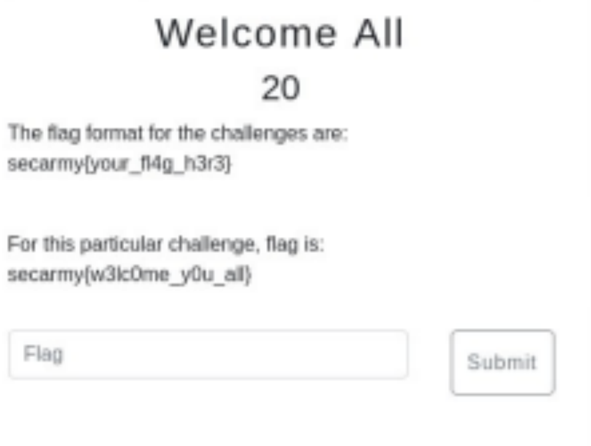
SECARMY CTF Write-up

Challenge1:- Welcome all:

Sol: click on the challenge and get the flag



Welcome All

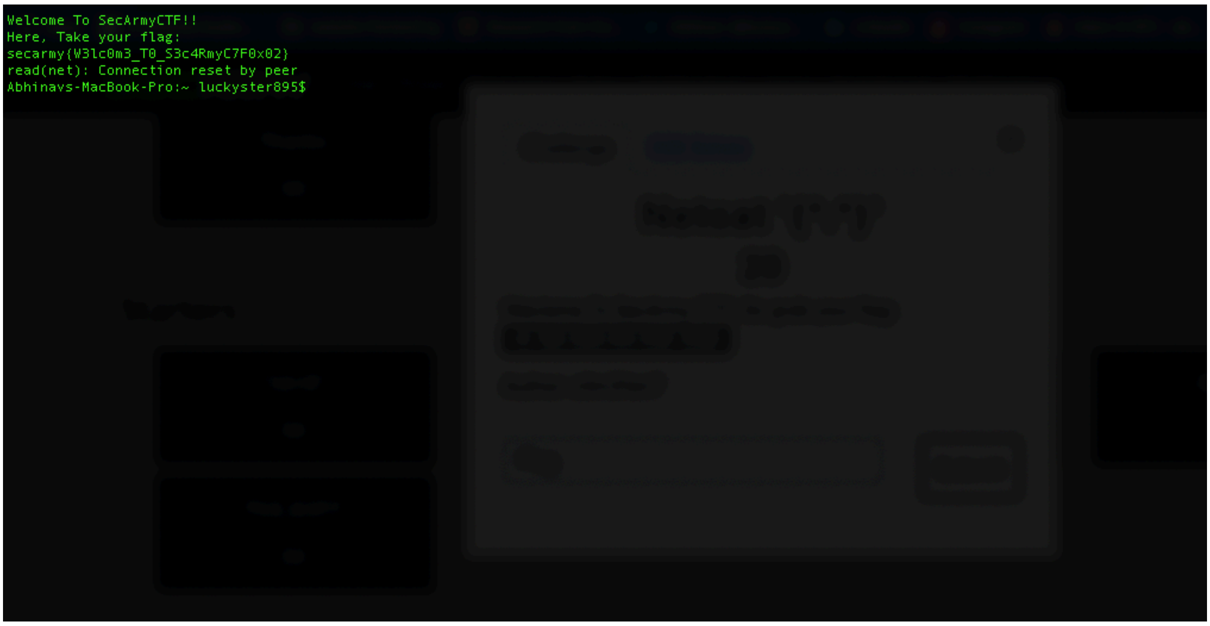
20

The flag format for the challenges are:
secarmy{your_fl4g_h3r3}

For this particular challenge, flag is:
secarmy{w3lc0me_y0u_all}

Challenge 2:- Netcat(nc)

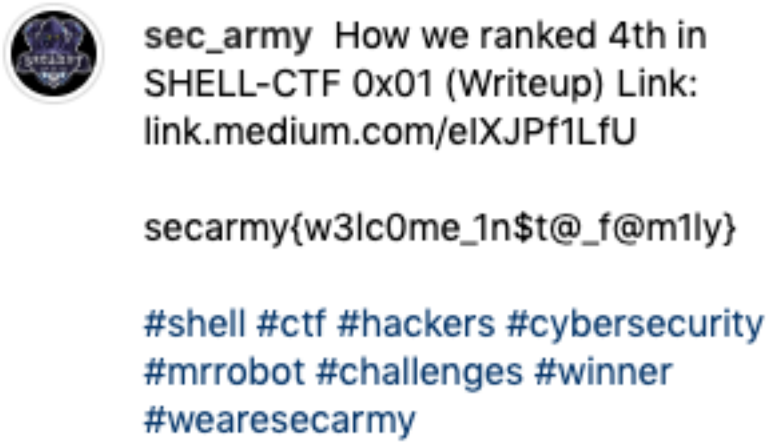
Sol:-   just use the command given “nc 68.183.44.136 2200”



```
Welcome To SecArmyCTF!!
Here, Take your flag:
secarmy{W3lc0m3_T0_S3c4RmyC7F0x02}
read(net): Connection reset by peer
Abhinavs-MacBook-Pro:~ luckyster895$
```

Challenge 3: Instafamous

Sol: checkout the first post on secarmy’s Instagram page



sec_army How we ranked 4th in SHELL-CTF 0x01 (Writeup) Link: link.medium.com/eIXJPf1LfU

secarmy{w3lc0me_1n$t@_f@m1ly}

#shell #ctf #hackers #cybersecurity #mrrobot #challenges #winner #wearesecarmy

Now coming to Starters challenge:

Challenge 2: Die Basis:

there was two text files in a zip :
1.file1.txt
2.file2.txt

file1.txt**: c2VjYXJteXtmbEBnXzFzXw==**
this was of base64 format on decoding we get "secarmy{fl@g_1s_ "

file2.txt: **L52GQM27MJAHGM35GMZA====**
this was of base32 format joining the text values of both the format we get the flag is:
secarmy{fl@g_1s_th3_b@s3}

Challenge 3: Easy Capture

Just unzip the flagmin.zip we get flagmain.txt and boom it's a binary format

01110011 01100101 01100011 01100001 01110010 01101101 01111001 01111011
01101000 00110011 01110010 00110011 01011111 01111001 00110000 01110101
01011111 01100011 01000000 01110000 01110100 01110101 01110010 00110011
01111101

Converting into text we get: secarmy{h3r3_y0u_c@ptur3}

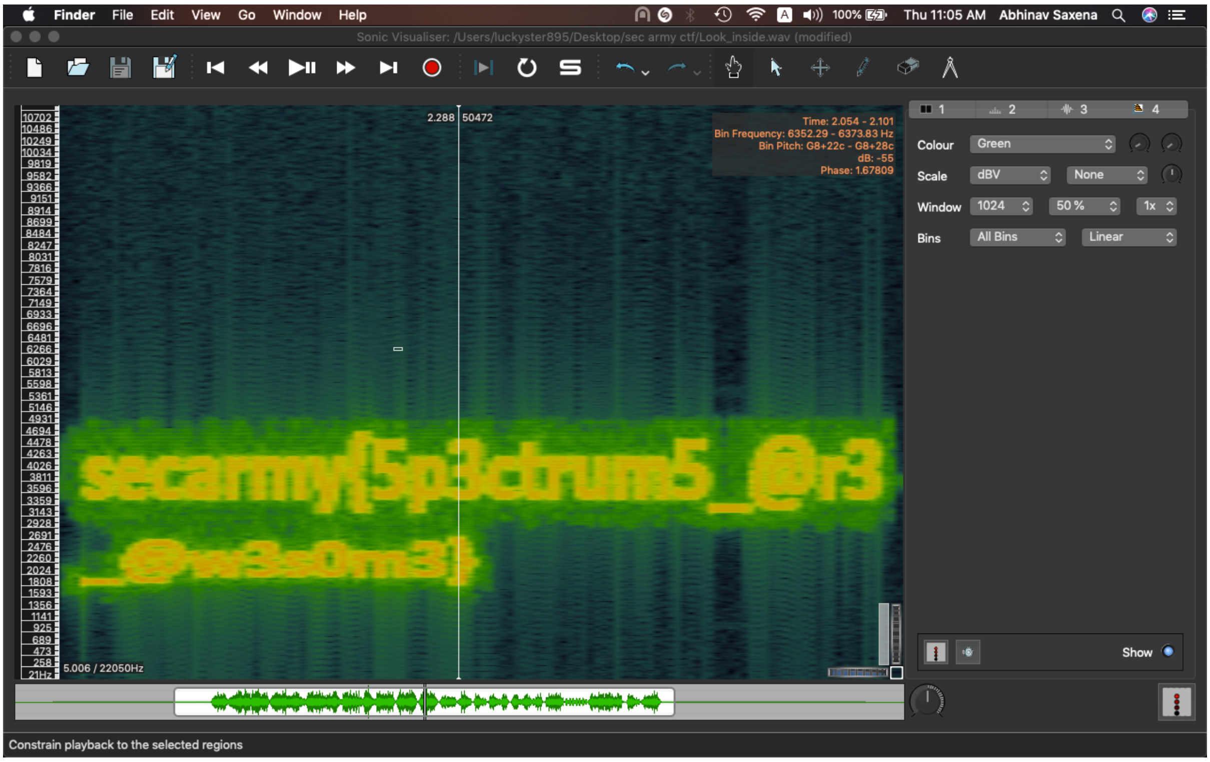
Now coming to Misc challenges

Challenge 1: Directories:

Question: It is a type of illusionary filesystem. It does not exist on a disk. Can U name it ?
Flag: secarmy{/proc}

Challenge 2: Look inside

we get a Look_inside.wav file



Sonic Visualiser: /Users/luckyster895/Desktop/sec army ctf/Look_inside.wav (modified)

Time: 2.054 - 2.101
Bin Frequency: 6352.29 - 6373.83 Hz
Bin Pitch: G8+22c - G8+28c
dB: -55
Phase: 1.67809

Colour Green

Scale dBV None

Window 1024 50 % 1x

Bins All Bins Linear

Constrain playback to the selected regions

Flag: secarmy{5p3ctrum5_@r3_@w3s0m3}

Challenge 3: Prizes

link: https://ctf.sec.army/prizes

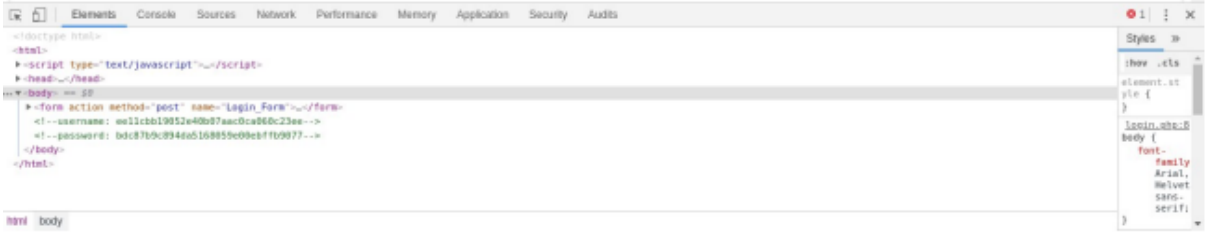
Now view the source code of this page and find “One step closer to the Prize”
you get: c2VjYXJteXtzMHVyYzNfaTVfbjNjZXM1YXJ5fQo= so this is base 64 encoded

decode to text and get the flag

Flag: secarmy{s0urc3_i5_n3ces5ary}

Challenge 4 : Web salad

open the Source of the page and you will see two hashes:



```
<!doctype html>
<html>
 ▶<script type="text/javascript">…</script>
 ▶<head>…</head>
…▼<body> == $0
   ▶<form action method="post" name="Login_Form">…</form>
    <!--username: ee11cbb19052e40b07aac0ca060c23ee-->
    <!--password: bdc87b9c894da5168059e00ebffb9077-->
  </body>
</html>
```

```
element.style {
}
login.php:8
body {
  font-
    family
    Arial,
    Helvet
    sans-
    serif;
}
```

it is md5 hash: decrypt it and get the
username: user
password: password1234
after login to page again go to inspect element
we get c2VjYXJteXt3M2JfYnVjazN0XzNuYzB1bjdlcjNkfQo=
Decode this string base 64 to text and we get the flag

FLAG: secarmy{w3b_buck3t_3nc0un7er3d}