

Universidade do Minho  
Mestrado Integrado em Engenharia Informática  
SSI-Trabalho Prático 2  
Grupo 1

João Bernardo Freitas a74814  
Rui Pereira pg42853

20 de Dezembro 2020

# Contents

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Parte A</b>	<b>4</b>
2.1	bancomontepio.pt . . . . .	4
2.2	porta188.com . . . . .	8
<b>3</b>	<b>Parte B</b>	<b>10</b>
3.1	Q1 . . . . .	13
3.2	Q2 . . . . .	15
3.3	Q3 . . . . .	18
3.3.1	Evento 1 . . . . .	19
3.3.2	Evento 2 . . . . .	20
3.4	Q4 . . . . .	21
3.5	Q5 . . . . .	21
3.5.1	Scan Final . . . . .	24
<b>4</b>	<b>Conclusão</b>	<b>25</b>

# 1 Introdução

Neste relatório vamos descrever os passos que seguimos para realizar o trabalho prático nº2 da cadeira de Segurança de Sistemas Informáticos.

Este trabalho prático centra-se em duas partes distintas:

- Colecta Passiva de Informações
- Colecta Activa de informações

Mais especificamente, para a primeira parte escolhemos dois endereços de empresas reais e através de métodos passivos vamos tentar obter informações que possam ser utilizadas para ataques. Na segunda parte vamos preparar duas máquinas virtuais, uma com a distribuição *Kali Linux* e a outra com **Metasploitable**, com o objectivo de atacarmos a segunda máquina virtual através da primeira.

## 2 Parte A

Esta primeira parte tem como objectivo a colecta passiva de informações de dois domínios utilizados por empresas sendo que as empresas que escolhemos foram o Montepio e a Portal88. Para completarmos este objectivo vamos utilizar a ferramenta **whois** para descobrir informações relativas a quem gere um certo domínio/endereço, a ferramenta **nslookup** para descobrirmos os endereços associados a cada domínio e por fim a internet para, por exemplo, através das propostas de emprego descobrir qual a infraestrutura interna utilizada por dita empresa.

### 2.1 bancomontepio.pt

Segue então a análise à empresa Montepio, que é uma das maiores instituições bancárias em Portugal e como tal, esperamos que tenha maior rigor em termos de segurança e privacidade, começando então pelo comando **whois bancomontepio.pt**.

```
joaob@PC-JB:/mnt/c/Users/joaob/Desktop$ whois bancomontepio.pt
Domain: bancomontepio.pt
Domain Status: Registered
Creation Date: 04/07/2018 17:59:48
Expiration Date: 03/07/2023 17:59:48
Owner Name: Caixa Economica Montepio Geral
Owner Address: Rua Aurea, 219-241
Owner Locality: Lisboa
Owner ZipCode: 1100-062
Owner Locality ZipCode: Lisboa
Owner Country Code: PT
Owner Email: dmcacador@montepio.pt,miguel.delgado@montepio.pt
Admin Name: Caixa Economica Montepio Geral
Admin Address: Rua Aurea, 219-241
Admin Locality: Lisboa
Admin ZipCode: 1100-062
Admin Locality ZipCode: Lisboa
Admin Country Code: PT
Admin Email: dmcacador@montepio.pt,miguel.delgado@montepio.pt
Name Server: ns4.bancomontepio.pt | IPv4: 194.65.117.193 and IPv6:
Name Server: ns5.bancomontepio.pt | IPv4: 85.88.143.129 and IPv6:
Name Server: ns2.bancomontepio.pt | IPv4: 195.47.200.12 and IPv6:
Name Server: ns1.bancomontepio.pt | IPv4: 195.47.200.11 and IPv6:
```

Figure 1: Resultado do comando **whois bancomontepio.pt**

Através deste comando podemos já obter algumas informações relativas ao Montepio, nomeadamente localizações físicas bem como emails e nomes dos funcionários que gerem este domínio. Neste caso podemos através de uma pesquisa no *Google* confirmar que o Miguel Delgado é actualmente um responsável regional do banco Montepio enquanto o Daniel Caçador é um **IT Security Manager** do Montepio. Através de apenas um comando já temos eventualmente dois possíveis funcionários por onde começar um ataque.

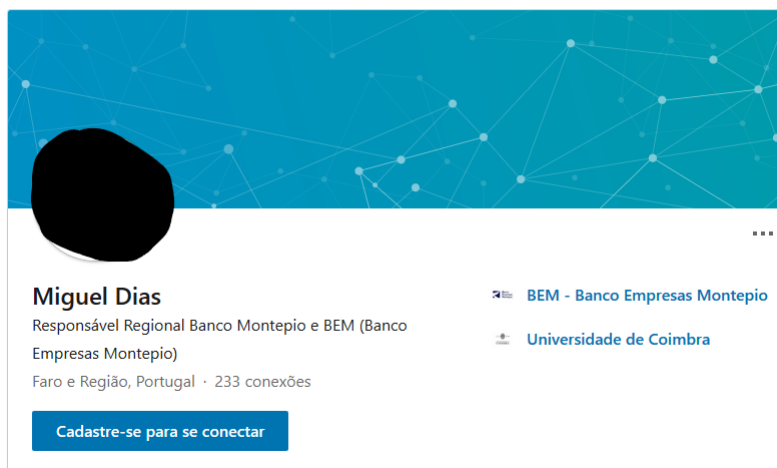


Figure 2: LinkedIn Miguel Delgado

---

## Daniel Caçador

---

**IT Security Manager @ Caixa Económica Montepio Geral**

Figure 3: Posição do Daniel Caçador na empresa

De seguida tentamos descobrir o endereço associado a esse domínio através do comando **nslookup bancomontepio.pt**.

```
joaob@PC-JB:/mnt/c/Users/joaob/Desktop$ nslookup bancomontepio.pt
Server:      172.25.192.1
Address:     172.25.192.1#53

Non-authoritative answer:
Name:   bancomontepio.pt
Address: 213.30.53.202
Name:   bancomontepio.pt
Address: 88.157.205.202
```

Figure 4: Resultado do comando **nslookup bancomontepio.pt**

Podemos agora tentar novamente o comando **whois 213.30.53.202** para verificar se obtemos novas informações.

```
person:      Rui Barbosa
address:     Les Palace: Rua Julio Dinis 158160
address:     4050318 Porto
address:     Portugal
phone:       +351924475389
mnt-by:      AS12353-MNT
nic-hdl:     RB22797-RIPE
created:     2017-05-03T09:13:38Z
last-modified: 2017-05-03T09:13:38Z
source:      RIPE # Filtered
```

Figure 5: Resultado do comando **whois 213.30.53.202**

Como podemos verificar temos um novo nome, sendo que este também inclui o número de telemóvel dele. Para confirmar decidimos novamente ir ao **LinkedIn**.

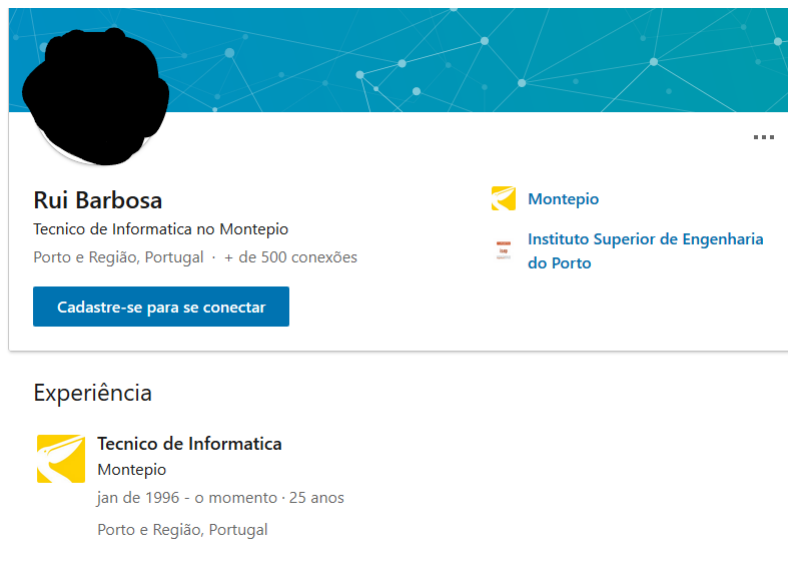


Figure 6: LinkedIn Rui Barbosa

Através destes métodos simples de colecta de informações já temos três nomes, um número de telemóvel, dois LinkedIns, todos disponíveis livremente na internet sendo que todos são eventuais vulnerabilidades.

Por fim fomos procurar ofertas de emprego que nos pudessem dar pistas para identificar a infraestrutura utilizada pelo Montepio, mas, felizmente para este, não foi possível encontrar nada de relevante, o que mostra que eles tiveram em atenção ao facto de que atacantes podem obter informações da infraestrutura utilizada que pode então ser uma vulnerabilidade.

## 2.2 porta188.com

Para a segunda empresa escolhemos a porta188.com, que é uma loja de roupa desportiva de Joane. Para a colecta de informações decidimos seguir o mesmo método utilizado em cima, começando então pelo comando **whois porta188.com**.

```
Domain Name: PORTA188.COM
Registry Domain ID: 1997283426_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.register.it
Registrar URL: http://we.register.it
Updated Date: 2020-02-28T00:00:00Z
Creation Date: 2020-03-08T00:00:00Z
Registrar Registration Expiration Date: 2022-01-25T00:00:00Z
Registrar: REGISTER S.P.A.
Registrar IANA ID: 168
Registrar Abuse Contact Email: abuse@register.it
Registrar Abuse Contact Phone: +39.05520021555
Reseller:
Domain Status: ok https://icann.org/epp#ok
Registry Registrant ID:
Registrant Name: REDACTED FOR PRIVACY
Registrant Organization: 3DS, Lda
Registrant Street: REDACTED FOR PRIVACY
Registrant City: REDACTED FOR PRIVACY
Registrant State/Province: BR
Registrant Postal Code: REDACTED FOR PRIVACY
Registrant Country: PT
Registrant Phone: REDACTED.FORPRIVACY
Registrant Phone Ext:
Registrant Fax: REDACTED.FORPRIVACY
```

Figure 7: Resultado do comando **whois porta188.com**

Como podemos verificar este comando não nos deu muita informação relativa ao domínio, sendo que todas as informações que seriam relevantes estão **redacted**.



Através do comando **nslookup** conseguimos novamente obter o endereço associado a este domínio.

```
joaob@PC-JB:/mnt/c/Users/joaob/Desktop$ nslookup porta188.com
Server:      172.25.192.1
Address:     172.25.192.1#53

Non-authoritative answer:
Name:   porta188.com
Address: 213.229.86.121
```

Figure 8: Resultado do comando **nslookup porta188.com**

Executando novamente o comando **whois** com o endereço obtemos o seguinte resultado.

```
role:          AS29550 Operators
address:       Simply Transit
address:       Unit 2
address:       Smallmead Road
address:       Reading
address:       Berkshire
address:       RG2 0QS
remarks:       For abuse please contact abuse@as29550.net
phone:         +44 (0)1628 777730
admin-c:       DD6881-RIPE
admin-c:       AJB5-RIPE
tech-c:        DD6881-RIPE
tech-c:        AJB5-RIPE
mnt-by:        AS29550-MNT
nic-hdl:       A0904-RIPE
created:       2010-03-25T17:02:11Z
last-modified: 2016-07-21T13:53:37Z
source:        RIPE # Filtered
abuse-mailbox: abuse@as29550.net
```

Figure 9: Resultado do comando **whois 213.229.86.221**

Felizmente para os donos desta empresa também não é possível obter muitas informações passivamente que possam ser utilizadas para ataques.

Algo que não foi surpreendente mas desapontante foi o facto de que a empresa com menos recursos é mais protegida que um dos maiores bancos nacionais, no que toca a informações que se podem obter na internet.

### 3 Parte B

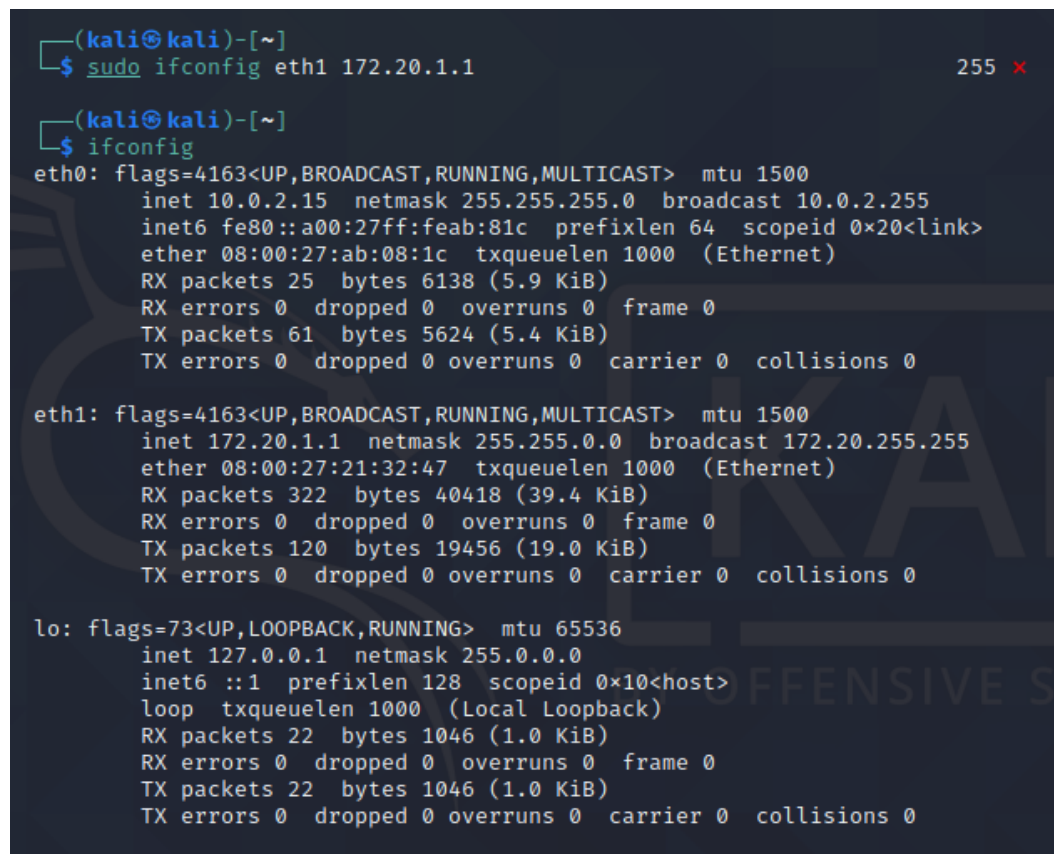
#### Setup

Antes de iniciarmos a resolução da Parte B é necessário preparar um ambiente de teste isolado. Para tal decidimos usar **VirtualBox** para configurar as duas imagens, uma com **Ubuntu** e outra com **Metaxploitable3**.

Para assegurarmos que o ambiente de teste é isolado configuramos ambas as máquinas de forma a que ambas possam comunicar através de uma rede interna e que máquina com sistema Auditor tenha ligação á rede.

Isso pode ser feito através da configuração correta das interfaces na **VirtualBox** e através de um comando em cada sistema.

No sistema auditor, a correr **Ubuntu** o comando a executar é **sudo ifconfig <interface> 172.20.1.1**.



```
(kali㉿kali)-[~]
$ sudo ifconfig eth1 172.20.1.1

(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:feab:81c prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:ab:08:1c txqueuelen 1000 (Ethernet)
    RX packets 25 bytes 6138 (5.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 61 bytes 5624 (5.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.20.1.1 netmask 255.255.0.0 broadcast 172.20.255.255
    ether 08:00:27:21:32:47 txqueuelen 1000 (Ethernet)
    RX packets 322 bytes 40418 (39.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 120 bytes 19456 (19.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 22 bytes 1046 (1.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 22 bytes 1046 (1.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 10: Configurar interface no sistema Auditor

*NOTA: Nesta fase inicial começamos por utilizar o **Kali Linux** como sistema Auditor mas da **Q1** para a frente mudamos para **Ubuntu***

No sistema alvo, o comando a executar é diferente visto que é uma máquina com **Windows**.

```
C:\Users\vagrant>netsh int ip set address "local area connection" static 172.20.1.2 255.255.255.0 172.20.1.1

C:\Users\vagrant>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::94f9:76a2:6c63:f67b%11
    IPv4 Address. . . . . : 172.20.1.2
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 172.20.1.1

Tunnel adapter Local Area Connection* 9:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :
```

Figure 11: Configurar interface no sistema Alvo

Por fim podemos confirmar que as máquinas conseguem comunicar uma com a outra através do comando **ping** e podemos também confirmar que a máquina alvo não tem ligação externa.

```
C:\Users\vagrant>ping 172.20.1.1

Pinging 172.20.1.1 with 32 bytes of data:
Reply from 172.20.1.1: bytes=32 time=1ms TTL=64
Reply from 172.20.1.1: bytes=32 time=1ms TTL=64

Ping statistics for 172.20.1.1:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
Control-C
^C
C:\Users\vagrant>ping www.google.pt
Ping request could not find host www.google.pt. Please check the name and try again.
```

Figure 12: Comunicação Alvo -> Auditor

```

(kali@kali)-[~]
$ ping 172.20.1.1
PING 172.20.1.1 (172.20.1.1) 56(84) bytes of data.
64 bytes from 172.20.1.1: icmp_seq=1 ttl=64 time=0.086 ms
64 bytes from 172.20.1.1: icmp_seq=2 ttl=64 time=0.104 ms
^C
--- 172.20.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1030ms
rtt min/avg/max/mdev = 0.086/0.095/0.104/0.009 ms

(kali@kali)-[~]
$ ping 172.20.1.2
PING 172.20.1.2 (172.20.1.2) 56(84) bytes of data.
64 bytes from 172.20.1.2: icmp_seq=1 ttl=128 time=1.69 ms
64 bytes from 172.20.1.2: icmp_seq=2 ttl=128 time=0.942 ms
^C
--- 172.20.1.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.942/1.316/1.690/0.374 ms

(kali@kali)-[~]
$ ping www.google.pt
PING www.google.pt (216.58.211.35) 56(84) bytes of data.
64 bytes from mad08s05-in-f3.1e100.net (216.58.211.35): icmp_seq=1 ttl=113 time=
30.5 ms
64 bytes from muc03s14-in-f35.1e100.net (216.58.211.35): icmp_seq=2 ttl=113 time
=18.6 ms
^C
--- www.google.pt ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 18.645/24.580/30.516/5.935 ms

```

Figure 13: Comunicação Auditor -> Alvo

Por fim também instalamos e preparamos todas as aplicações que vamos utilizar na máquina auditora para varrer a máquina alvo, nomeadamente o **Snort**, **Wireshark** e **Nessus**, podendo finalmente dar o ambiente de testes como preparado.

### 3.1 Q1

Através do comando **nmap -sSU -O 172.20.1.2** podemos verificar quais as portas que estão abertas bem como os serviços que estão a ser executados. Se algum destes serviços tiver vulnerabilidades pode ser um ponto de entrada neste sistema.

Para verificarmos as vulnerabilidades destes serviços fomos procurar **CVE-Common Vulnerability and Exposures** de cada um.

```
joao@joao:~$ sudo nmap -sSU -O 172.20.1.2
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-12 09:49 WET
Nmap scan report for 172.20.1.2
Host is up (0.13s latency).
Not shown: 1983 closed ports
PORT      STATE      SERVICE
22/tcp    open       ssh
135/tcp    open       msrpc
139/tcp    open       netbios-ssn
445/tcp    open       microsoft-ds
8383/tcp   open       m2mservices
49152/tcp  open       unknown
49153/tcp  open       unknown
49154/tcp  open       unknown
49155/tcp  open       unknown
49157/tcp  open       unknown
49158/tcp  open       unknown
49160/tcp  open       unknown
137/udp    open       netbios-ns
138/udp    open|filtered netbios-dgm
500/udp    open|filtered isakmp
4500/udp   open|filtered nat-t-ike
5355/udp    open|filtered llmnr
MAC Address: 08:00:27:D5:05:0B (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Microsoft Windows 7
OS CPE: cpe:/o:microsoft:windows_7::sp1
OS details: Microsoft Windows 7 SP1
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1582.04 seconds
```

Figure 14: Nmap do sistema Alvo

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	<a href="#">CVE-1999-0248</a>				1999-01-01	2008-09-05	10.0	None	Remote	Low	Not required	Complete	Complete	Complete
A race condition in the authentication agent mechanism of sshd 1.2.17 allows an attacker to steal another user's credentials.														
2	<a href="#">CVE-2001-0144</a>			Exec Code Overflow	2001-03-12	2018-05-02	10.0	Admin	Remote	Low	Not required	Complete	Complete	Complete
CORE SDI SSH1 CRC-32 compensation attack detector allows remote attackers to execute arbitrary commands on an SSH server or client via an integer overflow.														
3	<a href="#">CVE-2002-1645</a>			Exec Code Overflow	2002-11-25	2017-07-10	10.0	Admin	Remote	Low	Not required	Complete	Complete	Complete
Buffer overflow in the URL catcher feature for SSH Secure Shell for Workstations client 3.1 to 3.2.0 allows remote attackers to execute arbitrary code via a long URL.														

Figure 15: Vulnerabilidades do *ssh*

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	<a href="#">CVE-2018-8407</a>	200		+Info	2018-11-13	2018-12-13	2.1	None	Local	Low	Not required	Partial	None	None
An information disclosure vulnerability exists when "Kernel Remote Procedure Call Provider" driver improperly initializes objects in memory, aka "MSRPC Information Disclosure Vulnerability." This affects Windows 7, Windows Server 2012 R2, Windows RT 8.1, Windows Server 2008, Windows Server 2019, Windows Server 2012, Windows 8.1, Windows Server 2016, Windows Server 2008 R2, Windows 10, Windows 10 Servers.														

Figure 16: Vulnerabilidades do *msrpc*

<b>CVE-2020-10745</b>	A flaw was found in all Samba versions before 4.10.17, before 4.11.11 and before 4.12.4 in the way it processed NetBios over TCP/IP. This flaw allows a remote attacker could to cause the Samba server to consume excessive CPU use, resulting in a denial of service. This highest threat from this vulnerability is to system availability.	V3.1: <b>7.5 HIGH</b> V2.0: <b>7.5 HIGH</b>
<b>Published:</b>	Julho 07, 2020; 10:15:11 AM -0400	
<b>CVE-2020-13159</b>	Artica Proxy before 4.30.000000 Community Edition allows OS command injection via the Netbios name, Server domain name, dhclient_mac, Hostname, or Alias field. NOTE: this may overlap CVE-2020-10818.	V3.1: <b>9.0 CRITICAL</b> V2.0: <b>10.0 HIGH</b>
<b>Published:</b>	Junho 22, 2020; 2:15:11 PM -0400	
<b>CVE-2018-6766</b>	Swisscom TVMediaHelper 1.1.0.50 contains a vulnerability that could allow an unauthenticated, remote attacker to execute arbitrary code on the targeted system. This vulnerability exists due to the way .dll files are loaded. It allows an attacker to load a .dll of the attacker's choosing that could execute arbitrary code without the user's knowledge. The specific flaw exists within the handling of several DLLs (dwmapl.dll, PROPSYS.dll, cscapi.dll, SAMLIB.dll, netbios.dll, winhttp.dll, security.dll, ntmarta.dll, WindowsCodecs.dll, apphelp.dll) loaded by the SwisscomTVMediaHelper.exe process.	V3.0: <b>7.5 HIGH</b> V2.0: <b>7.2 HIGH</b>
<b>Published:</b>	Março 27, 2018; 1:29:00 PM -0400	

Figure 17: Vulnerabilidades do *netbios*

<b>CVE-2002-0597</b>	LANMAN service on Microsoft Windows 2000 allows remote attackers to cause a denial of service (CPU/memory exhaustion) via a stream of malformed data to microsoft-ds port 445.	V3.x:(not available) V2.0: <b>5.0 MEDIUM</b>
<b>Published:</b>	Junho 18, 2002; 12:00:00 AM -0400	

Figure 18: Vulnerabilidades do *microsoft-ds*

<b>CVE-2017-13690</b>	The IKEv2 parser in tcpdump before 4.9.2 has a buffer over-read in print-isakmp.c, several functions.	V3.0: <b>9.9 CRITICAL</b> V2.0: <b>7.5 HIGH</b>
<b>Published:</b>	Setembro 14, 2017; 2:29:03 AM -0400	
<b>CVE-2017-13689</b>	The IKEv1 parser in tcpdump before 4.9.2 has a buffer over-read in print-isakmp.c:ikev1_id_print().	V3.0: <b>9.9 CRITICAL</b> V2.0: <b>7.5 HIGH</b>
<b>Published:</b>	Setembro 14, 2017; 2:29:03 AM -0400	
<b>CVE-2017-13039</b>	The ISAKMP parser in tcpdump before 4.9.2 has a buffer over-read in print-isakmp.c, several functions.	V3.0: <b>9.9 CRITICAL</b> V2.0: <b>7.5 HIGH</b>
<b>Published:</b>	Setembro 14, 2017; 2:29:02 AM -0400	

Figure 19: Vulnerabilidades do *isakmp*

<b>CVE-2020-17467</b>	An issue was discovered in FNET through 4.6.4. The code for processing the hostname from an LLMNR request doesn't check for '\0' termination. Therefore, the deduced length of the hostname doesn't reflect the correct length of the actual data. This may lead to Information Disclosure in _fnet_llmnr_poll in fnet_llmnr.c during a response to a malicious request of the DNS class IN.	V3.x:(not available) V2.0:(not available)
<b>Published:</b> Dezembro 11, 2020; 6:15:13 PM -0500		
<b>CVE-2018-16525</b>	Amazon Web Services (AWS) FreeRTOS through 1.3.1, FreeRTOS up to V10.0.1 (with FreeRTOS+TCP), and WITTENSTEIN WHIS Connect middleware TCP/IP component allow remote attackers to execute arbitrary code or leak information because of a Buffer Overflow during parsing of DNS\LLMNR packets in prvParseDNSReply.	V3.0: <b>8.1 HIGH</b> V2.0: <b>6.8 MEDIUM</b>
<b>Published:</b> Dezembro 06, 2018; 6:29:00 PM -0500		
<b>CVE-2011-0657</b>	DNSAPI.dll in the DNS client in Microsoft Windows XP SP2 and SP3, Windows Server 2003 SP2, Windows Vista SP1 and SP2, Windows Server 2008 Gold, SP2, R2, and R2 SP1, and Windows 7 Gold and SP1 does not properly process DNS queries, which allows remote attackers to execute arbitrary code via (1) a crafted LLMNR broadcast query or (2) a crafted application, aka "DNS Query Vulnerability."	V3.x:(not available) V2.0: <b>7.5 HIGH</b>
<b>Published:</b> Abril 13, 2011; 2:55:01 PM -0400		

Figure 20: Vulnerabilidades do *llmnr*

## 3.2 Q2

Para executarmos uma varredura activa com **Nessus** temos primeiro que dar um alvo, ou lista de alvos, a varrer:

The screenshot shows the Nessus configuration interface for a new scan. The sidebar on the left has a 'BASIC' section expanded, showing 'General', 'Schedule', and 'Notifications'. The main configuration area has the following fields:

- Name:** Scan Inicial
- Description:** Scan inicial
- Folder:** My Scans (dropdown menu)
- Targets:** 172.20.1.2
- Buttons:** Upload Targets, Add File
- Footer:** Save, Cancel

Figure 21: Definições de varredura

Após fim da varredura obtemos o seguinte resultado:

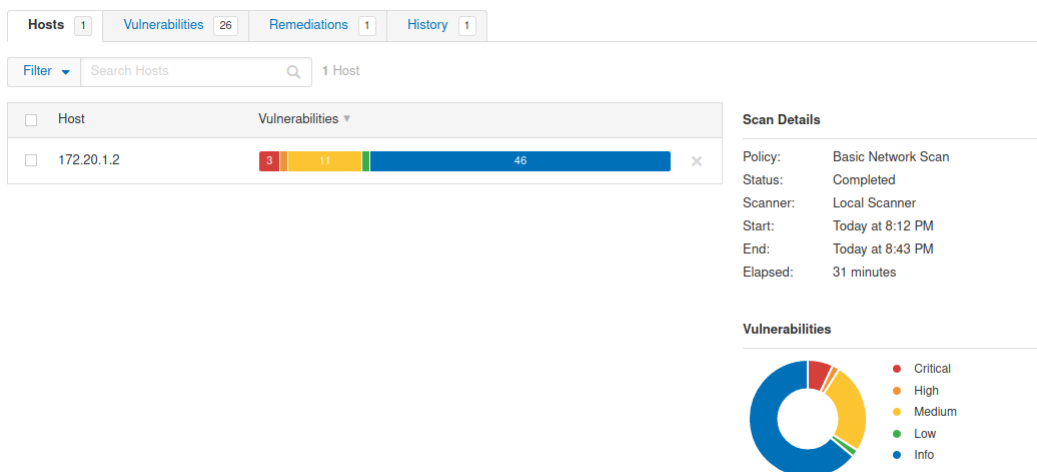


Figure 22: Resultado da varredura

Já aqui podemos as diferenças entre **Nmap** e **Nessus**, ao contrário do que teve de ser feito na **Q1**, o **Nessus** já nos dá as vulnerabilidades que ele detectou bem como o seu grau e passos a seguir para corrigir.

Esta diferença de resultados deve-se principalmente ao facto do **Nmap** ser apenas um *network scanner* e como tal a sua lista de vulnerabilidades para testar é pequena ou até inexistente, sendo que qualquer análise de vulnerabilidades utilizando apenas o **Nmap** irá certamente levar a que um elevado número de vulnerabilidades reportadas sejam falsos positivos/negativos.

O **Nessus**, sendo um *vulnerability scanner*, para além de fazer tudo o que o **Nmap** faz, vai também testar as portas abertas contra vulnerabilidades conhecidas. Isto tem como vantagem o facto de termos uma lista definitiva das vulnerabilidades a que a máquina alvo está sujeita, enquanto que com o **Nmap** temos só uma lista de vulnerabilidades a que o sistema pode ou não ser vulnerável.

Em termos de resultados, podemos ver que foram detectados:

- **3 Vulnerabilidades Críticas**
- **1 Vulnerabilidade Alta**
- **11 Vulnerabilidades Médias**
- **1 Vulnerabilidades Baixa**
- **46 Informações que podem ser úteis para futuros ataques**



Por fim, ao abrirmos cada uma das vulnerabilidades detectadas temos acesso a uma descrição da vulnerabilidade bem como a sua eventual correcção, algo que seria muito útil num ambiente real.

MEDIUM

TLS Version 1.0 Protocol Detection

< >

**Description**

The remote service accepts connections encrypted using TLS 1.0. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS like 1.2 and 1.3 are designed against these flaws and should be used whenever possible.

As of March 31, 2020, Endpoints that aren't enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors.

PCI DSS v3.2 requires that TLS 1.0 be disabled entirely by June 30, 2018, except for POS POI terminals (and the SSL/TLS termination points to which they connect) that can be verified as not being susceptible to any known exploits.

**Solution**

Enable support for TLS 1.2 and 1.3, and disable support for TLS 1.0.

**See Also**

<https://tools.ietf.org/html/draft-ietf-tls-oldversions-deprecate-00>

**Output**

TLSv1 is enabled and the server supports at least one cipher.

Port ▲	Hosts
3389 / tcp / msrdp	172.20.1.2

**Plugin Details**

Severity: Medium

ID: 104743

Version: 1.9

Type: remote

Family: Service detection

Published: November 22, 2017

Modified: March 31, 2020

**Risk Information**

Risk Factor: Medium

CVSS v3.0 Base Score 6.5

CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:N

CVSS Base Score: 6.1

CVSS Vector: CVSS2#AV:N/AC:H/Au:N/C:C/I:P/A:N

**Vulnerability Information**

Asset Inventory: True

Figure 23: Vulnerabilidade

### 3.3 Q3

Para esta questão começamos por analisar o ficheiro **alert.full**, que contém informações relativas aos pacotes que o **Snort** considerou como potenciais ataques.

Os eventos que escolhemos foram os seguintes:

Evento 1:

```
[**] [1:249:8] DDOS mstream client to handler [**]
[Classification: Attempted Denial of Service] [Priority: 2]
12/16-20:13:20.474155 172.20.1.1:5792 -> 172.20.1.2:15104
TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:48 DF
*****S* Seq: 0xD0D816F Ack: 0x0 Win: 0x1000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2000-0138]
[Xref => http://www.whitehats.com/info/IDS111]
```

Evento 2:

```
[**] [1:1384:8] MISC UPnP malformed advertisement [**]
[Classification: Misc Attack] [Priority: 2]
12/16-20:14:45.958499 172.20.1.1:62541 -> 172.20.1.2:1900
UDP TTL:64 TOS:0x0 ID:31337 IpLen:20 DgmLen:281
Len: 253
[Xref => http://www.microsoft.com/technet/security/bulletin/MS01-059.msp]
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2001-0877]
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=2001-0876]
[Xref => http://www.securityfocus.com/bid/3723]
```

Para identificarmos os pacotes dentro do **Wireshark** aplicamos os seguintes filtros com base na hora que o **Snort** registou o alerta:

```
frame.time == "Dec 16, 2020 20:13:20.474155"
frame.time == "Dec 16, 2020 20:14:45.958499"
```

frame.time == "Dec 16, 2020 20:14:45.958499"    frame.time == "Dec 16, 2020 20:13:20.474155"						
No.	Time	Source	Destination	Protocol	Length	Info
8242	47.467192	172.20.1.1	172.20.1.2	TCP	62	5792 -> 15104 [SYN] Seq=0 Win=4096 Len=0 MSS=1460 SACK_PERM=1
14967	132.951536	172.20.1.1	172.20.1.2	SSDP	295	NOTIFY * HTTP/1.1

Figure 24: Pacotes no **Wireshark**

### 3.3.1 Evento 1

Este primeiro evento foi detectado como **Denial of Service** que é um ataque que tenta sobrecarregar o sistema alvo com pedidos de conexão. Se o ataque tivesse sucesso este sistema iria ficar indisponível, algo que na vida real pode causar muito dano.

No analisador de tráfego podemos confirmar esta hipótese visto que o pacote detectado como tentativa de *denial of service* é precedido e procedido por vários outros pacotes semelhantes onde só mudam as portas de saída e de destino.

8240	47.466685	172.20.1.1	172.20.1.2	TCP	62 2241 → 3691 [SYN] Seq=0 Win=4096 Len=0 MSS=1460 SACK_PERM=1
8241	47.466938	172.20.1.1	172.20.1.2	TCP	62 58735 → 4030 [SYN] Seq=0 Win=4096 Len=0 MSS=1460 SACK_PERM=1
8242	47.467192	172.20.1.1	172.20.1.2	TCP	62 5792 → 15104 [SYN] Seq=0 Win=4096 Len=0 MSS=1460 SACK_PERM=1
8243	47.467452	172.20.1.1	172.20.1.2	TCP	62 21213 → 192 [SYN] Seq=0 Win=4096 Len=0 MSS=1460 SACK_PERM=1
8244	47.467705	172.20.1.1	172.20.1.2	TCP	62 23166 → 531 [SYN] Seq=0 Win=4096 Len=0 MSS=1460 SACK_PERM=1
8245	47.488503	172.20.1.1	172.20.1.2	TCP	62 31238 → 1096 [SYN] Seq=0 Win=4096 Len=0 MSS=1460 SACK_PERM=1
8246	47.489057	172.20.1.1	172.20.1.2	TCP	62 1079 → 1209 [SYN] Seq=0 Win=4096 Len=0 MSS=1460 SACK_PERM=1

Figure 25: DDOS detectado

Por fim o **Snort** também fornece o **CVE** respectivo, neste caso foi o *CVE-2000-0138*.

## CVE-2000-0138 Detail

### MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

### Current Description

A system has a distributed denial of service (DDOS) attack master, agent, or zombie installed, such as (1) Trinoo, (2) Tribe Flood Network (TFN), (3) Tribe Flood Network 2000 (TFN2K), (4) stacheldraht, (5) mstream, or (6) shaft.

Figure 26: CVE-2000-0138

Através do analisador de tráfego podemos também verificar que é um pacote **TCP** do tipo *SYN*, logo está a tentar iniciar uma conexão, com porta fonte **5792** e porta destino **15104**. Como é óbvio o pacote foi enviado da máquina auditora para a máquina alvo.

### 3.3.2 Evento 2

Este segundo evento foi detectado como *Universal Plug and Play* malformed advertisement.

O pacote foi novamente enviado pela máquina auditora para a máquina alvo, da porta **62541** para a porta **1900**. Ao contrário do evento anterior este usa **UDP** como protocolo de transporte sendo que o protocolo em si é **Simple Service Discovery Protocol** que é utilizado para descoberta de serviços numa rede.

É um pacote **NOTIFY**, que no protocolo **SSDP** é utilizado para anunciar a criação ou remoção de serviços numa rede.

O **Snort** detetou duas vulnerabilidades para este pacote, nomeadamente **CVE-2001-0876** e **CVE-2001-0877**.

#### CVE-2001-0876 Detail

##### MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

#### Current Description

Buffer overflow in Universal Plug and Play (UPnP) on Windows 98, 98SE, ME, and XP allows remote attackers to execute arbitrary code via a NOTIFY directive with a long Location URL.

Figure 27: **CVE-2001-0876**

#### CVE-2001-0877 Detail

##### MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

#### Current Description

Universal Plug and Play (UPnP) on Windows 98, 98SE, ME, and XP allows remote attackers to cause a denial of service via (1) a spoofed SSDP advertisement that causes the client to connect to a service on another machine that generates a large amount of traffic (e.g., chargen), or (2) via a spoofed SSDP announcement to broadcast or multicast addresses, which could cause all UPnP clients to send traffic to a single target system.

Figure 28: **CVE-2002-0877**

### 3.4 Q4

A diferença entre os resultados do **IDS** e do **Nessus** pode ser devido ao facto do **IDS** alertar sobre qualquer tráfego que considere anómalo segundo as regras definidas esteja a porta aberta ou fechada, enquanto o **Nessus** apenas detecta vulnerabilidades de portas que estejam abertas.

### 3.5 Q5

Para a resolução desta questão começamos por escolher 3 vulnerabilidades com graus diferentes de gravidade, sendo que procuramos também as formas de corrigir cada uma.

CRITICAL

MS14-066: Vulnerability in Schannel Could Allow Remote Code Exec...

< >

**Description**

The remote Windows host is affected by a remote code execution vulnerability due to improper processing of packets by the Secure Channel (Schannel) security package. An attacker can exploit this issue by sending specially crafted packets to a Windows server.

Note that this plugin sends a client Certificate TLS handshake message followed by a CertificateVerify message. Some Windows hosts will close the connection upon receiving a client certificate for which it did not ask for with a CertificateRequest message. In this case, the plugin cannot proceed to detect the vulnerability as the CertificateVerify message cannot be sent.

**Solution**

Microsoft has released a set of patches for Windows 2003, Vista, 2008, 7, 2008 R2, 8, 2012, 8.1, and 2012 R2.

**See Also**

<http://www.nessus.org/u?64e97902>

**Output**

No output recorded.

Port <sup>A</sup>	Hosts
3389 / tcp / msrdp	172.20.1.2

**Plugin Details**

Severity: Critical

ID: 79638

Version: 1.146

Type: remote

Family: Windows

Published: December 1, 2014

Modified: November 16, 2020

**Risk Information**

Risk Factor: Critical

CVSS v3.0 Base Score: 8.8

CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

CVSS v3.0 Temporal Vector: CVSS:3.0/E:F/RL:O/RC:C

CVSS v3.0 Temporal Score: 8.2

CVSS Base Score: 10.0

CVSS Temporal Score: 8.3

CVSS Vector: CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C

CVSS Temporal Vector: CVSS2#E:F/RL:OF/RC:C

**Vulnerability Information**

Figure 29: MS14-066

21

HIGH
MS12-020: Vulnerabilities in Remote Desktop Could Allow Remote C...
<
>

### Plugin Details

Severity:	High
ID:	58435
Version:	1.59
Type:	remote
Family:	Windows
Published:	March 22, 2012
Modified:	September 14, 2020

### Risk Information

Risk Factor: High

CVSS Base Score: 9.3

CVSS Temporal Score: 7.3

CVSS Vector: CVSS2#AV:N/AC:M/Au:N/C:C/I:C/A:C

CVSS Temporal Vector:

CVSS2#E:POC/RL:OF/RC:C

IAVM Severity: I

### Vulnerability Information

CPE: cpe:/o:microsoft:windows

cpe:/a:microsoft:remote\_desktop\_protocol

Exploit Available: true

### Description

An arbitrary remote code vulnerability exists in the implementation of the Remote Desktop Protocol (RDP) on the remote Windows host. The vulnerability is due to the way that RDP accesses an object in memory that has been improperly initialized or has been deleted.

If RDP has been enabled on the affected system, an unauthenticated, remote attacker could leverage this vulnerability to cause the system to execute arbitrary code by sending a sequence of specially crafted RDP packets to it.

This plugin also checks for a denial of service vulnerability in Microsoft Terminal Server.

Note that this script does not detect the vulnerability if the 'Allow connections only from computers running Remote Desktop with Network Level Authentication' setting is enabled or the security layer is set to 'SSL (TLS 1.0)' on the remote host.

### Solution

Microsoft has released a set of patches for Windows XP, 2003, Vista, 2008, 7, and 2008 R2.

Note that an extended support contract with Microsoft is required to obtain the patch for this vulnerability for Windows 2000.

### See Also

<https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2012/ms12-020>

### Output

No output recorded.

Figure 30: MS12-020

MEDIUM

TLS Version 1.0 Protocol Detection

< >

Plugin Details

Severity: Medium

ID: 104743

Version: 1.9

Type: remote

Family: Service detection

Published: November 22, 2017

Modified: March 31, 2020

Description

The remote service accepts connections encrypted using TLS 1.0. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS like 1.2 and 1.3 are designed against these flaws and should be used whenever possible.

As of March 31, 2020, Endpoints that aren't enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors.

PCI DSS v3.2 requires that TLS 1.0 be disabled entirely by June 30, 2018, except for POS POI terminals (and the SSL/TLS termination points to which they connect) that can be verified as not being susceptible to any known exploits.

Solution

Enable support for TLS 1.2 and 1.3, and disable support for TLS 1.0.

See Also

<https://tools.ietf.org/html/draft-ietf-tls-oldversions-deprecate-00>

Risk Information

Risk Factor: Medium

CVSS v3.0 Base Score 6.5

CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:N

CVSS Base Score: 6.1

CVSS Vector: CVSS2#AV:N/AC:H/Au:N/C:C/I:P/A:N

Vulnerability Information

Asset Inventory: True

Output

TLSv1 is enabled and the server supports at least one cipher.

Port	Hosts
3389 / tcp / msrdp	172.20.1.2

Figure 31: TLS Version

Para eliminar estas vulnerabilidades temos que aplicar as seguintes correções na máquina alvo:

- **MS14-066**- Instalar actualizações *KB3018238* e *KB2992611*
- **MS12-020**- Instalar atualização *KB2621440*
- **TLS-1.0**- Forçar a utilização do **TLS1.2** através de ficheiros que alteram os registos






 Enable-TLS12-TLS11-Windows.reg	20/12/2020 00:01	Entradas de registo	2 KB
 Enable-TLS12-Windows.reg	20/12/2020 00:00	Entradas de registo	2 KB
 windows6.1-kb2621440-x64_0f4492f612ea59c386a59e587d71ee3ae5d0f475.msu	19/12/2020 23:52	Pacote Autónomo...	1 400 KB
 Windows6.1-KB2992611-x64.msu	19/12/2020 23:41	Pacote Autónomo...	5 351 KB
 Windows6.1-KB3018238-x64.msu	19/12/2020 23:41	Pacote Autónomo...	338 KB

Figure 32: Correções a aplicar

### 3.5.1 Scan Final

Infelizmente foi-nos impossível realizar o scan final para verificar se as correcções aplicadas funcionavam ou não, visto que numa das máquinas dos alunos o **Nessus** não estava a funcionar correctamente e no computador do outro a máquina alvo não detectava a pen **USB** que continha os ficheiros que corrigiam as vulnerabilidades, apesar das definições do controlador **USB** na **Vbox** terem sido alteradas.



Figure 33: Definições do controlador USB na Vbox

- USB1.1- Não detectava USB
- USB2.0- Não iniciava
- USB3.0- Não detectava USB

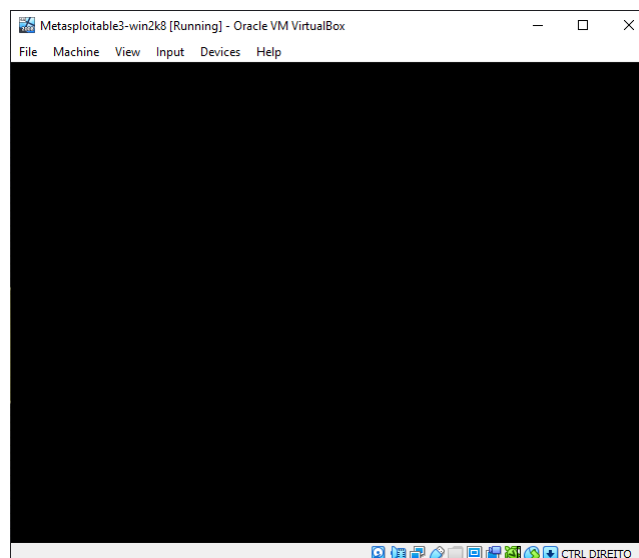


Figure 34: Máquina alvo com controlador USB 2.0

Apesar de tudo, assumindo que os ficheiros corrigiam de facto as vulnerabilidades, o resultado esperado seria que as vulnerabilidades detectadas desaparecessem sendo provavelmente substituídas por novas vulnerabilidades.



## 4 Conclusão

Na realização deste trabalho foi possível adquirir conhecimentos sobre as formas de encontrar vulnerabilidades nos vários tipos de software abordados neste trabalho. Foi possível também conhecermos vários tipos de *scanners*, bem como as suas diferenças, e assim encontrar um leque maior de vulnerabilidades em cada sistema analisado. Na nossa realização do trabalho também exploramos maneiras de tentar salvaguardar estes *exploits*.

Em suma, acreditamos que todos os objectivos que nos foram propostos foram alcançados com a excepção do scan final, sendo que só não foi realizada por problemas alheios aos alunos. É também de apontar que a performance da máquina alvo deixa muito a desejar, por exemplo para abrir o *file explorer* demora cerca de 15 minutos e só para ligar a máquina alvo demora cerca de 1 hora.