

Incognito 2.0 CTF

CHALLENGES' WRITE-UPS



17-18 APRIL 2021

Team Kali Billi



Contents

EASY	3
Sanity Check	3
Sponsor.....	4
Alphanumerals	4
Voyage.....	5
Not Common.....	6
Simple Login	7
EZ.....	8
Access My Contract	10
Library.....	11
MEDIUM	12
RedBull Gives You	12
Heads-UP.....	13
Astral Plane	14
Cold VVars	16



Sponsor

Challenge 201 Solves

Sponsor
1

Ask for flag on the incognito channel of our test server
<https://discord.gg/PytHH8yn>

=flag

Flag Submit

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_On_Di5c0rd}`

Alphanumerals

Challenge 107 Solves

Alphanumerals
100

Someone said this is based on alphanumerals, I doubt it.

856419 1078119 885788332 18698824

Flag Submit

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

Challenge 33 Solved

Voyage
95

Each Assassin is good but what if they get together... Note- Flag is not in regular format

[bit.ly/3wxy3nK](#) [bit.ly/2Q1R80m](#) [bit.ly/3rZ4pEm](#)

Flag

Submit

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.

```
root@kali:~# curl https://bit.ly/3wxy3nK
<html>
<head><title>Bitly</title></head>
<body><a href="https://cutt.ly/ClickB" moved here</a></body>
</html>root@kali:~https://bit.ly/2Q1R80m2Q1R80m
<html>
<head><title>Bitly</title></head>
<body><a href="https://cutt.ly/4it3d_" moved here</a></body>
</html>root@kali:~https://bit.ly/3rZ4pEm3rZ4pEm
<html>
<head><title>Bitly</title></head>
<body><a href="https://cutt.ly/Ass4ss1n" moved here</a></body>
</html>root@kali:~#
root@kali:~#
```

- 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{C11ckB4it3d_Ass4ss1n}`



Not Common

Challenge 68 Solves ✕

Not Common

50

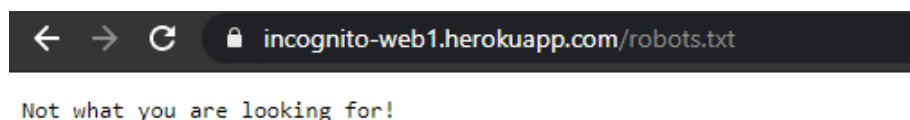
It is important but yet people haven't made it a common practice.

<https://incognito-web1.herokuapp.com/>

Flag Submit

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{FOR_Bug_Hunt3rs}`



Simple Login

Challenge 58 Solves *

Simple Login

84

Challenge Link: <https://incognito-web.herokuapp.com/>

View Hint

Flag Submit

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.

```
Request to https://incognito-web.herokuapp.com:443 [52.50.65.57]
Forward Drop Intercept i... Action Open Bro... Comment this item
Pretty Raw ln Actions v
1 POST / HTTP/1.1
2 Host: incognito-web.herokuapp.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 21
9 Origin: https://incognito-web.herokuapp.com
10 Connection: close
11 Referer: https://incognito-web.herokuapp.com/
12 Upgrade-Insecure-Requests: 1
13 X-Forwarded-For: 127.0.0.1
14
15 user=admin&pass=admin
```

```
Request to https://incognito-web.herokuapp.com:443 [52.50.65.57]
Forward Drop Intercept i... Action Open Bro... Comment this item
Pretty Raw ln Actions v
1 PUT / HTTP/1.1
2 Host: incognito-web.herokuapp.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 21
9 Origin: https://incognito-web.herokuapp.com
10 Connection: close
11 Referer: https://incognito-web.herokuapp.com/
12 Upgrade-Insecure-Requests: 1
13 X-Forwarded-For: 127.0.0.1
14
15 user=admin&pass=admin
```

Flag = `ICTF{NOT_S0_S3Cur3d_853541}`



EZ

Challenge 88 Solves

Ez
10

Let's begin the blockchain party shall we?

rinkeby address = '0x475276B94C298134D07bF0d71EbAcB08eB3D6d5F'

q2.sol

Flag

Submit

Steps:

- This challenge is based on crypto currency. Here we are given a rinkeby address = '0x475276B94C298134D07bF0d71EbAcB08eB3D6d5F'
- 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.

Transaction Details

Overview State

[This is a Rinkeby Testnet transaction only]

Transaction Hash: 0xd47c87659caf5a3ccf74b630987a498856d559c1f4c53c7c7dd29d86f695771

Status: Success

Block: 8424486 31538 Block Confirmations

Timestamp: 5 days 11 hrs ago (Apr-16-2021 07:27:41 PM +UTC)

From: 0x55590dc461ce79eb2280cd1446932b46112af9

To: [Contract 0x475276b94c298134d07bf0d71ebacbd8eb3d6d5f Created]

Value: 0 Ether (\$0.00)

Transaction Fee: 0.000135127 Ether (\$0.00)

Gas Price: 0.000000001 Ether (1 Gwei)

[Click to see More](#)

Transaction Details

Overview State

Advanced A set of information that represents the current **state** is updated when a transaction takes place on the network. The below is a summary of those changes :

Address	Before	After	State Difference
0x00	1,633.460147173534734069 Eth	1,633.460282300534734069 Eth	▲ 0.000135127
0x475276b94c298134d07bf0d71ebacbd8eb3d6d5f	0 Eth Nonce: 0	0 Eth Nonce: 1	
0x55590dc461ce79eb2280cd1446932b46112af9	3.338433713927926623 Eth Nonce: 186	3.338298586927926623 Eth Nonce: 187	▼ 0.000135127

- 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

Challenge 48 Solves ✕

Access My Contract

21

Let me see you access this!

rinkeby address = "0x7A9bFC829D2df5B9Abc9097E9a1265b7C193DD2e"

View Hint

q1.sol

Flag

Submit

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

Challenge 38 Solves ✕

Library
244

<https://tryhackme.com/jr/incognito1>

Flag Submit

Steps:

- This challenge was a boot2root challenge hosted on [TryHackMe](https://tryhackme.com).
- 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 ...

Challenge 105 Solved

Redbull gives you ...
100

crypt.png

Flag

Submit

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	-	=	\
☹	☹	☹	☹	☹	☹	☹	☹	☹	☹	☹	☹	☹	☹
~	!	@	#	\$	%	^	&	*	()	_	+	
☹	☹	☹	☹	☹	☹	☹	☹	☹	☹	☹	☹	☹	☹
[]	{	}	;	'	:	"	,	.	/	<	>	?
☹	☹	☹	☹	☹	☹	☹	☹	☹	☹	☹	☹	☹	☹

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

Flag = `ICTF{WIINGS_NOT_WINGS}`



Heads-UP

Challenge 18 Solved

Heads-UP

148

What does this show?

View Hint

headup.txt

Flag Submit

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.

```
headup.txt •
1 43.39893, 82.86786
2 42.75697, 82.95575
3 39.71761, 82.69208
4 38.96995, 81.11005
5 46.86946, 82.25263
6 47.99899, 79.44013
7 45.03653, 54.47919
8 52.22264, 52.63349
9 52.70457, 55.70966
10 57.5231, 58.43427
11 57.33384, 53.33661
12 45.59278, 68.54169
13 42.8859, 72.93622
14 33.44275, 71.52997
15 32.48413, 67.04755
16 31.59008, 62.38935
17 21.302, 80.23115
18 17.99163, 57.99482
19 42.82146, 57.46747
```

- 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.



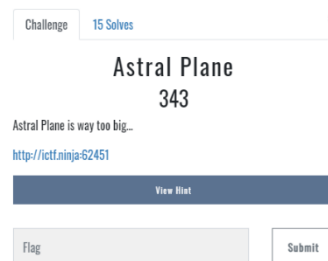


- **Orion Belt** had the same structure as those marks on the google map.
- So, the key was "orion" now we used `stegsnow` to decrypt the whitespace.
- `stegsnow -C -p orion headup.txt`

Flag = `ICTF{Fr0m_Out3r_W0RLD}`

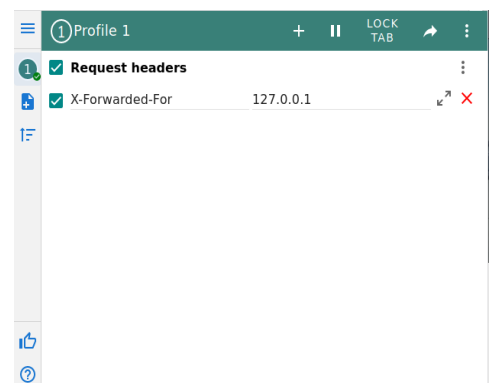


Astral Plane



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.





```
1
2
3 <!DOCTYPE html>
4 <!--[if lt IE 7 ]> <html lang="en" class="ie6 ielt8"> <![endif]-->
5 <!--[if IE 7 ]> <html lang="en" class="ie7 ielt8"> <![endif]-->
6 <!--[if IE 8 ]> <html lang="en" class="ie8"> <![endif]-->
7 <!--[if (gte IE 9)|!(IE)]><!--> <html lang="en"> <!--<![endif]-->
8 <head>
9 <meta charset="utf-8">
10 <title>Paper Stack</title>
11 <link rel="stylesheet" type="text/css" href="/static/style.css" />
12 </head>
13 <div class="login">
14 <h1>Login to Web App</h1>
15 <form method="post" action="/admin/passd">
16 <!--<p><input type="text" name="username" value="" placeholder="Username"></p-->
17 <p><input type="password" name="password" value="" placeholder="Password"></p>
18 <p class="remember_me">
19 <label>
20 <input type="checkbox" name="remember_me" id="remember_me">
21 Remember me on this computer
22 </label>
23 </p>
24 <p class="submit"><input type="submit" name="commit" value="Login"></p>
25 </form>
26 </div>
27
28 <!--<div class="login-help">
29 <p>Forgot your password? <a href="/forgoott">Click here to reset it</a>.</p>
30 </div-->
31 </html>
32
```

- `/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

Challenge 7 Solved

Cold VVars

548

<https://tryhackme.com/jr/coldvvars>

Submit contents of root.txt enclosed in [CTF].
Ex:[CTF{dadasdq23424rfdsaadfas}]

Flag

Submit

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.

```
ArthurMorgan@incognito:/$ env
env
APACHE_LOG_DIR=/var/log/apache2
LANG=en_US.UTF-8
INVOCAATION_ID=69e89c8185964fdc9d688cb3a75f5c0d
APACHE_LOCK_DIR=/var/lock/apache2
XDG_SESSION_ID=c1
USER=ArthurMorgan
PWD=/
HOME=/home/ArthurMorgan
JOURNAL_STREAM=9:20171
APACHE_RUN_GROUP=www-data
APACHE_RUN_DIR=/var/run/apache2
APACHE_RUN_USER=www-data
OPEN_PORT=4545
MAIL=/var/mail/ArthurMorgan
SHELL=/bin/sh
APACHE_PID_FILE=/var/run/apache2/apache2.pid
SHLVL=2
LOGNAME=ArthurMorgan
XDG_RUNTIME_DIR=/run/user/1001
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
_=/usr/bin/env
```

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

```
ArthurMorgan@incognito:/$ nc -lnvp 4545
nc -lnvp 4545
Listening on [0.0.0.0] (family 0, port 4545)
Connection from 127.0.0.1 44160 received!

ideaBox
1.Write
2.Delete
3.Steal other's Trash
4.Show'nExit
```

- 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}`