Incognito 2.0 CTF

CHALLENGES' WRITE-UPS

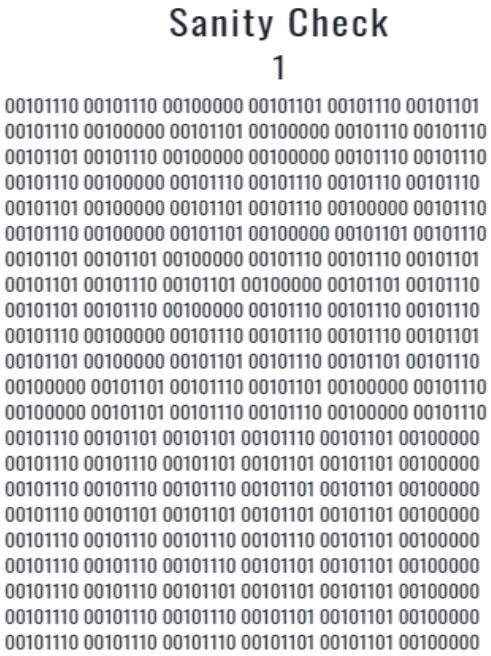
17-18 APRIL 2021

Team Kali Billi

Contents

EASY

Sanity Check



Steps:

- You will find these binary digits in the question.

```
00101110 00101110 00100000 00101101 00101110 00101101 00101110 00100000
00101101 00100000 00101110 00101110 00101101 00101110 00100000 00100000
00101110 00101110 00101110 00100000 00101110 00101110 00101110 00101110
00101101 00100000 00101101 00101110 00100000 00101110 00101110 00100000
00101101 00100000 00101101 00101110 00101101 00101101 00100000 00101110
00101110 00101101 00101101 00101110 00101101 00100000 00101101 00101110
00101101 00101110 00100000 00101110 00101110 00101110 00101110 00100000
00101110 00101110 00101110 00101101 00101101 00100000 00101101 00101110
00101101 00101110 00100000 00101101 00101110 00101101 00100000 00101110
00100000 00101101 00101110 00101110 00100000 00101110 00101110 00101101
00101101 00101110 00101101 00100000 00101110 00101110 00101101 00101101
00101101 00100000 00101110 00101110 00101110 00101101 00101101 00100000
00101110 00101101 00101101 00101101 00101101 00100000 00101110 00101110
00101110 00101110 00101101 00100000 00101110 00101110 00101110 00101101
00101101 00100000 00101110 00101110 00101101 00101101 00101101 00100000
00101110 00101110 00101110 00101101 00101101 00100000 00101110 00101110
00101110 00101101 00101101 00100000
```

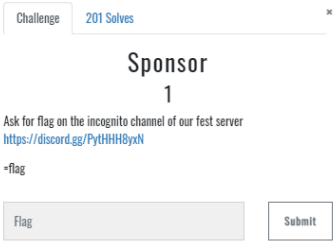
- Convert them and you will find a morse code like this:

```
.. -.-. - ..-.  ... ....- -. .. - -.-- ..--.- -.-. .... ...-- -.-. -.- .
-.. ..--.- ..--- ...-- .---- ....- ...-- ..--- ...-- ...--
```

- After converting this morse code into text from any online converter you will get your flag.

Flag: `ICTF{s4nity_ch3cked_23143233}`

Sponsor

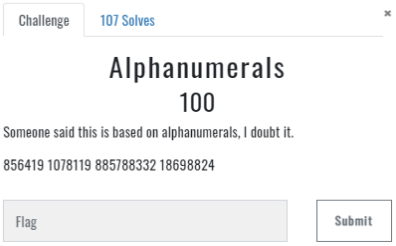


Steps:

- Join the discord channel given in the description of the challenge.
- Go to the `#incognito` channel and type `=flag`.
- The bot will dm you the flag.

Flag = `ICTF{Fl4g_F0und_0n_Di5c0rd}`

Alphanumerals



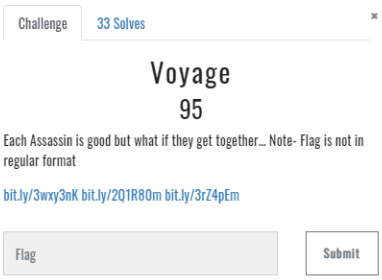
Steps:

- Alphanumerals means an encoding based on alphabets and numbers.
- The given alphanumeric string is a `Base36 string`, so we found an online base36 decoder.
- After decoding this string, we found our flag.

Flag = `ICTF{n3vr_ending_b4s3s}`

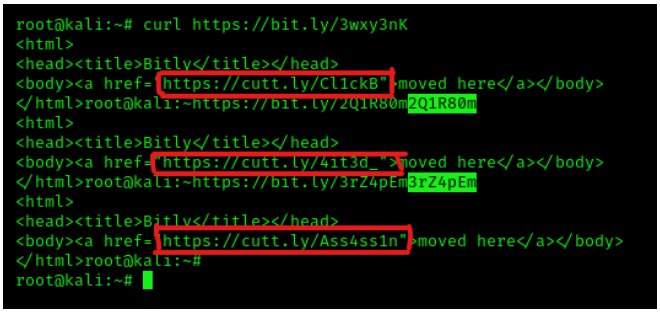
(We had first blood in this)

Voyage



Steps:

- There are 3 bit.ly links given, so let's try to visit them.
- When we visit those 3 links, it redirected us to some website with wallpapers. So, no luck with visiting those links directly.
- Then after using `curl` command with those links and we found something.



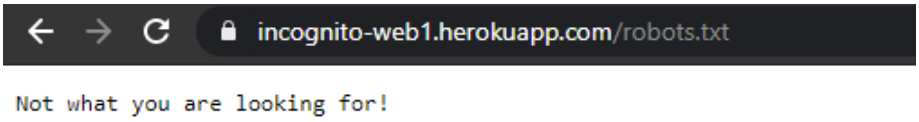
- After looking closely, you can see that those links were redirected from cutt.ly links. After concatenating those 3 redirected links we found our flag.

Flag = `ICTF{Cl1ckB4it3d_Ass4ss1n}`

Not Common

Steps:

- After visiting the link mentioned in the challenge, all you can see is an image of a robot.
- As the challenge mentions that it is a most common practice that people need to do it and we see a robot, so we decided to check for `/robots.txt`.
- But it was a rabbit hole.



- We are back to square one. Now we did a directory scan on the given site in the hope to find some admin panel or something.
- After some seconds of scan on `dirbuster` tool in Linux, we found a `/security.txt` file.
- After visiting the `https://incognito-web1.herokuapp.com/security.txt` we found the flag.

Flag = `ICTF{F0R_Bug_Hunt3rs}`

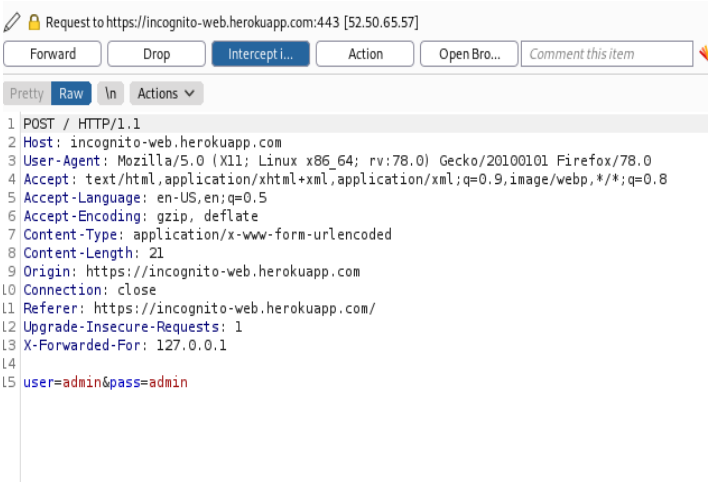
Simple Login

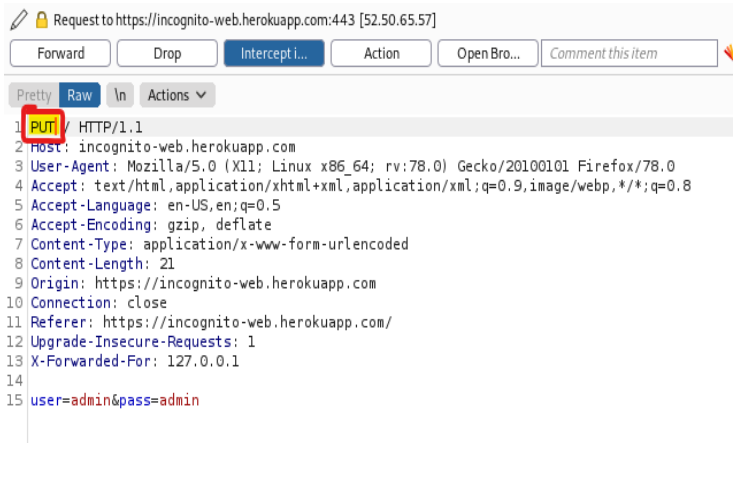
Steps:

- After visiting the link given in the challenge we see a login page so we try the most common username and password that is `admin:admin`.
- It throws this message:

> **You are not who you say to be, bcz he has put some extra measures to prevent this scenario**

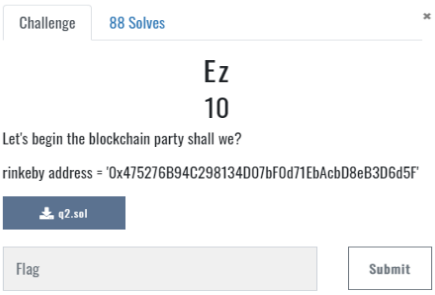
- Now there is a hint in this statement, the keyword like `"PUT"` so we tried to capture the request on `burpsuite` and see a POST request being sent.
- Using the hint from the keyword `"Put"`, we try to change the header of the request from `post to put` and we get the flag.





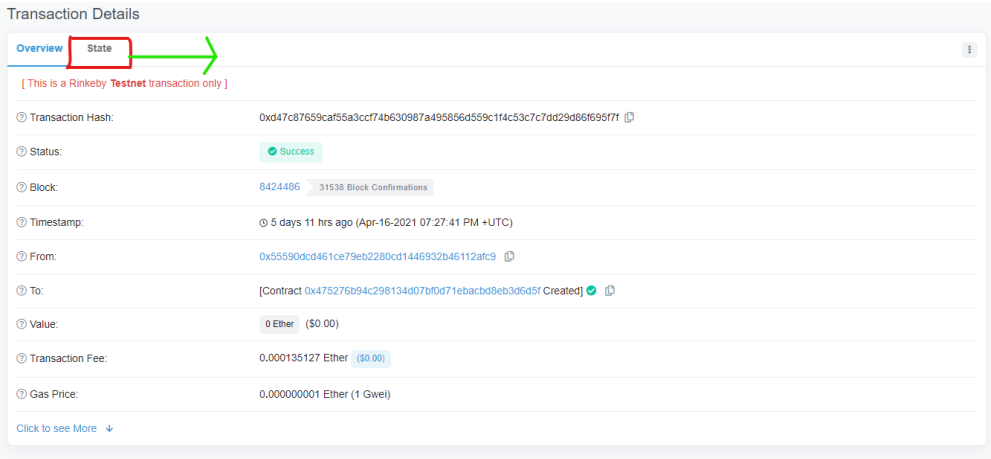
Flag = `ICTF{N0T_S0_S3Cur3d_853541}`

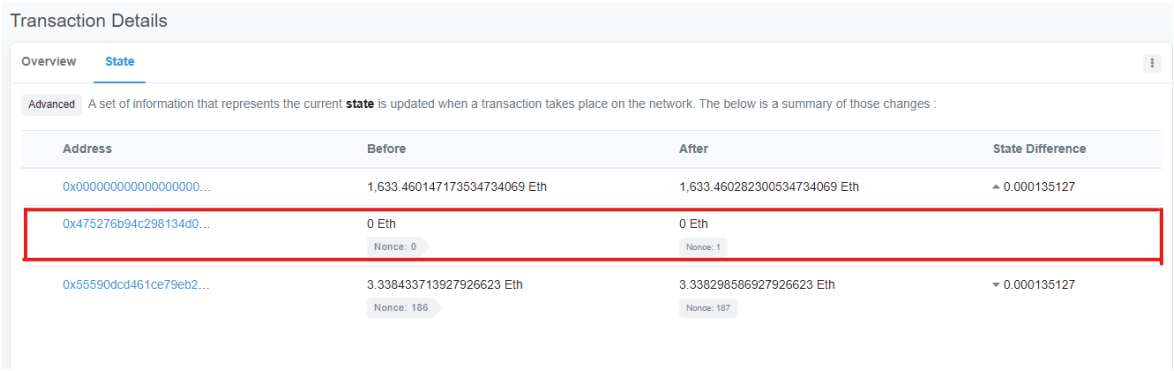
EZ



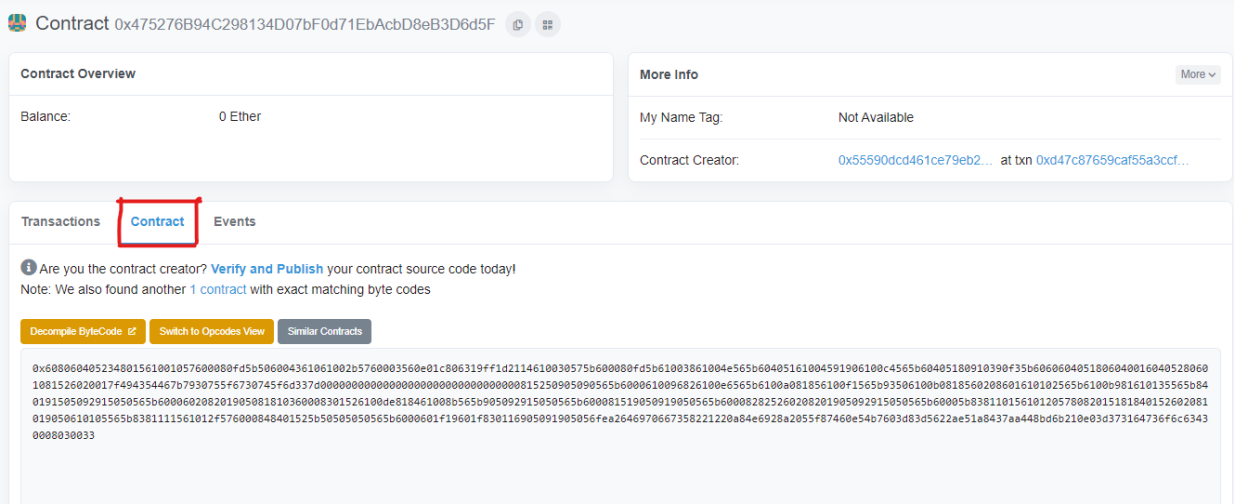
Steps:

- This challenge is based on crypto currency. Here we are given a rinkeby address = `'0x475276B94C298134D07bF0d71EbAcbD8eB3D6d5F'`
- Now after learning about rinkeby and contracts in cryptocurrencies, we can across a website which gave us information about contracts and transactions by using the address given above.
- Using https://rinkeby.etherscan.io/ we searched for this address and found a transaction detail.
- View those details and go to the `"State"` section.





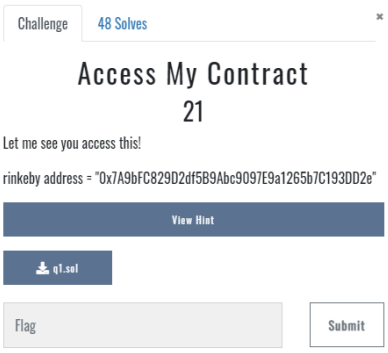
- Now visit the second contract in the `"State"` section and then open the contract part in that.



- You will see a `contract ByteCode`, now click on the `Decompile ByteCode` and then the ByteCode will be decompiled, and you will see the flag in the decompiled section.

Flag = `ICTF{y0u_g0t_m3}`

Access My Contract



Steps:

- Just like the last Challenge "EZ" we were given a Rinkeby Address = '0x7A9bFC829D2df5B9Abc9097E9a1265b7C193DD2e'
- Following all the steps of the above challenge we come up to ByteCode Decompiler and after decompiling we get a huge code.
- We try to google some of the lines of code to find out what that is and when we google `revert with 'NH{q', 34`, we find a website with the same code and a flag in it.

  Flag = `ICTF{Crypt0_is_Fun}`

Library

Steps:

- This challenge was a boot2root challenge hosted on TryHackMe.
- We needed to find 2 files from this machine. User.txt and the other was Root.txt.
- So, first we did a basic `nmap` scan, but we found nothing but only 1 open http port which indicated to visit the website on that port.
- There was a login page. We tried some common usernames and passwords and then we tried to perform a sql injection and this worked and gave us a successful login.
  `username: admin and password: admin' or '1'='1`
- So, now we are logged in as an admin so we can make changes. There are some names of the books given and we can add more books to it.
- We add a new book but instead of uploading an image as book cover, we try to upload `a php reverse shell` and to our luck it gets uploaded.
- Now we run a `dirbuster` scan to find out some directories where we can find our reverse shell and we find it on `$ip/assets/img/reverseshell.php.`
- Getting a successful reverse shell using netcat command by `nc -nvp <port>`
- We are now logged in as `www-data` and we see that there was a user named `'cirius'` so we try to login into that with some common passwords.
- The password used to login into `cirius` was **`password.`**
- Visiting the `/home/cirius` directory we see a `user.txt` file and when we open it, we found the first check point.
- After checking permissions of this user by `sudo -l`, we get to know that we have permission to run all commands on this system.
- So, we change the user to root by using `sudo su -` and then we are root.
- We find the root.txt file into `/root directory`. The hash in root.txt was the flag.

  Flag = `ICTF{d9539e12946736ee8d1e6e0a18f2596c}`

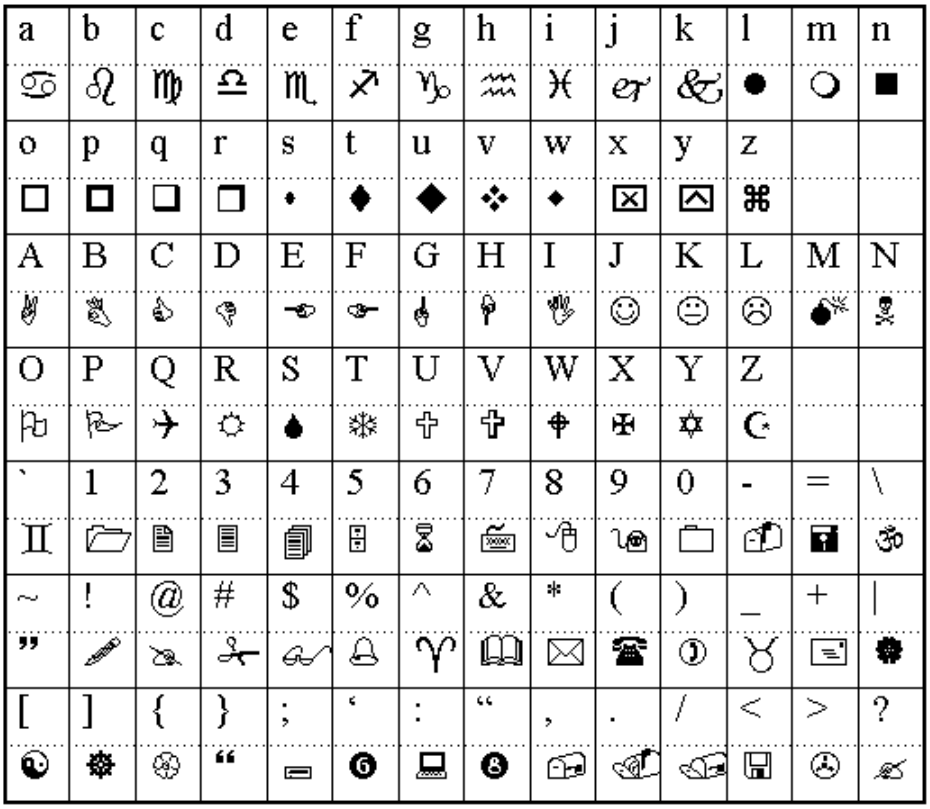
MEDIUM

RedBull Gives You ...

Steps:

- The image given in the question is this.

- After searching on google about `"Wingdings Characters"` (because Redbull gives you **wings**) we get the characters mapped with those symbols.

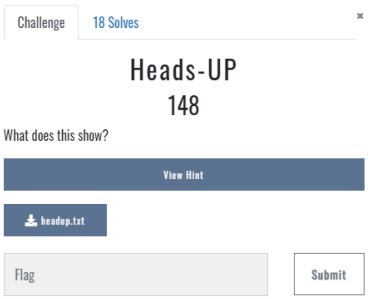


a	b	c	d	e	f	g	h	i	j	k	l	m	n
♋	♌	♍	♎	♏	♐	♑	♒	♓	🙵	🙶	●	❍	■
o	p	q	r	s	t	u	v	w	x	y	z		
□	❑	❑	❒	⬩	⧫	◆	❖	⬩	⊠	⊡	⌘		
A	B	C	D	E	F	G	H	I	J	K	L	M	N
✌	🤘	👌	👎	👈	👉	👆	👆	🖐	☺	😐	☹	💣	☠
O	P	Q	R	S	T	U	V	W	X	Y	Z		
🏳	🏴	✈	☼	💧	❄	✞	✞	✠	✠	✡	☪		
`	1	2	3	4	5	6	7	8	9	0	-	=	\
♊	🗁	🗎	🗏	🗐	🗄	🖴	🖮	🖰	🖲	🗀	🗃	🖬	🕉
~	!	@	#	$	%	^	&	*	(	)	_	+	|
”	🖉	✒	✂	🖋	🕭	♈	🕮	🖂	☎	✆	♉	🖹	✿
[	]	{	}	;	‘	:	“	,	.	/	<	>	?
☯	☸	❁	“	▭	❻	💻	❽	🖃	🖃	🖃	🖫	🕸	🖉

- After mapping the correct sequence of symbols with respective character we get the flag.

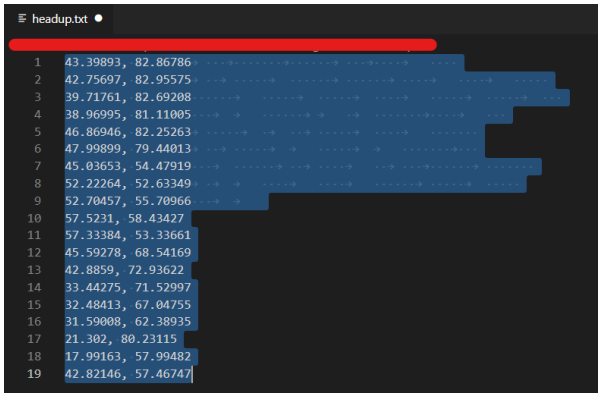
Flag = `ICTF{WIINGS_NOT_WINGS}`

Heads-UP



Steps:

- The headup.txt file given with the challenge contained some co-ordinates/location points given to us.
- But after thorough inspection we found out that it was not the only thing given to us in the text file but there was some whitespace cryptography in that file.

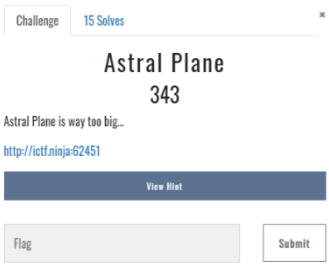


- We knew that we had to use `Stegsnow` tool to decrypt this whitespace, but we also knew that we need a key to do it.
- Plot all the co-ordinates on a map using google earth, it wasn't very clear, but it was forming some image.
- The challenge name said "**Heads-UP**", so we figured out about the constellations and we looked up at google.

- **Orion Belt** had the same structure as those marks on the google map.
- So, the key was `"orion"` now we used `stegsnow` to decrypt the whitespace.
- `stegsnow -C -p orion headup.txt`

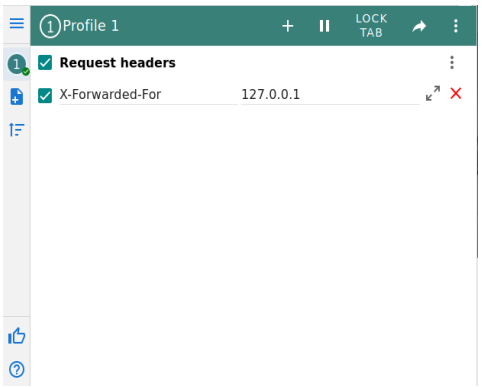
Flag = `ICTF{Fr0m_Out3r_W0RLD}`

Astral Plane



Steps:

- Visiting the link and there was nothing. Source code had nothing in it.
- After doing the Directory scanning, we got an `/admin` page.
- But the admin page was not accessible/forbidden.
- Capturing the request on `burpsuite` and see that there was a `GET` request forwarded to the server.
- Using a `firefox extension` named `"ModHeader"` to forward a `"X-Forwarded-for"` header request to the server using the localhost so that the server thinks of us as the admin or the main user.
- It allowed us to access the `/admin` page and now we are at admin login page.
- When we see the page source of that login page, we find something. It was a path mentioned in the forgot password section.



```
<!DOCTYPE html>
<!--[if lt IE 7 ]> <html lang="en" class="ie6 ielt8"> <![endif]-->
<!--[if IE 7 ]>    <html lang="en" class="ie7 ielt8"> <![endif]-->
<!--[if IE 8 ]>    <html lang="en" class="ie8"> <![endif]-->
<!--[if (gte IE 9)|!(IE)]><!--> <html lang="en"> <!--<![endif]-->
<head>
<meta charset="utf-8">
<title>Paper Stack</title>
<link rel="stylesheet" type="text/css" href="/static/style.css" />
</head>
<div class="login">
  <h1>Login to Web App</h1>
  <form method="post" action="/admin/passd">
    <!--<p><input type="text" name="username" value="" placeholder"Username"></p>-->
    <p><input type="password" name="password" value="" placeholder="Password"></p>
    <p class="remember_me">
      <label>
        <input type="checkbox" name="remember_me" id="remember_me">
        Remember me on this computer
      </label>
    </p>
    <p class="submit"><input type="submit" name="commit" value="Login"></p>
  </form>
</div>

<!--<div class="login-help">
  <p>Forgot your password? <a href="/forgoott">Click here to reset it</a>.</p>
</div>-->
</html>
```

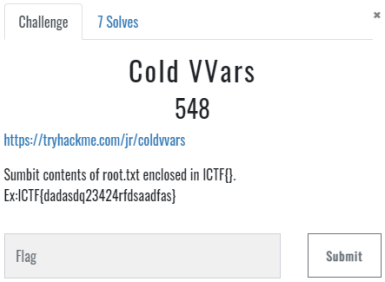
- `/forgoott` was the path mentioned in the page source.
- When visited http://ictf.ninja:62451/forgoott we found an image which displayed some piece of code which said to validate the password if the

```
app.post('/admin/passd',function(req,res){
    if(req.body.password){
        var pass=req.body.password;
        console.log(pass);
        if(pass=='safebanãna'){
```

  password was "safebanãna".
- When we tried to enter this password, it gave us an error of wrong password.
- So now we searched about Unicode and UTF-8 because of that "ã" and why it isn't accepting it.
- We found out that this character has 2 unicode values which is accepted everywhere. So, we tried both the characters from this website. https://fr.wikipedia.org/wiki/%C3%83
- And we got the password as "safebanãna" with the second "ã" in this website.
- After successful login on that admin page we got our flag.

Flag = ICTF{Un1c0d3_FUN_452735}

Cold VVars



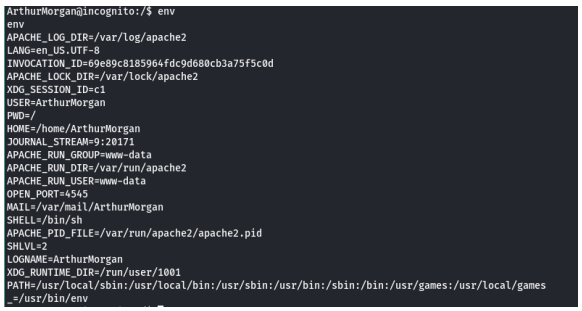
Steps:

- This was another `boot2root` machine which was hosted on `TryHackMe.`
- After booting up the machine we scan for open ports using `nmap.`
- Nmap shows `4` open ports. We found a website hosted on `port 8082` and it was a login page.
- This `website/login` page was vulnerable to `XPath injection.`
- So when we used a query like `" or 1=1 or "` it gave us all the login details of many users.

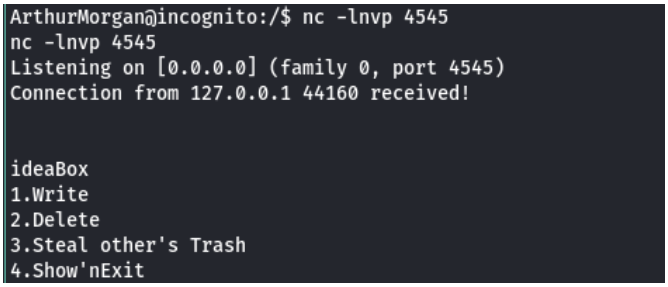
Username Password
Tove Jani
Godzilla KONGistheKING
SuperMan snyderCut
ArthurMorgan DeadEye

- Out of all these, the details of `"ArthurMorgan"` worked.
- We logged in via `smb port` and using AuthurMorgan's username and password.
- We uploaded a `reverse shell on /dev/reverse.php` and got the shell in the smb port.
- We were logged in as `www-data` and we tried to spawn a `tty shell` into it by `python3 -c 'import pty;pty.spawn("/bin/bash")'`
- Now we switched user and used the password of ArthurMorgan to do so.

- Checked the `env` and found an `open port 4545`.



- We used netcat to connect to that port by `nc -lvnp 4545.`



- The 4th option gives us `vi terminal` and we can turn it to shell.
- Then we got a reverse shell on this port.
- Use TMUX session by `tmux attach-session -t 0` and close the tmux window and you will see 1 window with root.
- You will get root.txt file after this and the hash in it will be the flag.

Flag = `ICTF{42f191b937ea71cd2052a06a7a08585a}`