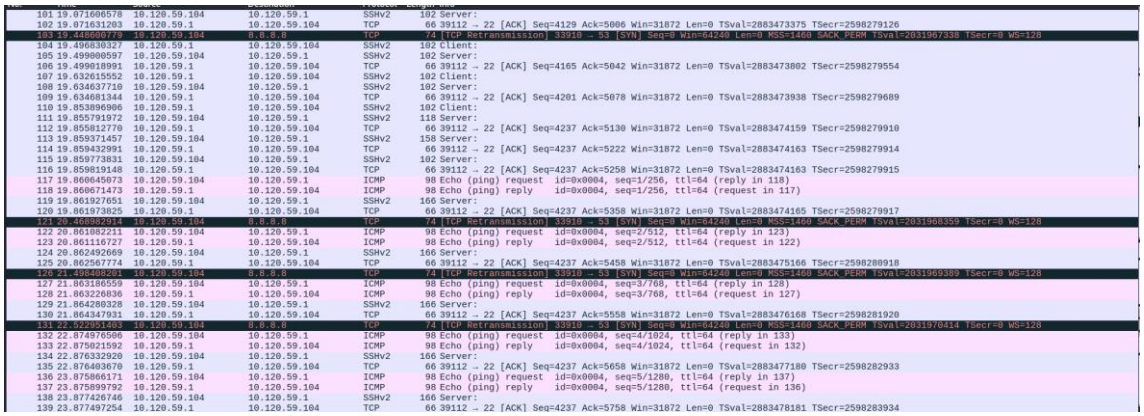


Figura 35- Fluxo de dados ssh

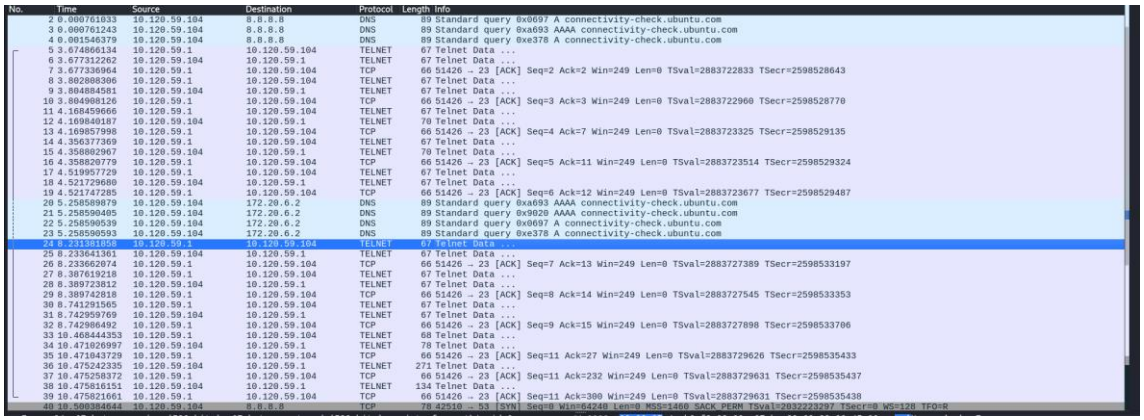


Figura 36- Fluxo de dados telnet

5. Diferenças entre *telnet* e *ssh*

As principais diferenças são as seguintes:

- *Telnet* é um serviço para terminal virtual, enquanto o *ssh* é um serviço que permite executar comandos em uma máquina remota.
- *Telnet* é vulnerável a ataques de segurança;
- O *ssh* usa a porta 22 e o *telnet* a porta 23;
- O *ssh* transmite os dados em formato criptográfico.
- O *ssh* é adequado para redes públicas, já o *telnet* para redes privada.

6. Diferença entre um acesso *HTTP / HTPPS*

Inicialmente, através do *Ubuntu*, acedeu-se a dois sites com acessos diferentes, um *HTTP* e outro *HTTPS*.

Enquanto se acede aos diferentes serviços, o *Kali,* analisava o tráfego através do *wireshark* para ser possível determinar as diferenças entre eles.

Abaixo, são indicadas as capturas para um acesso *HTTP* e para um acesso *HTTPS*.

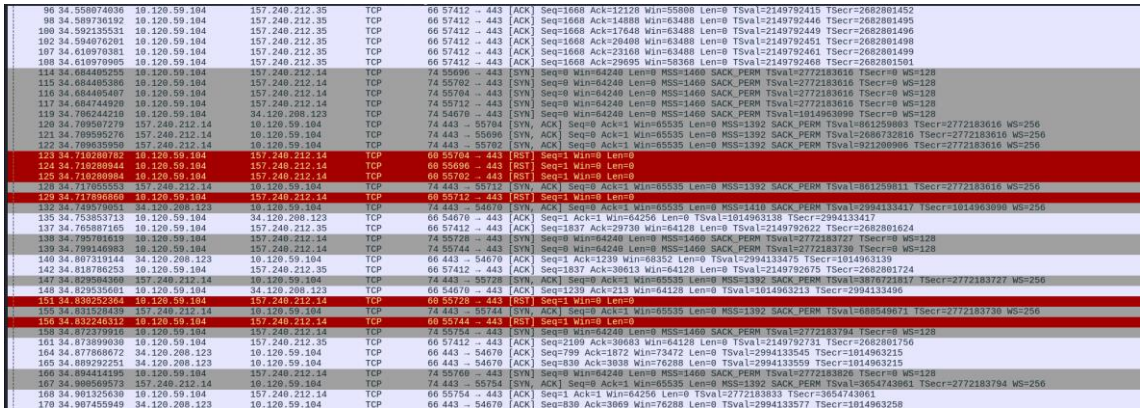


Figura 37- Captura HTTPS

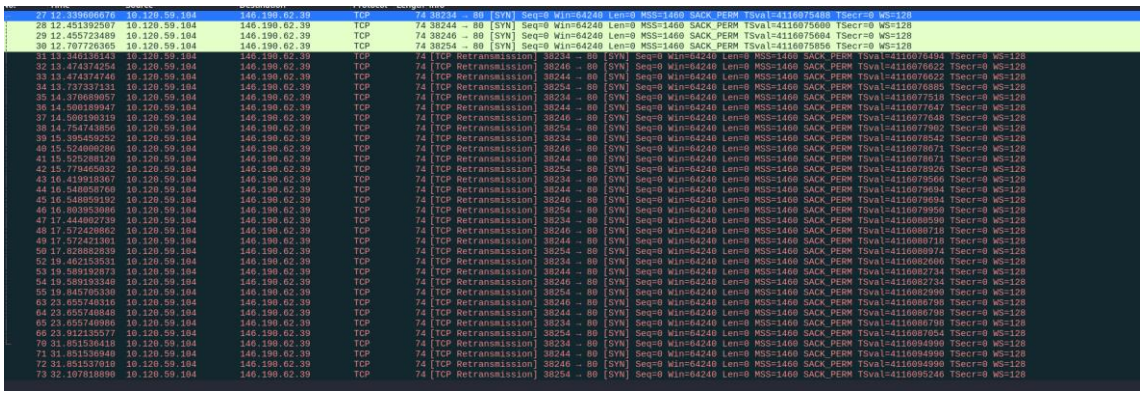


Figura 38- Captura HTTPS

As principais diferenças entre estes dois protocolos são as seguintes:

- *HTTPS* fornece uma conexão criptográfica segura;
- *HTTPS* evita a divulgação de senhas, números de cartões de crédito...;
- Sites *HTTPS* tende a ser mais rápidos do que sites *HTTP*;

No geral as mudanças mais significativas são em termos de segurança, sendo o *HTTPS* mais seguro que o *HTTP*.

Hacking

Neste último tema, iremos abordar três diferentes ferramentas, o *nmap*, o *OpenVAS* e o *Metasploit*.

O *nmap* é um comando Linux que permite efetuar o scan de endereços IP e portas em uma rede e detetar aplicações instaladas.

O *OpenVAS* é um framework de vários serviços e ferramentas que oferece uma solução de varredura e gerenciamento de vulnerabilidade.

O *Metasploit* um projeto de segurança de computadores que fornece informações sobre vulnerabilidades de segurança e ajuda em testes de penetração e desenvolvimento de assinaturas IDS.

1. NMAP

Começou-se por instalar o *nmap* usando o comando abaixo indicado:

sudo apt-get install nmap

De seguida, mapeou-se a rede para ser possível descobrir as portas abertas.

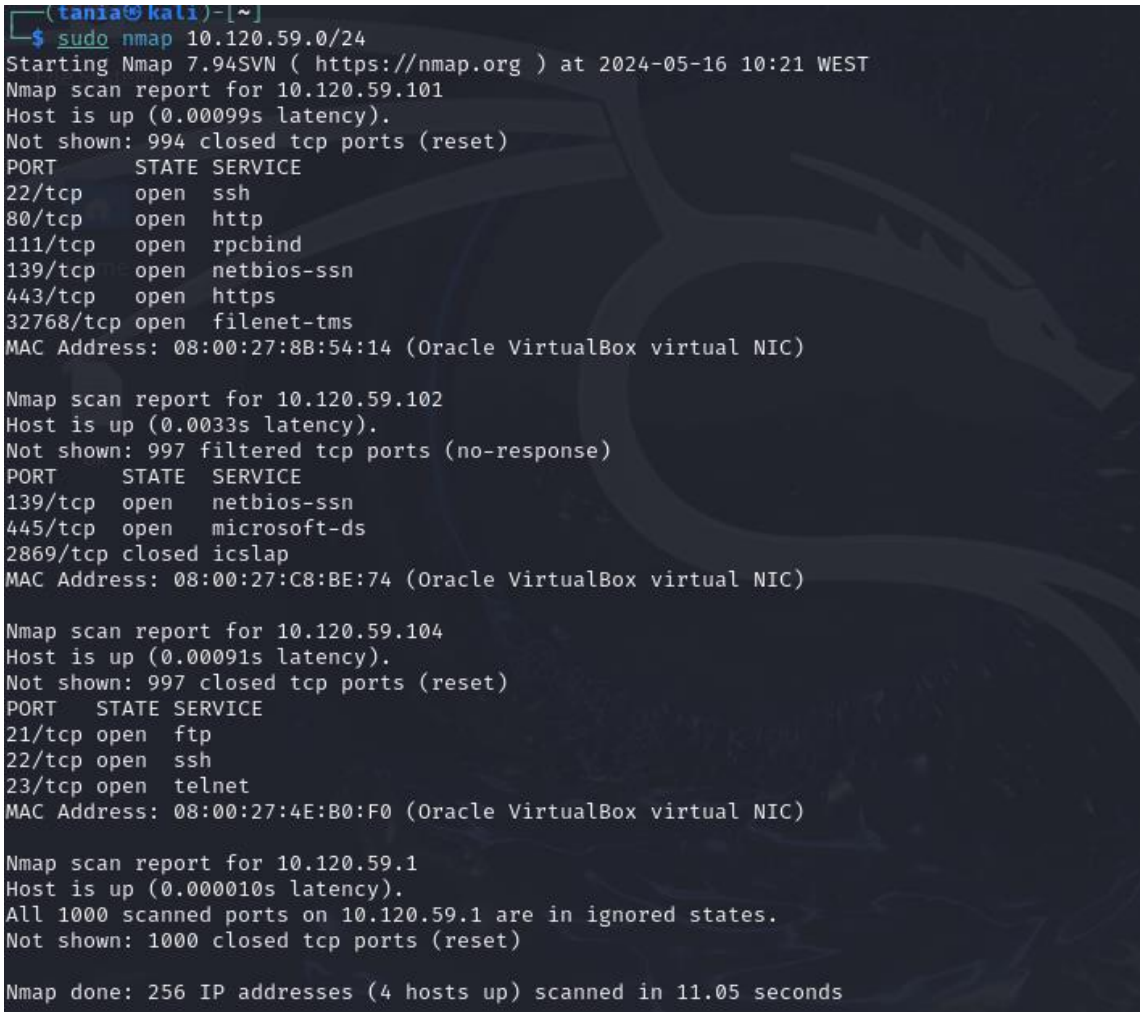


Figura 39- Demonstração das portas abertas

É possível analisar que o serviço *TCP* está a ser utilizado em todas as máquinas, *kali, ubuntu, kioptrix*.

Figura 40- Demonstração dos sistemas operativos

É possível perceber que o sistema operativo do *Windows XP* é o *Microsoft Windows Server 2003*, enquanto a máquina restante se trata de *Linux,* no *Kioptrix (Linux 2.4.9-2.4.18)* e no *Ubuntu (Linux 4.15-5.8)*.

2. OpenVAS

Primeiramente, instalou-se esta ferramenta usando o seguinte comando:

```
sudo apt install docker.io -y
```

Após executado, o próximo passo consistiu no início automático do serviço Docker durante a inicialização do sistema e, também, na verificação do estado, ou seja, se realmente se encontrava ativo.

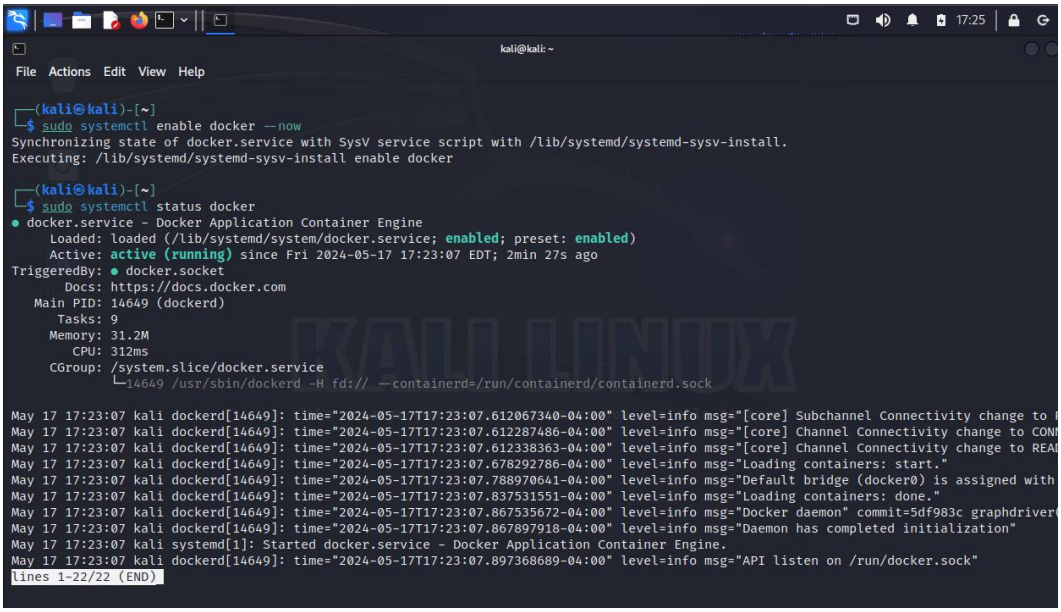


Figura 41- Inicio automático (Docker)

No passo seguinte, verificou-se a versão do Docker.

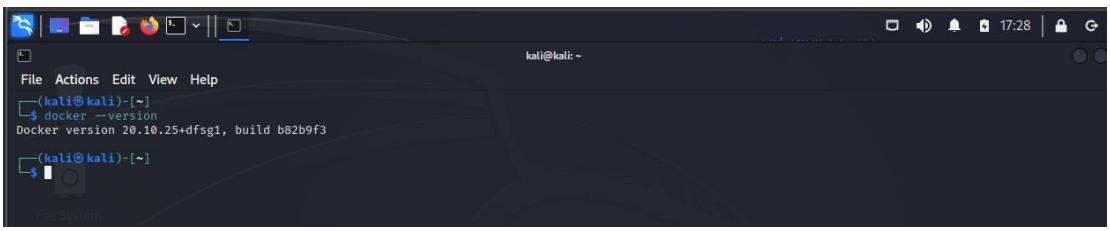


Figura 42- Verificação da versão

De seguida, efetuou-se o *download* do Docker Hub para o sistema local.

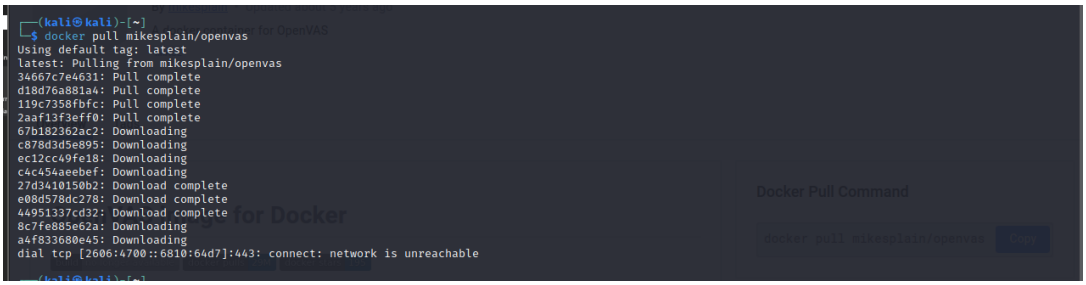


Figura 43- Download da imagem Docker Hub

Por fim, executou-se o comando abaixo que permite o serviço *openvas* seja executado em segundo plano enquanto a porta 443 seja mapeada para a porta 443 do host, ou seja, permite com que o *OpenVAS* seja acedido através de uma página web.

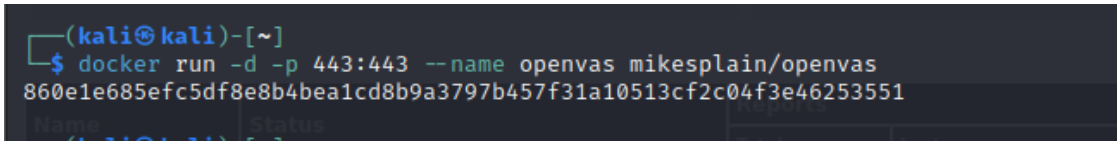


Figura 44- Execução OpenVAS

Por fim, abriu-se o *OpenVAS* através da página *web*, indicou-se o respetivo *username* e *password*, criou-se um target com a rede que se deseja encontrar as vulnerabilidades e, por fim, uma *task* para ser possível executar e verificar todas as vulnerabilidades existentes.

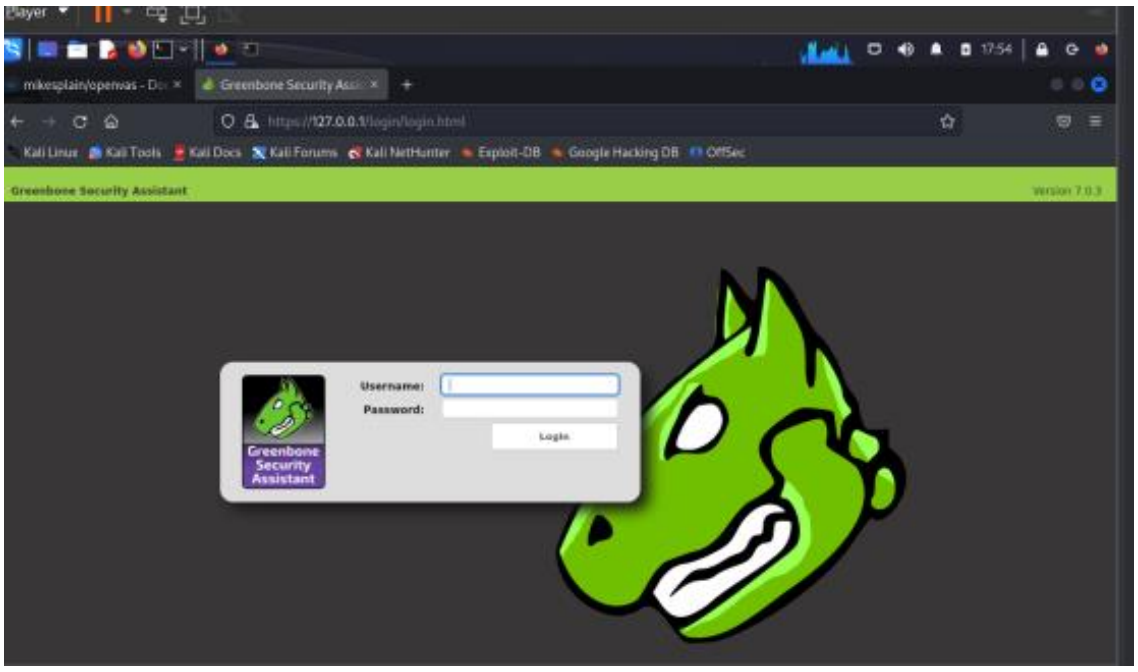


Figura 45- Autenticação

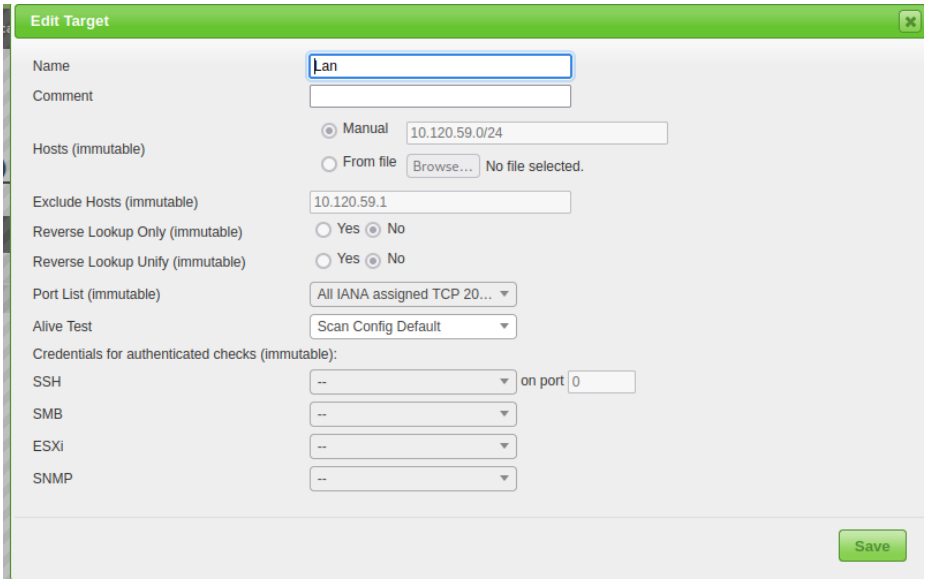


Figura 46- Criação de um target

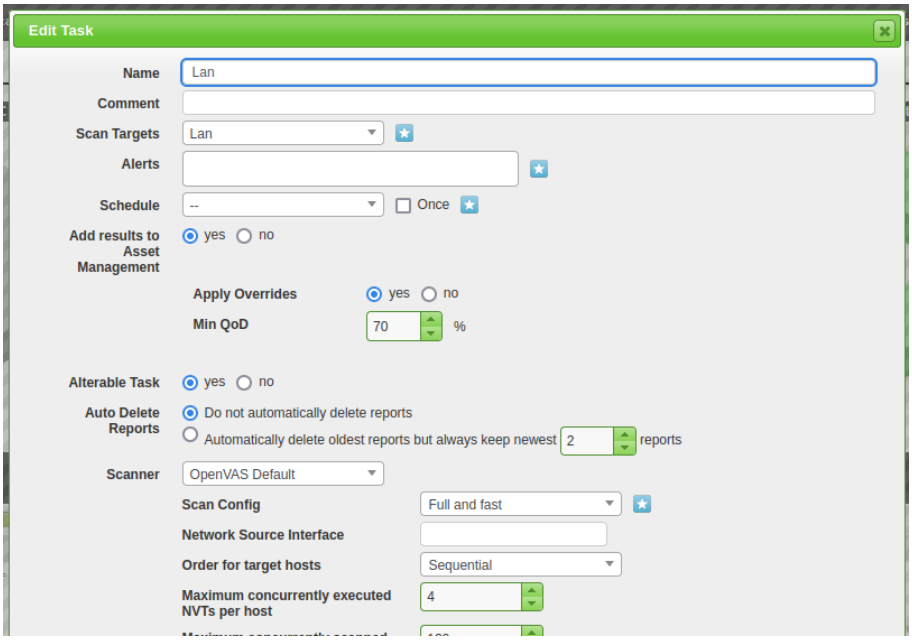


Figura 47- Criação de uma task

Em seguida, seleciona-se a tarefa (*task)* e clinica-se em *start*. Após este passo é mostrado um *dashboard* com a classificação do risco das vulnerabilidades encontradas, baixo, médio ou alto.

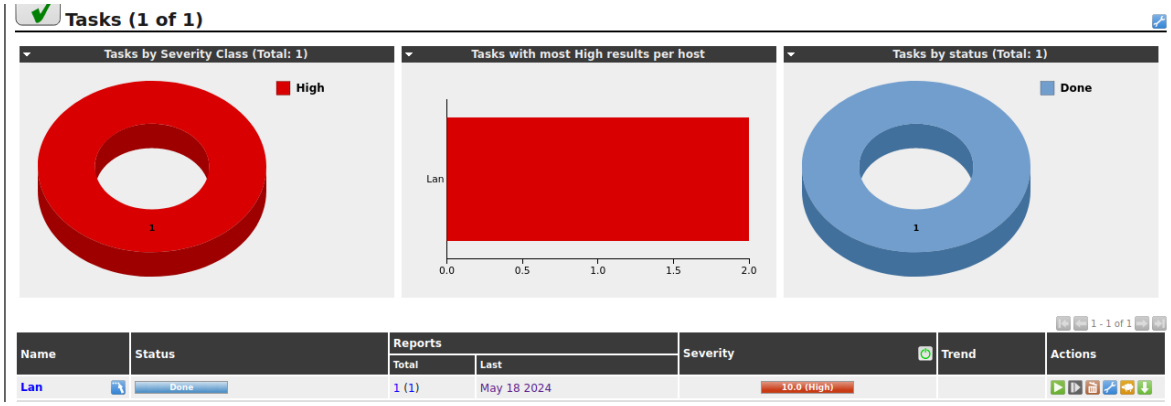


Figura 48- Dashboard

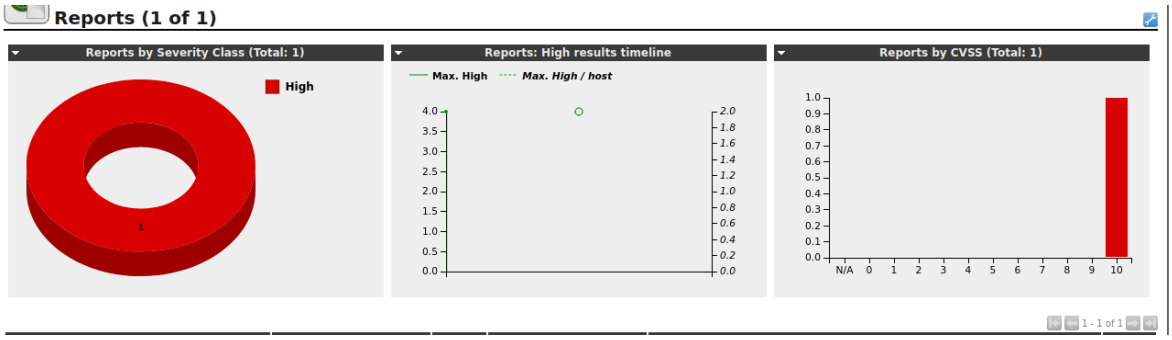


Figura 49- Dashboard 2

Na figura seguinte, são apresentadas mais informações relativas ao IP da máquina e o sistema operativo.

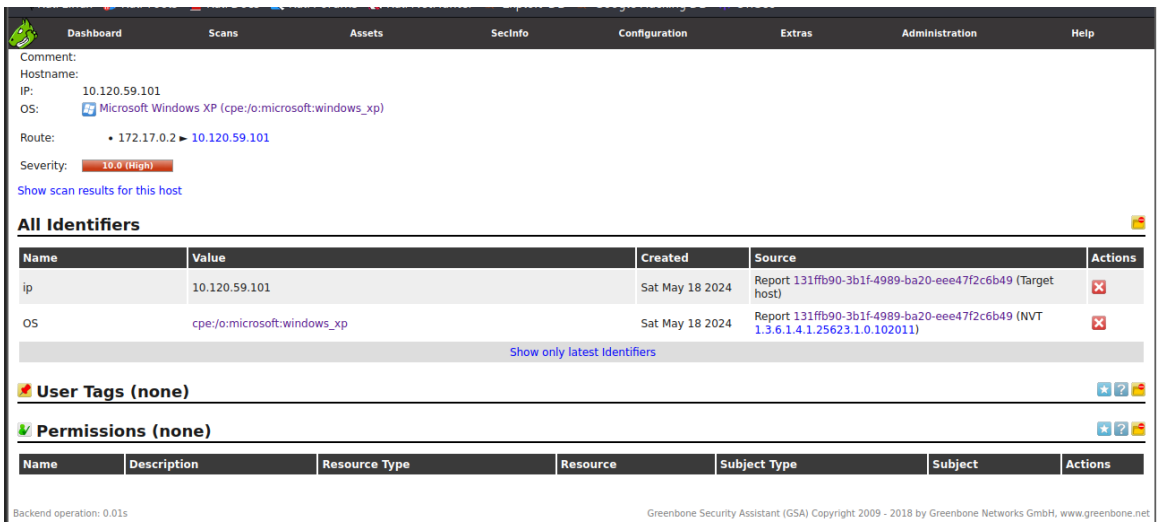


Figura 50- Informações adicionais

Abaixo são indicadas todas as vulnerabilidades encontradas.

Report: Results (4 of 21)

ID: 131ffb90-3b1f-4989-ba20-eee47f2c6b49
Modified: Sat May 18 09:47:03 2024
Created: Sat May 18 09:43:24 2024
Owner: admin

1 - 4 of 4

Vulnerability	Severity	QoD	Host	Location	Actions
OS End Of Life Detection	10.0 (High)	80%	10.120.59.101	general/tcp	
Microsoft Windows SMB Server NTLM Multiple Vulnerabilities (971468)	10.0 (High)	98%	10.120.59.101	445/tcp	
Vulnerabilities in SMB Could Allow Remote Code Execution (958687) - Remote	10.0 (High)	98%	10.120.59.101	445/tcp	
Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)	9.3 (High)	95%	10.120.59.101	445/tcp	

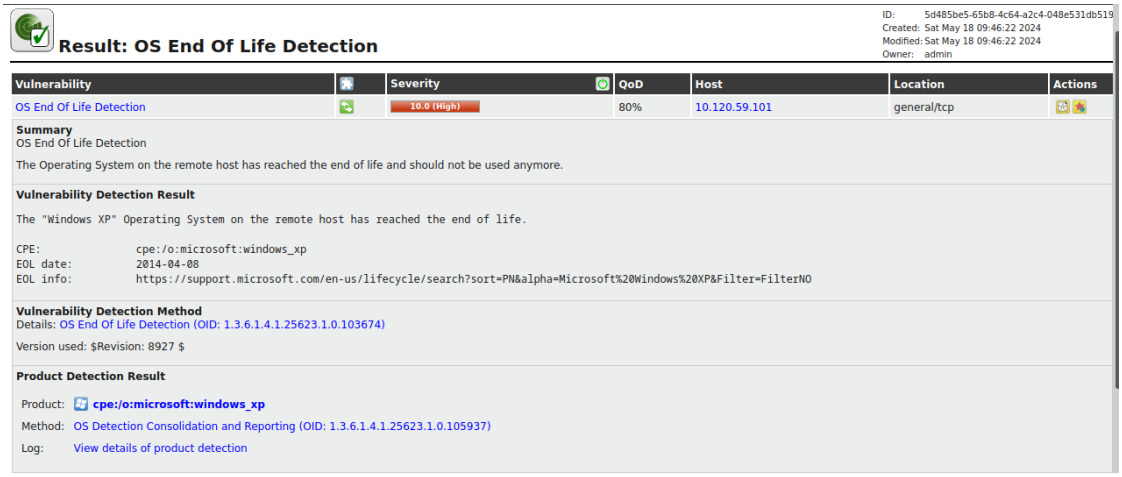
(Applied filter:autofp=0 apply_overrides=1 notes=1 overrides=1 result_hosts_only=1 first=1 rows=100 sort-reverse=severity levels=hml min_qod=70)

1 - 4 of 4

Figura 51- Reports

De realçar que não foram encontradas vulnerabilidades na máquina virtual, *Ubuntu*, pois possui a versão mais recente. No entanto, no *kioptrix* não foi possível encontrar devido a uma falha de conectividade entre as máquinas *kali* e *ubuntu*, no caso do Gonçalo, contudo, no caso da Tânia a ferramenta não concluía a instalação.

As seguintes imagens referem-se ao *report* de cada vulnerabilidade encontrada.



Result: OS End Of Life Detection

ID: 5d485be5-65b8-4c64-a2c4-048e531db519
Created: Sat May 18 09:46:22 2024
Modified: Sat May 18 09:46:22 2024
Owner: admin

Vulnerability	Severity	QoD	Host	Location	Actions
OS End Of Life Detection	10.0 (High)	80%	10.120.59.101	general/tcp	

Summary
OS End Of Life Detection

The Operating System on the remote host has reached the end of life and should not be used anymore.

Vulnerability Detection Result

The "Windows XP" Operating System on the remote host has reached the end of life.

CPE: cpe:/o:microsoft:windows_xp
EOL date: 2014-04-08
EOL info: https://support.microsoft.com/en-us/lifecycle/search?sort=PN&alpha=Microsoft%20Windows%20XP&Filter=FilterNO

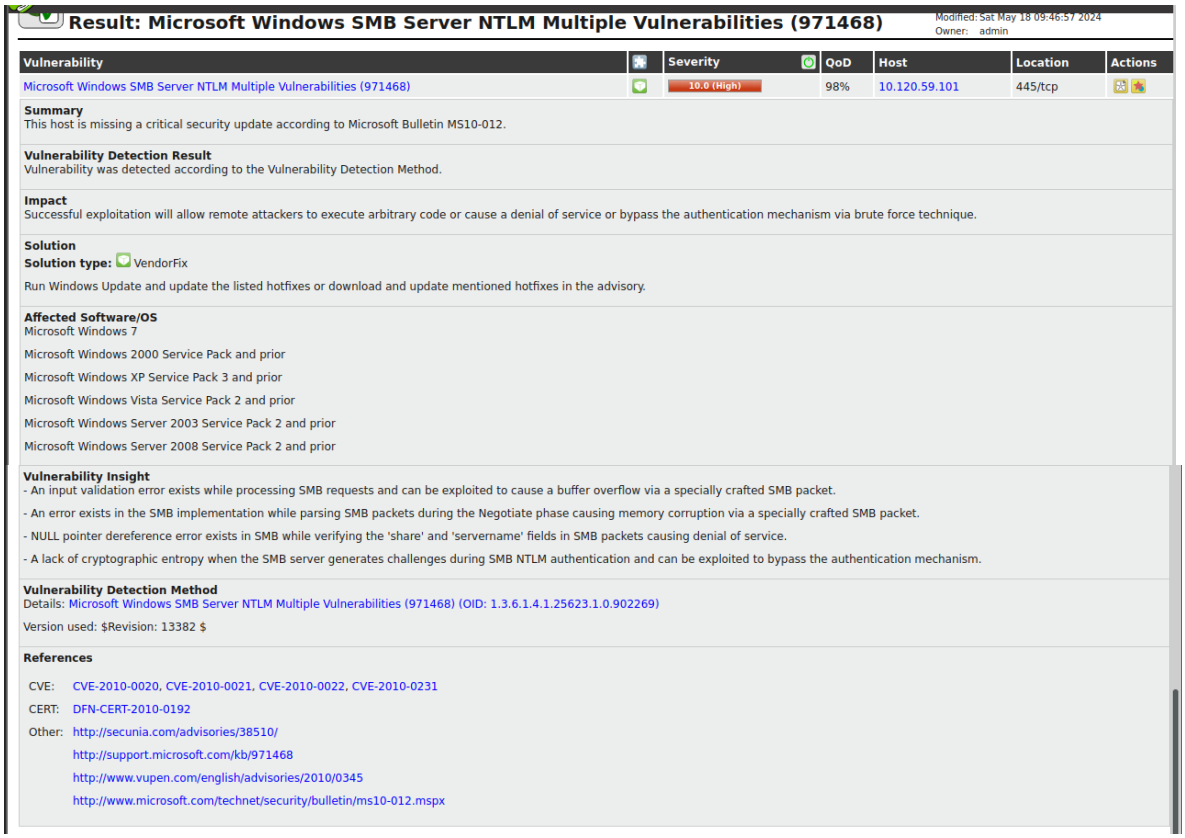
Vulnerability Detection Method
Details: OS End Of Life Detection (OID: 1.3.6.1.4.1.25623.1.0.103674)

Version used: $Revision: 8927 $

Product Detection Result

Product: cpe:/o:microsoft:windows_xp
Method: OS Detection Consolidation and Reporting (OID: 1.3.6.1.4.1.25623.1.0.105937)
Log: View details of product detection

Figura 52- "OS End Of Life Detection" Report



Result: Microsoft Windows SMB Server NTLM Multiple Vulnerabilities (971468)

Modified: Sat May 18 09:46:57 2024
Owner: admin

Vulnerability	Severity	QoD	Host	Location	Actions
Microsoft Windows SMB Server NTLM Multiple Vulnerabilities (971468)	10.0 (High)	98%	10.120.59.101	445/tcp	

Summary
This host is missing a critical security update according to Microsoft Bulletin MS10-012.

Vulnerability Detection Result
Vulnerability was detected according to the Vulnerability Detection Method.

Impact
Successful exploitation will allow remote attackers to execute arbitrary code or cause a denial of service or bypass the authentication mechanism via brute force technique.

Solution
Solution type: VendorFix
Run Windows Update and update the listed hotfixes or download and update mentioned hotfixes in the advisory.

Affected Software/OS
Microsoft Windows 7
Microsoft Windows 2000 Service Pack and prior
Microsoft Windows XP Service Pack 3 and prior
Microsoft Windows Vista Service Pack 2 and prior
Microsoft Windows Server 2003 Service Pack 2 and prior
Microsoft Windows Server 2008 Service Pack 2 and prior

Vulnerability Insight
- An input validation error exists while processing SMB requests and can be exploited to cause a buffer overflow via a specially crafted SMB packet.
- An error exists in the SMB implementation while parsing SMB packets during the Negotiate phase causing memory corruption via a specially crafted SMB packet.
- NULL pointer dereference error exists in SMB while verifying the 'share' and 'servername' fields in SMB packets causing denial of service.
- A lack of cryptographic entropy when the SMB server generates challenges during SMB NTLM authentication and can be exploited to bypass the authentication mechanism.

Vulnerability Detection Method
Details: Microsoft Windows SMB Server NTLM Multiple Vulnerabilities (971468) (OID: 1.3.6.1.4.1.25623.1.0.902269)
Version used: $Revision: 13382 $

References

CVE: CVE-2010-0020, CVE-2010-0021, CVE-2010-0022, CVE-2010-0231
CERT: DFN-CERT-2010-0192
Other: http://secunia.com/advisories/38510/
http://support.microsoft.com/kb/971468
http://www.vupen.com/english/advisories/2010/0345
http://www.microsoft.com/technet/security/bulletin/ms10-012.mspx

Figura 53- "Microsoft Windows SMB Server NTLM Multiple Vulnerabilites" Report

Owner: admin

Result: Vulnerabilities in SMB Could Allow Remote Code Execution (958687) - Remote

Vulnerability	Severity	QoD	Host	Location	Actions
Vulnerabilities in SMB Could Allow Remote Code Execution (958687) - Remote	10.0 (High)	98%	10.120.59.101	445/tcp	

Summary
This host is missing a critical security update according to Microsoft Bulletin MS09-001.

Vulnerability Detection Result
Vulnerability was detected according to the Vulnerability Detection Method.

Impact
Successful exploitation could allow remote unauthenticated attackers to cause denying the service by sending a specially crafted network message to a system running the server service.

Solution
Solution type: VendorFix
Run Windows Update and update the listed hotfixes or download and update mentioned hotfixes in the advisory

Affected Software/OS
Microsoft Windows 2K Service Pack 4 and prior.
Microsoft Windows XP Service Pack 3 and prior.
Microsoft Windows 2003 Service Pack 2 and prior.

Vulnerability Insight
The issue is due to the way Server Message Block (SMB) Protocol software handles specially crafted SMB packets.

Vulnerability Detection Method
Details: Vulnerabilities in SMB Could Allow Remote Code Execution (958687) - Remote (OID: 1.3.6.1.4.1.25623.1.0.900233)
Version used: $Revision: 12602 $

References

CVE: CVE-2008-4114, CVE-2008-4834, CVE-2008-4835
BID: 31179
Other: http://www.milw0rm.com/exploits/6463
http://www.microsoft.com/technet/security/bulletin/ms09-001.mspx

Figura 54- "Vulnerabilities in SMB Could Allow Remote Code Execution" Report

Result: Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)

ID: bd828204-c7f0-4273-9be9-9eae297f0a87
Created: Sat May 18 09:47:00 2024
Modified: Sat May 18 09:47:00 2024
Owner: admin

Vulnerability	Severity	QoD	Host	Location	Actions
Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)	9.3 (High)	95%	10.120.59.101	445/tcp	

Summary
This host is missing a critical security update according to Microsoft Bulletin MS17-010.

Vulnerability Detection Result
Vulnerability was detected according to the Vulnerability Detection Method.

Impact
Successful exploitation will allow remote attackers to gain the ability to execute code on the target server, also could lead to information disclosure from the server.

Solution
Solution type: VendorFix
Run Windows Update and update the listed hotfixes or download and update mentioned hotfixes in the advisory

Affected Software/OS
Microsoft Windows 10 x32/x64 Edition Microsoft Windows Server 2012 Edition Microsoft Windows Server 2016 Microsoft Windows 8.1 x32/x64 Edition Microsoft Windows Server 2012 R2 Edition Microsoft Windows 7 x32/x64 Edition Service Pack 1 Microsoft Windows Vista x32/x64 Edition Service Pack 2 Microsoft Windows Server 2008 R2 x64 Edition Service Pack 1 Microsoft Windows Server 2008 x32/x64 Edition Service Pack 2

Vulnerability Insight
Multiple flaws exist due to the way that the Microsoft Server Message Block 1.0 (SMBv1) server handles certain requests.

Vulnerability Detection Method
Send the crafted SMB transaction request with fid = 0 and check the response to confirm the vulnerability.
Details: Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389) (OID: 1.3.6.1.4.1.25623.1.0.810676)
Version used: $Revision: 11874 $

References
CVE: CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0147, CVE-2017-0148
BID: 96703, 96704, 96705, 96707, 96709, 96706
CERT: CB-K17/0435, DFN-CERT-2017-0448
Other: https://support.microsoft.com/en-in/kb/4013078
https://technet.microsoft.com/library/security/MS17-010
https://github.com/rapid7/metasploit-framework/pull/8167/files

Figura 55-"Microsoft Windows SMB Server Multiple Vulnerabilites- Remote" Report

3. *Metasploit*

Inicialmente, começou-se por atualizar a lista de pacotes e instalar o *Metasploit Framework* na máquina virtual, *Kali*. Para tal, usufruiu-se dos seguintes comandos:

```
sudo apt update
sudo apt install metasploit-framework
```

Em seguida, acedeu-se ao *metasploit*, executando:

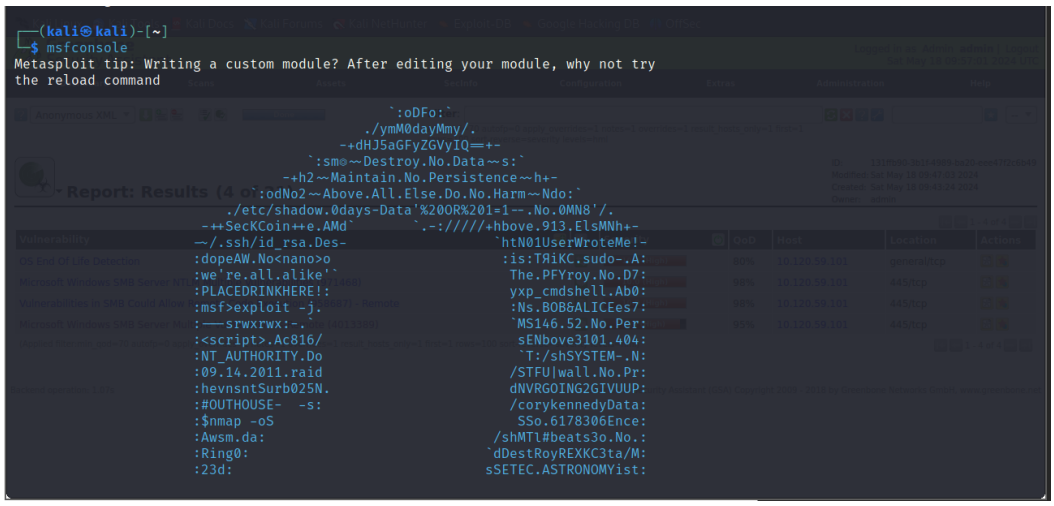


Figura 56- "msfconsole"

- *Windows XP*

Primeiramente, procurou-se por vulnerabilidades através do *metaspoit*, obtendo os seguintes resultados.

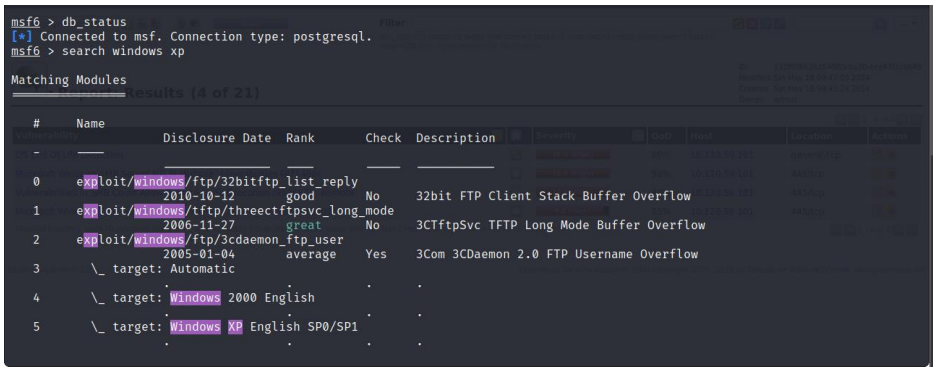


Figura 57- Vulnerabilidades

Em seguida, utilizou-se um dos *exploits* encontrados, configurou-se o IP do alvo, através do *RHOST* e, também, o IP local, através do *LHOST*.

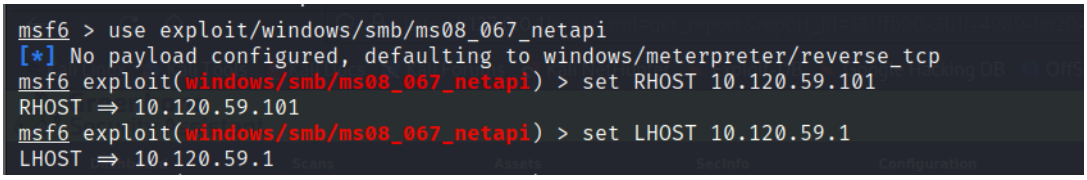


Figura 58- Configurações

Configurou-se, também, a porta de escuta, neste caso, porta 4444.

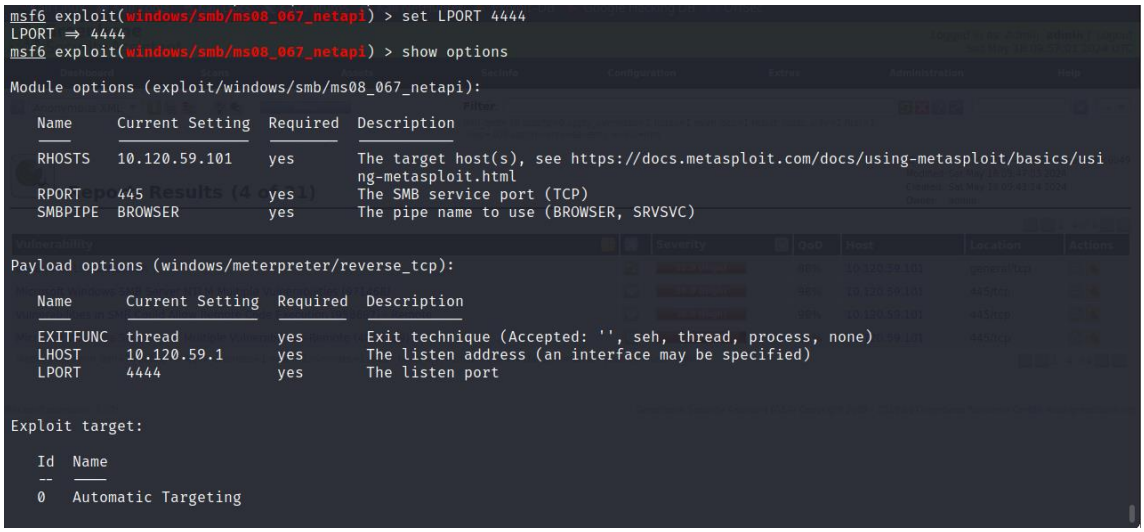


Figura 59- Configurações 2

Por fim, executamos o *exploit*, de forma a verificar o acesso à máquina virtual que, neste caso, se trata do *Windows XP*.

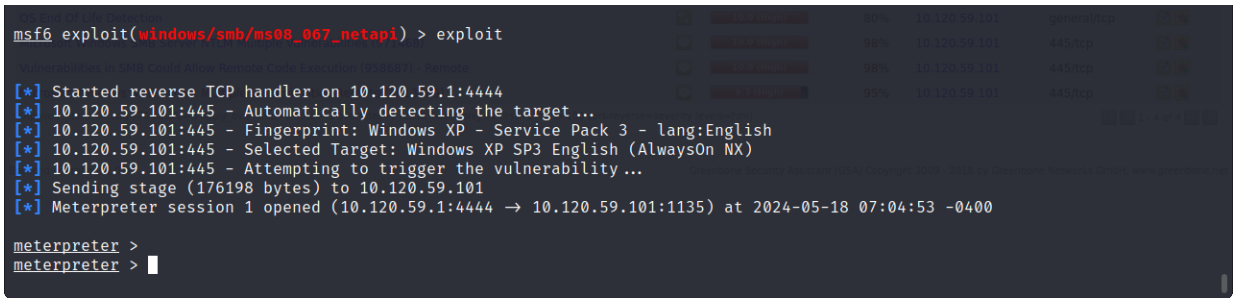


Figura 60- Execução do exploit

Abaixo são indicados alguns testes para validar o acesso à máquina, como por exemplo, a listagem de processos ativos na máquina alvo e a obtenção de informações do sistema.

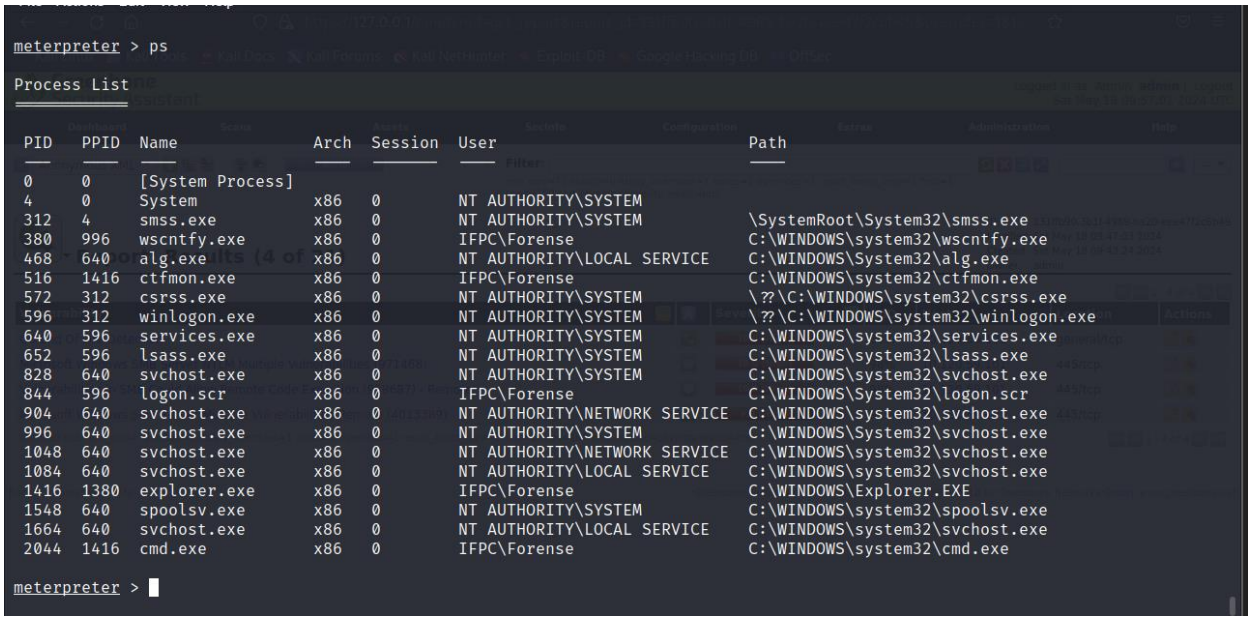


Figura 61- Listagem dos processos ativos

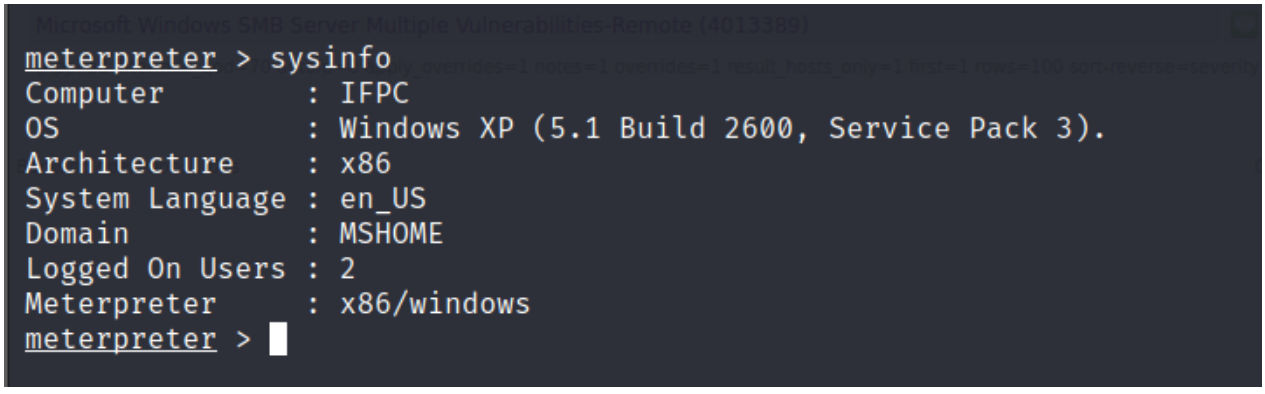


Figura 62- Informações do sistema

- *Kioptrix*

Neste caso, como não foi possível descobrir as vulnerabilidades através do *OpenVAS*, pesquisou-se por vulnerabilidades conhecidas, encontrando um exemplo no site seguinte: https://medium.com/@enzobomfim2003/explora%C3%A7%C3%A3o-da-m%C3%A1quina-kioptrix1-11fa4dd15b7f

Abaixo são indicadas as imagens do processo, relativamente semelhantes ao anterior, e uma validação do acesso à máquina, executando o comando *ping*.

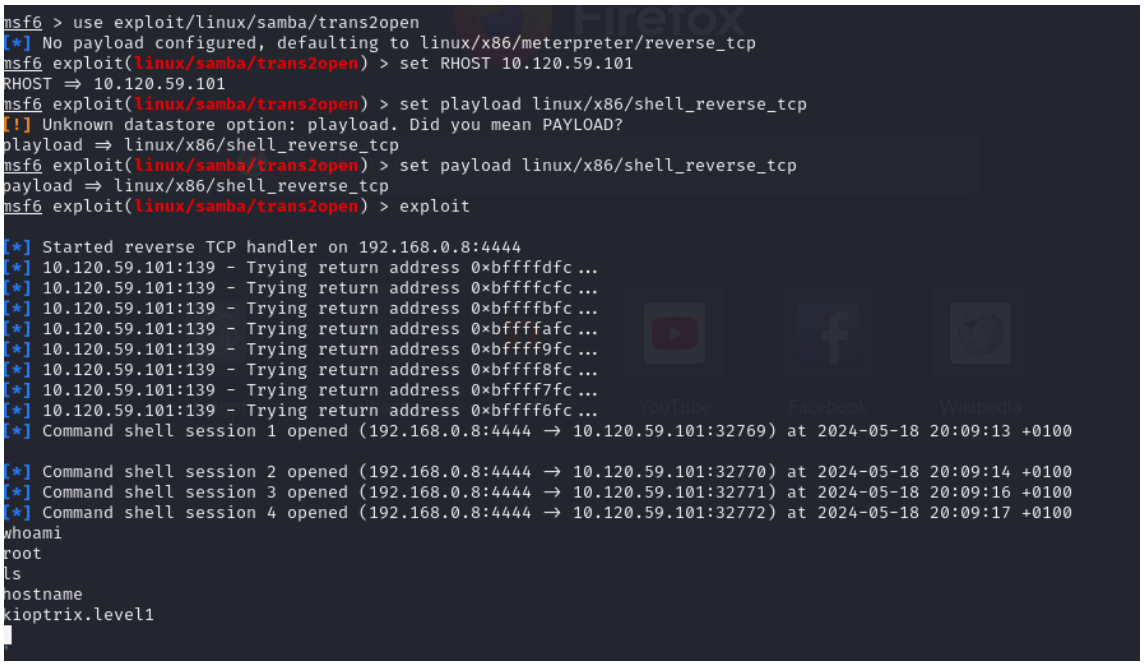


Figura 63- Configurações

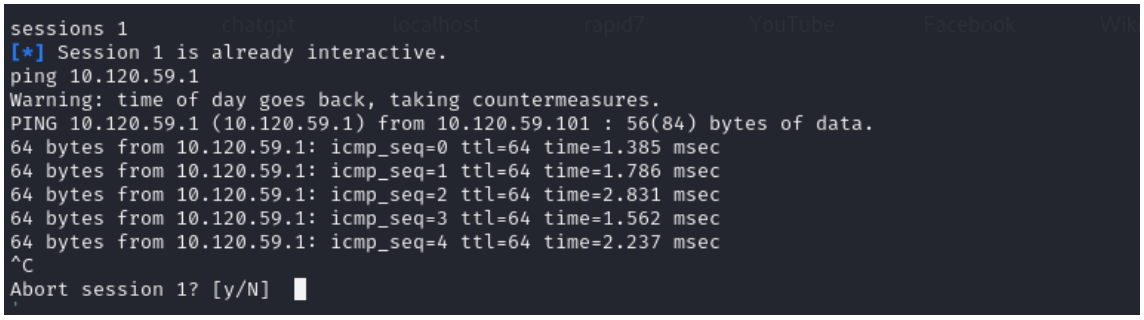


Figura 64- Validação através do comando ping

Conclusão

Apesar de ter sido um trabalho ligeiramente complexo, foi bastante satisfatório para aplicação e, de certa forma, aprendizagem de alguns conceitos aprendidos nas aulas. Contudo, as maiores dificuldades centraram-se na parte inicial, e, também, na instalação da ferramenta *OpenVAS* que, apesar das dificuldades conseguiu-se concluir com algum sucesso.

Bibliografia

https://www.bosontreinamentos.com.br/linux/servidor-dhcp-no-linux/

https://pplware.sapo.pt/internet/http-e-https-descubra-as-diferencas/

https://www.untanglebrasil.com.br/telnet-e-ssh-diferencas-e-qual-o-melhor/

https://www.youtube.com/watch?v=bBXCrnin3DY