TEH3261/ FTEH3261
ETHICAL HACKING AND SECURITY ASSESSMENT
PROJECT 3 – 10%

Lab Session : 1BV
Team Name : 3AM Hackers
Team Leader : 1191302861 Mayar Abdulmalik M Shenawi
Group Member 1 : 1181102166 Ahmed Aldughaither
Group Member 2 : 1191302763 Al Ghamdi Omar Saeed O
Group Member 3 : 1191201179 Rian Tiew Ming Sheen
Group Member 4 : 1171103833 Loo Wei Jun

Instructions:

1. Form a group of **3-5** students, remain in the same group for all projects and *Capture-the-Flag* (CTF) – lab tests.
2. Complete **ALL** the questions below by attaching a snapshot of your screen result as evidence.
3. Only **1 copy** to submit by the Group Leader.
4. Submit latest by **8 April 2022, 5pm** to the MMLS Assignment Repository.

Familiarize with *Capture-the-Flag (CTF)* – Realistic Mission

Platform: Hack This Site (https://hackthissite.org/)

1. Continue your exercise at **hackthissite.org**

2. **Login** to your account.

3. On the left tab of the main page, in the **challenges** section, click " **Realistic Missions**"

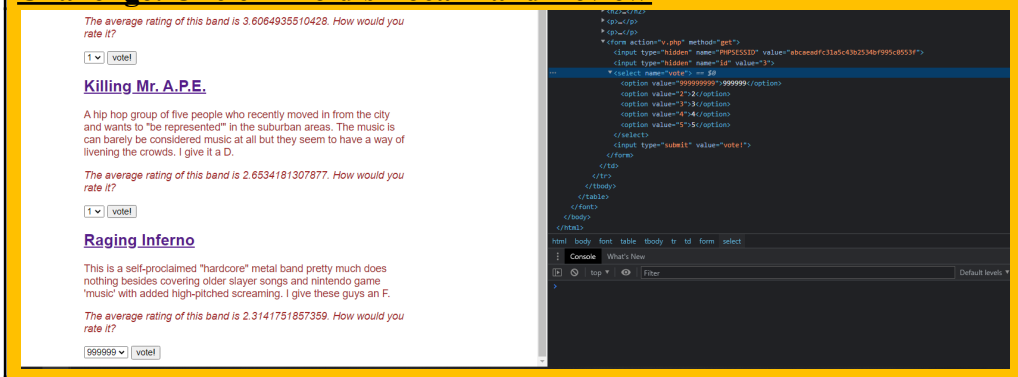
4. There are **sixteen (16)** Realistic Missions which attempt to mimic real, moderate to difficult hacking, in real life situations. Solve at least **six (6)** puzzles. You get **0.5 point** per level completed.

Show your steps on how you solved the challenge. [3%]

Notes: your account name must be shown when you snapshot your answer – to prove that you attempted the challenges/ puzzles by using your account.

Prepared by: SY Ooi

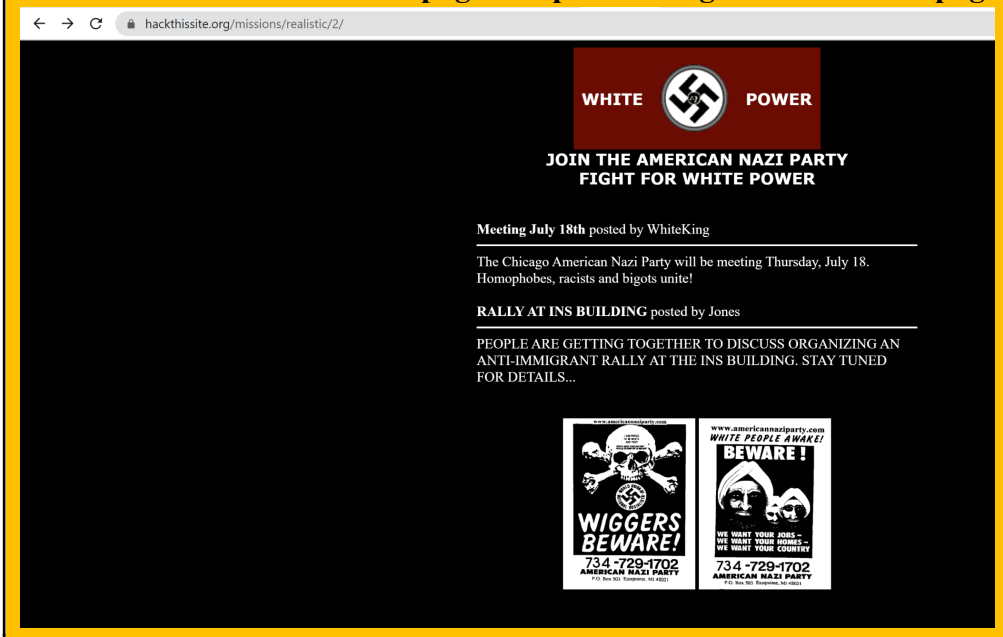
Challenge: Uncle Arnold's Local Band Review



You right click>inspect>change the value of raging inferno vote into any high number and click vote

Challenge: Chicago American Nazi Party

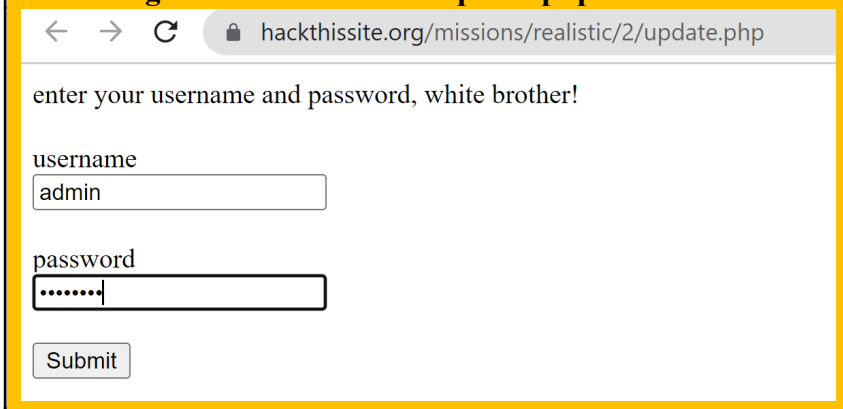
To access their administrator page and post messages to their main page.



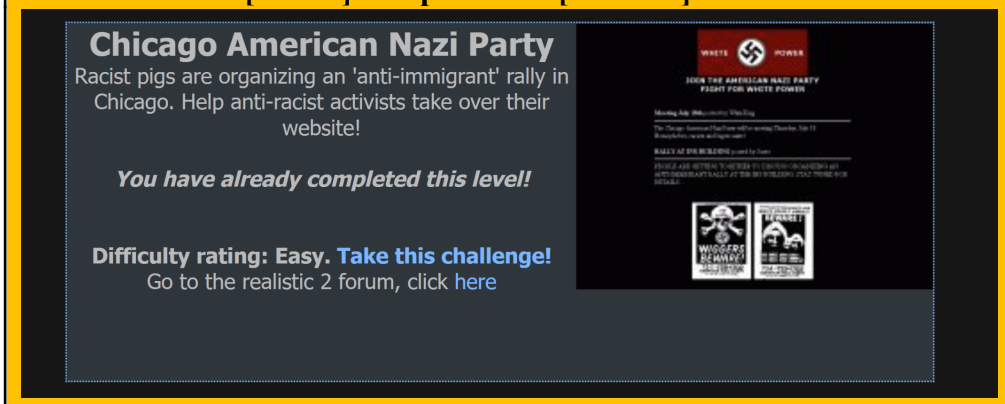
The website shown.

Prepared by: SY Ooi

“View Page Source” Click on “update.php” link



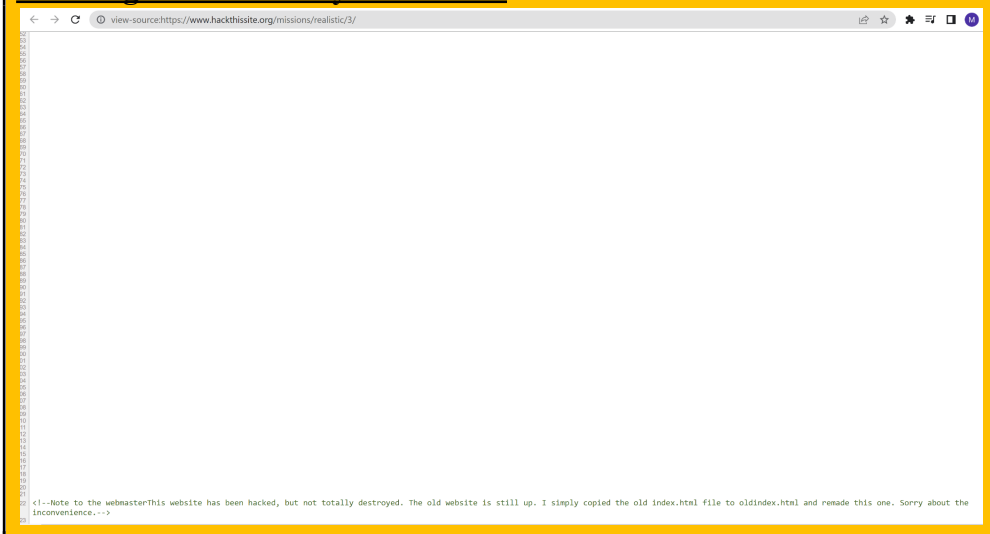
Enter username [admin] and password [‘or‘1=1‘] to access the website



Challenge completed.

Prepared by: SY Ooi

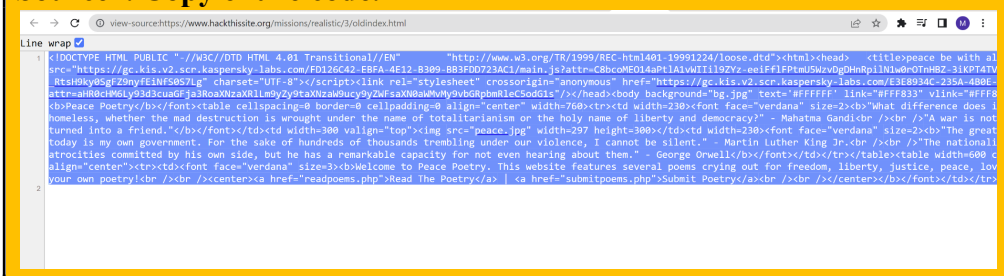
Challenge: Peace Poetry: HACKED



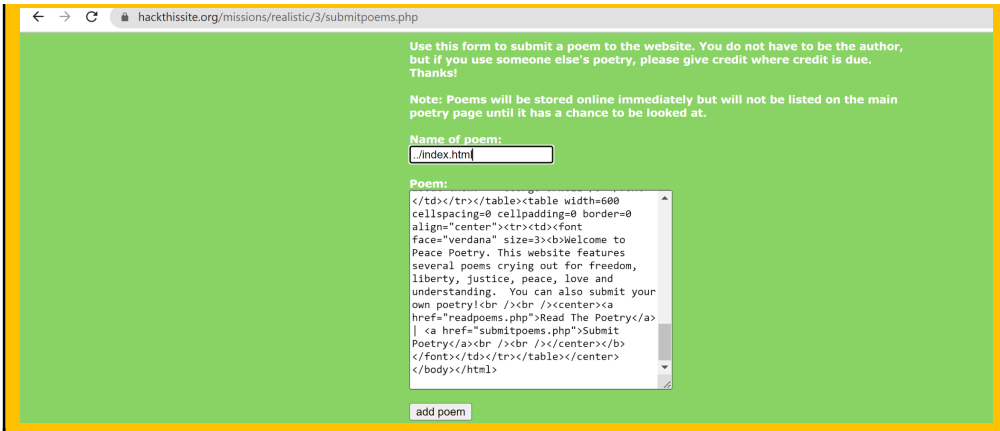
Click on the website and "View Page Source" and you will find "oldindex.html"

Add to the website URL

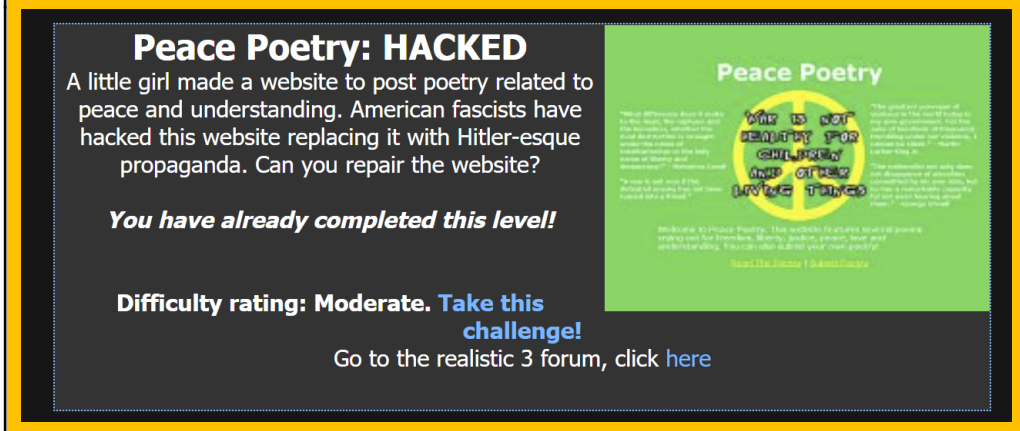
You will reach another page, right-click on the website and "View Page Source". Copy entire code.



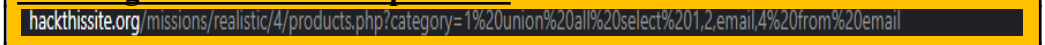
Prepared by: SY Ooi



Choose "Submit Poetry", paste the code in the poem. Use name "../index.html"



Challenge: Fischer Animal products



First you have to click on any product
then you notice the link is a category
by playing with the category format, you are able to get the emails by change it into UNION ALL SELECT null, *, null, null FROM email;
it will display the emails

Prepared by: SY Ooi

2

jsmith@uic.edu

2

3ambeer@graffiti.net

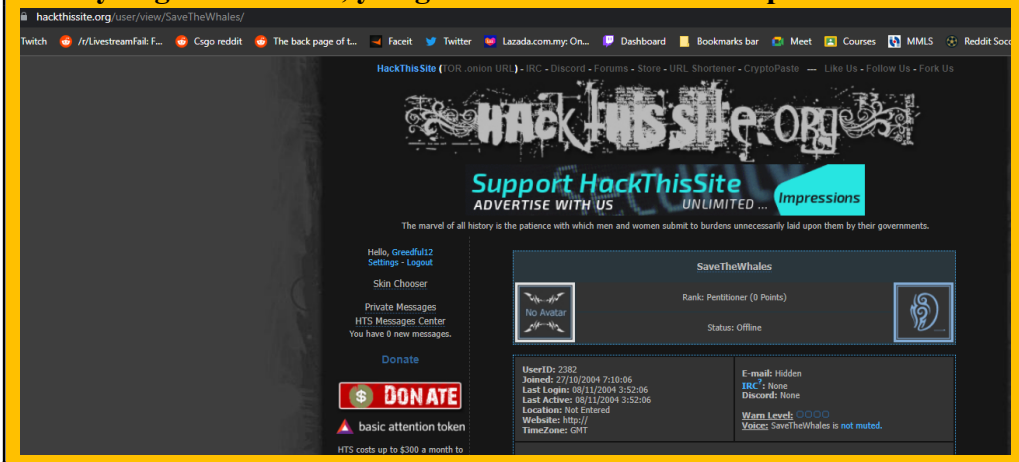
2

shootfirst@yahoo.com

2

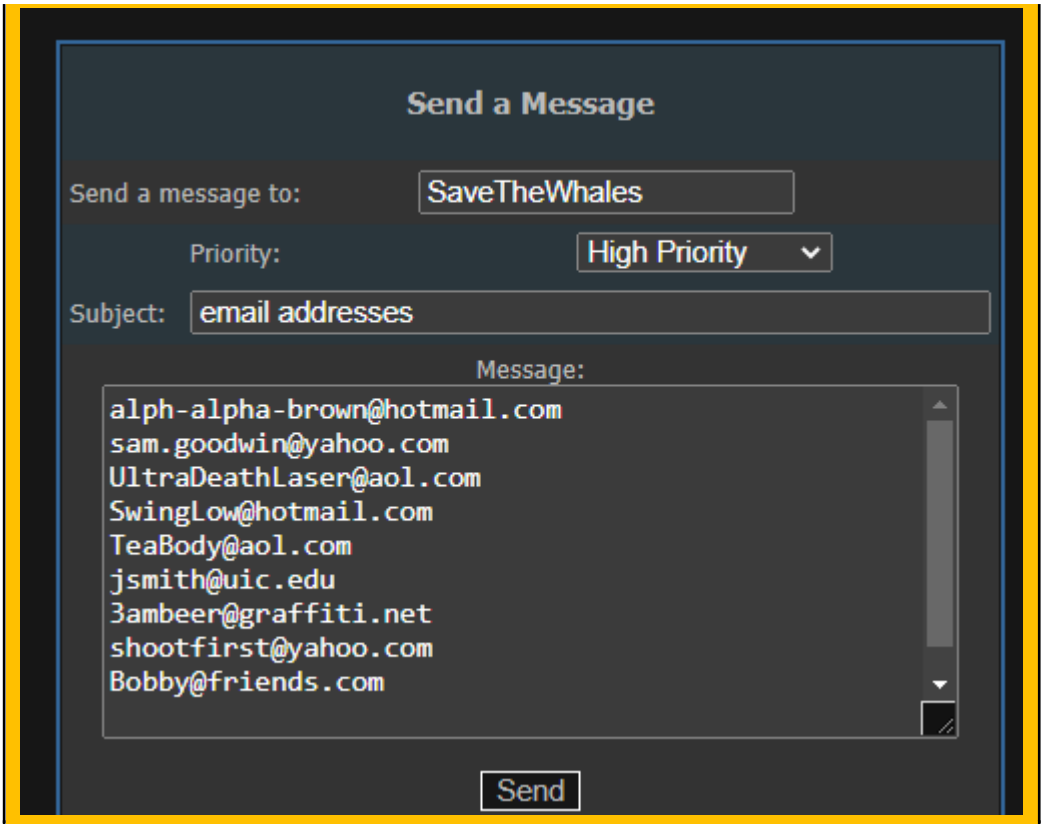
Bobby@friends.com

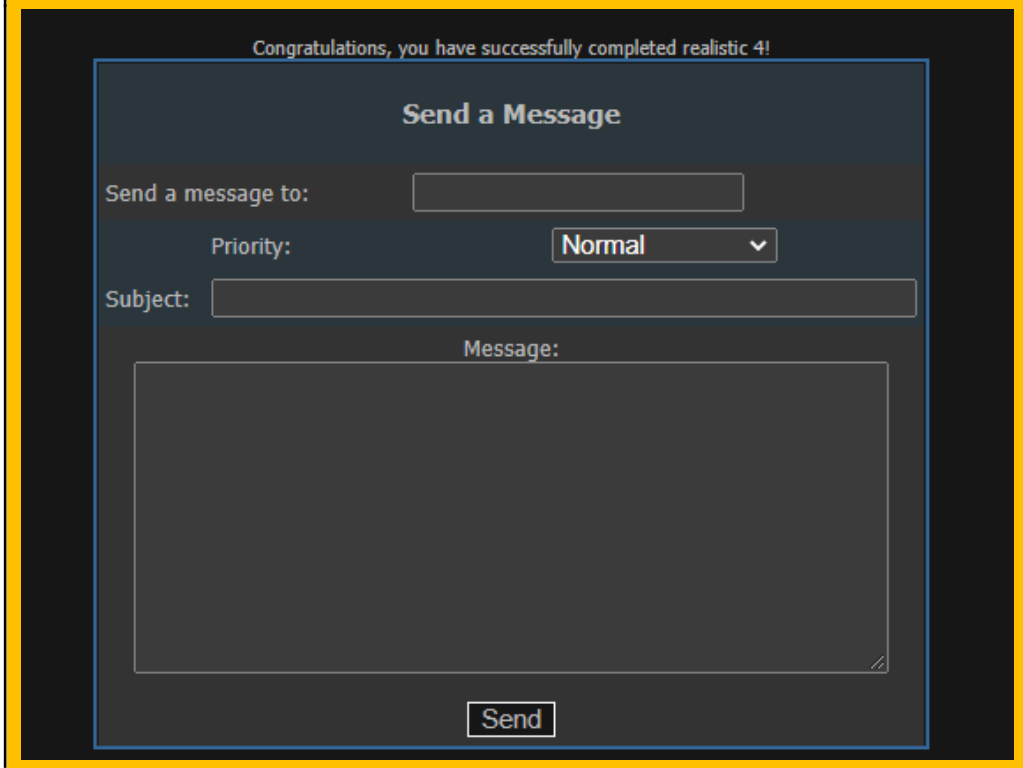
After you get the emails, you go to the SaveTheWhales profile



You click in his name and email him addresses

Prepared by: SY Ooi

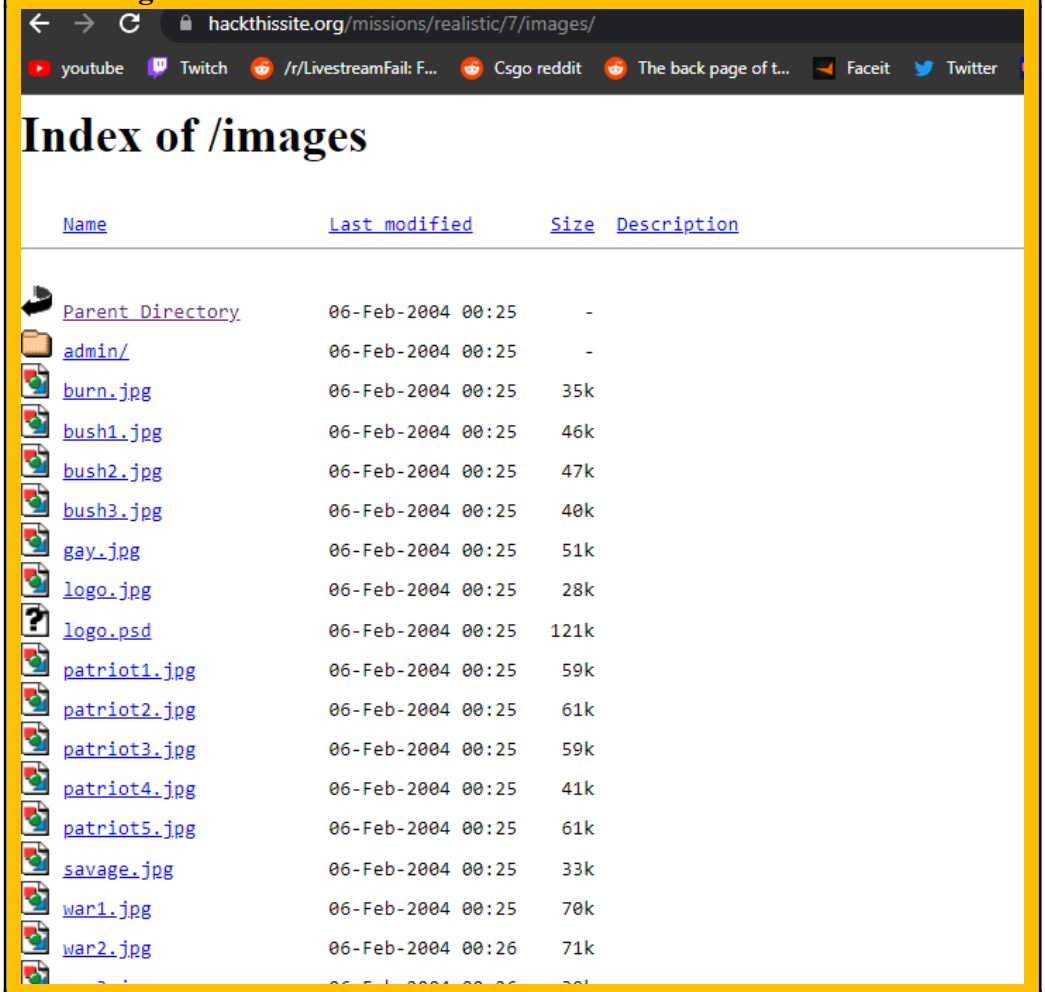




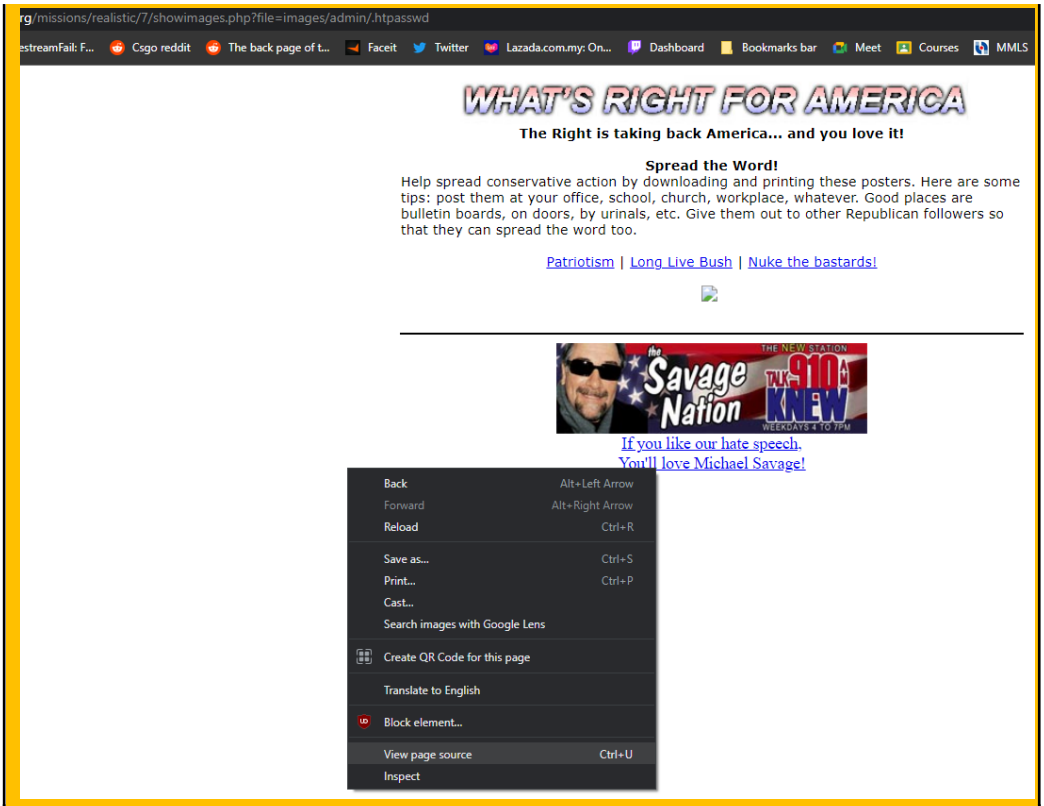
After you email, it should say you completed realistic challenge 4

Prepared by: SY Ooi

Challenge 5
what's right for america

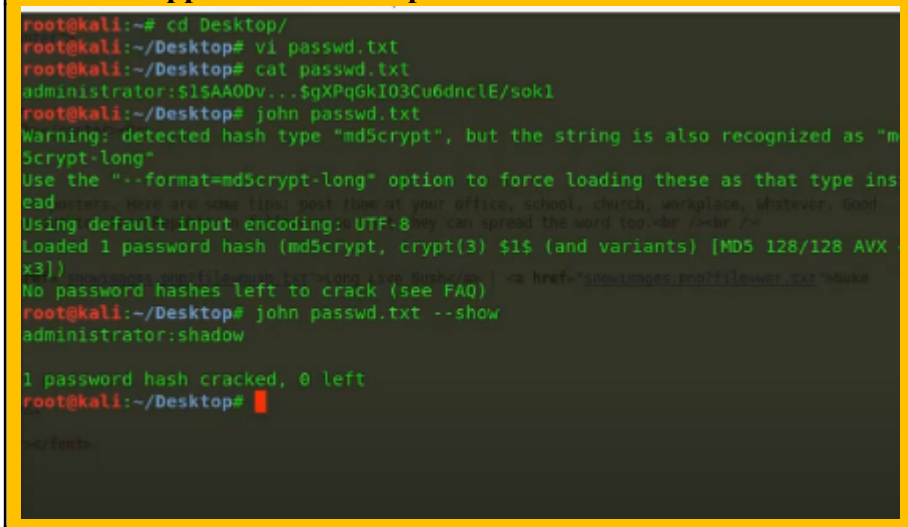


Check images directory



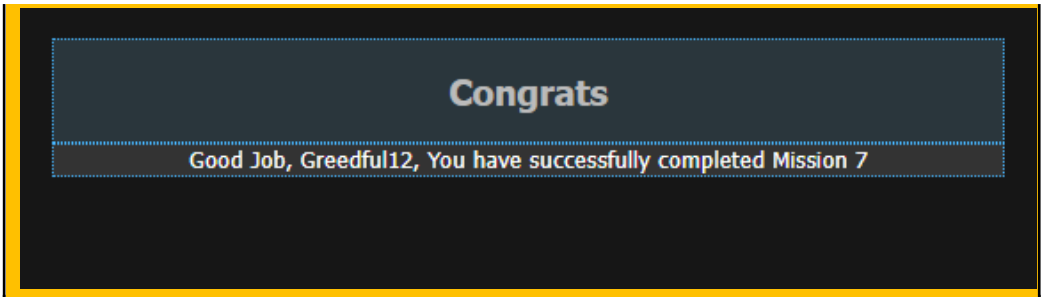
View page source for info from .htpasswd

Get the password
John the ripper to crack the password

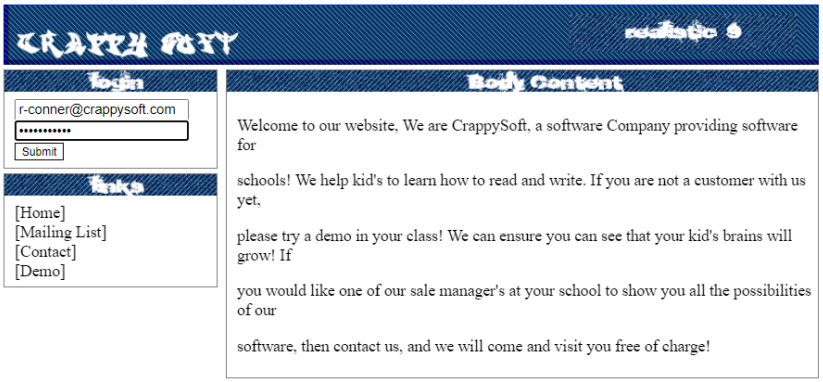


login using administrator and password shadow to complete

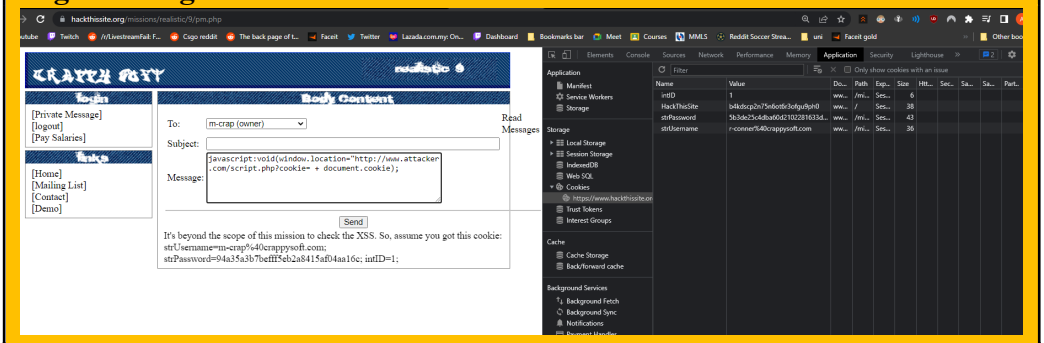
Prepared by: SY Ooi



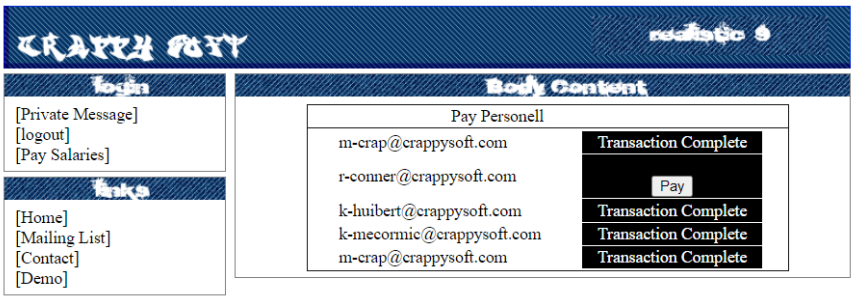
Challenge 6 crappysoft software



Login from given info

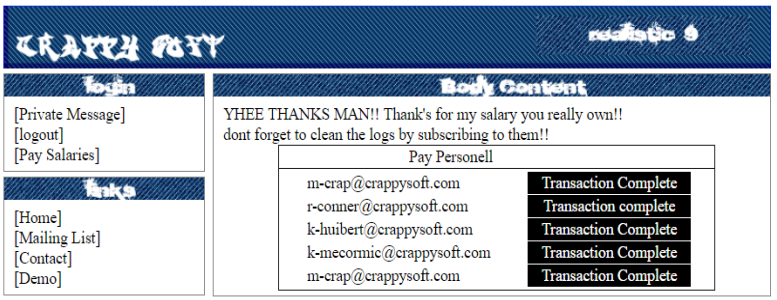


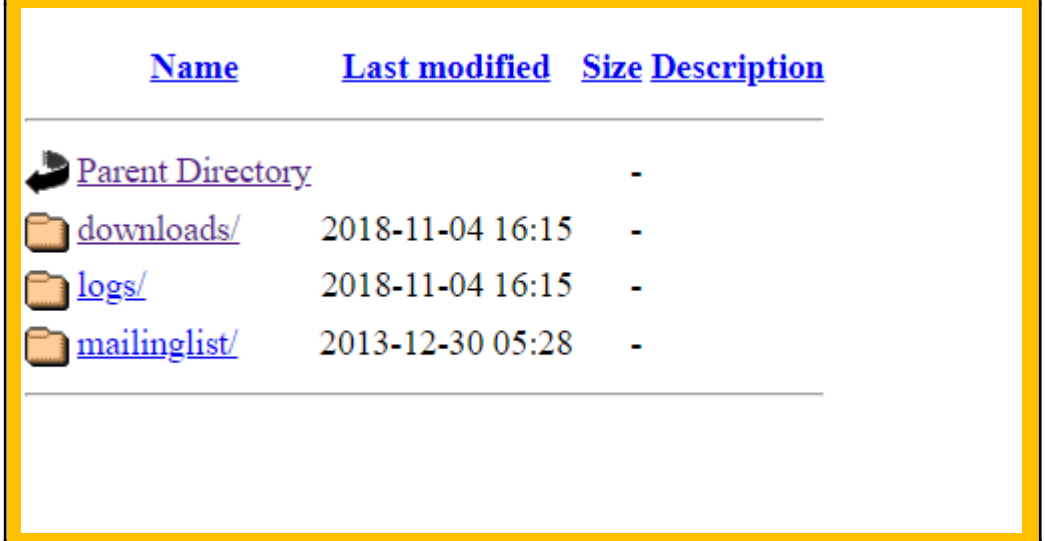
Change cookies after sending XSS attack and getting the info to replace int ID/username/password



Click pay in pay salaries after getting permission to view it

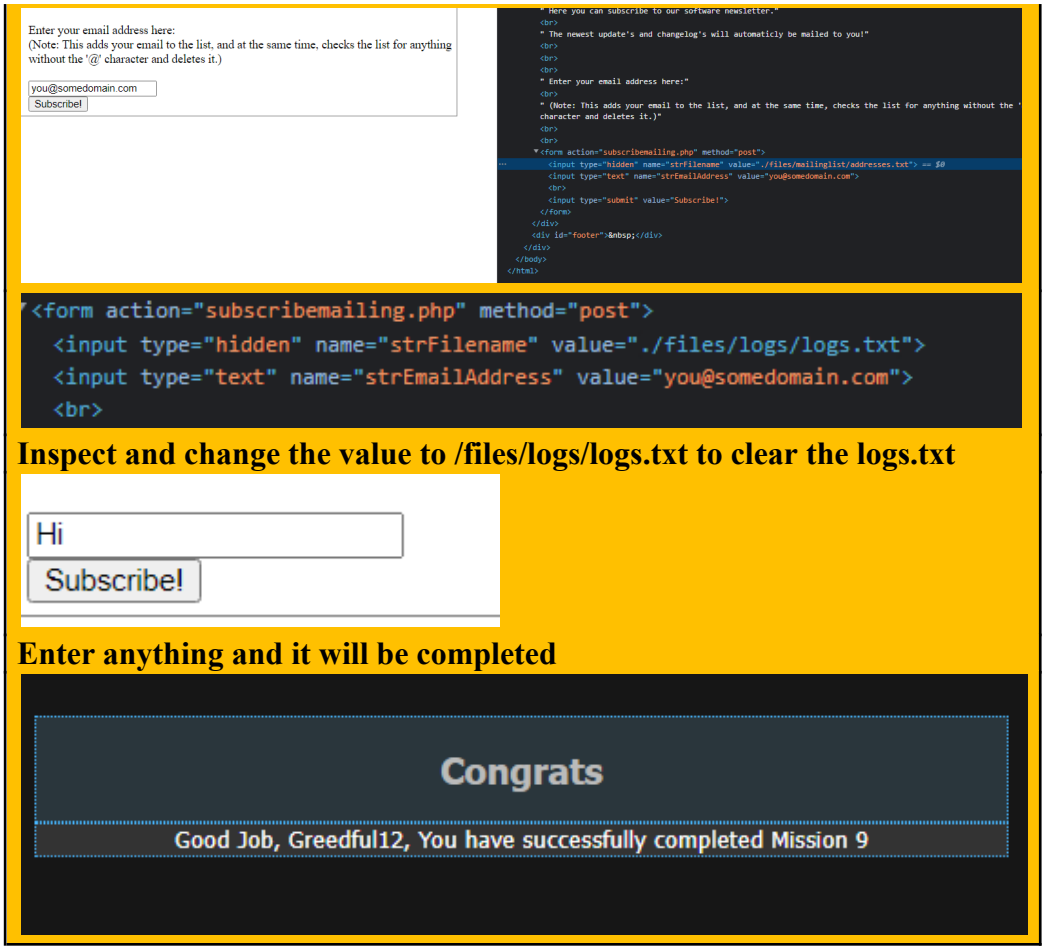
Prepared by: SY Ooi





After you find logs.txt in /downloads/

Prepared by: SY Ooi

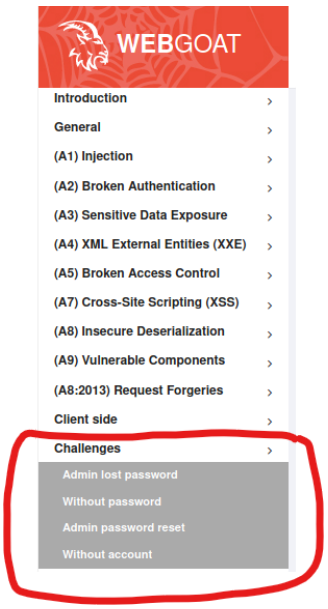


Application hacking – test vulnerabilities commonly found in Java-based applications

Platform: WebGoat

1. Refer to the Lab 10, continue your exercise at **WebGoat**.
2. **Login** to your account.
3. On the left tab of the main page, solve these **4 tasks** in the **challenges** section:
    a. Admin lost password
    b. Without password
    c. Admin password reset
    d. Without account

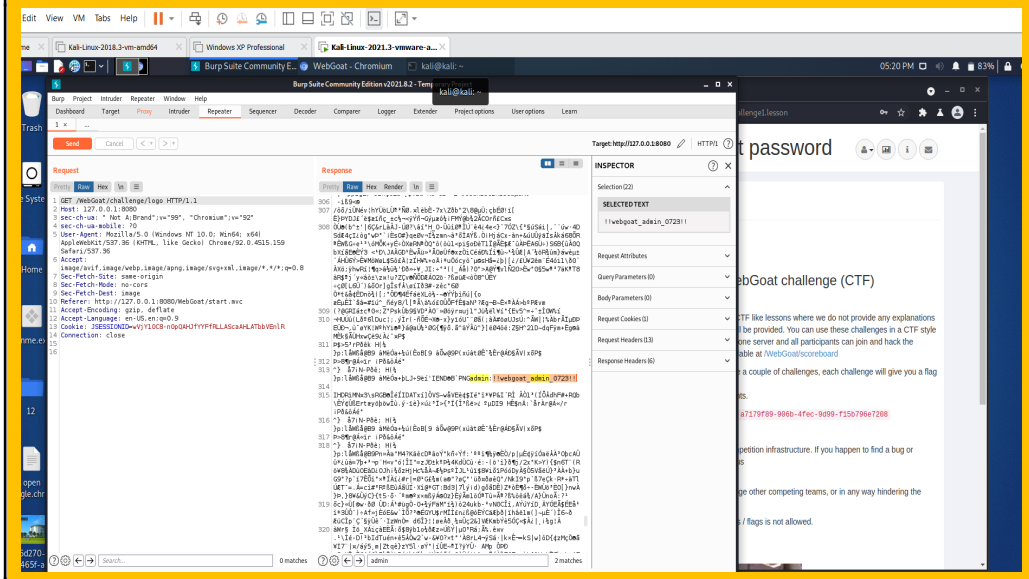
Prepared by: SY Ooi



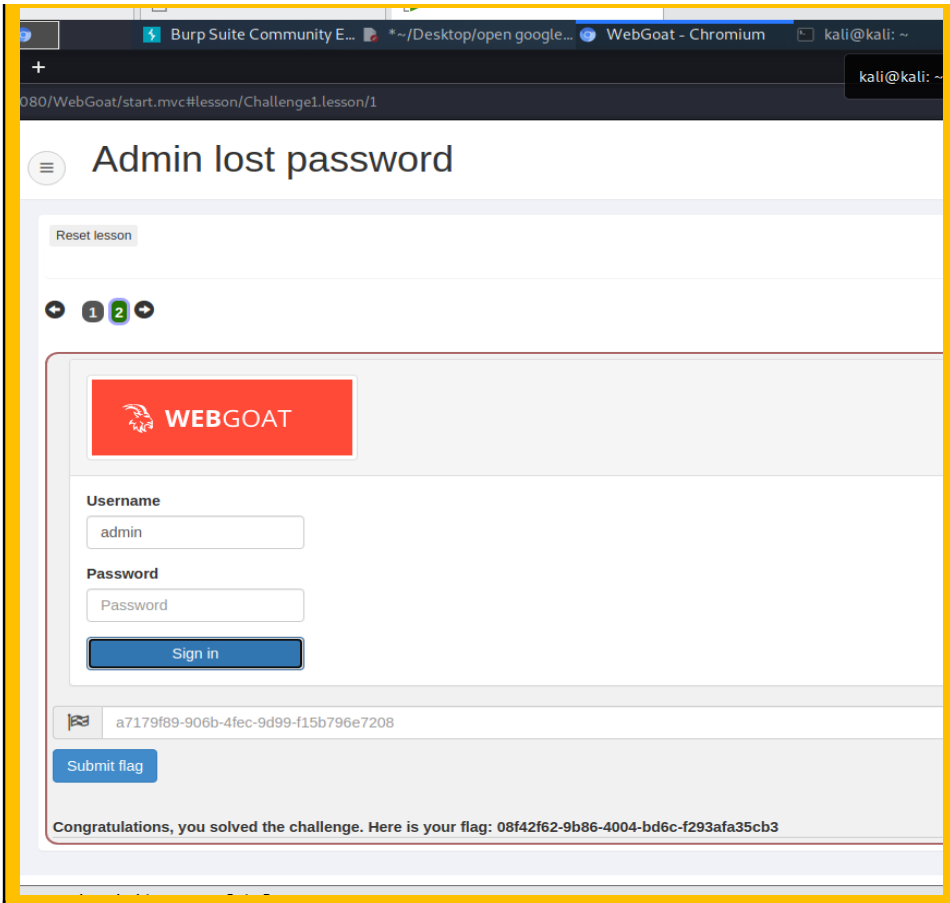
Each task will earn 0.5%.

Admin Lost password : Challenge 1

Configure burpsuite and proxy settings , find GET/WebGoat/challenge/logo HTTP/1.1 and send the request to repeater

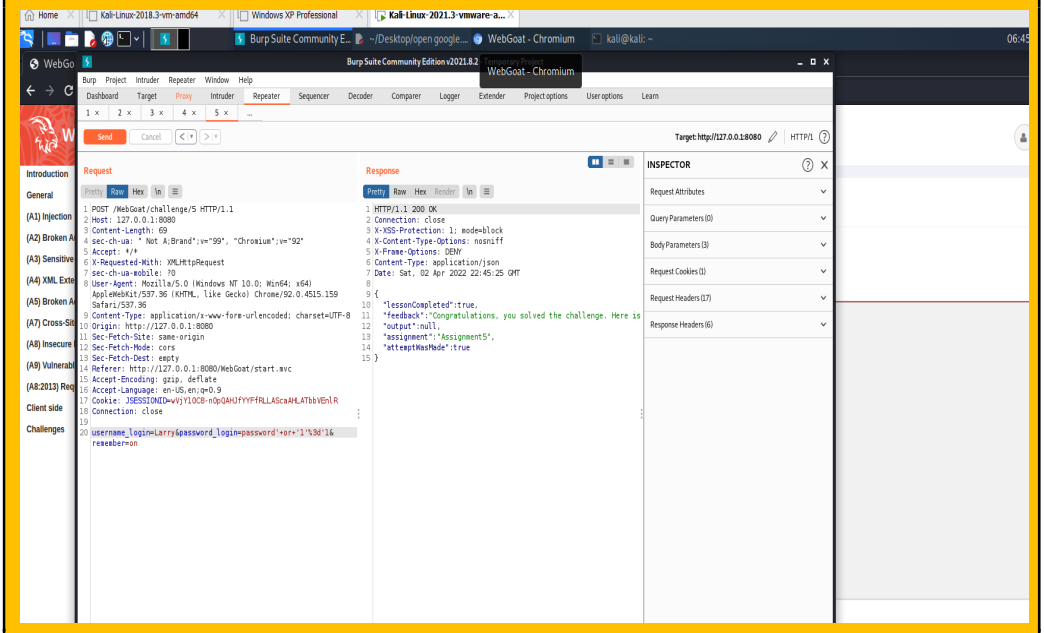


Prepared by: SY Ooi

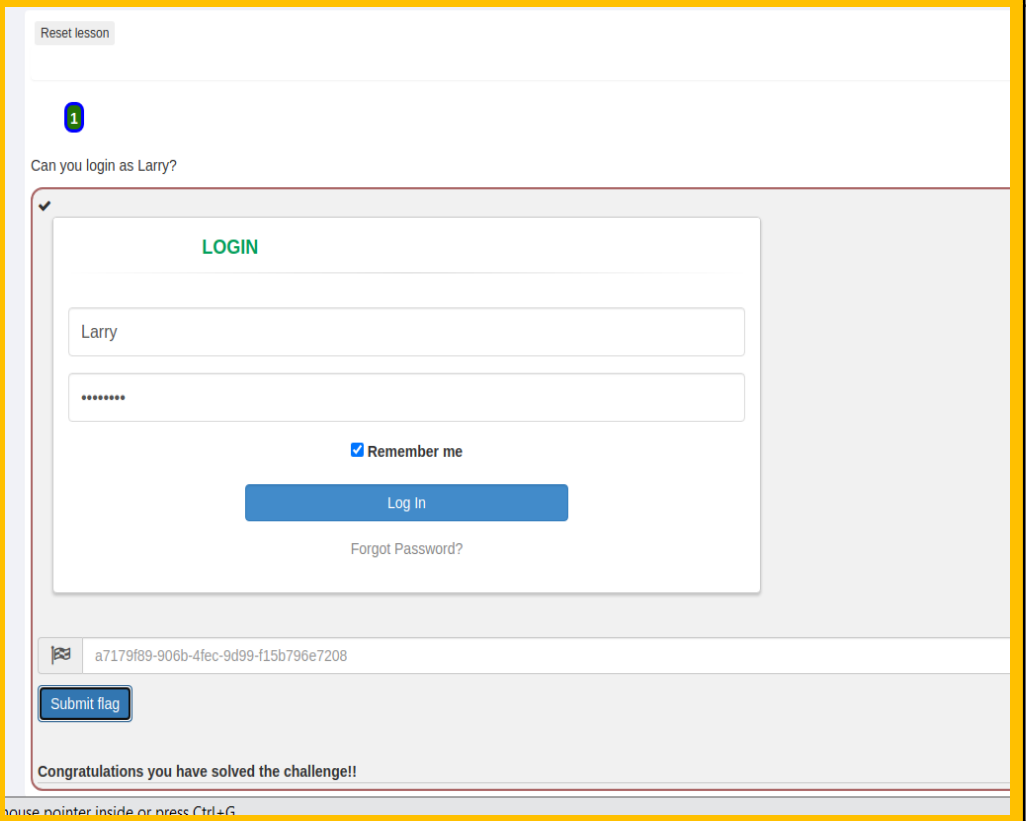


Challenges Without Password Can you login as Larry?

Configure Burpsuite and proxy setting:
- Put userName= Larry,
Find POST/WebGoat/challenge/5 HTTP/1.1 then Send Repeater

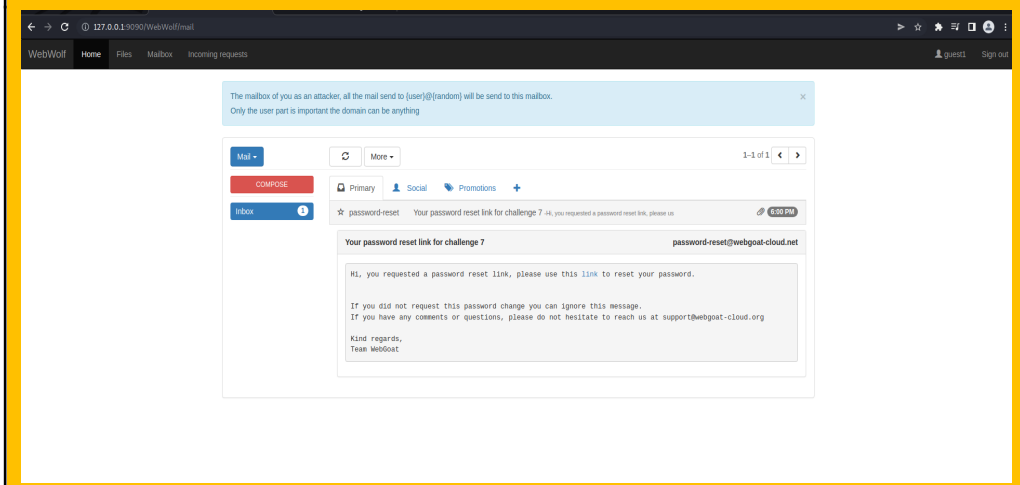


Prepared by: SY Ooi

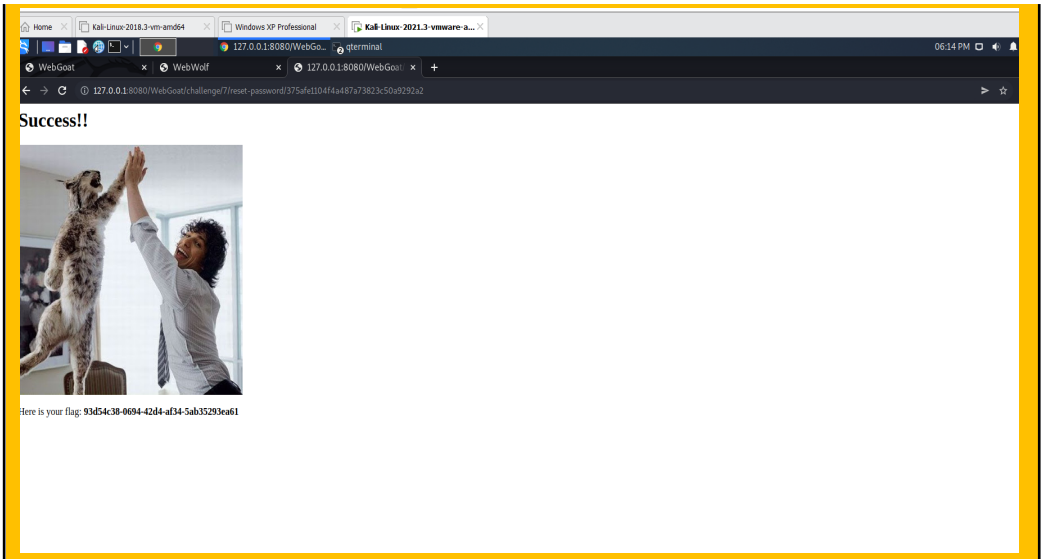


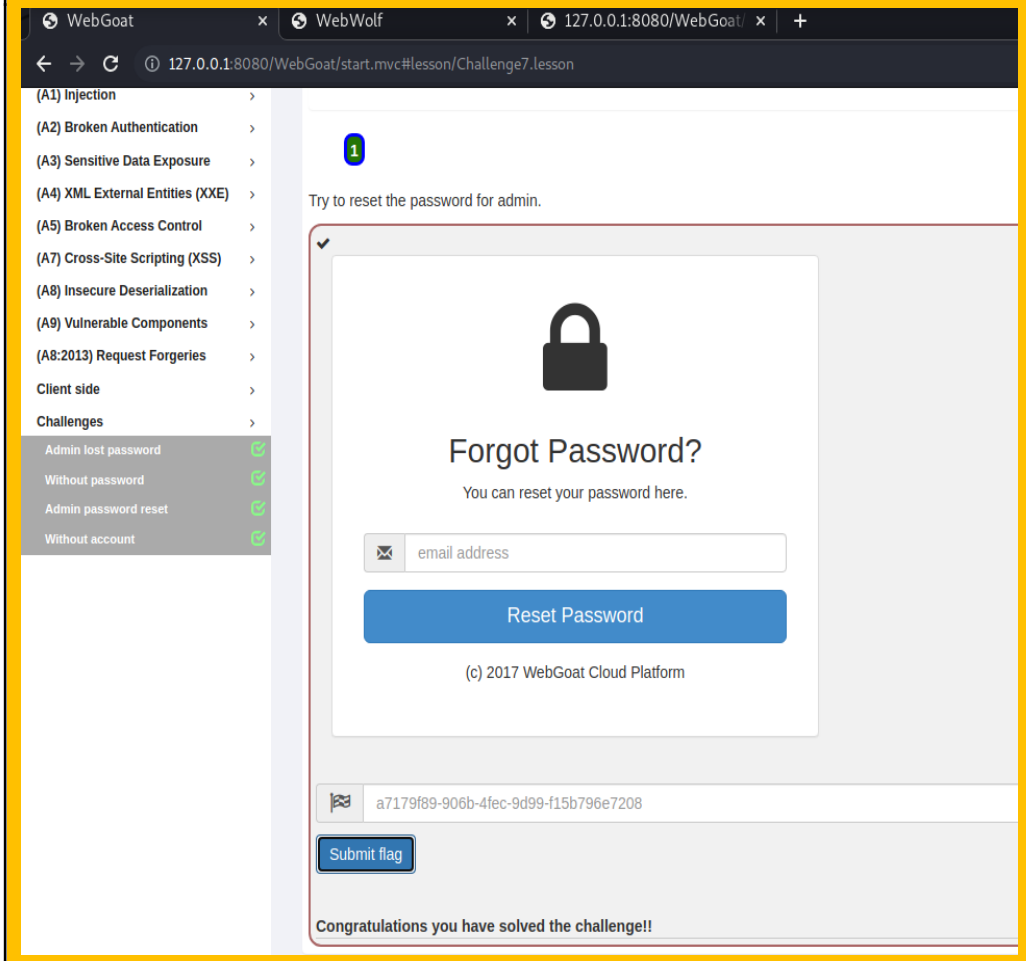
Challenges Admin password reset :

Reset then go the the WebWolf check your email
we have to find the correct <Hash Value> to reset the admin
then Check the source code : https://github.com/...
observe the code:
import org.owasp.webgoat.challenges.SolutionConstants;
Get the hash and complete the solution.



Prepared by: SY Ooi

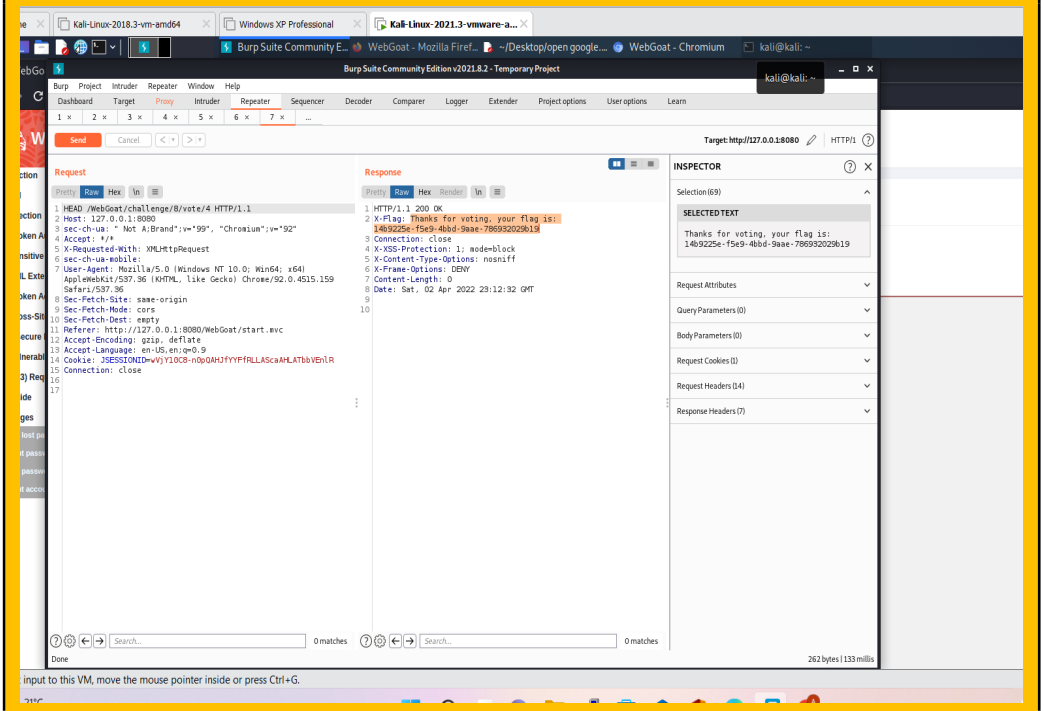




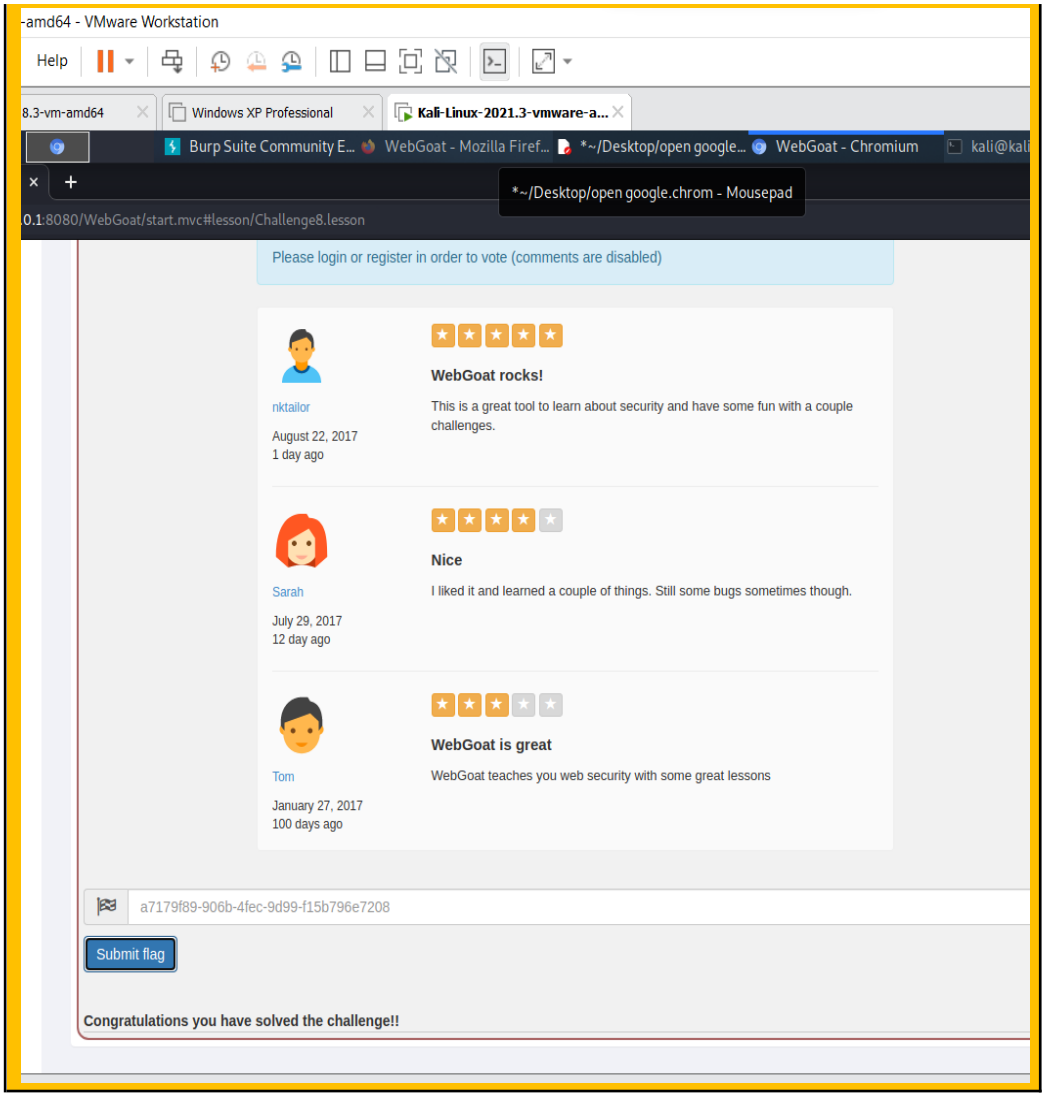
Prepared by: SY Ooi

Challenges Without Account :

Configure Burpsuite and Proxy Settings:
Find GET/WebGoat/challenge/8/vote/4 HTTP/1.1
Then change GET to HEAD



Prepared by: SY Ooi



Familiarize with *Capture-the-Flag (CTF)* – Stego Mission

Platform: Hack This Site (https://hackthissite.org/)

1. Continue your exercise at **hackthissite.org**

2. **Login** to your account.

3. On the left tab of the main page, in the **challenges** section, click " **Stego Missions**"

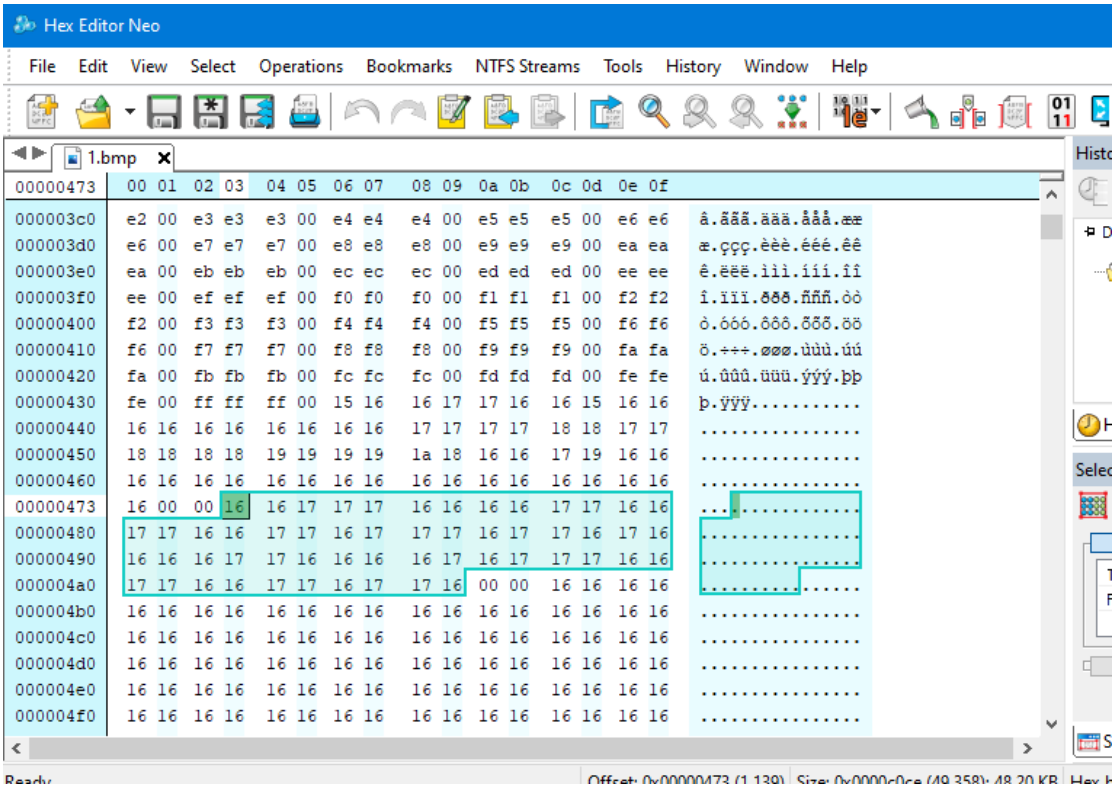
4. There are **seventeen (17)** Stego Missions. The goal in these missions is to extract the hidden message from the media file provided. Solve at least **six (6)** puzzles. You get one point per level completed.

Prepared by: SY Ooi

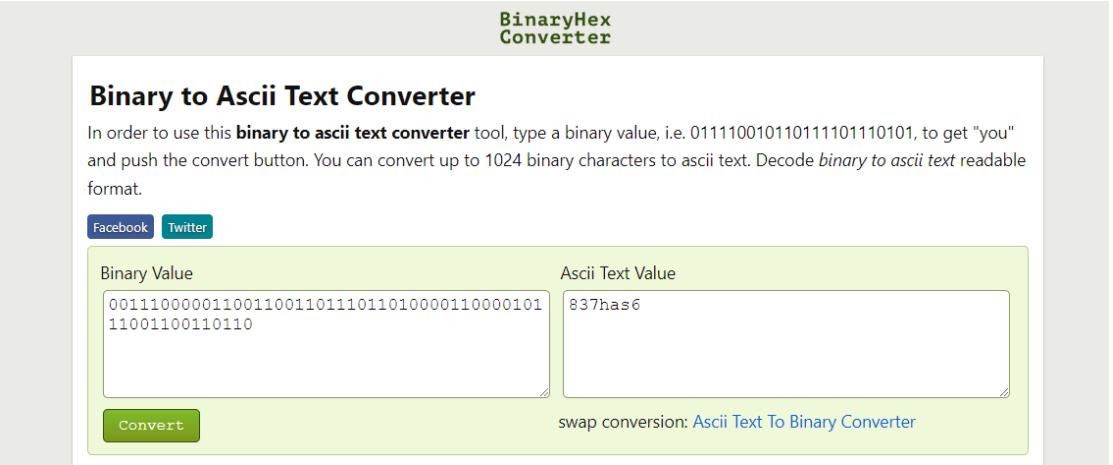
Show your steps on how you solved the challenge. [3%]

Notes: your account name must be shown when you snapshot your answer – to prove that you attempted the challenges/ puzzles by using your account.

1. **Mission 1**

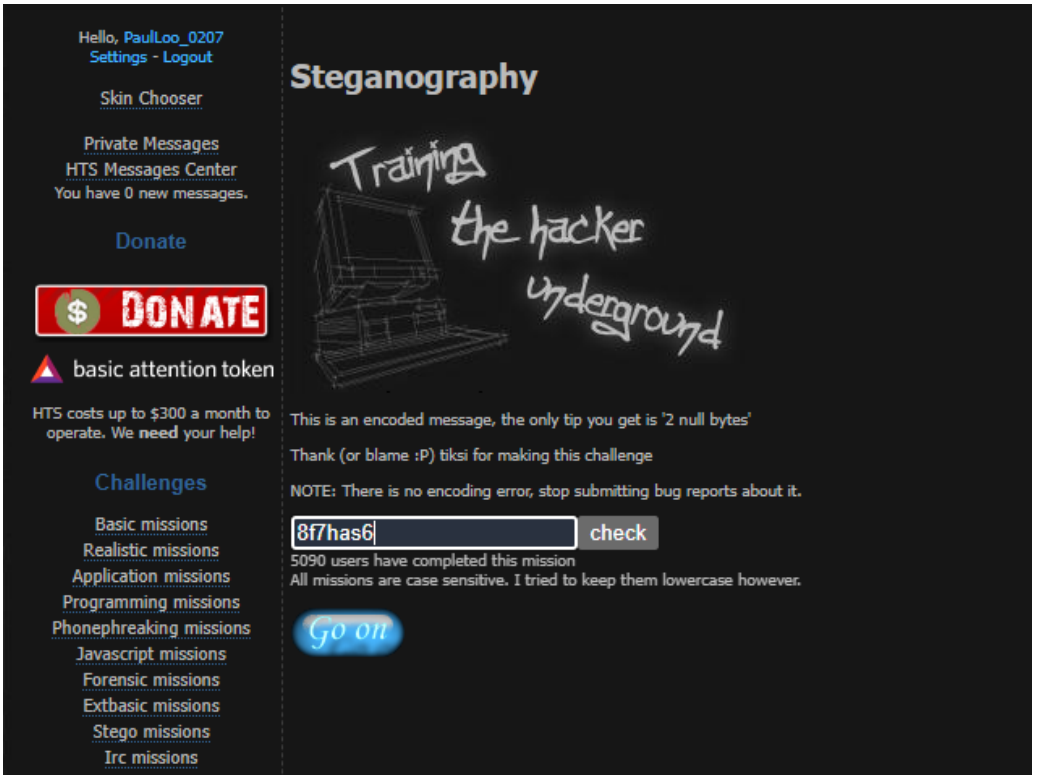


Download the image and copy the binary.



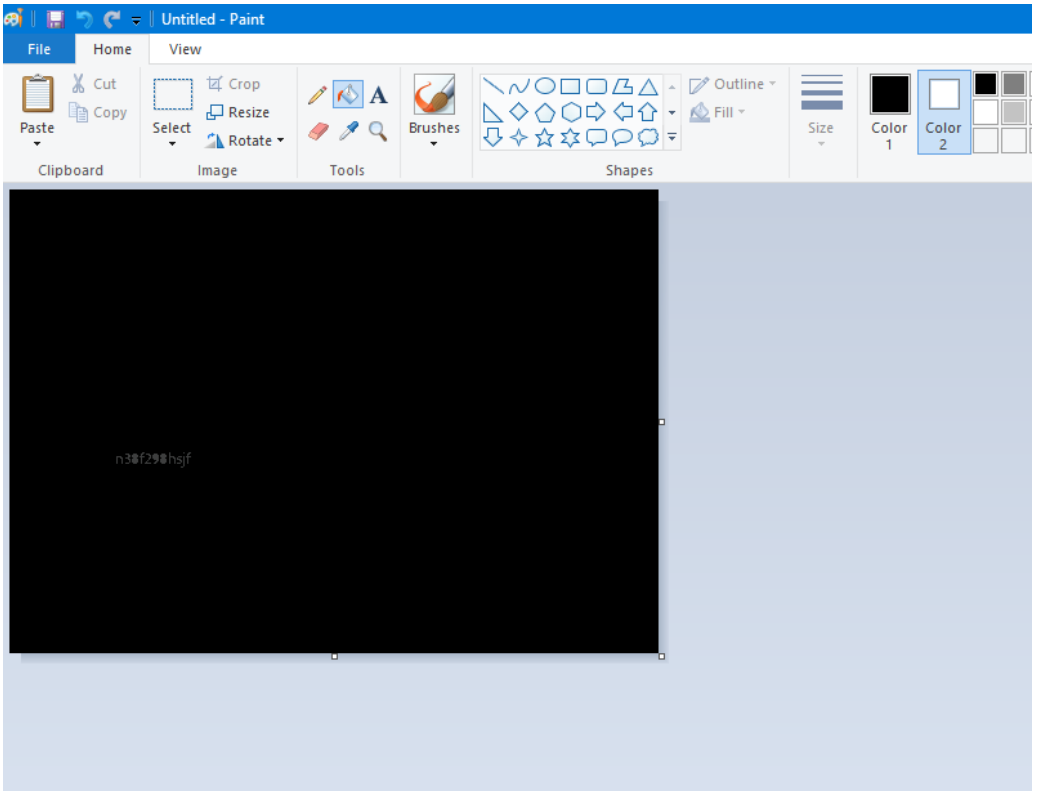
Use converter to convert binary to Ascii value.

Prepared by: SY Ooi



Type in and check, mission done.

2. Mission 3

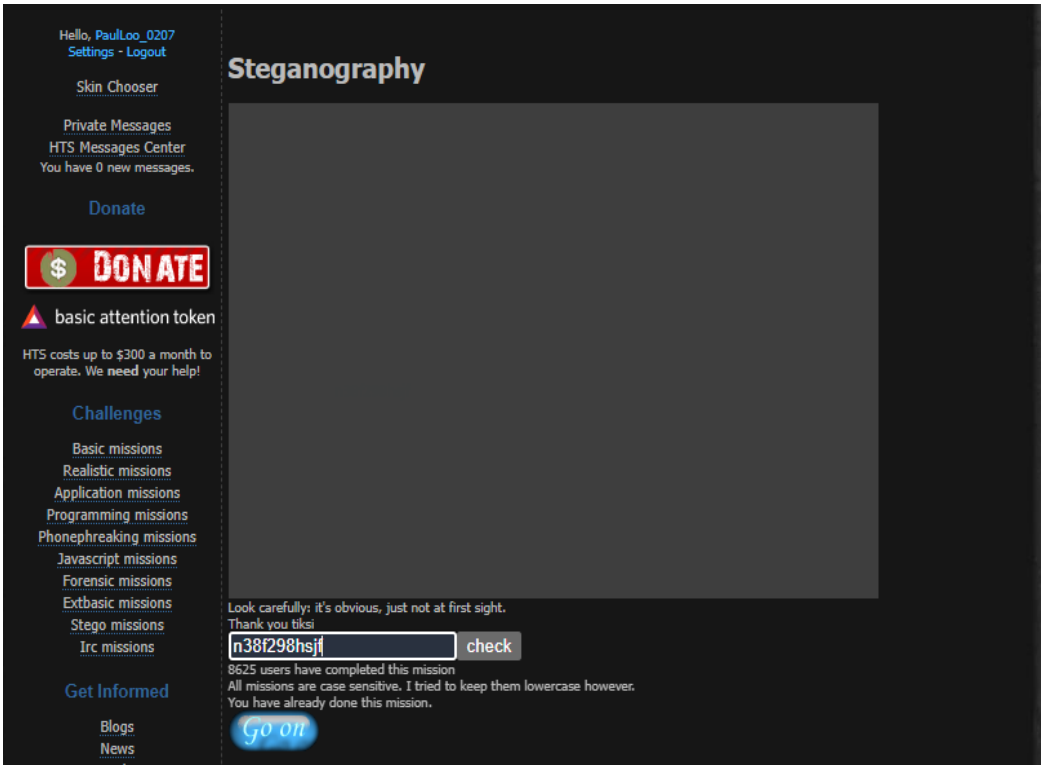


.

Prepared by: SY Ooi

Copy and Paste the picture to Paint. Use color 2 as white.

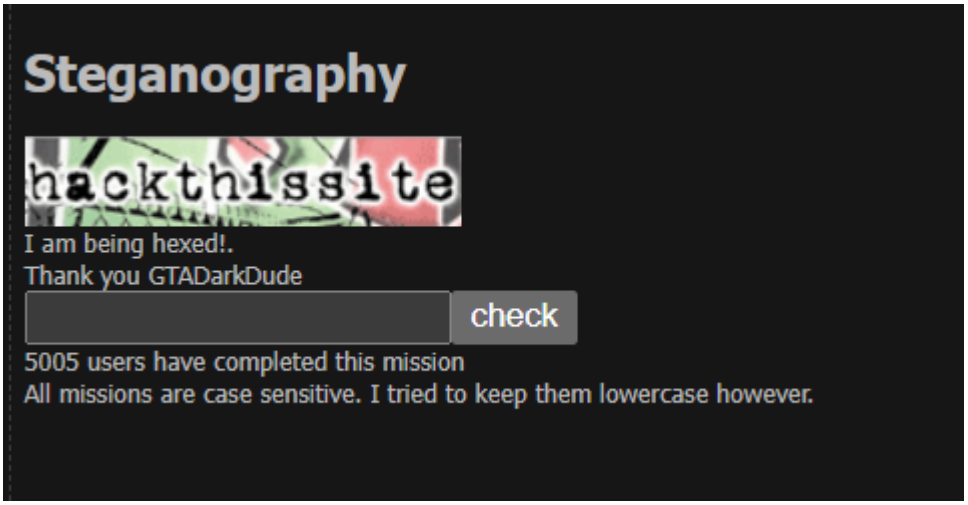
Zoom in and there is a clue there.



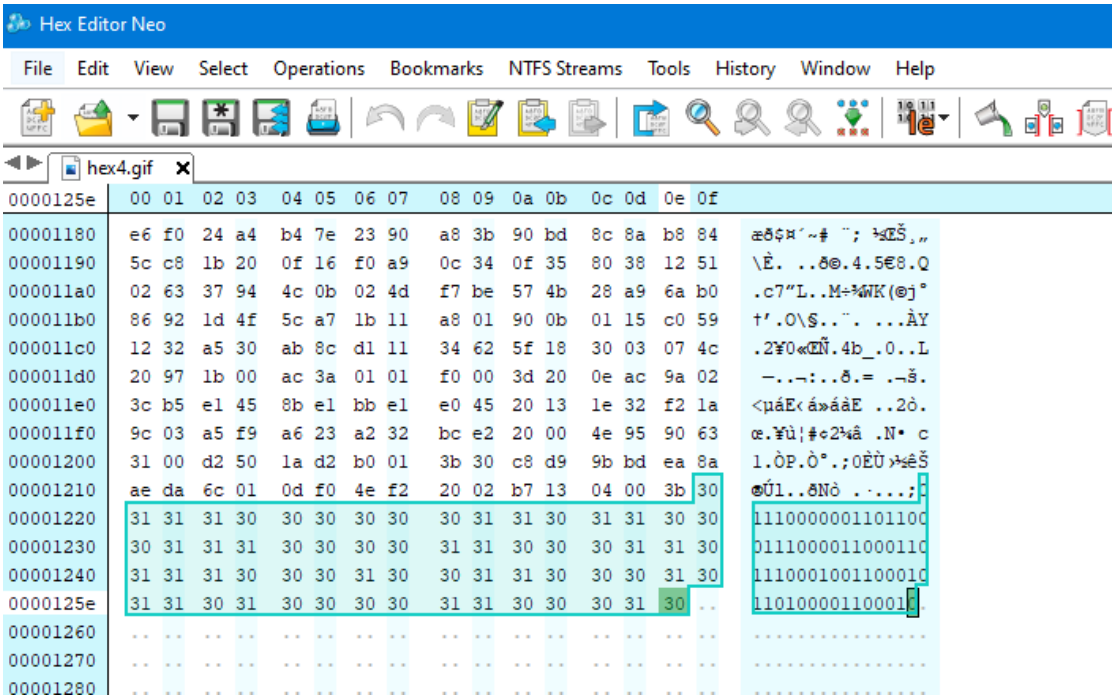
Key in the clue and completed.

Prepared by: SY Ooi

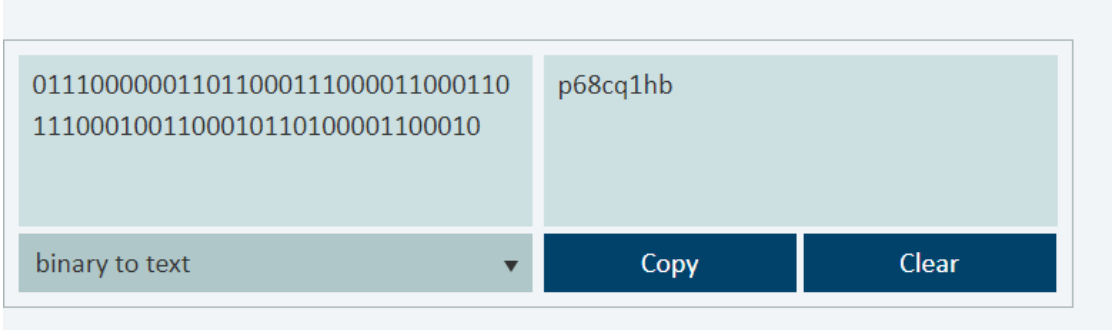
3. Mission 4



Download the image



Open the file and copy copy paste the hex code

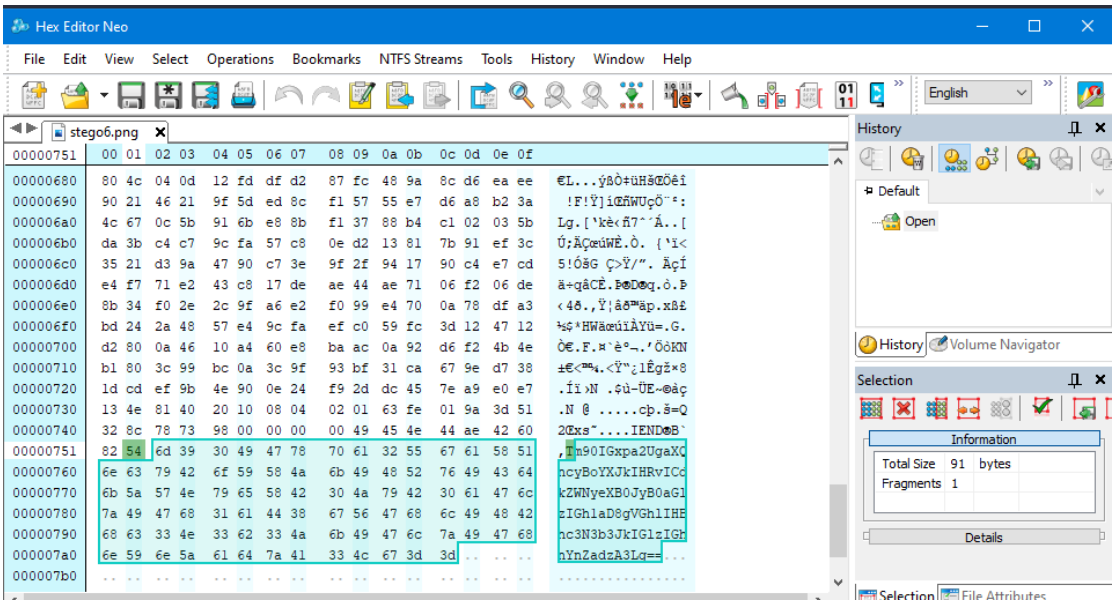


Copy the code and convert into text

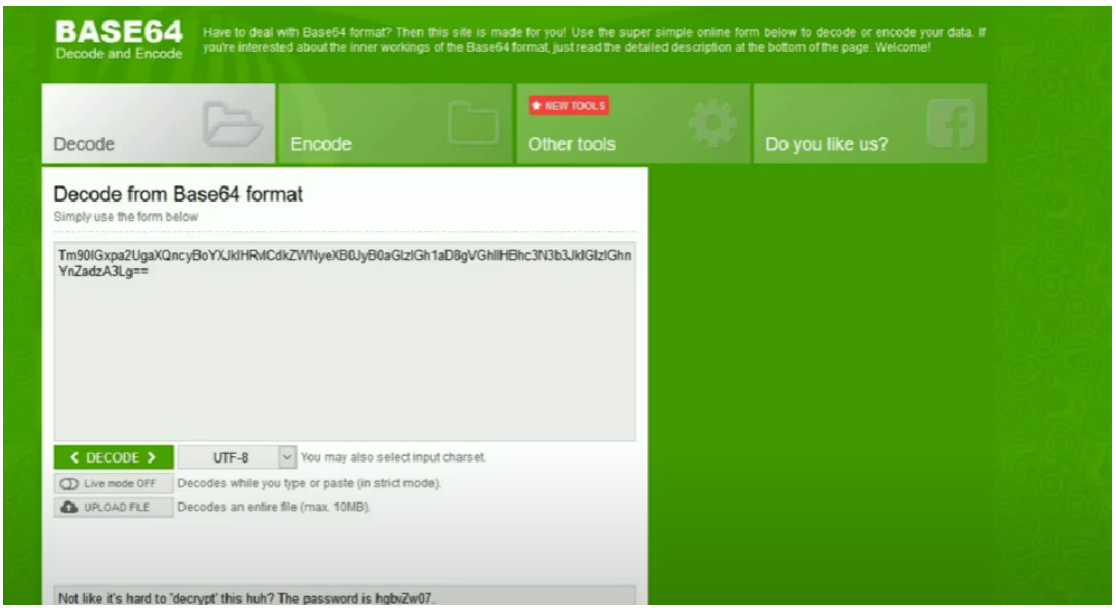
Prepared by: SY Ooi

Copy the text and key in it, check it and mission solved

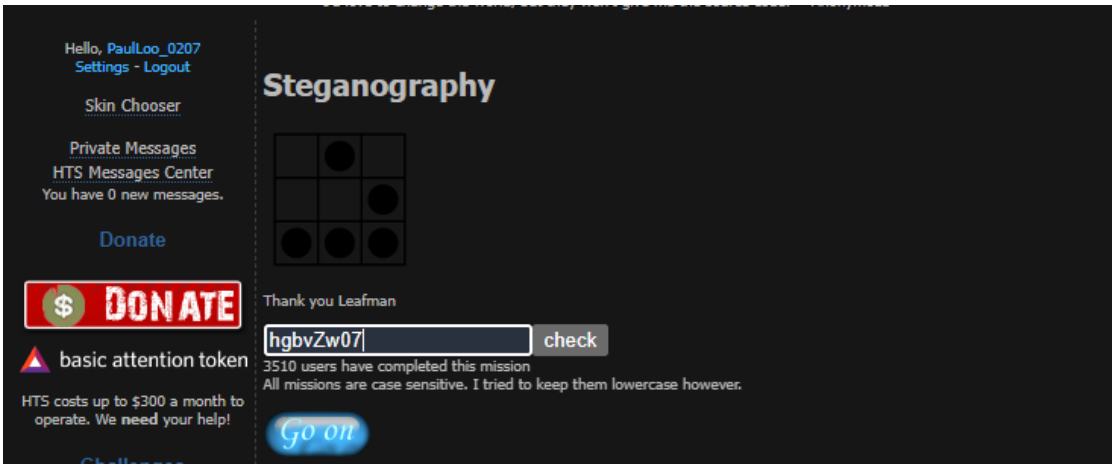
4. Mission 6



Download the image and open at hex dex



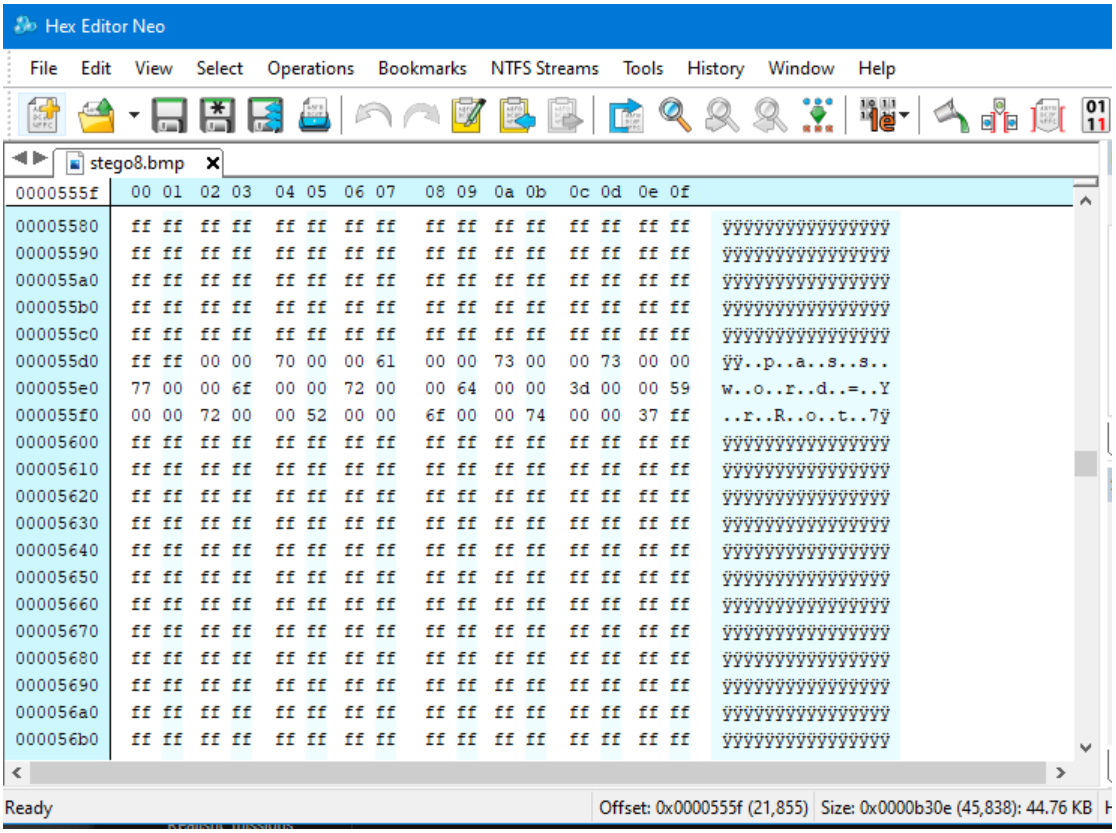
Copy the text and decode it



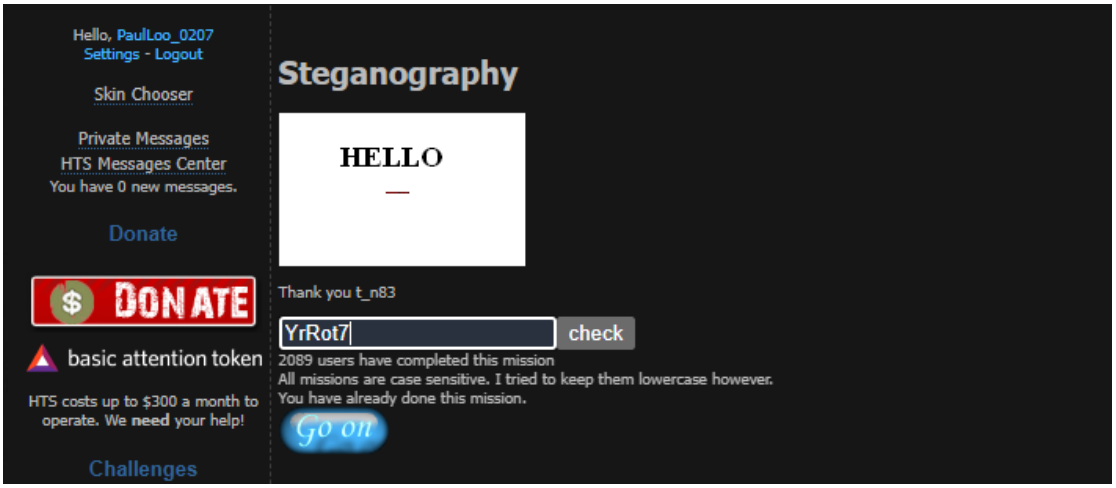
Type in and check, mission done

Prepared by: SY Ooi

5. Mission 8



Download the image, open the file and go to line 55d0. Passwords are shown there.



Type in and the password is there

Prepared by: SY Ooi

6. **Mission 10**

A second version of Bacon's cipher uses a unique code f

Letter	Code	Binary	Letter	Code	Binary
A	aaaaa	00000	N	abbab	01101
B	aaaab	00001	O	abbba	01110
C	aaaba	00010	P	abbbb	01111
D	aaabb	00011	Q	baaaa	10000
E	aabaa	00100	R	baaab	10001
F	aabab	00101	S	baaba	10010
G	aabba	00110	T	baabb	10011
H	aabbb	00111	U	babaa	10100
I	abaaa	01000	V	babab	10101
J	abaab	01001	W	babba	10110
K	ababa	01010	X	babbb	10111
L	ababb	01011	Y	bbaaa	11000
M	abbaa	01100	Z	bbaab	11001

Refer to bacon's cipher on wikipedia

After cracking the code, answer will be "thepasswordisnothere"

Type nothere and check it, mission done.

Password cracking

Machine: you can decide if you want to use Kali/ Windows, but the hashes that we are going to crack are the 3 hashes which you can found from the Windows XP (P3, P5, and P7)

Tool 1: THC Hydra

1. Explore the usage of THC Hydra in solving the similar problem as in Lab 12 (the 3 XP credentials).

Show your steps on how you used it. [1%]

```
┌──(kali㉿kali)-[~]
└─$ useradd -m p3
useradd: user 'p3' already exists

┌──(kali㉿kali)-[~]
└─$ useradd -m p5
useradd: user 'p5' already exists

┌──(kali㉿kali)-[~]
└─$ useradd -m p7
useradd: user 'p7' already exists
```

Add users p3 with abc password, add user p5 with abcde password, and Add user p7 with abcdefg password

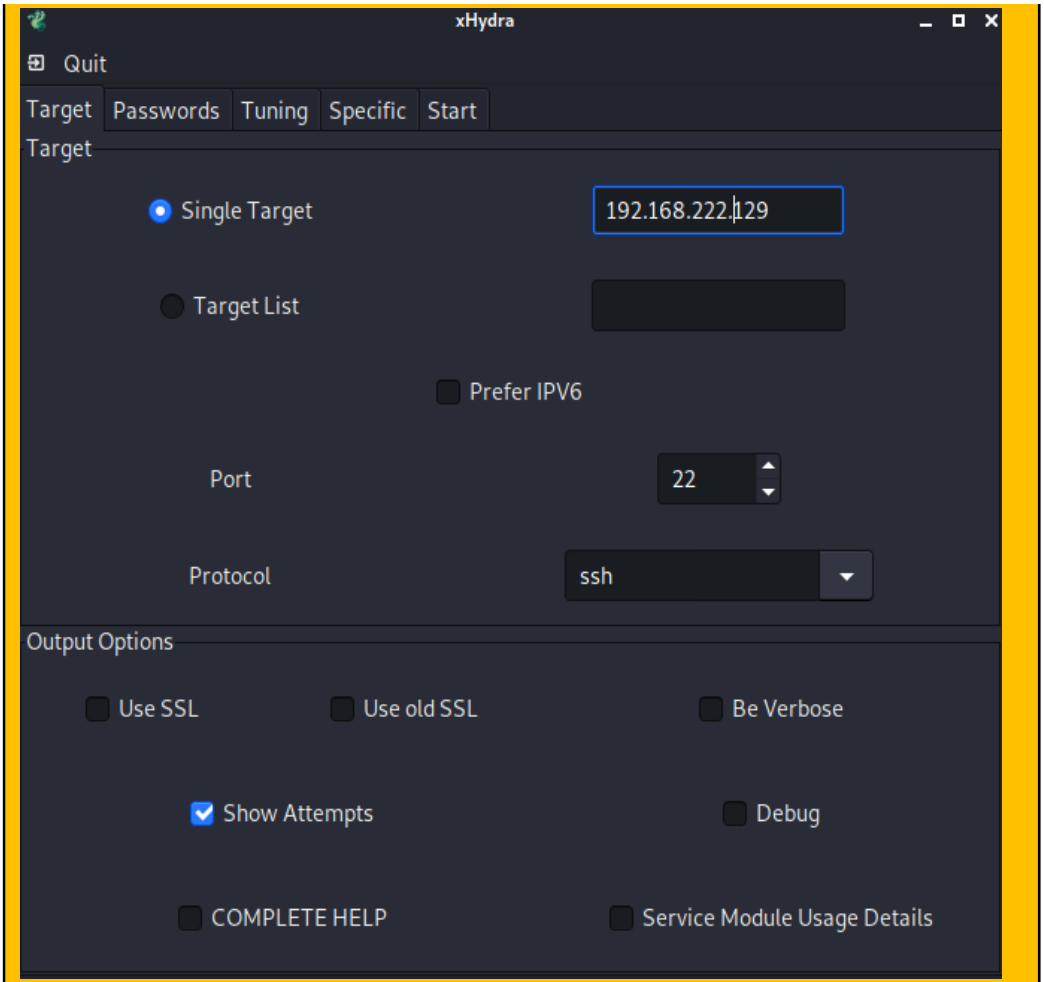
```
┌──(kali㉿kali)-[~]
└─$ sudo crunch 3 3 abc -o /home/kali/Desktop/numbriclist2.tx
Crunch will now generate the following amount of data: 108 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 27

crunch: 100% completed generating output

┌──(kali㉿kali)-[~]
└─$ sudo crunch 7 7 abcdefg -o /home/kali/Desktop/numbriclist.tx
Crunch will now generate the following amount of data: 6588344 bytes
6 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 823543

crunch: 100% completed generating output
```

```
┌──(kali㉿kali)-[~]
└─$ sudo crunch 5 5 abcde -o /home/kali/Desktop/numbriclist3.tx
[sudo] password for kali:
Crunch will now generate the following amount of data: 18750 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 3125

crunch: 100% completed generating output
```

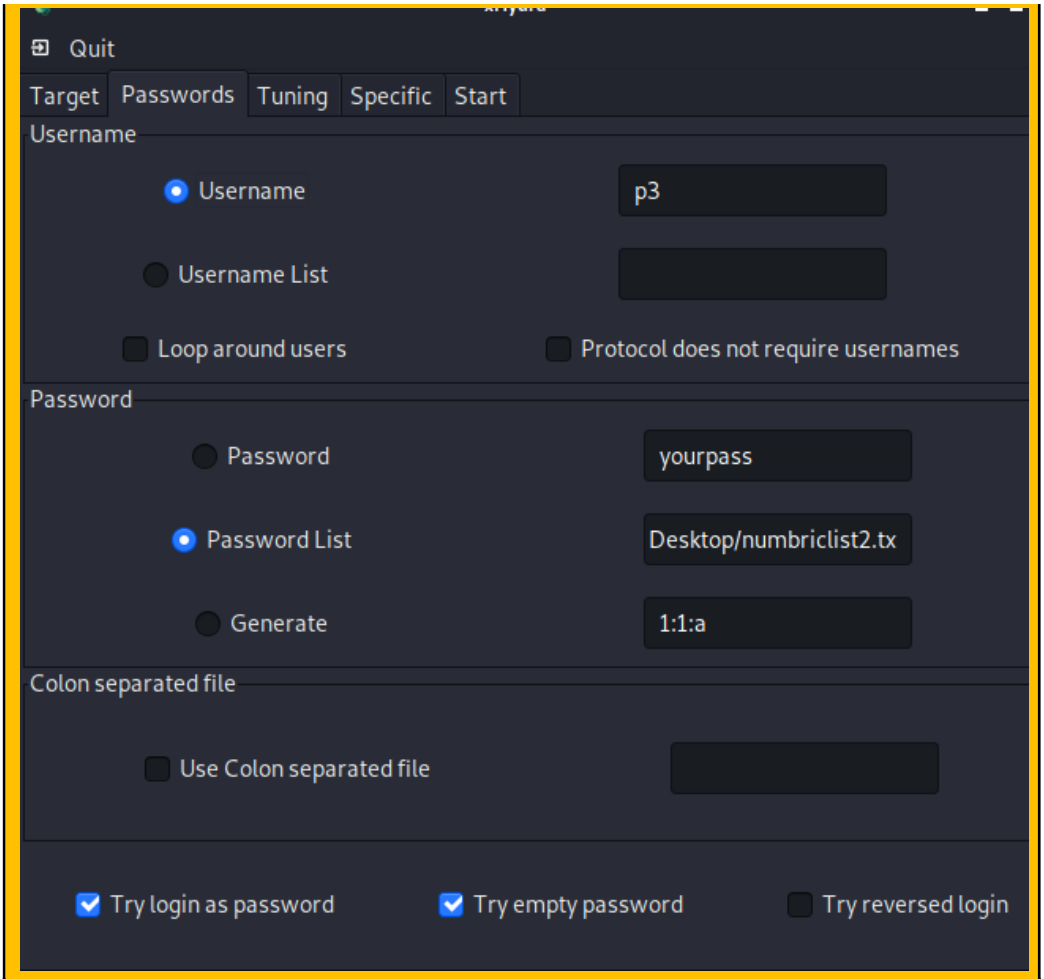
Create a numeric list for each user and set the to desktop

```
┌──(kali㉿kali)-[~]
└─$ systemctl start ssh.socket

┌──(kali㉿kali)-[~]
└─$ nmap 192.168.222.129 -p 22
Starting Nmap 7.91 ( https://nmap.org ) at 2022-04-10 21:01 EDT
Nmap scan report for 192.168.222.129
Host is up (0.00012s latency).

PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 2.18 seconds
```

Turn on SSH temporary and check if its open



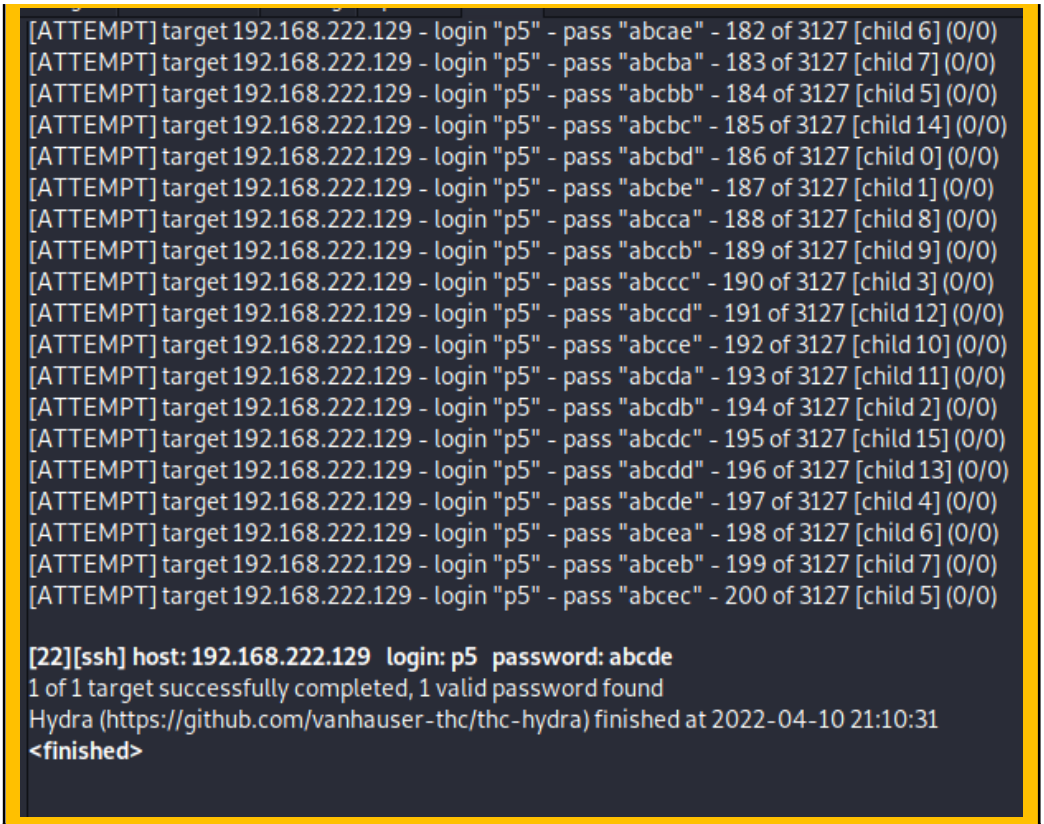
Set the kali ip and port 22 and protocol ssh with show attempts



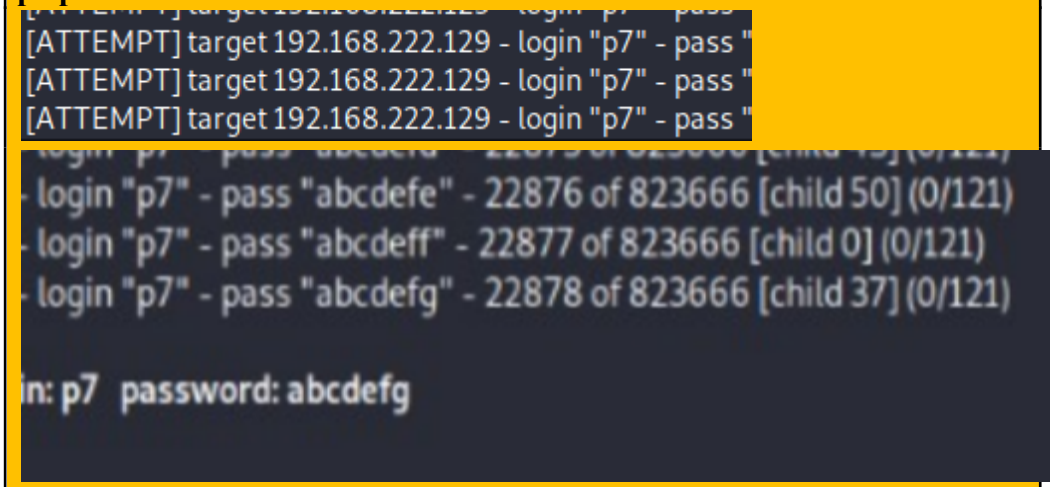
Set username p3 and password list to the numeric list you created for each user

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-10 21:02:24
[DATA] max 4 tasks per 1 server, overall 4 tasks, 29 login tries (l:1/p:29), ~8 tries per task
[DATA] attacking ssh://192.168.222.129:22/
[ATTEMPT] target 192.168.222.129 - login "p3" - pass "p3" - 1 of 29 [child 0] (0/0)
[ATTEMPT] target 192.168.222.129 - login "p3" - pass "" - 2 of 29 [child 1] (0/0)
[ATTEMPT] target 192.168.222.129 - login "p3" - pass "aaa" - 3 of 29 [child 2] (0/0)
[ATTEMPT] target 192.168.222.129 - login "p3" - pass "aab" - 4 of 29 [child 3] (0/0)
[ATTEMPT] target 192.168.222.129 - login "p3" - pass "aac" - 5 of 29 [child 1] (0/0)
[ATTEMPT] target 192.168.222.129 - login "p3" - pass "aba" - 6 of 29 [child 0] (0/0)
[ATTEMPT] target 192.168.222.129 - login "p3" - pass "abb" - 7 of 29 [child 1] (0/0)
[ATTEMPT] target 192.168.222.129 - login "p3" - pass "abc" - 8 of 29 [child 2] (0/0)
[ATTEMPT] target 192.168.222.129 - login "p3" - pass "aca" - 9 of 29 [child 3] (0/0)

[22][ssh] host: 192.168.222.129   login: p3   password: abc
```

Start it and you should see the password for p3



p5 password



p7 password

Tool 2: RainbowCrack

2. Explore the usage of RainbowCrack in solving the similar problem as in Lab 12 (the 3 XP credentials).

Show your steps on how you used it. [1%]
1st, install rainbow crack

```
└─# sudo apt-get install rainbowcrack                                    1 ×
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following NEW packages will be installed:
  rainbowcrack
0 upgraded, 1 newly installed, 0 to remove and 135 not upgraded.
Need to get 130 kB of archives.
After this operation, 506 kB of additional disk space will be used.
Get:1 http://kali.cs.nctu.edu.tw/kali kali-rolling/main amd64 rainbowcrack am
d64 1.8-0kali1 [130 kB]
Fetched 130 kB in 2s (79.2 kB/s)
Selecting previously unselected package rainbowcrack.
(Reading database ... 268043 files and directories currently installed.)
Preparing to unpack .../rainbowcrack_1.8-0kali1_amd64.deb ...
Unpacking rainbowcrack (1.8-0kali1) ...
Setting up rainbowcrack (1.8-0kali1) ...
Processing triggers for kali-menu (2021.3.3) ...
```

Then, generate rainbow table

```
┌──(root💀kali)-[/usr/share/rainbowcrack]
└─# rtgen ntlm loweralpha-numeric 1 7 0 3800 3554432 0
rainbow table ntlm_loweralpha-numeric#1-7_0_3800×3554432_0.rt parameters
hash algorithm:         ntlm
hash length:            16
charset name:           loweralpha-numeric
charset data:           abcdefghijklmnopqrstuvwxyz0123456789
charset data in hex:    61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72
 73 74 75 76 77 78 79 7a 30 31 32 33 34 35 36 37 38 39
charset length:         36
plaintext length range: 1 - 7
reduce offset:          0×00000000
plaintext total:        80603140212
```

```
655360 of 3554432 rainbow chains generated (1 m 39.2 s)
786432 of 3554432 rainbow chains generated (1 m 30.4 s)
917504 of 3554432 rainbow chains generated (1 m 29.4 s)
1048576 of 3554432 rainbow chains generated (1 m 29.7 s)
1179648 of 3554432 rainbow chains generated (1 m 32.7 s)
1310720 of 3554432 rainbow chains generated (1 m 31.1 s)
1441792 of 3554432 rainbow chains generated (1 m 29.2 s)
1572864 of 3554432 rainbow chains generated (1 m 29.8 s)
1703936 of 3554432 rainbow chains generated (1 m 29.0 s)
1835008 of 3554432 rainbow chains generated (1 m 33.8 s)
1966080 of 3554432 rainbow chains generated (1 m 29.7 s)
2097152 of 3554432 rainbow chains generated (1 m 29.2 s)
2228224 of 3554432 rainbow chains generated (1 m 29.4 s)
2359296 of 3554432 rainbow chains generated (1 m 32.8 s)
2490368 of 3554432 rainbow chains generated (1 m 31.0 s)
2621440 of 3554432 rainbow chains generated (1 m 29.7 s)
2752512 of 3554432 rainbow chains generated (1 m 29.5 s)
2883584 of 3554432 rainbow chains generated (1 m 33.4 s)
3014656 of 3554432 rainbow chains generated (1 m 29.4 s)
3145728 of 3554432 rainbow chains generated (1 m 13.5 s)
3276800 of 3554432 rainbow chains generated (1 m 4.3 s)
3407872 of 3554432 rainbow chains generated (1 m 3.9 s)
3538944 of 3554432 rainbow chains generated (1 m 3.2 s)
3554432 of 3554432 rainbow chains generated (0 m 7.5 s)
```

Then, redirect back to the rainbow crack file directory to sort the rainbow table.

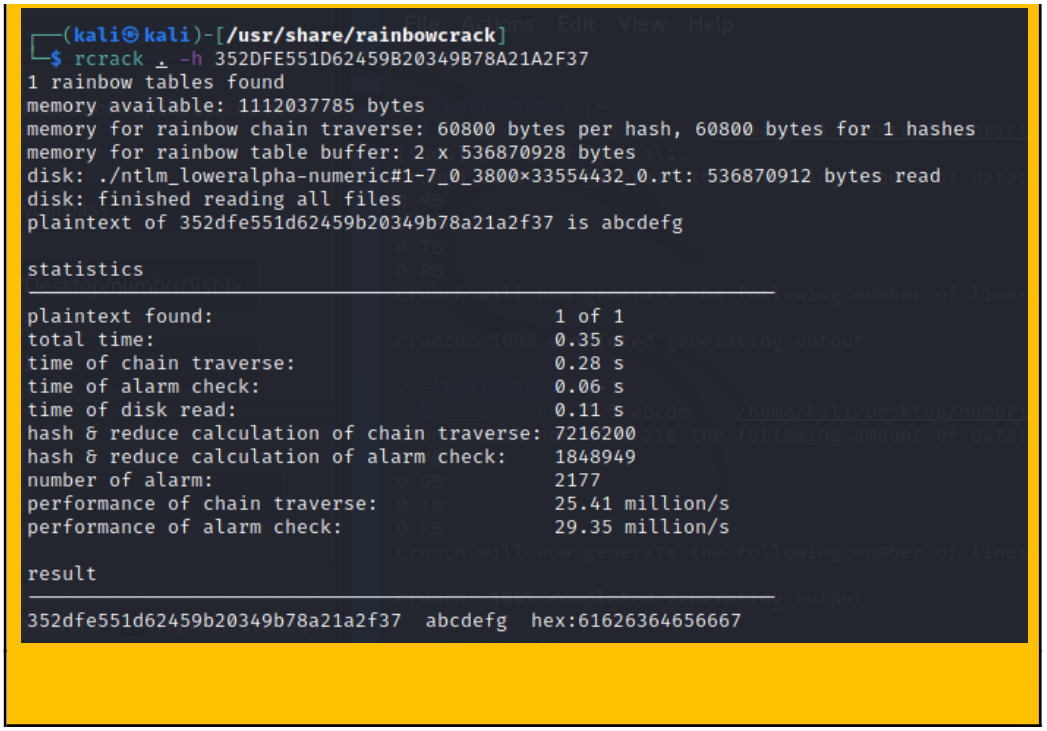
```
┌──(root💀kali)-[/usr/share/rainbowcrack]
└─# rtsort .
./ntlm_loweralpha-numeric#1-7_0_3800x3554432_0.rt:
3012726784 bytes memory available
loading data ...
sorting data ...
writing sorted data ...
```

```
┌──(kali㉿kali)-[/usr/share/rainbowcrack]
└─$ rcrack . -h E0FBA38268D0EC66EF1CB452D5885E53
1 rainbow tables found
memory available: 1113594265 bytes
memory for rainbow chain traverse: 60800 bytes per hash, 60800 bytes for 1 hashes
memory for rainbow table buffer: 2 x 536870928 bytes
disk: ./ntlm_loweralpha-numeric#1-7_0_3800x33554432_0.rt: 536870912 bytes read
disk: finished reading all files
plaintext of e0fba38268d0ec66ef1cb452d5885e53 is abc

statistics
--------------------------------------------------------------------------------
plaintext found:                        1 of 1
total time:                             0.29 s
time of chain traverse:                 0.29 s
time of alarm check:                    0.00 s
time of disk read:                      0.13 s
hash & reduce calculation of chain traverse: 7216200
hash & reduce calculation of alarm check:    5973
number of alarm:                        133
performance of chain traverse:          24.88 million/s
performance of alarm check:             5.97 million/s

result
--------------------------------------------------------------------------------
e0fba38268d0ec66ef1cb452d5885e53  abc  hex:616263
```

p3

```
┌──(kali㉿kali)-[/usr/share/rainbowcrack]
└─$ rcrack . -h 3F5156E39D9C989C2609FD8329A46CA4
1 rainbow tables found
memory available: 1113204326 bytes
memory for rainbow chain traverse: 60800 bytes per hash, 60800 bytes for 1 hashes
memory for rainbow table buffer: 2 x 536870928 bytes
disk: ./ntlm_loweralpha-numeric#1-7_0_3800x33554432_0.rt: 536870912 bytes read
disk: finished reading all files
plaintext of 3f5156e39d9c989c2609fd8329a46ca4 is abcde

statistics
--------------------------------------------------------------------------------
plaintext found:                        1 of 1
total time:                             0.31 s
time of chain traverse:                 0.31 s
time of alarm check:                    0.00 s
time of disk read:                      0.12 s
hash & reduce calculation of chain traverse: 7216200
hash & reduce calculation of alarm check:    6457
number of alarm:                        128
performance of chain traverse:          22.98 million/s
performance of alarm check:             6.46 million/s

result
--------------------------------------------------------------------------------
3f5156e39d9c989c2609fd8329a46ca4  abcde  hex:6162636465
```

p5


┌──(kali㉿kali)-[/usr/share/rainbowcrack]
└─$ rcrack . -h 352DFE551D62459B20349B78A21A2F37
1 rainbow tables found
memory available: 1112037785 bytes
memory for rainbow chain traverse: 60800 bytes per hash, 60800 bytes for 1 hashes
memory for rainbow table buffer: 2 x 536870928 bytes
disk: ./ntlm_loweralpha-numeric#1-7_0_3800×33554432_0.rt: 536870912 bytes read
disk: finished reading all files
plaintext of 352dfe551d62459b20349b78a21a2f37 is abcdefg
statistics
plaintext found: 1 of 1
total time: 0.35 s
time of chain traverse: 0.28 s
time of alarm check: 0.06 s
time of disk read: 0.11 s
hash & reduce calculation of chain traverse: 7216200
hash & reduce calculation of alarm check: 1848949
number of alarm: 2177
performance of chain traverse: 25.41 million/s
performance of alarm check: 29.35 million/s
result
352dfe551d62459b20349b78a21a2f37 abcdefg hex:61626364656667