



Universidade do Minho
Escola de Engenharia

UNIVERSIDADE DO MINHO

TRABALHO PRÁTICO

TP 2 - Parte B: PenTest - Scanning

Autores:

Bruno Rodrigues: pg41660

Carlos Alves pg41840

1 ÍNDICE

2	INTRODUÇÃO	3
3	- Questão 1	4
3.1	nmap -sn	4
3.2	nmap -sU	4
3.3	nmap -sT.....	6
3.4	Nmap -sS	7
4	Questão 2	9
4.1	Nmap -sA (scan ACK)	9
4.2	Nmap -sF	11
4.3	Nmap -sN	13
4.4	Nmap -sX.....	15
4.5	Diferenças ente -sF, -sN e -sX.....	16
5	Questão 3	18
6	Questão 4	19
6.1	Openssh 4.7p1.....	19
6.2	Apache http Server 2.2.8	20
6.3	UnrealIRCd	20
7	Questão 5	21
8	Questão 6	21
9	Questão 7	24
10	Questão 8.....	25
10.1	HTTP TRACK/TRACE Methods Allowed	25
10.2	rexecd Service Detection	26
10.3	Bind Shell Backdoor Detection	27
10.4	VNC Server 'password' Password	27
11	CONCLUSÃO	29
12	Bibliografia	30

2 INTRODUÇÃO

Este trabalho prático sugerido na unidade curricular Tecnologia de Segurança, tem por objetivo fazer uso do ambiente instalado e configurado para realizar atividades voltadas para *Penetration Testing*, tendo como base os exercícios descritos no enunciado do trabalho prático 2 – B.

Inicialmente foi então necessário instalar e configurar um ambiente de *Penetration Testing* (Trabalho Prático 2 – A), ao realizar as diversas configurações das várias componentes foram-nos surgindo alguns problemas, principalmente com a conectividade entre o **KALI** e o **Metasploitable2**.

Neste documento encontram-se as respostas de forma objetiva e com imagens para comprovar os resultados.

Máquina usada nos testes

192.168.56.4 - Endereço IP

192.168.56.6 - O endereço de IP do VM metasploitable2

Interface – vboxnet0

3 - QUESTÃO 1

3.1 NMAP -SN

O Comando envia um ping a todos os hosts disponíveis, e mostra aqueles que responderam.

Na figura em baixo podemos verificar que ao realizar um “ping” ao endereço 192.168.56.6, o comando -sn devolve que o host está ativo, e indica-nos também o endereço MAC da máquina.

```
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-28 18:30 WET
Nmap scan report for 192.168.56.6
Host is up (0.0015s latency).
MAC Address: 08:00:27:61:F1:C0 (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
```

Figura. 1 – nmap -sn 192.168.56.6

No **Wireshark**, configurado para observar tráfego na interface vboxnet0, podemos ver que o mesmo é bastante reduzido, sendo que só dois pacotes são transmitidos durante o scan.

Podemos verificar também que durante o scan é utilizado o protocolo **ARP** (*Address Resolution Protocol*). Para descobrir o endereço físico do computador a quem queremos comunicar, o protocolo **ARP** envia uma mensagem *Broadcast* a questionar a quem pertence o endereço de IP, sendo que a mensagem é respondida pelo recipiente do pedido (neste caso, a máquina com o IP 192.168.56.6), retornando o seu endereço MAC.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	0a:00:27:00:00:00	Broadcast	ARP	42	Who has 192.168.56.6? Tell 192.168.56.4
2	0.001521630	PcsCompu_61:f1:c0	0a:00:27:00:00:00	ARP	42	192.168.56.6 is at 08:00:27:61:f1:c0

3.2 NMAP -sU

Este comando faz um scan **UDP**, isto é, vai analisar portas **UDP** e descobrir quais estão abertas.

O pedido é feito através do envio de um cabeçalho **UDP** vazio a cada porta.

Caso a porta esteja fechada, é reportado um erro **ICMP**, que nos diz que a porta está inalcançável, caso esteja aberta, existirá trocas de pacotes.

Na figura em baixo podemos ver quais as portas **UDP** que se encontram abertas.

```
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-28 19:38 WET
Nmap scan report for 192.168.56.6
Host is up (0.00037s latency).
Not shown: 992 closed ports
PORT      STATE      SERVICE
53/udp    open       domain
68/udp    open|filtered dhcpcd
69/udp    open|filtered tftp
111/udp   open       rpcbind
137/udp   open       netbios-ns
138/udp   open|filtered netbios-dgm
786/udp   open|filtered concert
2049/udp  open       nfs
MAC Address: 08:00:27:61:F1:C0 (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 1075.08 seconds
```

Figura. 2 – nmap -sU 192.168.56.6

Podemos reparar também que existem algumas portas que estão marcadas como “open|filtered”.

Quando o **Nmap** é incapaz de determinar se uma porta está aberta ou filtrada, o **Nmap** coloca as portas num estado “open | filtered”.

Isto pode acontecer em scans onde portas abertas não dão nenhuma resposta, ou que um filtro de pacotes descartou a resposta que tenha sido pedida.

Portanto não sabe com certeza se a porta está aberta ou se está sendo filtrada.

A figura em baixo demonstra uma fração do tráfego gerado pelo scan.

No.	Time	Source	Destination	Protocol	Length	Info
593	206.881365054	192.168.56.4	192.168.56.6	UDP	42	40352 → 19 Len=0
594	206.881682890	192.168.56.6	192.168.56.4	ICMP	70	Destination unreachable (Port unreachable)
595	207.682281104	192.168.56.4	192.168.56.6	UDP	42	40352 → 34433 Len=0
596	207.682645420	192.168.56.6	192.168.56.4	ICMP	70	Destination unreachable (Port unreachable)
597	208.483143589	192.168.56.4	192.168.56.6	DNS	54	Server status request 0x0000
598	208.483424288	192.168.56.6	192.168.56.4	DNS	54	Server status request response 0x0000 Not implemented
599	208.483450703	192.168.56.4	192.168.56.6	ICMP	82	Destination unreachable (Port unreachable)

Figura. 3 – Excerto de tráfego

Nesta figura temos o exemplo entre uma porta que está fechada, e uma porta que está aberta.

Podemos reparar que quando existe um pedido **UDP** para a porta 19, é retornado um erro **ICMP**, que nos diz que a porta está fechada.

Por outro lado, quando existe um pedido para a porta 53(as linhas 597 e 598), podemos reparar que retorno de pacote, o que nos diz que a porta está aberta.

3.3 NMAP –ST

O **Nmap** pede ao sistema operativo para estabelecer uma conexão com a máquina e porta alvos enviando uma chamada de sistema `connect()`, que é a mesma chamada que browsers e outras aplicações de rede usam para estabelecer uma conexão. Este modo utiliza a API de Sockets de Berkeley, que invés de ler as respostas em pacotes em estado bruto(raw), o **Nmap** utiliza esta API para obter informações do estado de cada tentativa de conexão.

Na figura em baixo podemos ver as portas TCP que estão abertas no endereço IP 192.168.56.6

```
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-28 22:12 WET
Nmap scan report for 192.168.56.6
Host is up (0.00020s latency)
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    40 open  ftp
22/tcp    41 open  ssh
23/tcp    42 open  telnet
25/tcp    43 open  smtp
53/tcp    44 open  domain
80/tcp    45 open  http
111/tcp   46 open  rpcbind
139/tcp   47 open  netbios-ssn
445/tcp   48 open  microsoft-ds
512/tcp   49 open  exec
513/tcp   50 open  login
514/tcp   51 open  shell
1099/tcp  52 open  rmiregistry
1524/tcp  53 open  ingreslocknet0
2049/tcp  54 open  acnfame: vboxnet0
2121/tcp  55 open  atccproxy-ftp
3306/tcp  56 open  mysql
5432/tcp  57 open  postgresql
5900/tcp  58 open  vnc
6000/tcp  59 open  x11
6667/tcp  60 open  irc
8009/tcp  61 open  ajp13
8180/tcp  62 open  unknown
MAC Address: 08:00:27:61:F1:C0 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

Figura. 4 – nmap -sT 192.168.56.6

Uma porta **TCP** retorna **SYN/ACK** caso a porta esteja aberta, e um **RST** (reset) caso não esteja aberta.

Como podemos reparar em baixo, no exemplo de tráfego gerado pelo scan e recolhido pelo **Wireshark**, na linha 13 e 17 é retornado um **RST** para as portas 995 e 42000, indicando que essas portas estão fechadas, e na linha 14 e 16 é retornado um **SYN/ACK**, que nos indica que as portas 445 e 5900 se encontram abertas.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.047532992	192.168.56.4	192.168.56.6	TCP	74	42000 → 445 [SYN] Seq=0 Win=64
5	0.047562463	192.168.56.4	192.168.56.6	TCP	74	53292 → 5900 [SYN] Seq=0 Win=6
6	0.047606582	192.168.56.4	192.168.56.6	TCP	74	50796 → 80 [SYN] Seq=0 Win=642
7	0.047664234	192.168.56.4	192.168.56.6	TCP	74	48812 → 1723 [SYN] Seq=0 Win=6
8	0.047704498	192.168.56.4	192.168.56.6	TCP	74	33270 → 23 [SYN] Seq=0 Win=642
9	0.047741657	192.168.56.4	192.168.56.6	TCP	74	60570 → 22 [SYN] Seq=0 Win=642
10	0.047777845	192.168.56.4	192.168.56.6	TCP	74	54622 → 993 [SYN] Seq=0 Win=64
11	0.047810907	192.168.56.4	192.168.56.6	TCP	74	52562 → 3389 [SYN] Seq=0 Win=6
12	0.047843508	192.168.56.4	192.168.56.6	TCP	74	51274 → 8888 [SYN] Seq=0 Win=6
13	0.047943583	192.168.56.6	192.168.56.4	TCP	54	995 → 60282 [RST, ACK] Seq=1 A
14	0.048007283	192.168.56.6	192.168.56.4	TCP	74	445 → 42000 [SYN, ACK] Seq=0 A
15	0.048030374	192.168.56.4	192.168.56.6	TCP	66	42000 → 445 [ACK] Seq=1 Ack=1
16	0.048088315	192.168.56.6	192.168.56.4	TCP	74	5900 → 53292 [SYN, ACK] Seq=0
17	0.048101229	192.168.56.4	192.168.56.6	TCP	66	42000 → 445 [RST, ACK] Seq=1 A

Figura. 5 – Excerto tráfego

3.4 NMAP -sS

O -sS tem o mesmo objetivo que o -sT, sendo que este também é utilizado para encontrar portas TCP abertas. Ao contrário do comando -sT, o -sS não conclui a conexão com a porta TCP que está a ser verificada, enviando um pacote SYN como se fosse abrir uma conexão real, e depois espera pela resposta.

Na figura em baixo podemos ver as portas TCP que estão abertas no endereço IP 192.168.56.6, igual a -sT.

```

root@Bruno:~# nmap -sS 192.168.56.6
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-28 22:51 WET
Nmap scan report for 192.168.56.6
Host is up (0.00029s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslocknet0
2049/tcp  open  acnfsame: vboxnet0
2121/tcp  open  atccproxy-ftp: ethernet (1)
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  tx11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 08:00:27:61:F1:C0 (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.26 seconds

```

Figura. 6 – nmap -sS 192.168.56.6

No exemplo de tráfego mostrado a seguir, podemos verificar o envio de pacotes SYN para o endereço de ip 192.168.56.6, como também a resposta por parte deste, nas linhas 13 e 15 podemos verificar que é retornado uma flag SYN/ACK, que nos indica que as portas 21 e 53 estão abertas, e nas linhas a vermelho o retorno de flags RST, que significa que as portas estão fechadas.

12	0.088228120	192.168.56.4	192.168.56.6	TCP	58 54565 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
13	0.088341064	192.168.56.6	192.168.56.4	TCP	58 53 → 54565 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
14	0.088367170	192.168.56.4	192.168.56.6	TCP	54 54565 → 53 [RST] Seq=1 Win=0 Len=0
15	0.088401075	192.168.56.6	192.168.56.4	TCP	58 21 → 54565 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
16	0.088412691	192.168.56.4	192.168.56.6	TCP	54 54565 → 21 [RST] Seq=1 Win=0 Len=0
17	0.088437805	192.168.56.6	192.168.56.4	TCP	54 3389 → 54565 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
18	0.088461258	192.168.56.6	192.168.56.4	TCP	54 1720 → 54565 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Figura. 7 – Excerto tráfego

Nas figuras em baixo podemos ver a diferença de tráfego gerado pelos scans -sT e -sS:

Podemos verificar que como no scan -sT é realizado uma conexão com a porta alvo, é gerado mais tráfego de rede:

35	0.048447963	192.168.56.4	192.168.56.6	TCP	74 56020 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2620406817 TSecr=0 WS=1024
47	0.048705671	192.168.56.6	192.168.56.4	TCP	74 139 → 56020 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=1310716 TSecr=2620406817 WS=32
48	0.048718728	192.168.56.4	192.168.56.6	TCP	66 56020 → 139 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2620406818 TSecr=1310716
49	0.048730877	192.168.56.4	192.168.56.6	TCP	66 56020 → 139 [RST, ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2620406818 TSecr=1310716

Figura. 8 – Excerto tráfego scan (-sT)

Enquanto que no scan -sS, como não estabelece uma conexão real com a porta alvo, o tráfego gerado é menor:

30	0.088726099	192.168.56.4	192.168.56.6	TCP	58 54565 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
51	0.088893135	192.168.56.6	192.168.56.4	TCP	58 445 → 54565 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
52	0.088904143	192.168.56.4	192.168.56.6	TCP	54 54565 → 445 [RST] Seq=1 Win=0 Len=0

Figura. 9 – Excerto tráfego scan(-sS)

4 QUESTÃO 2

4.1 NMAP -sA (SCAN ACK)

O scan -sA, ao contrário dos outros scans, não serve para verificar se as portas estão abertas ou fechadas, mas para mapear conjuntos de regras de firewall, determinando quais as portas que estão filtradas ou não.

Na figura em baixo podemos verificar que todas as portas no VMmetasploitable2 se encontram não filtradas.

```
root@Bruno:~# nmap -sA 192.168.56.6
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-01 19:47 WET
Nmap scan report for 192.168.56.6
Host is up (0.00036s latency).
All 1000 scanned ports on 192.168.56.6 are unfiltered
MAC Address: 08:00:27:61:F1:C0 (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds
```

Figura. 10 – nmap -sA 192.168.56.6

Quando uma porta se encontra não filtrada, significa que a porta está acessível, mas o scan ACK (-sA) não determina se estas se encontram abertas ou fechadas.

Na figura em baixo podemos verificar que todas as portas no endereço IP 45.33.32.156, exceto uma, se encontram não filtradas.

```
root@Bruno:~# nmap -sA 45.33.32.156
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-27 16:26 WET
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.21s latency).
Not shown: 999 unfiltered ports
PORT      STATE      SERVICE
19/tcp    filtered  chargen
Nmap done: 1 IP address (1 host up) scanned in 2.48 seconds
```

Figura. 11 – nmap -sA 45.33.32.156

Quando uma porta se encontra filtrada, significa que o scan não conseguiu determinar se a porta está acessível, devido a uma filtragem de pacotes que se encontra ativa que impedem a sondagem. Isto pode ser devido a uma firewall, ou regras de router.

Podemos reparar na figura a seguir, que demonstra uma fração do tráfego captado pelo Wireshark, o envio de pacotes com a flag ACK para o endereço ip 192.168.56.6, e o retorno de pacotes RST, tanto por portas abertas e fechadas.

8	0.084298723	192.168.56.4	192.168.56.6	TCP	54 34455 → 199 [ACK] Seq=1 Ack=1 Win=1024 Len=0
9	0.084309315	192.168.56.4	192.168.56.6	TCP	54 34455 → 993 [ACK] Seq=1 Ack=1 Win=1024 Len=0
10	0.084319845	192.168.56.4	192.168.56.6	TCP	54 34455 → 25 [ACK] Seq=1 Ack=1 Win=1024 Len=0
11	0.084330752	192.168.56.4	192.168.56.6	TCP	54 34455 → 1720 [ACK] Seq=1 Ack=1 Win=1024 Len=0
12	0.084347588	192.168.56.4	192.168.56.6	TCP	54 34455 → 23 [ACK] Seq=1 Ack=1 Win=1024 Len=0
13	0.084568710	192.168.56.6	192.168.56.4	TCP	54 256 → 34455 [RST] Seq=1 Win=0 Len=0
14	0.084625031	192.168.56.6	192.168.56.4	TCP	54 143 → 34455 [RST] Seq=1 Win=0 Len=0
15	0.084654747	192.168.56.6	192.168.56.4	TCP	54 554 → 34455 [RST] Seq=1 Win=0 Len=0
16	0.084684986	192.168.56.6	192.168.56.4	TCP	54 80 → 34455 [RST] Seq=1 Win=0 Len=0
17	0.084713364	192.168.56.6	192.168.56.4	TCP	54 113 → 34455 [RST] Seq=1 Win=0 Len=0

Figura.12 - Excerto de tráfego

O mesmo acontece para o endereço de ip 45.33.32.156:

17	0.564488492	192.168.1.72	45.33.32.156	TCP	54 47449 → 22 [ACK] Seq=1 Ack=1 Win=1024 Len=0
18	0.564497310	192.168.1.72	45.33.32.156	TCP	54 47449 → 111 [ACK] Seq=1 Ack=1 Win=1024 Len=0
19	0.564506196	192.168.1.72	45.33.32.156	TCP	54 47449 → 3389 [ACK] Seq=1 Ack=1 Win=1024 Len=0
20	0.564516886	192.168.1.72	45.33.32.156	TCP	54 47449 → 3306 [ACK] Seq=1 Ack=1 Win=1024 Len=0
21	0.564529699	192.168.1.72	45.33.32.156	TCP	54 47449 → 1720 [ACK] Seq=1 Ack=1 Win=1024 Len=0
22	0.799863395	45.33.32.156	192.168.1.72	TCP	54 23 → 47449 [RST] Seq=1 Win=0 Len=0
23	0.800723619	45.33.32.156	192.168.1.72	TCP	54 1723 → 47449 [RST] Seq=1 Win=0 Len=0
24	0.800744776	45.33.32.156	192.168.1.72	TCP	54 443 → 47449 [RST] Seq=1 Win=0 Len=0
25	0.800750162	45.33.32.156	192.168.1.72	TCP	54 53 → 47449 [RST] Seq=1 Win=0 Len=0
26	0.800754968	45.33.32.156	192.168.1.72	TCP	54 8080 → 47449 [RST] Seq=1 Win=0 Len=0

Figura.13 - Excerto de tráfego

É de notar também, que a porta que se encontra filtrada, a porta 19, não retorna nenhum pacote como resposta ao envio do pacote ACK, por isso a porta fica marcada como filtrada.

O Wireshark tem uma ferramenta que nos permite seguir o fluxo de pacotes trocados pelas portas, e através dela podemos verificar que a porta 19 não retorna uma resposta:

No.	Time	Source	Destination	Protocol	Length	Info
263	2.189220234	192.168.1.72	45.33.32.156	TCP	54	47449 → 19 [ACK] Seq=1 Ack=1 Win=1024 Len=0

Figura.14 –Transmissão de pacotes para porta 19

Em contraste com outras portas que não se encontram filtradas:

No.	Time	Source	Destination	Protocol	Length	Info
12	0.564423241	192.168.1.72	45.33.32.156	TCP	54	47449 → 23 [ACK] Seq=1 Ack=1 Win=1024 Len=0
22	0.799863395	45.33.32.156	192.168.1.72	TCP	54	23 → 47449 [RST] Seq=1 Win=0 Len=0

Figura.15 –Troca de pacotes para porta 23

4.2 NMAP –SF

O nmap -sF (FIN scan) verifica se existe portas abertas | filtradas, enviando uma FIN flag para a porta.

Num serviço normal, uma flag FIN é enviada quando uma conversa entre portas terminou, sendo que uma porta que está fechada retornará uma RST flag, enquanto que uma porta aberta não retorna nada, visto que a ligação terminou, sendo o pacote descartado.

Então, se a porta retornar um RST flag, significa que a porta está fechada, se a porta não retornar nada, significa que esta está aberta, ou filtrada, sendo que este scan não consegue diferenciar.

Na figura em baixo podemos verificar as portas TCP que se encontram abertas | filtradas no endereço ip 192.168.56.6:

```

root@Bruno:~# nmap -sF 192.168.56.6
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-01 19:48 WET
Nmap scan report for 192.168.56.6
Host is up (0.0027s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open|filtered  ftp
22/tcp    open|filtered  ssh
23/tcp    open|filtered  telnet
25/tcp    open|filtered  smtp
53/tcp    open|filtered  domain
80/tcp    open|filtered  http
111/tcp   open|filtered  rpcbind
139/tcp   open|filtered  netbios-ssn
445/tcp   open|filtered  microsoft-ds
512/tcp   open|filtered  exec
513/tcp   open|filtered  login
514/tcp   open|filtered  shell (336 bits), 42 bytes captured (336
1099/tcp  open|filtered  rmiregistry:00 (0a:00:27:00:00:00), Dst:
1524/tcp  open|filtered  ingreslockquest)
2049/tcp  open|filtered  nfs
2121/tcp  open|filtered  ccproxy-ftp
3306/tcp  open|filtered  mysql
5432/tcp  open|filtered  postgresql
5900/tcp  open|filtered  vnc
6000/tcp  open|filtered  X11
6667/tcp  open|filtered  irc
8009/tcp  open|filtered  ajp13
8180/tcp  open|filtered  unknown
MAC Address: 08:00:27:61:F1:C0 (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 1.42 seconds

```

Figura. 16 – nmap -sF 192.168.56.6

No endereço de ip 45.33.32.156 todas as portas encontram-se abertas | filtradas

```
root@Bruno:~# nmap -sF 45.33.32.156
Starting Nmap 7.80 (https://nmap.org) at 2019-11-27 16:28 WET
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.19s latency).
All 1000 scanned ports on scanme.nmap.org (45.33.32.156) are open|filtered
Nmap done: 1 IP address (1 host up) scanned in 17.35 seconds
```

Figura. 17 – nmap -sF 45.33.32.156

Em baixo podemos ver um extrato do tráfego de rede gerado pelo scan para o endereço 192.168.56.6, onde se verifica o envio de pacotes com a flag FIN:

10	0.080626347	192.168.56.4	192.168.56.6	TCP	54 35979 → 1720 [FIN] Seq=1 Win=1024 Len=0
11	0.080645599	192.168.56.4	192.168.56.6	TCP	54 35979 → 995 [FIN] Seq=1 Win=1024 Len=0
12	0.080677560	192.168.56.4	192.168.56.6	TCP	54 35979 → 80 [FIN] Seq=1 Win=1024 Len=0
13	0.080843580	192.168.56.6	192.168.56.4	TCP	54 143 → 35979 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
14	0.080921721	192.168.56.6	192.168.56.4	TCP	54 110 → 35979 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
15	0.080972849	192.168.56.6	192.168.56.4	TCP	54 256 → 35979 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0

Figura. 18 – Excerto de tráfego

Por exemplo, a porta 80 não retorna pacotes, o que nos indica que esta encontra-se aberta | filtrada

No.	Time	Source	Destination	Protocol	Length	Info
12	0.080677560	192.168.56.4	192.168.56.6	TCP	54	35979 → 80 [FIN] Seq=1 Win=1024 Len=0

Figura.19 –Troca de pacotes para porta 60

Enquanto que a porta 143, na linha 13, retorna uma flag RST, logo encontra-se fechada:

No.	Time	Source	Destination	Protocol	Length	Info
3	0.080487859	192.168.56.4	192.168.56.6	TCP	54	35979 → 143 [FIN] Seq=1 Win=1024 Len=0
13	0.080843580	192.168.56.6	192.168.56.4	TCP	54	143 → 35979 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0

Figura.20 –Troca de pacotes para porta 143

Na figura em baixo podemos observar o extrato de tráfego de rede gerado para o endereço 45.33.32.156, diferenciando-se do tráfego de rede gerado pelo endereço de ip 192.16.56.6 pelo facto de não observarmos pacotes com a flag RST. Podemos então concluir que todas as portas estão abertas | filtradas.

21	1.021857357	192.168.1.72	45.33.32.156	TCP	54 64844 → 5900 [FIN] Seq=1 Win=1024 Len=0
22	1.021891135	192.168.1.72	45.33.32.156	TCP	54 64844 → 25 [FIN] Seq=1 Win=1024 Len=0
23	1.021903212	192.168.1.72	45.33.32.156	TCP	54 64844 → 443 [FIN] Seq=1 Win=1024 Len=0
24	1.021913499	192.168.1.72	45.33.32.156	TCP	54 64844 → 199 [FIN] Seq=1 Win=1024 Len=0
25	1.021924354	192.168.1.72	45.33.32.156	TCP	54 64844 → 113 [FIN] Seq=1 Win=1024 Len=0
26	1.021934488	192.168.1.72	45.33.32.156	TCP	54 64844 → 256 [FIN] Seq=1 Win=1024 Len=0

Figura.21 –Exemplo de tráfego

Se seguirmos o fluxo da troca de pacotes podemos ver o fluxo de pacotes para todas as portas é igual ao exemplo a seguir:

No.	Time	Source	Destination	Protocol	Length	Info
21	1.021857357	192.168.1.72	45.33.32.156	TCP	54	64844 → 5900 [FIN] Seq=1 Win=1024 Len=0

Figura.22–Troca de pacotes para porta 5900

4.3 NMAP -sN

O Null Scan (-sN) é um tipo de scan onde são enviados para as portas pacotes sem nenhuma flag.

O objetivo deste scan é verificar que portas se encontram abertas.

Como o pacote enviado não tem flags implementadas, quando é recebido por uma porta aberta é descartado. Se a porta tiver fechada, um pacote RST é enviado como resposta.

Em baixo podemos ver o resultado do scan para o ip 192.168.56.6:

```

root@Bruno:~# nmap -sN 192.168.56.6
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-01 19:49 WET
Nmap scan report for 192.168.56.6:
Host is up (0.00064s latency).
Not shown: 6977 closed ports
PORT      STATE SERVICE
21/tcp    0 open|filtered ftp
22/tcp    7 open|filtered ssh
23/tcp    8 open|filtered telnet
25/tcp    9 open|filtered smtp
53/tcp    0 open|filtered domain
80/tcp    1 open|filtered http
111/tcp   0 open|filtered rpcbind
139/tcp   0 open|filtered netbios-ssn
445/tcp   0 open|filtered microsoft-ds
512/tcp   0 open|filtered exec
513/tcp   0 open|filtered login
514/tcp   0 open|filtered shell (336 bits), 42 bytes captured (336 b
1099/tcp  0 open|filtered rmiregistry:00 (0a:00:27:00:00:00), Dst: t
1524/tcp  0 open|filtered ingreslockquest)
2049/tcp  0 open|filtered nfs
2121/tcp  0 open|filtered ccproxy-ftp
3306/tcp  0 open|filtered mysql
5432/tcp  0 open|filtered postgresql
5900/tcp  0 open|filtered vnc
6000/tcp  0 open|filtered X11
6667/tcp  0 open|filtered irc
8009/tcp  0 open|filtered ajp13
8180/tcp  0 open|filtered unknown
MAC Address: 08:00:27:61:F1:C0 (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 1.44 seconds

```

Figura.23 –Scan -sN 192.168.56.6

E aqui para o endereço ip 45.33.32.156:

```
root@Bruno:~# nmap -sN 45.33.32.156 -o Port: 60730, Dst Port: 443, Seq: 1, Ac
Starting Nmap 7.800 (https://nmap.org) at 2019-11-27 16:30 WET
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).
All 1000 scanned ports on scanme.nmap.org (45.33.32.156) are open|filtered
Nmap done: 1 IP address (1 host up) scanned in 18.06 seconds
```

Figura.24 –Scan -sN 45.33.32.156

Em baixo podemos ver o tráfego de rede gerado por este scan para o endereço 192.168.56.6:

9	0.080432236	192.168.56.4	192.168.56.6	TCP	54	36558 → 995 [<None>] Seq=1 Win=1024 Len=0
10	0.080444890	192.168.56.4	192.168.56.6	TCP	54	36558 → 135 [<None>] Seq=1 Win=1024 Len=0
11	0.080457632	192.168.56.4	192.168.56.6	TCP	54	36558 → 5900 [<None>] Seq=1 Win=1024 Len=0
12	0.080472186	192.168.56.4	192.168.56.6	TCP	54	36558 → 53 [<None>] Seq=1 Win=1024 Len=0
13	0.080713302	192.168.56.6	192.168.56.4	TCP	54	993 → 36558 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
14	0.080776396	192.168.56.6	192.168.56.4	TCP	54	443 → 36558 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
15	0.080807924	192.168.56.6	192.168.56.4	TCP	54	8888 → 36558 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
16	0.080837891	192.168.56.6	192.168.56.4	TCP	54	995 → 36558 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Figura.25 –Excerto de trafego

Conseguimos ver que são enviados pacotes sem nenhuma flag imposta, e também conseguimos ver que existe portas que retornam um RST flag, que nos indica que estão fechadas.

Em baixo podemos verificar que a porta 53 não retorna nenhum pacote:

No.	Time	Source	Destination	Protocol	Length	Info
12	0.080472186	192.168.56.4	192.168.56.6	TCP	54	36558 → 53 [<None>] Seq=1 Win=1024 Len=0

Figura.26 –Troca de pacotes para a porta 53

Em contraste com uma porta que retorna um pacote RST:

No.	Time	Source	Destination	Protocol	Length	Info
9	0.080432236	192.168.56.4	192.168.56.6	TCP	54	36558 → 995 [<None>] Seq=1 Win=1024 Len=0
16	0.080837891	192.168.56.6	192.168.56.4	TCP	54	995 → 36558 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Figura.27 –Troca de pacotes para a porta 995

A seguir podemos ver o tráfego de rede gerado por este scan para o endereço 45.33.32.156:

15	1.309632556	192.168.1.72	45.33.32.156	TCP	54	45722 → 995 [<None>] Seq=1 Win=1024 Len=0
16	1.309662867	192.168.1.72	45.33.32.156	TCP	54	45722 → 993 [<None>] Seq=1 Win=1024 Len=0
17	1.309672341	192.168.1.72	45.33.32.156	TCP	54	45722 → 25 [<None>] Seq=1 Win=1024 Len=0
18	1.309680114	192.168.1.72	45.33.32.156	TCP	54	45722 → 256 [<None>] Seq=1 Win=1024 Len=0
19	1.309688447	192.168.1.72	45.33.32.156	TCP	54	45722 → 80 [<None>] Seq=1 Win=1024 Len=0
20	1.309696443	192.168.1.72	45.33.32.156	TCP	54	45722 → 22 [<None>] Seq=1 Win=1024 Len=0

Figura.28 –Excerto de tráfego

Que não retorna nenhum pacote RST, sendo que todas as portas estão abertas | filtradas.

4.4 NMAP -sX

O Xmas Scan (-sX) é um tipo de scan onde são enviados para as portas pacotes com três flags: FIN, PSH e URG.

O objetivo deste scan é verificar que portas se encontram abertas.

Se a porta tiver fechada, um pacote RST é enviado como resposta.

Em baixo podemos ver o resultado do scan -sX para o endereço de ip 192.168.56.6:

```

root@Bruno:~# nmap -sX 192.168.56.6
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-01 19:50 WET
Nmap scan report for 192.168.56.6: c0 0a:00:27:00:00:00 ARP
Host is up (0.0019s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    6 open|filtered ftp
22/tcp    7 open|filtered ssh
23/tcp    8 open|filtered telnet
25/tcp    9 open|filtered smtp
53/tcp    0 open|filtered domain
80/tcp    1 open|filtered http
111/tcp   0 open|filtered rpcbind
139/tcp   0 open|filtered netbios-ssn
445/tcp   0 open|filtered microsoft-ds
512/tcp   0 open|filtered exec
513/tcp   0 open|filtered login
514/tcp   0 open|filtered shell (336 bits), 42 bytes captured (336
1099/tcp  0 open|filtered rmiregistry:00 (0a:00:27:00:00:00), Dst:
1524/tcp  0 open|filtered pingreslockquest)
2049/tcp  0 open|filtered nfs
2121/tcp  0 open|filtered ccproxy-ftp
3306/tcp  0 open|filtered mysql
5432/tcp  0 open|filtered postgresql
5900/tcp  0 open|filtered vnc
6000/tcp  0 open|filtered X11
6667/tcp  0 open|filtered irc
8009/tcp  0 open|filtered ajp13
8180/tcp  0 open|filtered unknown
MAC Address: 08:00:27:61:F1:C0 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.43 seconds

```

Figura.29 –Scan -sX 192.168.56.6

E para o endereço 45.33.32.156:

```

root@Bruno:~# nmap -sX 45.33.32.156
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-27 16:32 WET
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.19s latency).
All 1000 scanned ports on scanme.nmap.org (45.33.32.156) are open|filtered
Nmap done: 1 IP address (1 host up) scanned in 15.61 seconds

```

Figura.30 –Scan -sX 45.33.32.156

Em baixo vemos uma fração do tráfego captado:

No.	Time	Source	Destination	Protocol	Length	Info
10	0.084089033	192.168.56.4	192.168.56.6	TCP	54	46207 → 22 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
11	0.084096168	192.168.56.4	192.168.56.6	TCP	54	46207 → 993 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
12	0.084103154	192.168.56.4	192.168.56.6	TCP	54	46207 → 110 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
13	0.084239090	192.168.56.6	192.168.56.4	TCP	54	3389 → 46207 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
14	0.084293062	192.168.56.6	192.168.56.4	TCP	54	1720 → 46207 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
15	0.084322832	192.168.56.6	192.168.56.4	TCP	54	1723 → 46207 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0

Figura.31 –Excerto de tráfego

Podemos ver que os pacotes são enviados com as três flags, também podemos ver que há portas que retornam pacotes com RST flags, que nos indica que estão fechadas.

Em baixo podemos ver o tráfego de rede gerado para o endereço de ip 45.33.32.156:

No.	Time	Source	Destination	Protocol	Length	Info
19	0.526265093	192.168.1.72	45.33.32.156	TCP	54	64659 → 995 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
20	0.526279439	192.168.1.72	45.33.32.156	TCP	54	64659 → 113 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
21	0.526294203	192.168.1.72	45.33.32.156	TCP	54	64659 → 110 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
22	0.526310174	192.168.1.72	45.33.32.156	TCP	54	64659 → 3306 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
23	0.526340843	192.168.1.72	45.33.32.156	TCP	54	64659 → 1025 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
24	0.526366907	192.168.1.72	45.33.32.156	TCP	54	64659 → 5900 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0

Figura.32 –Excerto de tráfego

Que não retorna nenhum pacote RST, sendo que todas as portas estão abertas | filtradas.

4.5 DIFERENÇAS ENTE -SF, -SN E -SX

Os três scans aqui referidos têm exatamente o mesmo comportamento, sendo que a diferença são as flags dos pacotes. Estes scans servem para descobrir quais as portas que se encontram abertas no sistema-alvo.

Um pacote sem nenhuma flag imposta, como é o caso do Null scan, aparecerá assim se inspecionarmos com o wireshark:

```

▼ Flags: 0x000 (<None>)
 000. .... = Reserved: Not set
...0 .... = Nonce: Not set
.... 0... = Congestion Window Reduced (CWR): Not set
.... .0.. = ECN-Echo: Not set
.... ..0. = Urgent: Not set
.... ...0 = Acknowledgment: Not set
.... .... 0... = Push: Not set
.... ..... 0.. = Reset: Not set
.... ..... ..0. = Syn: Not set
.... ..... ...0 = Fin: Not set
[TCP Flags: .....]
```

Figura.33 – Flags ativas Scan Null

Um pacote com as flags FIN, PSH e URG , como no Xmas Scan:

```

▼ Flags: 0x029 (FIN, PSH, URG)
  000. .... = Reserved: Not set
  ...0 .... = Nonce: Not set
  .... 0... = Congestion Window Reduced (CWR): Not set
  .... .0.. = ECN-Echo: Not set
  .... ..1. = Urgent: Set
  .... ...0 = Acknowledgment: Not set
  .... .... 1... = Push: Set
  .... .... .0.. = Reset: Not set
  .... .... ..0. = Syn: Not set
  > .... .... ...1 = Fin: Set
    
```

Figura.34 – Flags ativas Scan Xmas

E como no FIN scan:

```

Flags: 0x001 (FIN)
  000. .... = Reserved: Not set
  ...0 .... = Nonce: Not set
  .... 0... = Congestion Window Reduced (CWR): Not set
  .... .0.. = ECN-Echo: Not set
  .... ..0. = Urgent: Not set
  .... ...0 = Acknowledgment: Not set
  .... .... 0... = Push: Not set
  .... .... .0.. = Reset: Not set
  .... .... ..0. = Syn: Not set
  > .... .... ...1 = Fin: Set
    
```

Figura.35 – Flags ativas Scan FIN

5 QUESTÃO 3

O Sistema Operativo utilizado é Linux

```

root@Bruno:~# nmap -O 192.168.56.3
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-27 16:35 WET
Nmap scan report for 192.168.56.3
Host is up (0.00028s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 08:00:27:61:C0 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
29, 0.01s elapsed (1 total hosts)
OS detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 1.83 seconds
  
```

Figura. 36 – nmap -O 192.168.56.3

Utilizando a identificação da *stack fingerprinting* do TCP/IP é possível detetar de forma remota o SO. Ao usar o comando *nmap -o*, o próprio **nmap** envia alguns pacotes TCP/UDP ao *host* remoto e examina quase todos os bits das respostas. Deste modo, o **Nmap** compara os resultados com o banco de dados “*nmap-os-fingerprints*” com mais de 1500 identificações de SO publicamente conhecidas, caso encontre alguma correspondência é exibido os detalhes do SO, SO CPE, geração do SO e ainda o tipo de dispositivo. Neste caso, o **Nmap** conseguiu identificar o SO, mas caso não acontecesse seria possível que nós fornecêssemos as informações do nosso SO à base de dados do **Nmap**, deste modo estaríamos a contribuir para identificações futuras.

É possível utilizar diferentes opções para detetar SO como por exemplo: “*--osscan-limit*” que faz com que o **Nmap** não tente detetar SO contra *hosts* que não correspondam a este critério. Permite economizar algum tempo.

6.2 APACHE HTTP SERVER 2.2.8

CVE-2018-1312 – No Apache httpd 2.2.8, ao gerar um desafio de autenticação HTTP Digest, o nonce enviado para impedir ataques de resposta não foi gerado corretamente usando uma seed pseudoaleatória. Num cluster de servidores usando uma configuração de autenticação Digest comum, as solicitações HTTP podem ser repetidas nos servidores por um invasor sem deteção.

CVE-2017-7679 – No Apache httpd 2.2.x antes de 2.2.33 e 2.4.x antes de 2.4.26, o mod_mime pode ler um byte após o final de um buffer ao enviar um cabeçalho de resposta malicioso do Tipo de Conteúdo.

CVE-2016-8612 – O servidor **HTTP Apache** mod_cluster anterior à versão httpd 2.4.23 é vulnerável a uma Validação de entrada imprópria na lógica de análise de protocolo no balanceador de carga, resultando numa falha de Segmentation Fault no processo httpd.

6.3 UNREALIRCd

CVE-2017-13649 – O UnrealIRCd 4.0.13 e versões anteriores criam um arquivo PID depois de remover privilégios de uma conta “não raiz”, o que permitir que usuários locais eliminem processos arbitrários, aproveitando o acesso a essa conta “não raiz” para modificação do arquivo PID antes que um script raiz execute uma "interrupção" comando “cat / nome do caminho”.

CVE-2016-7144 – A função m_authenticate em modules / m_sasl.c no UnrealIRCd antes de 3.2.10.7 e 4.x antes de 4.0.6 permite que invasores remotos falsifiquem impressões digitais de certificado e, por consequência, efetuam login como outro usuário por meio de um parâmetro criado, AUTHENTICATE.

CVE-2013-7384 – O UnrealIRCd 3.2.10 antes do 3.2.10.2 permite que invasores remotos causem uma negação de serviço (“desreferência” e falha de apontador NULL) por meio de vetores não especificados, relacionados ao SSL.

7 QUESTÃO 5

Ao analisarmos a segurança do sistema metasploitable2, utilizando a ferramenta OpenVAS conseguimos verificar que este encontra-se com enumeras falhas de segurança, de diversos níveis de gravidade. Ao analisarmos mais analiticamente o gráfico abaixo apresentado podemos verificar que o sistema encontra-se com dezassete vulnerabilidades de risco elevado, trinta e três vulnerabilidades de risco medio e apenas duas de baixo risco.

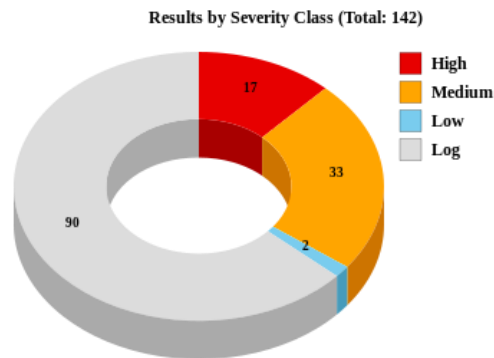


Figura. 38 – Resultados

8 QUESTÃO 6

A seguinte figura é um excerto de um dos alertas gerados pela ferramenta Snort.

```
[**] [1:1564:6] WEB-MISC login.htm access [**]
[Classification: access to a potentially vulnerable web application] [Priority: 2]
12/04-16:35:40.085758 192.168.56.4:60831 -> 192.168.56.6:80
TCP TTL:64 TOS:0x0 ID:6918 IpLen:20 DgmLen:364 DF
***A**** Seq: 0xFA5BEABF Ack: 0x2C2E7498 Win: 0x1AE0 TcpLen: 32
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-1533][Xref => http://www.securityfocus.com/bid/665]
```

Figura. 39 – Excerto do relatório - snort

O modem Dicon LAN ISDN da Eicon Technology permite que um atacante remoto cause uma negação de serviço por meio de uma senha longa no arquivo login.htm em seu serviço HTTP.

Como pode ser observado na figura abaixo, esta vulnerabilidade está classificada com um Base Score de 7.5 High e um subscore de explorabilidade de 10.0.

CVSS v2.0 Severity and Metrics:	
Base Score: 7.5 HIGH	
Vector: (AV:N/AC:L/Au:N/C:P/I:P/A:P)	
Impact Subscore: 6.4	
Exploitability Subscore: 10.0	
<hr/>	
Access Vector (AV): Network	
Access Complexity (AC): Low	
Authentication (AU): None	
Confidentiality (C): Partial	
Integrity (I): Partial	
Availability (A): Partial	
Additional Information:	
Allows unauthorized disclosure of information	
Allows unauthorized modification	
Allows disruption of service	

Figura. 40 – CVSS v2.0 Métricas

O **Snort** identifica este tráfego como anómalo devido ao fato que existe uma falha de segurança associada à tentativa de enviar uma chave longa para o ficheiro login.htm, sendo assim possível causar a negação de serviços. Forçando o modem a reiniciar.

No.	Time	Source	Destination	Protocol	Length	Info
13862	518.122226013	192.168.56.4	192.168.56.6	TCP	74	60831 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=104058433 TSecr=0 WS=1024
13863	518.122361860	192.168.56.6	192.168.56.4	TCP	74	80 → 60831 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=28574 TSecr=104058433 WS=32
13864	518.122383012	192.168.56.4	192.168.56.6	TCP	66	60831 → 80 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=104058433 TSecr=28574
13865	518.122653859	192.168.56.4	192.168.56.6	HTTP	378	GET /tkset/login.html HTTP/1.1
13866	518.122766687	192.168.56.6	192.168.56.4	TCP	66	80 → 60831 [ACK] Seq=1 Ack=313 Win=6880 Len=0 TSval=104058433 TSecr=28574
13867	518.122984919	192.168.56.6	192.168.56.4	HTTP	586	HTTP/1.1 404 Not Found (text/html)
13868	518.122995500	192.168.56.4	192.168.56.6	TCP	66	60831 → 80 [ACK] Seq=313 Ack=521 Win=64512 Len=0 TSval=104058434 TSecr=28574
13869	518.124597840	192.168.56.4	192.168.56.6	TCP	66	60831 → 80 [FIN, ACK] Seq=313 Ack=521 Win=64512 Len=0 TSval=104058435 TSecr=28574
13870	518.124780052	192.168.56.6	192.168.56.4	TCP	66	80 → 60831 [FIN, ACK] Seq=521 Ack=314 Win=6880 Len=0 TSval=28575 TSecr=104058435
13871	518.124795212	192.168.56.4	192.168.56.6	TCP	66	60831 → 80 [ACK] Seq=314 Ack=522 Win=64512 Len=0 TSval=104058436 TSecr=28575

Figura. 41 – Tráfego analisado pela ferramenta **Wireshark**

Esta vulnerabilidade pode ser explorada quando um utilizador remoto conecta-se à porta HTTP e envia uma solicitação GET (como se verifica no tráfego obtido, imagem acima) no formato seguinte:

<http://192.168.56.6/login.htm?password=0123456789012345678901234567890123456789>

2º Evento

```
[**] [1:1748:8] FTP command overflow attempt [**]
[Classification: Generic Protocol Command Decode] [Priority: 3]
12/04-16:33:01.600549 192.168.56.4:42131 -> 192.168.56.6:21
TCP TTL:64 TOS:0x0 ID:58265 IpLen:20 DgmLen:986 DF
***AP*** Seq: 0x128422A3 Ack: 0x98D6B395 Win: 0x3F TcpLen: 32
TCP Options (3) => NOP NOP TS: 103899948 12544
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2002-0606][Xref => http://www.securityfocus.com/bid/4638]
```

Figura. 42 – Excerto do relatório - **Snort**

Buffer overflow no servidor FTP do 3Cdaemon 2.0 permite que atacantes remotos causem uma negação de serviço (falha) e possivelmente executem código arbitrário por meio de comandos longos, como login.

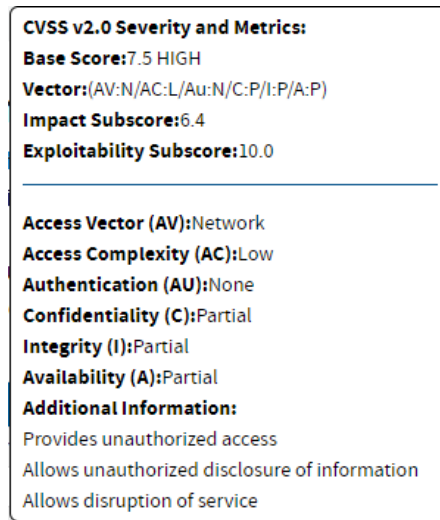


Figura. 43 – CVSS v2.0 Métricas

Como pode ser observado na figura abaixo, esta vulnerabilidade está classificada com um Base Score de 6.5 *High* e um subscore de explorabilidade de 10.0.

Este tráfego “apontado” pelo **Snort** como anómalo, diz-nos que ocorreu diversos pedidos de acesso de modo a ocorrer transferência de ficheiros (Login). E que foi inserido um comando que não foi reconhecido, talvez esta seja o indício de se tratar da vulnerabilidade descrita anteriormente.

No.	Time	Source	Destination	Protocol	Length	Info
12114	359.635820685	192.168.56.4	192.168.56.6	TCP	74	42131 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=103899947 TSecr=0 WS=1024
12115	359.635937179	192.168.56.6	192.168.56.4	TCP	74	21 → 42131 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=12544 TSecr=103899947 WS=32
12116	359.635953330	192.168.56.4	192.168.56.6	TCP	66	42131 → 21 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=103899947 TSecr=12544
12117	359.637061425	192.168.56.6	192.168.56.4	FTP	86	Response: 220 (vsFTPd 2.3.4)
12118	359.637077070	192.168.56.6	192.168.56.4	TCP	66	42131 → 21 [ACK] Seq=1 Ack=21 Win=64512 Len=0 TSval=103899948 TSecr=12544
12119	359.637444434	192.168.56.4	192.168.56.6	FTP	1000	Request: \026\003\001\003\241\001\000\003\235\003\001\347\337\275MrBnaLdHBYW6pXVe438101EDxq4\000\0038\000\000\0...
12120	359.637532486	192.168.56.6	192.168.56.4	TCP	66	21 → 42131 [ACK] Seq=21 Ack=935 Win=7680 Len=0 TSval=12544 TSecr=103899948
12121	359.637585567	192.168.56.6	192.168.56.4	FTP	104	Response: 530 Please login with USER and PASS.
12122	359.637591237	192.168.56.4	192.168.56.6	TCP	66	42131 → 21 [ACK] Seq=935 Ack=59 Win=64512 Len=0 TSval=103899948 TSecr=12544
12123	359.637690242	192.168.56.6	192.168.56.4	FTP	104	Response: 530 Please login with USER and PASS.
12124	359.637694527	192.168.56.4	192.168.56.6	TCP	66	42131 → 21 [ACK] Seq=935 Ack=97 Win=64512 Len=0 TSval=103899949 TSecr=12544
12125	359.637756868	192.168.56.6	192.168.56.4	FTP	104	Response: 530 Please login with USER and PASS.
12126	359.637760567	192.168.56.4	192.168.56.6	TCP	66	42131 → 21 [ACK] Seq=935 Ack=135 Win=64512 Len=0 TSval=103899949 TSecr=12544
12127	361.638636943	192.168.56.4	192.168.56.6	TCP	66	42131 → 21 [FIN, ACK] Seq=935 Ack=135 Win=64512 Len=0 TSval=103901950 TSecr=12544
12129	361.639045696	192.168.56.6	192.168.56.4	FTP	76	Response: 500 OOPS:
12130	361.639082599	192.168.56.4	192.168.56.6	TCP	54	42131 → 21 [RST] Seq=936 Win=0 Len=0

Figura. 44 – Tráfego analisado pela ferramenta **Wireshark**

Foram verificadas diversas formas de explorar esta vulnerabilidade, o envio de uma quantidade absurda de dados ao servidor *ftp* pode desencadear uma condição de *overflow* com base na pilha. Mas neste caso, o envio de dados aleatórios pode causar falha no aplicativo. Todas estas formas descritas podem permitir a um utilizador mal-intencionado executar código no servidor.

9 QUESTÃO 7

Ao comparar os resultados obtidos é possível observar que no Scan Report do **OpenVAS**, as atualizações de segurança do fornecedor não são confiáveis; as substituições estão ativadas; quando um resultado tem uma substituição, este relatório usa a ameaça de substituição. Informações sobre substituições estão incluídas no Scan Report. Em comparação com o **Snort**, o **OpenVAS** não mostra detalhes de todos os problemas encontrados, é apenas listado os hosts que produziram problemas; Problemas classificados com nível de ameaça “Log” não são exibidos; Os problemas com nível de ameaça “Debug” e “False Positive” também não são exibidos neste Scan Report. Algo a salientar também é que neste Scan Report os resultados com uma QoD inferior a 70 não são mostrados.

Host	High	Medium	Low	Log	False Positive
192.168.56.6	3	21	2	0	0
Total: 1	3	21	2	0	0


Figura. 45 – Tabela Scan Report

Essencialmente, o bom de fazer uso do **Nessus/OpenVAS** é que, diferente dos restantes scanners de segurança de rede tradicionais que apenas se focam nos serviços que escutam na rede, estes também se concentram nos hosts locais. Deste modo, ele pode até determinar se há patches ausentes, em que sistemas estão sendo executados (Windows, Unix...). Além disso, o Scanner do **OpenVAS** executa os testes de vulnerabilidade de rede que são servidos com atualizações diárias, fornecidas principalmente pelo Feed NVT **OpenVAS**.



Em relação ao **Snort** este pode detetar por anomalias em conformidade com os resultados da rotina da rede, fazendo assim um perfil. Deste modo caso surja algum evento discrepante, o sistema identifica a anomalia como uma potencial intrusão. Também pode ser baseado em assinaturas, isto é, ao configurar previamente o sistema, o **Snort** vai tomar como alvo, atividades que se enquadram como maliciosas. Só que desta forma é necessário que tenhamos um conhecimento amplo das ameaças virtuais existentes. Por último a detenção baseada em comportamento identifica os ataques em comparação do comportamento do utilizador. Este método permite ao administrador definir quais os indícios que possam indicar uma possível invasão à rede.

10 QUESTÃO 8

10.1 HTTP TRACK/TRACE METHODS ALLOWED



Result: Cleartext Transmission of Sensitive Information via HTTP

Vulnerability	Severity	QoD	Host	Location	Actions
Cleartext Transmission of Sensitive Information via HTTP Summary The host / application transmits sensitive information (username, passwords) in cleartext via HTTP. Vulnerability Detection Result The following input fields were identified (URL:input name): http://192.168.56.6/phpMyAdmin/:pma_password http://192.168.56.6/phpMyAdmin/70-A:pma_password http://192.168.56.6/tikiwiki/tiki-install.php:pass http://192.168.56.6/tikiwiki/bin/view/TWiki/TWikiUserAuthentication:oldpassword Impact An attacker could use this situation to compromise or eavesdrop on the HTTP communication between the client and the server using a man-in-the-middle attack to get access to sensitive data like usernames or passwords. Solution Solution type: Workaround Enforce the transmission of sensitive data via an encrypted SSL/TLS connection. Additionally make sure the host / application is redirecting all users to the secured SSL/TLS connection before allowing to input sensitive data into the mentioned functions.	4.9 (Medium)	80%	192.168.56.6	80/tcp	 

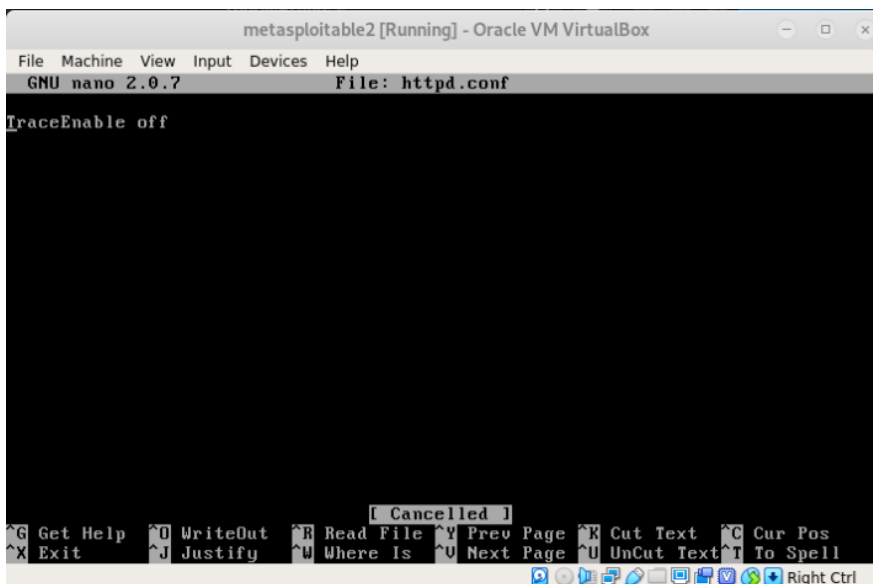
ID: fc149516-924d-4bfb-93bb-0a4e015d7ea1
 Created: Wed Dec 4 16:44:39 2019
 Modified: Wed Dec 4 16:44:39 2019
 Owner: admin

Figura. 46 – Resultado da vulnerabilidade HTTP TRACK/TRACE Methods Allowed

O host/aplicação transmitem informação sensível como por exemplo, usernames, passwords em texto limpo via HTTP.

O TRACE é ativado por padrão em uma instalação apache. Existem duas maneiras de corrigir. O primeiro foi adicionar a diretiva *TraceEnable* ao httpd.conf e defini-lo com o valor OFF.

A segunda forma de corrigir é criar uma regra “*mod_rewrite*” que desabilitará os métodos http, que segundo a nossa pesquisa é bastante popular e funciona com qualquer versão do apache que suporte *mod_rewrite*.



```

metasploitable2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
GNU nano 2.0.7 File: httpd.conf


TraceEnable off

[Cancelled]
Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page UnCut Text To Spell
  
```


Figura. 47 – Solução no httpd.conf

10.2 REXECD SERVICE DETECTION

Esta vulnerabilidade consiste basicamente na deteção do estado do serviço rexecd, onde o host remoto está a correr-lo. Assim sendo, o servidor rexec não está permitindo conexões deste host.




Result: rexec Passwordless / Unencrypted Cleartext Login

Vulnerability	Severity	QoD	Host	Location	Actions
rexec Passwordless / Unencrypted Cleartext Login	10.0 (High)	80%	192.168.56.6	512/tcp	

Summary
This remote host is running a rexec service.

Vulnerability Detection Result
The rexec service is not allowing connections from this host.

Solution
Solution type:  Mitigation
Disable the rexec service and use alternatives like SSH instead.

ID: f6bc6a55-ce98-4f40-a4f6-7aeb5186f9de
Created: Wed Dec 4 16:39:25 2019
Modified: Wed Dec 4 16:39:25 2019
Owner: admin

Figura. 48 – Resultado da vulnerabilidade rexec Passwordless

A solução enquadrava-se na desabilitação do serviço rexec e usar SSH em vez de rexec. Usando o comando `nmap -sV 192.168.56.6`, podemos verificar que a porta/serviço já não se encontram.

```

root@bruno:~# nmap -sV 192.168.56.6
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-05 17:09 WET
Nmap scan report for 192.168.56.6
Host is up (0.00023s latency).
Not shown: 983 closed ports
PORT      STATE SERVICE        VERSION
22/tcp    open  ssh            OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
25/tcp    open  smtp           Postfix smtpd
53/tcp    open  domain         ISC BIND 9.4.2
80/tcp    open  http           Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind        2 (RPC #100000)
139/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
1099/tcp  open  java-rmi       GNU Classpath grmiregistry
2049/tcp  open  nfs            2-4 (RPC #100003)
2121/tcp  open  ftp           ProFTPD 1.3.1
3306/tcp  open  mysql          MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql     PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc            VNC (protocol 3.3)
6000/tcp  open  x11            (access denied)
6667/tcp  open  irc            UnrealIRCd
8009/tcp  open  ajp13         Apache Jserv (Protocol v1.3)
8180/tcp  open  http          Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:61:F1:C0 (Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Linux, Unix; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.76 seconds
  
```

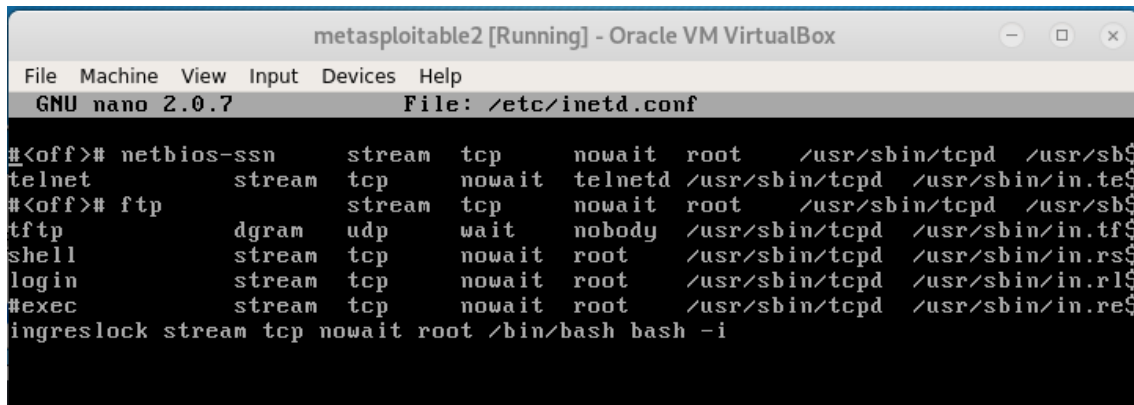
Figura. 49 – `nmap -sV 192.168.56.6`

```

msfadmin@metasploitable:~$ sudo kill $(sudo lsof -t -i:512)
msfadmin@metasploitable:~$
  
```

Figura. 50 – Solução provisória

Algo que achamos curioso, foi que ao terminar o processo acima referido na porta 512 esta por alguma razão o mesmo efeito foi observado na porta 1524. Consequentemente a vulnerabilidade VNC Server ‘password’ Password deixou de estar presente. Mas com o `sudo kill` verificamos que era apenas provisório.



```

metasploitable2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
GNU nano 2.0.7 File: /etc/inetd.conf

#<off># netbios-ssn      stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/tcpd
telnet                stream  tcp    nowait  telnetd  /usr/sbin/tcpd  /usr/sbin/in.telnetd
#<off># ftp             stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.ftpd
tftp                  dgram  udp    wait    nobody   /usr/sbin/tcpd  /usr/sbin/in.tftpd
shell                 stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rshd
login                 stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rlogind
#exec                 stream  tcp    nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rexecd
ingreslock            stream  tcp    nowait  root    /bin/bash      bash -i

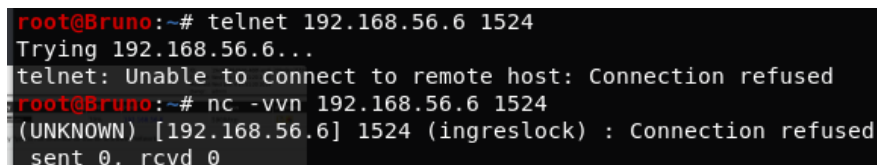
```

Figura. 51 – Solução para a vulnerabilidade rexec

Ao comentar a linha exec no ficheiro inetd.conf, observada na imagem acima, conseguimos resolver esta vulnerabilidade.

10.3 BIND SHELL BACKDOOR DETECTION

Como podemos ver na figura abaixo, ao executarmos os comandos telnet 192.168.56.6 1524, onde este tenta efetuar uma ligação que é finalizada com uma conexão refusada para a porta indicada pelo nmap. O comando a seguir nc -vvn 192.168.56.6 1524 também retorna a mesma informação, conexão refusada.



```

root@Bruno:~# telnet 192.168.56.6 1524
Trying 192.168.56.6...
telnet: Unable to connect to remote host: Connection refused
root@Bruno:~# nc -vvn 192.168.56.6 1524
(UNKNOWN) [192.168.56.6] 1524 (ingreslock) : Connection refused
sent 0, rcvd 0


```

Figura. 52

Deste modo, entendemos que a vulnerabilidade se encontra resolvida.

10.4 VNC SERVER 'PASSWORD' PASSWORD

Nesta vulnerabilidade o servidor VNC em execução no host remoto é protegido com uma senha fraca. Um invasor remoto não autenticado pode explorar isso para assumir o controlo do sistema.



Result: VNC Brute Force Login

ID: 0bfc01d4-3f39-42e4-8d70-b66b365e964c
Created: Thu Dec 5 18:58:01 2019
Modified: Thu Dec 5 18:58:01 2019
Owner: admin







Vulnerability		Severity		QoD	Host	Location	Actions
VNC Brute Force Login		9.9 (High)		95%	192.168.56.6	5900/tcp	 
<h2>Summary</h2> <p>Try to log in with given passwords via VNC protocol.</p>							
<h2>Vulnerability Detection Result</h2> <p>It was possible to connect to the VNC server with the password: password</p>							
<h2>Solution</h2> <p>Solution type:  Mitigation</p> <p>Change the password to something hard to guess or enable password protection at all.</p>							

Figura. 53 – Resultado da vulnerabilidade VNC Brute Force Login.

Para mitigar a vulnerabilidade do VNS, nós modificamos a password para uma mais forte, utilizando todo o tipo de caracteres (Maiúsculas, minúsculas, números, pontuação).

Vulnerability	Severity	QoD	Host
OS End Of Life Detection	10.0 (High)	80%	192.168.56.6
TWiki XSS and Command Execution Vulnerabilities	10.0 (High)	80%	192.168.56.6
Java RMI Server Insecure Default Configuration Remote Code Execution Vulnerability	10.0 (High)	95%	192.168.56.6
Distributed Ruby (dRuby/DRb) Multiple Remote Code Execution Vulnerabilities	10.0 (High)	99%	192.168.56.6
DistCC Remote Code Execution Vulnerability	9.3 (High)	99%	192.168.56.6
MySQL / MariaDB weak password	9.0 (High)	95%	192.168.56.6
PostgreSQL weak password	9.0 (High)	99%	192.168.56.6
rlogin Passwordless / Unencrypted Cleartext Login	7.5 (High)	70%	192.168.56.6

Figura.54 – Vulnerabilidades solucionadas

Após realizarmos as tarefas dos passos anteriores para solucionar as vulnerabilidades indicadas no enunciado, voltamos a verificar as vulnerabilidades ainda presentes e com isto concluímos que as vulnerabilidades indicadas foram solucionadas com sucesso.

11 CONCLUSÃO

Neste trabalho tivemos a oportunidade de trabalhar com diferentes ferramentas e aplicações desenhadas para realizar testes de penetração, sendo que o principal objetivo deste trabalho seria experimentá-las e pôr em prática as potencialidades destas ferramentas na análise ao sistema Metasploitable2.

O objetivo do trabalho era analisar o sistema Metasploitable2, sendo que depois teríamos de identificar riscos de segurança que este sistema propositadamente continha, e ao mesmo tempo, descrever as funcionalidades das várias ferramentas que eram postas em prática. Por último teríamos de descobrir algumas das falhas que o sistema continha, e tentar resolvê-las.

Ao longo da realização do trabalho fomos identificando e descrevendo de forma objetiva o que achávamos pertinente para a resposta das várias perguntas propostas. É do nosso entender também que respondemos a todas estas questões da melhor forma que sabíamos, e como tal acreditamos que o objetivo do trabalho foi cumprido com sucesso.

Por último, gostaríamos de acrescentar que a realização deste trabalho permitiu-nos adquirir maiores competências e entendimento na área de segurança, como também entender melhor o que é lecionado na aula, bem como um melhor entendimento do que é e do que se trata testes de penetração.

12 BIBLIOGRAFIA

(s.d.). Obtido de OpenVAS: <http://openvas.org/>

(2019). Obtido de guru99: <https://www.guru99.com/wireshark-passwords-sniffer.html>

Alassouli, D. H. (2018). *Part 7: Sniffer and Phishing Hacking*. Kindle.

Rahalkar, S. (2018). *Quick Start Guide to Penetration Testing: With NMAP, OpenVAS and Metasploit*. APress.

Snort. (s.d.). Obtido de Snort: <https://www.snort.org/>

Obtido de Nmap: <https://nmap.org/book/man.html>