

Report - Universal Cereal Bus

Table of Contents

- [4 Stage = 4x Better](#)
- [All The Wrappers](#)
- [Ancient Rome](#)
- [Ball Goes Brrrr](#)
- [Change the World](#)
- [Connect Home](#)
- [First Day at Uni!](#)
- [Intelli-telnet](#)
- [Keep Me Posted](#)
- [Postman Pat knows everything about 13 Rotten Apples](#)
- [Remote Access 1](#)
- [Remote Access 2](#)
- [RTSP That Thing](#)
- [Smart Camera Secret File](#)
- [The Reality Deep Down in our Hearts](#)
- [Time for a Little Magic Trick](#)
- [To Kill a Blue Bird](#)
- [Uh Oh](#)
- [Way Back Home](#)
- [Where's Jeffery?](#)
- [Where's Wally](#)
- [Work work work work work](#)

4 Stage = 4x Better

We are given the link:

```
https://chal.hackmac.xyz:30105
```

Hello Contestant!

This challenge is a multi-stage challenge. You are required to solve four stages of crypto mini challenges.
Be sure to make note of each correct input, as all four will make up the flag.

Good Luck and click the button below to continue!

[Continue](#)

The title tells us we have 4 stages to get through.

Stage 1:

Welcome to Stage 1!
Don't forget to make note of your answer...

Ciphertext is: **Usfaobm**
Here's your hint:



[Enter your answer here](#)

[Submit](#)

A caesar salad can only mean one thing - Caesar cipher. Running it through CyberChef (rotating through all the numbers until we reached a word) gives us:

Germany

Recipe	Input
<p>ROT13</p> <p><input type="checkbox"/> Rotate lower case chars</p> <p><input type="checkbox"/> Rotate upper case chars</p> <p>Amount 12</p>	Usfaobm
	<p>Output</p> <p>Germany</p>

Stage 2:

Welcome to Stage 2!

Don't forget to make note of your answer...

Ciphertext is:

.— / . / — / . / .. / .-. / .

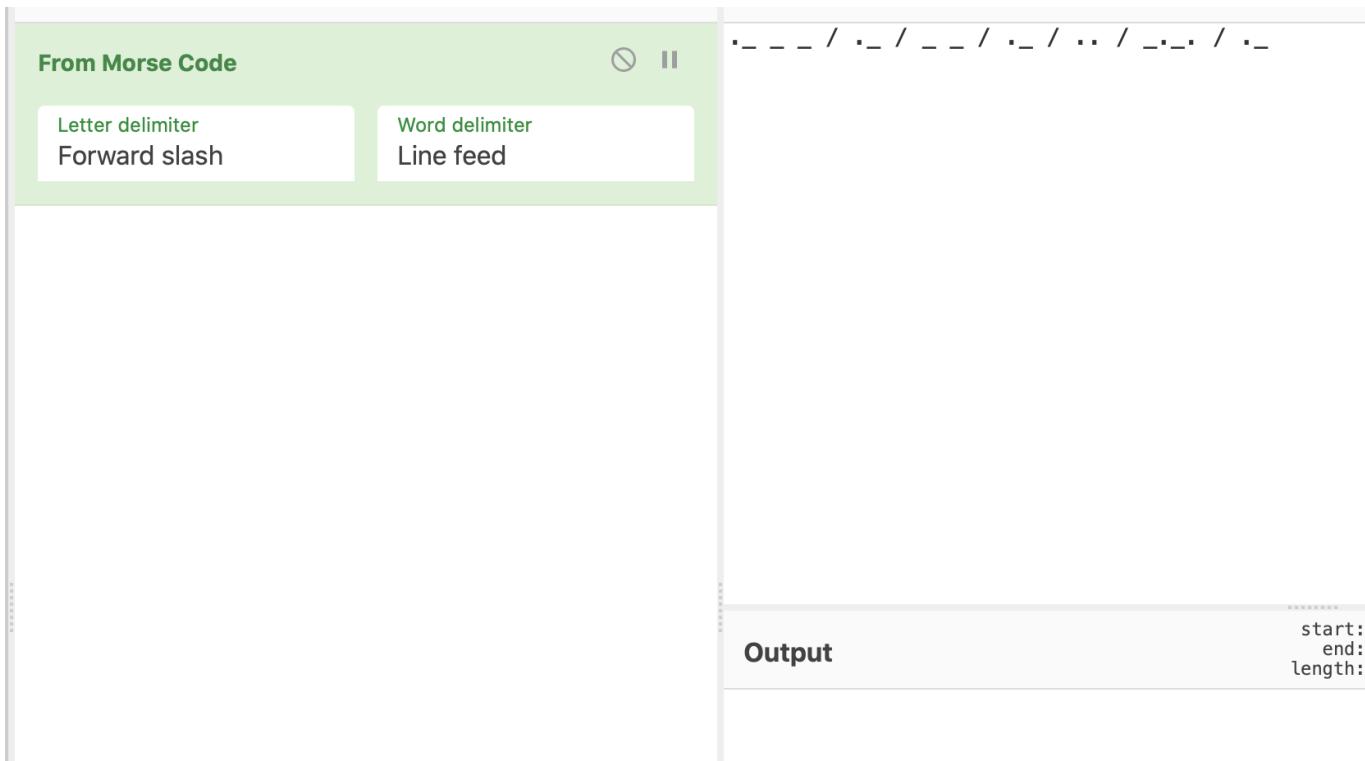
Enter your answer here

Submit

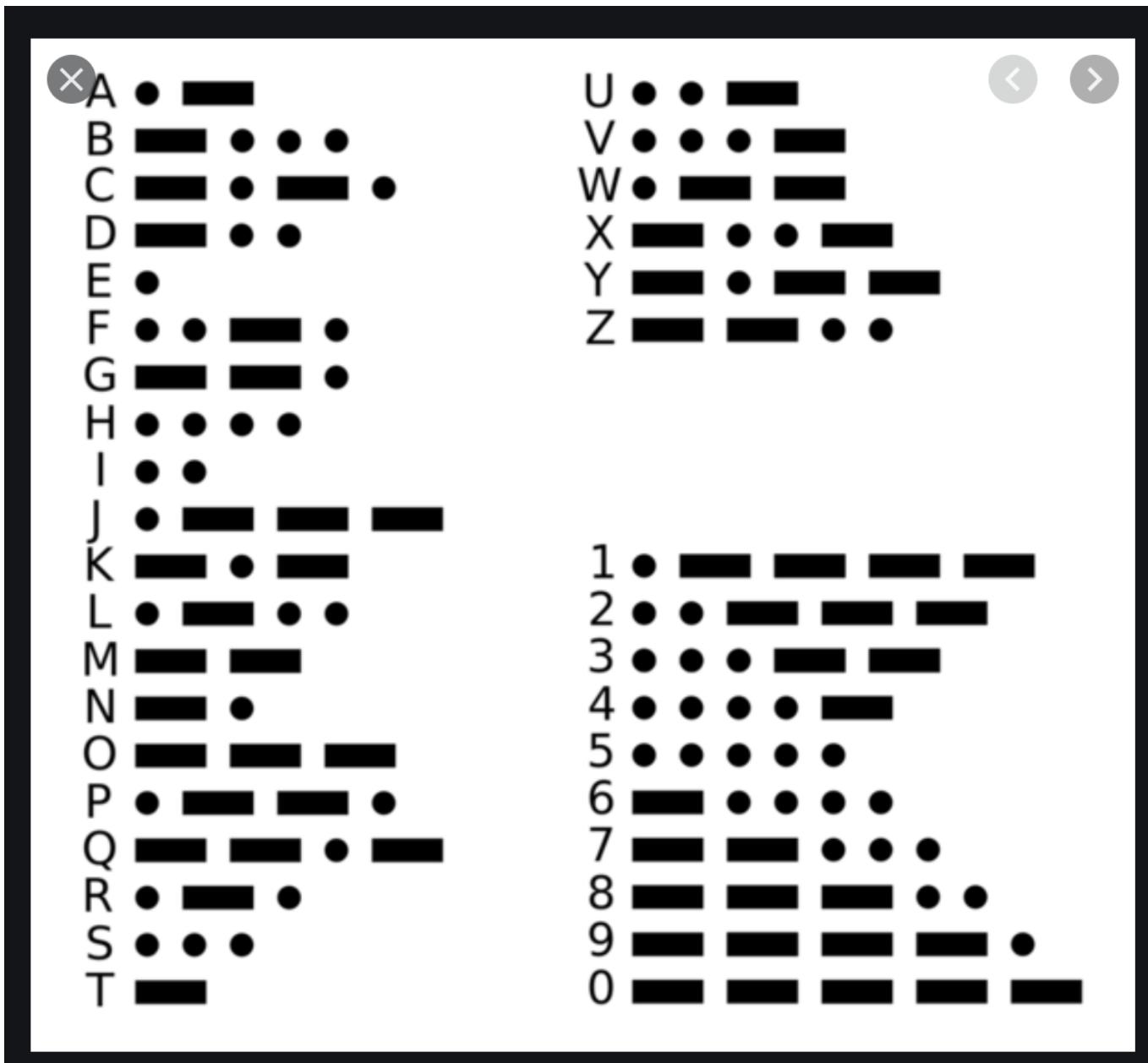
This definitely looks like morse code but running it through CyberChef gives us something that doesn't look right (confirmed by trying to submit it in the final flag).

Recipe		Input
From Morse Code	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	.— / . / — / . / .. / .-. / .
<input type="checkbox"/> Letter delimiter Space	<input type="checkbox"/> Word delimiter Line feed	
Output		
ATTATTAICA		

Even trying to change the delimiter to a forward slash did not provide a result:



So, by manually checking each one against the morse code lookup table:



We got:

Jamaica

Stage 3:

Welcome to Stage 3!
Don't forget to make note of your answer...

Ciphertext is: **Pntnpvvyiia**
Here's your hint: **Pingpingpin**

Enter your answer here

Submit

This definitely looks like a substitution cipher, but it wasn't immediately clear that the 'hint' was actually a 'key'. After trying a couple of manual substitutions, we tried the Vigenere decode on it and it returned:

Afghanistan

Recipe		Input
Vigenère Decode	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Pntnpvvyiia
Key		
Pingpingpin		
Output		
		Afghanistan

Stage 4:**Welcome to Stage 4!***Don't forget to make note of your answer...***Ciphertext is: 4647d00cf81f8fb0ab80f753320d0fc9****Here's your hint:****Enter your answer here****Submit**

The weapon appears to be an MP5, and a hash brown likely means it relates to a hash. So by running the text we were given, on an MD5 hash decode lookup

```
https://md5.gromweb.com/?md5=4647d00cf81f8fb0ab80f753320d0fc9
```

We get:

Indonesia

MD5 reverse for 4647d00cf81f8fb0ab80f753320d0fc9

The MD5 hash:

4647d00cf81f8fb0ab80f753320d0fc9

was successfully reversed into the string:

Indonesia

Feel free to provide some other MD5 hashes you would like to try to reverse.

The next page shows us that the words we found are the flag:

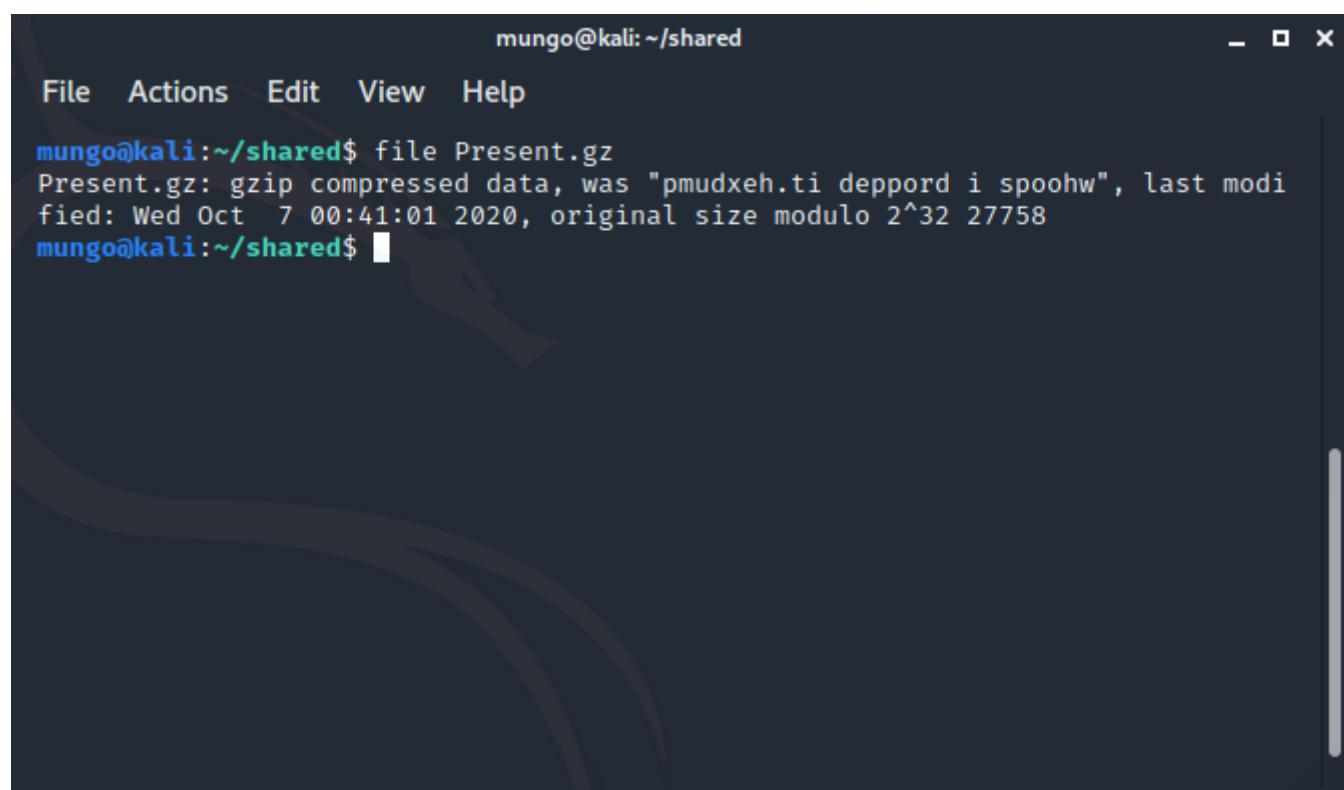
Congratulations!

You successfully captured the flag
Put all four correct answers together for the flag

```
hackmac{GermanyJamaicaAfghanistanIndonesia}
```

All The Wrappers

We are given the file `Present.gz` which appears to be a `gzipped` compressed file. To confirm, we will run the `file` command:

A screenshot of a terminal window titled "mungo@kali: ~/shared". The window has standard window controls (minimize, maximize, close) at the top right. The terminal menu bar shows "File Actions Edit View Help". The command "file Present.gz" is entered and its output is displayed:

```
mungo@kali:~/shared$ file Present.gz
Present.gz: gzip compressed data, was "pmudxeh.ti deppord i spoohw", last modified: Wed Oct 7 00:41:01 2020, original size modulo 2^32 27758
mungo@kali:~/shared$
```

The terminal background is dark, and there is a faint watermark of a person's face in the center.

This confirms that it is indeed a gzip file. We also get a hint for a later phase of the challenge in the previous filename, `pmudxeh.ti deppord i spoohw`, which is `whoops i dropped it.hxdump` in reverse. For the meantime, we will unzip the file using the `gunzip` command:

```
$ gunzip Present.gz
```

This gives us an uncompressed file. We'll run `file` again to see what to do with it:

mungo@kali:~/shared\$ file Present
Present: ASCII text
mungo@kali:~/shared\$

Running `cat` to display the contents of the file reveals it is indeed a hexdump.

mungo@kali:~/shared\$
File Actions Edit View Help
00001520 20 30 31 30 30 30 31 30 31 20 30 31 30 31 30 30 | 01000101 010100
00001530 30 31 20 30 31 30 30 30 31 30 30 20 30 31 30 30 | 01 01000100 0100
00001540 30 31 31 31 20 30 31 30 30 31 31 30 31 20 30 31 | 0111 01001101 01
00001550 30 31 30 30 31 30 20 30 31 30 30 30 30 30 31 20 | 010010 01000001
00001560 30 31 30 30 30 31 31 31 20 30 31 30 31 30 31 30 | 01000111 0101010
00001570 31 20 30 31 30 31 31 30 31 30 20 30 31 30 31 30 | 1 01011010 01010
00001580 30 31 31 20 30 31 30 30 30 30 31 20 30 31 30 | 011 01000001 010
00001590 30 31 31 31 30 20 30 31 30 30 31 30 31 30 20 30 | 01110 01001010 0
000015a0 31 30 31 30 30 30 31 20 30 31 30 30 30 31 30 31 | 1010001 01000101
000015b0 20 30 31 30 30 30 30 30 31 20 30 31 30 31 31 30 | 01000001 010110
000015c0 31 30 20 30 31 30 31 30 31 30 30 20 30 31 30 30 | 10 01010100 0100
000015d0 30 31 30 31 20 30 31 30 30 31 30 30 31 20 30 31 | 0101 01001001 01
000015e0 30 30 30 30 31 30 20 30 31 30 31 30 31 31 30 20 | 0000010 01010110
000015f0 30 31 30 30 30 31 31 31 20 30 31 30 30 30 31 30 | 01000111 0100010
00001600 31 20 30 31 30 31 30 30 30 31 20 30 31 30 30 30 | 1 01010001 01000
00001610 31 30 30 20 30 31 30 30 31 30 31 31 20 30 31 30 | 100 01001011 010
00001620 30 31 31 31 30 20 30 31 30 31 31 30 30 31 20 30 | 01110 01011001 0
00001630 30 31 31 31 31 30 31 | 0111101|mungo@kali
:~/shared\$

To reverse the hexdump into what appears to be binary, `xxd` with the `-r` option will give us a file with the original data:

```
$ xxd -r > file.txt
```

Now that we have this binary data, we can use the web tool CyberChef to decode it. With the binary input from the file, we can convert it:

1. From Binary:

```
GU2CANBZEAZTEIBVGIQDKNJAGMZCANJSEA2TIIBTGIQDKNBAHE3SAMZSEA2TIIBUHEQDGMRAGUYSANJQEA
ZTEIBVGEQDIOJAGMZCANJUEA2TMIBTGIQDKMZAGU3SAMZSEA2TCIBVGQEQDGMRAGU2SANJSEAZTEIBVGQGD
SOBAGMZCANJUEA2TAIBTGIQDKMJAGQ4SAMZSEA2TCIBVG4QDGMRAGUYSANJTEAZTEIBVGQQDKMBAGMZCAN
JREA2TCIBTGIQDKMZAGU2CAMZSEA2TIIBVGQEQDGMRAGU2CANJQEAZTEIBVGIQDKNJAGMZCANJUEA4TSIBT
GIQDKNJAGUYCAMZSEA2TGIBZG4QDGMRAGUZSANJUEAZTEIBVGEQDKNZAGMZCANJREA2TGIBTGIQDKNBAGU
YCAMZSEA2TCIBVGQEQDGMRAGU2SANJXEAZTEIBVGMQDSNZAGMZCANJTEA2TMIBTGIQDKMRAGEYDCIBTGIQDKNBAHE4SAM
ZSEA2TIIBVGAQDGMRAGU2CAMJQGEQDGMRAGUZSANJQEAZTEIBVGEQDKNY=
```

The = sign at the end of the string indicates it's a number system of a higher base, such as base32 or base64. As there are only upper-case characters visible, we can assume base32.

2. From Base32:

```
54 49 32 52 55 32 52 54 32 54 97 32 54 49 32 51 50 32 51 49 32 54 56 32 53 57 32
51 51 32 55 52 32 54 98 32 54 50 32 51 49 32 51 57 32 51 53 32 54 50 32 51 51 32
53 54 32 54 54 32 54 50 32 52 55 32 54 99 32 55 50 32 53 97 32 53 54 32 51 57 32
51 53 32 54 50 32 51 51 32 53 54 32 55 57 32 53 56 32 51 51 32 52 50 32 55 57 32
53 97 32 53 56 32 52 101 32 54 99 32 54 50 32 54 101 32 53 50 32 51 57
```

As there are instances of the digit 9, this can't be octal. There are no letters, so it probably isn't hexadecimal. We will assume this is in decimal.

3. From Decimal:

```
61 47 46 6a 61 32 31 68 59 33 74 6b 62 31 39 35 62 33 56 66 62 47 6c 72 5a 56 39
35 62 33 56 79 58 33 42 79 5a 58 4e 6c 62 6e 52 39
```

This looks like hexadecimal, as there are no letters above f.

4. From Hexadecimal:

```
aGFja21hY3tgb195b3VfbG1rZV95b3VyX3ByZXNlbnR9
```

We can see a combination of digits, uppercase and lowercase alphabetical characters, so this could be base64.

5. From Base64:

```
hackmac{do_like_your_present}
```

There's our flag!

This is what the final CyberChef window looks like:

The screenshot shows the CyberChef interface with a recipe for decoding binary data. The 'Input' section contains a large block of binary code (length: 5687 lines: 1). The 'Output' section shows the decoded text: "hackmac{do_you_like_your_present}".

Recipe

- From Binary**
 - Delimiter: Space
- From Base32**
 - Alphabet: A-Z2-7=
- Remove non-alphabet chars
- From Decimal**
 - Delimiter: Space
 - Support signed values
- From Hex**
 - Delimiter: Auto
- From Base64**
 - Alphabet: A-Za-z0-9+=
- Remove non-alphabet chars

Input
length: 5687
lines: 1

Output
start: 0 time: 4ms
end: 33 length: 33
length: 33 lines: 1

```
01000111 01010101 00110010 01000011 01000001 01001110 01000010 01011010
01000101 01000001 01011010 01010100 01000101 01001001 01000010 01010110
01000111 01001001 01010001 01000100 01001011 01001110 01001010 01000001
01000111 01001101 01011010 01000011 01000001 01001110 01001010 01010011
01000101 01000001 00110010 01010100 01001001 01001001 01000010 01010100
01000111 01001001 01010001 01000100 01001011 01001110 01001010 01000001
01001000 01000101 00110011 01010011 01000001 01001101 01011010 01010011
01000101 01000001 00110010 01010100 01001001 01001001 01000010 01010101
01000111 01010101 01011001 01010011 01000001 01001110 01001010 01010001
01000101 01000001 01011010 01010001 01000001 01001101 01010010 01010110
01000111 01000101 01011010 01010001 01000001 01001101 01010010 01010110
01000111 01000101 01011010 01010001 01000001 01001101 01010010 01010110
01000101 01000001 01011010 01010001 01000001 01001101 01010010 01010110
01000111 01000101 01011001 01010011 01000001 01001111 01001010 01000001
01000111 01000101 01011010 01000011 01000001 01001110 01001010 01010101
01000101 01000001 00110010 01010100 01001001 01001001 01000010 01010100
01000111 01000101 01010001 01000100 01001101 01001010 01010001 01010101
01000111 01000101 01010001 01000100 01001101 01001010 01010001 01010101
01000101 01000001 01010001 01000100 01001101 01001010 01010001 01010101
01000111 01000101 01010001 01000100 01001101 01001010 01010001 01010101
01000111 01000101 01010001 01000100 01001101 01001010 01010001 01010101
01000101 01000001 01010001 01000100 01001101 01001010 01010001 01010101
01000111 00110100 01010001 01000100 01001101 01001010 01010001 01010100
01000111 01010101 01011001 01010011 01000001 01001110 01001010 01010100
01000101 01000001 01011010 01010000 01000101 01001001 01000010 01010110
01000111 01010001 01010001 01000100 01001011 01001101 01000001 01000001
01000111 01001101 01011010 01000011 01000001 01001110 01001010 01010010
A10AA1A1 A10AAA1A1 AA11AA1A A1A101AA A1000011 A1001101 01001010 01010010
A10AA1A1 A10AAA1A1 AA11AA1A A1A101AA A1000011 A1001101 01001010 01010010
A10AA1A1 A10AAA1A1 AA11AA1A A1A101AA A1000011 A1001101 01001010 01010010
```

Ancient Rome

Since the challenge is in the crypto section, the name hints that we are potentially dealing with a Caesar cipher. We are given the link:

<https://chal.hackmac.xyz:30106>

Encryption Challenge

In this challenge, the encryption key changes every 15 seconds...
So don't take too long to figure it out

10 seconds left!

The encrypted password is : ngiqkxsgt

Your answer here

Submit

Want a hint?

Want a hint?

"We should totally just stab Caesar!"

By having CyberChef loaded, and quickly putting the encrypted password in, the only thing left is to shift by a random number (or seemingly random, as it changed with each encrypted password we tried), in less than 15 seconds, until we are left with a word that makes sense:

The screenshot shows a web application interface for a ROT13 cipher. At the top left, the text "ROT13" is displayed in green. To the right are two checkboxes: "Rotate lower case chars" (unchecked) and "Rotate upper case chars" (unchecked). Below these is a "Amount" input field containing the value "11", with up and down arrow buttons to its right. On the far right, there is a "Stop" button (a circle with a slash) and a "Run" button (two vertical bars). The main area of the application shows the input text "hackerman" and the resulting output text "wprztgbpc".

The next page shows us that the word we found is the flag:

A large green banner with white text. The main text reads "Congratulations!" in a large font. Below it, in a smaller font, is the message: "You successfully captured the flag" and "The password you decrypted is the flag for this challenge".

```
hackmac{hackerman}
```

Ball Goes Brrrr

We knew that Jeffery's twitter gave us insight into his love for basketball. Checking his following again, we see there is only one Australian account:

The image shows a Twitter profile page for the account @UBLofficial. The header photo features three basketball players: one in a black shirt holding a large white banner that reads "UBL" in orange, "THE WINNING TEAM" in black, "TEN THOUSAND DOLLARS" in black, and "Major Sponsor AMP" in black; another player in a black shirt is partially visible behind him; and a third player in a green jersey with the number 15 and "SYDNEY COBRAS" on it is smiling. Below the header are two circular buttons: one with three dots and another with the word "Follow".

Ultimate Basketball

@UBLofficial

Sydney basketball is about to heat up this Summer with UBL Season 3, starting 29th August 2015. check out: facebook.com/UltimateBballL... for updates and news

⌚ Sydney ⚡ UltimateBasketball.com.au 📅 Joined July 2013

209 Following 142 Followers

Not followed by anyone you're following

Tweets **Tweets & replies** **Media** **Likes**

Initially, we tried **sydney** as this was the location on their twitter page, however this did not work.

Going to their linked website:

<https://parramattawildcats.basketball/ubl>

Revealed the account was for the **Parramatta Wild Cats**.

However, **parramatta** also did not work.

Scrolling down, we found out they were playing at **Lidcombe**:

The screenshot shows the Parramatta Wildcats basketball website. At the top, there's a navigation bar with links for 'LEARN TO PLAY', 'COMPETITIONS', 'UBL', 'ONLINE SERVICES', and 'ABOUT US'. Social media icons for YouTube, Instagram, Twitter, and Facebook are also present. A logo for 'DOOLEYS LIDCOMBE CATHOLIC CLUB' is on the right.

Below the navigation, there's a section with three white boxes containing text and arrows:

- with sessions running most weekdays, and games on Friday nights for Juniors.**
- Parramatta Basketball Association. Competitions available for Senior & Youth Men. This is a perfect stepping stone to get you to the next level.**

On the left, there's a 'Major Sponsor' section featuring the 'DOOLEYS' logo. On the right, there's a 'Partners' section featuring logos for 'be.basketball' and 'SportsTG Let's Win'.

At the bottom, there's a footer menu with four sections: 'GENERAL INFORMATION', 'LEARN TO PLAY', 'COVID-19 SAFETY', and 'CONTACT US'. Each section lists various links.

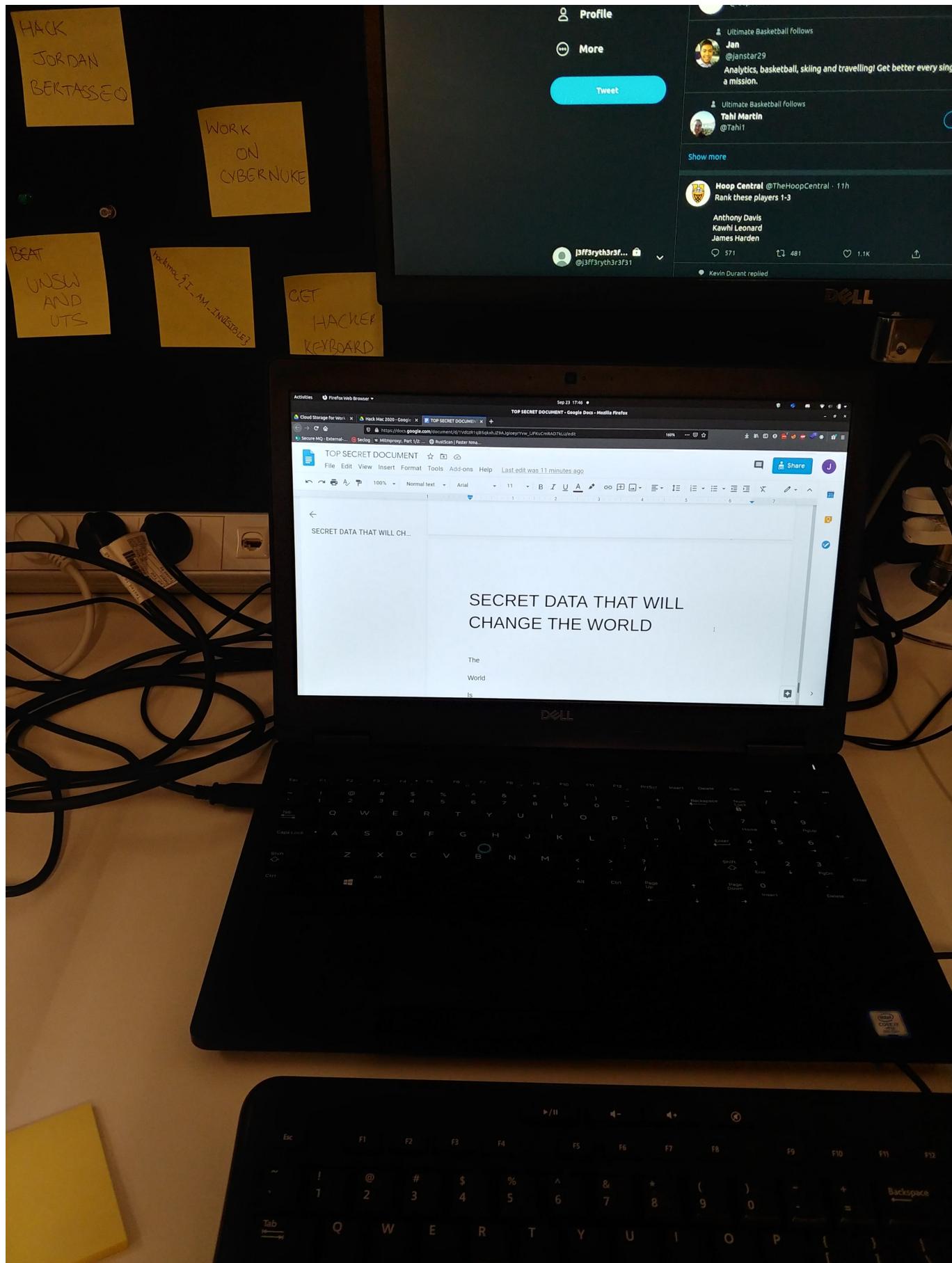
GENERAL INFORMATION	LEARN TO PLAY	COVID-19 SAFETY	CONTACT US
Competitions UBL Court Hire History of the Wildcats Frequently Asked Questions Privacy Policy	Working with Children Coaching Application Referees Active Kids Vouchers Learn to Play Holiday Camps	COVID Visitor Registration Form COVID Safe Venue	info@parramattawildcats.basketball 02-9646-3840 PO Box 415 Lidcombe NSW 1825 Auburn Basketball Centre Church Street Lidcombe NSW 2141 VIEW MAP

However, this also did not work. Thankfully, trying **Auburn** worked.

```
hackmac{auburn}
```

Change the World

Checking the same image from the challenge **Keep Me Posted**:



We see there is a Google Doc file open on the laptop, and for the most part, the link is visible:

1VdtzR1sjB5qkxhJZ9AJg*oeyrYvw_*JFKuCmRAD7kLU

The two * signs resemble characters that could be lowercase i, uppercase I, or lowercase l

This meant 6 possible permutations. A quick Excel concatenation:

```
=CONCATENATE(A1,B1,C1,D1,E1,F1)
```

A	B	C	D	E	F	G
https://docs.google.com/document/d/	1VdtzR1sjB5qkxhJZ9AJg	i	oeyrYvw_	i	JFKuCmRAD7kLU	https://docs.google.com/document/d/1VdtzR1sjB5qkxhJZ9AJgioeyrYvw_1JFKuCmRAD7kLU
https://docs.google.com/document/d/	1VdtzR1sjB5qkxhJZ9AJg	i	oeyrYvw_	I	JFKuCmRAD7kLU	https://docs.google.com/document/d/1VdtzR1sjB5qkxhJZ9AJgioeyrYvw_IJFKuCmRAD7kLU
https://docs.google.com/document/d/	1VdtzR1sjB5qkxhJZ9AJg	i	oeyrYvw_	I	JFKuCmRAD7kLU	https://docs.google.com/document/d/1VdtzR1sjB5qkxhJZ9AJgioeyrYvw_IJFKuCmRAD7kLU
https://docs.google.com/document/d/	1VdtzR1sjB5qkxhJZ9AJg	I	oeyrYvw_	i	JFKuCmRAD7kLU	https://docs.google.com/document/d/1VdtzR1sjB5qkxhJZ9AJgioeyrYvw_iJFKuCmRAD7kLU
https://docs.google.com/document/d/	1VdtzR1sjB5qkxhJZ9AJg	I	oeyrYvw_	i	JFKuCmRAD7kLU	https://docs.google.com/document/d/1VdtzR1sjB5qkxhJZ9AJgioeyrYvw_ijFKuCmRAD7kLU
https://docs.google.com/document/d/	1VdtzR1sjB5qkxhJZ9AJg	I	oeyrYvw_	i	JFKuCmRAD7kLU	https://docs.google.com/document/d/1VdtzR1sjB5qkxhJZ9AJgioeyrYvw_1JFKuCmRAD7kLU
		lowercase i				
		lowercase l				
		uppercase i				

After trying some links, one worked:

```
https://docs.google.com/document/d/1VdtzR1sjB5qkxhJZ9AJgioeyrYvw_1JFKuCmRAD7kLU/edit
```

Ctrl+F for **hackmac{** led to a flag in the hackmac format:

ALERT: SECRET DATA BE CAREFUL
AI FRT· SFCRFT DATA RF CARFFU II

hackmac{you_should_not_be_here}

Connect Home

This challenge was very simple, and was required to have access to the rest of the 'Hack the CISO' challenges. It is more of a tutorial to gain access to the CISO network through a VPN, rather than a challenge.

The flag for this challenge is the IP address given in the challenge description, written in the flag's format.

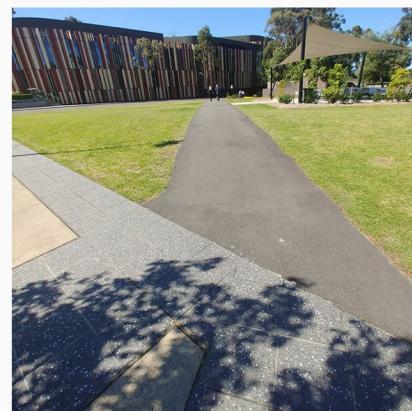
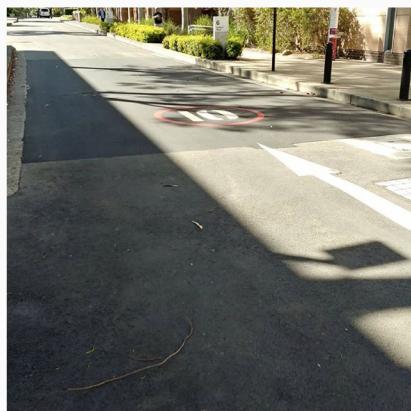
```
hackmac{137.111.189.100}
```

First Day at Uni!

We are given Jeffery's instagram page:

```
https://www.instagram.com/j3ff3ryth3r3f3r33/
```

He has about 6 posts that appear to be relevant to MQ:



By looking at each one, we find that he is 2 minutes late to class in this one:



A photograph of the same brick building with a paved walkway and bushes, identical to the one above. This version includes a screenshot of the page's source code.

```
<!DOCTYPE html>
<html lang="en" class="js logged-in client-root js-focus-visible sDN5V">
  <head></head>
  <body class="style">
    <div id="react-root">
      <section class="jEogl E3X2T">
        <main class="SCxLW o6ar " role="main">
          <div class="Kj7hi yJx9Gc">
            <div class="l1TEKPx">
              <article class="0BXij M9sTE h0YNM SgTZ1 " role="presentation" tabindex="-1">
                <header class="Ppjfr UE9AK wd0Qh"></header>
                <div class="MEAGs"></div>
                <div class="97aPb wKw0K"></div>
                <div class="e02As ">
                  <section class="1pm_3lqm"></section>
                  <section class="EPK_vgnzN"></section>
                  <div class="taWk"></div>
                  <div class="K_0BX NnvRm">
                    <a class="c-Y17" href="/p/Cfdip61Nha/" tabindex="0">
                      <time class="lo9PC Nzbs5" datetime="2020-09-23T02:03:14.000Z" title="Sep 23, 2020">September 23</time> = $0
                    </a>
                  </div>
                </div>
              </article>
            </div>
            <div class="Igw0E TwRSI eGOV_ _4EzTm IM32b "></div>
            <div class="Z666a"></div>
          </main>
          <nav class="NCx7H jLuN9 "></nav>
          <footer class="8Rnq9 _3Laht " role="contentinfo"></footer>
        </div>
        <script type="text/javascript">window._pendingAdditionalData(["/p/Cfdip61Nha"]);</script>
        <link rel="stylesheet" href="/static/bundles/es6/ConsumerUICommons.css/8f7fcbb7c148.css" type="text/css" crossorigin="anonymous">
        <link rel="stylesheet" href="/static/bundles/es6/Consumer.css/9a51ba5d97e.css" type="text/css" crossorigin="anonymous">
        <script type="text/javascript"></script>
        <script type="text/javascript" src="/static/bundles/es6/Vendor.js/c911f5848b87.js" crossorigin="anonymous"></script>
        <script type="text/javascript" src="/static/bundles/es6/en_US.js/7b83afbf0fb.js" crossorigin="anonymous"></script>
        <script type="text/javascript" src="/static/bundles/es6/en_en_US.js/7b83afbf0fb.js" crossorigin="anonymous"></script>
      </section>
    </div>
```

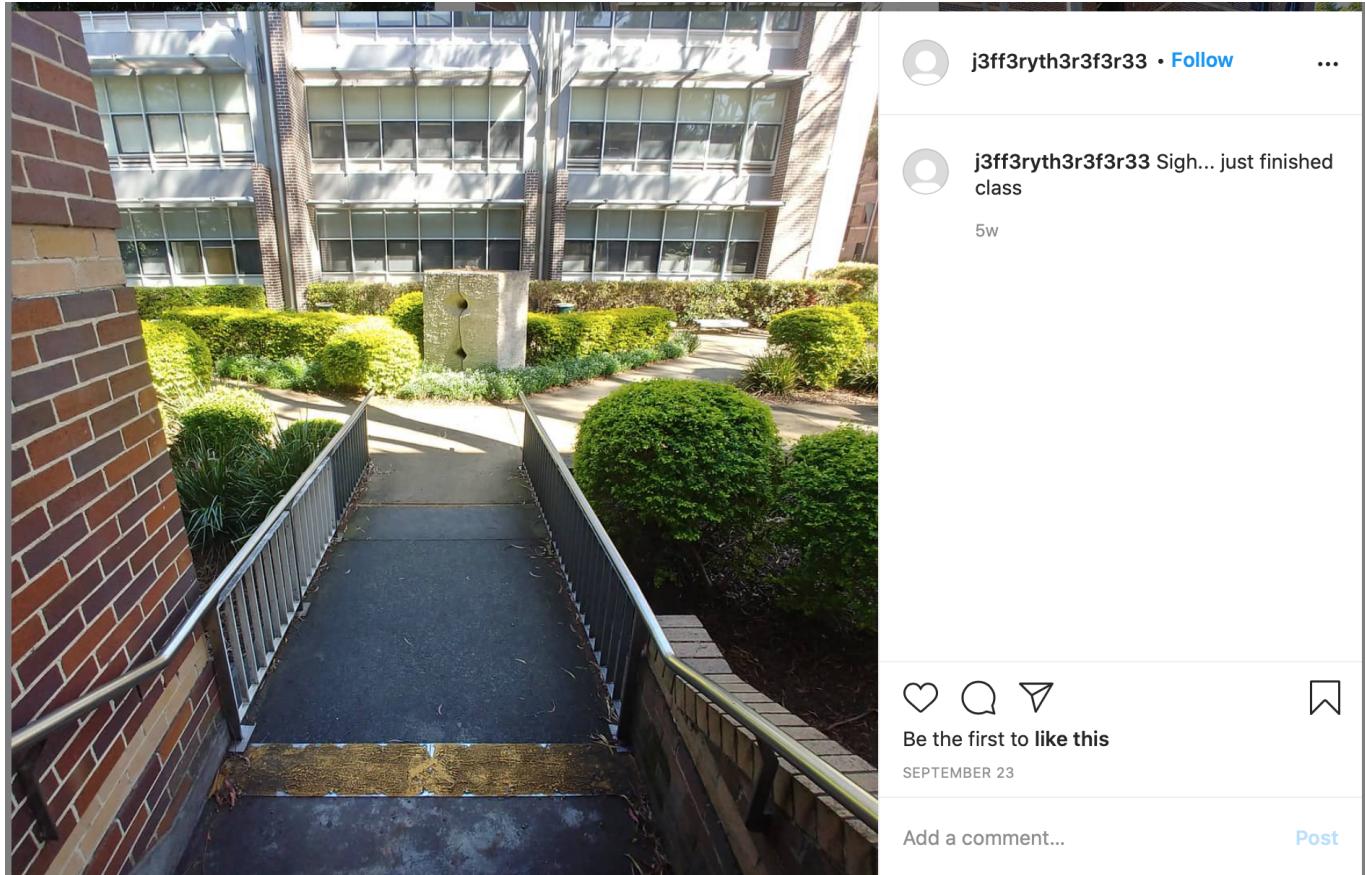
Be the first to like this

SEPTEMBER 23

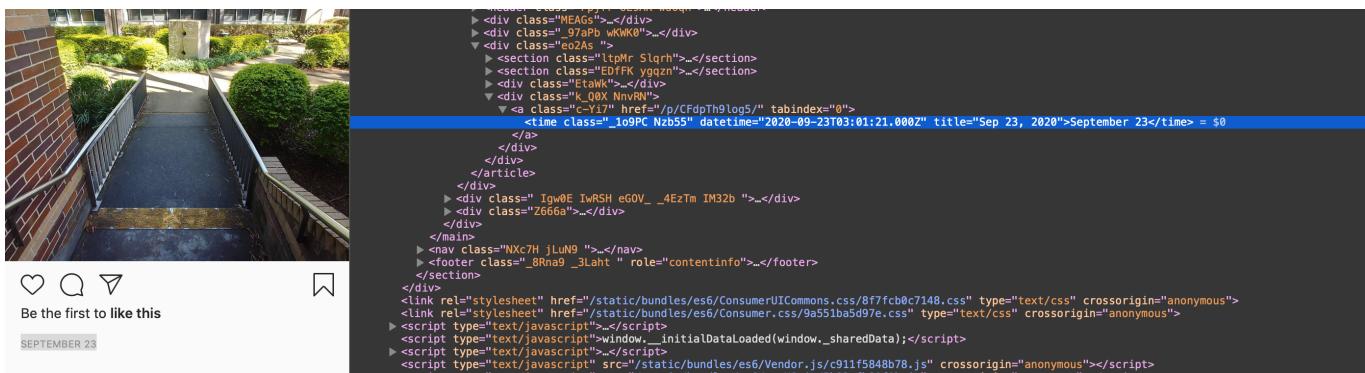
By inspecting element, we see this was taken at around 2:00am on the 23rd of September. However, it definitely doesn't look like it's 2:00am (and it'd be unlikely his class would be on then). A google search shows that this time is likely to be GMT, 11 hours behind AEST (since the picture was before daylight savings, we don't have to worry about AEDT).

This means his class started at 1PM.

It's like to assume it either finished at 2PM or 3PM, however, another post shows him leaving class:



By inspecting element again, we see it was an hour class, and he finished at 2PM.



Additionally, this looks like it's around 9 Wallys Walk, and since we're told to use the old building number, we know he's had class at E6A.

Arranging all this information into the flag format, we know the flag is:

```
hackmac{E6A_1PM_2PM}
```

Intelli-telnet

This challenge is relatively simple. The description gives us a great deal of information:

- The telnet password is the device's default
- The device is a Zmodo camera

- The flag is the password (in the flag format)

A quick google search shows reports of a telnet password for this device being

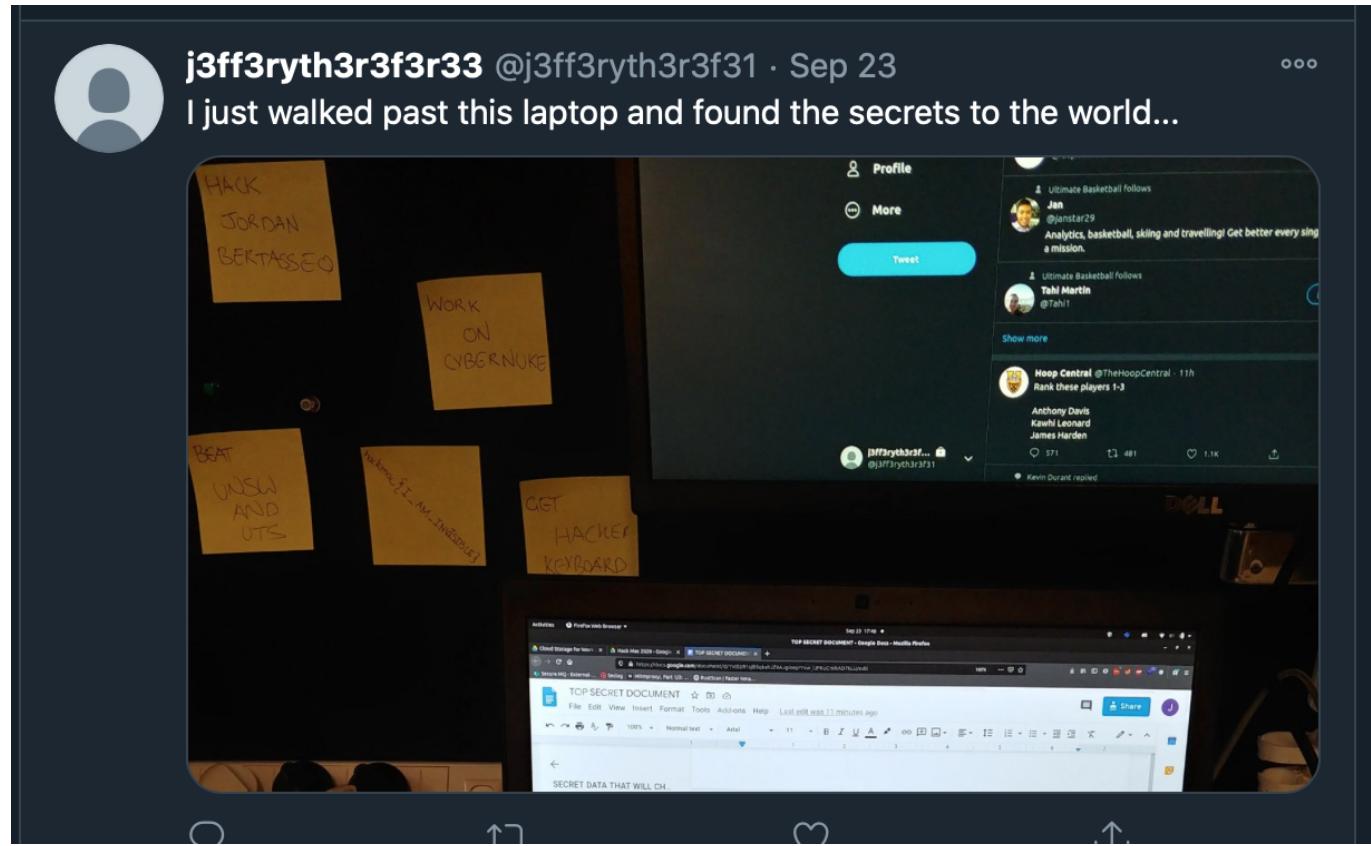
```
zmodo19820816
```

Attempting to connect to the host with this password shows that this is the correct password, so the flag is:

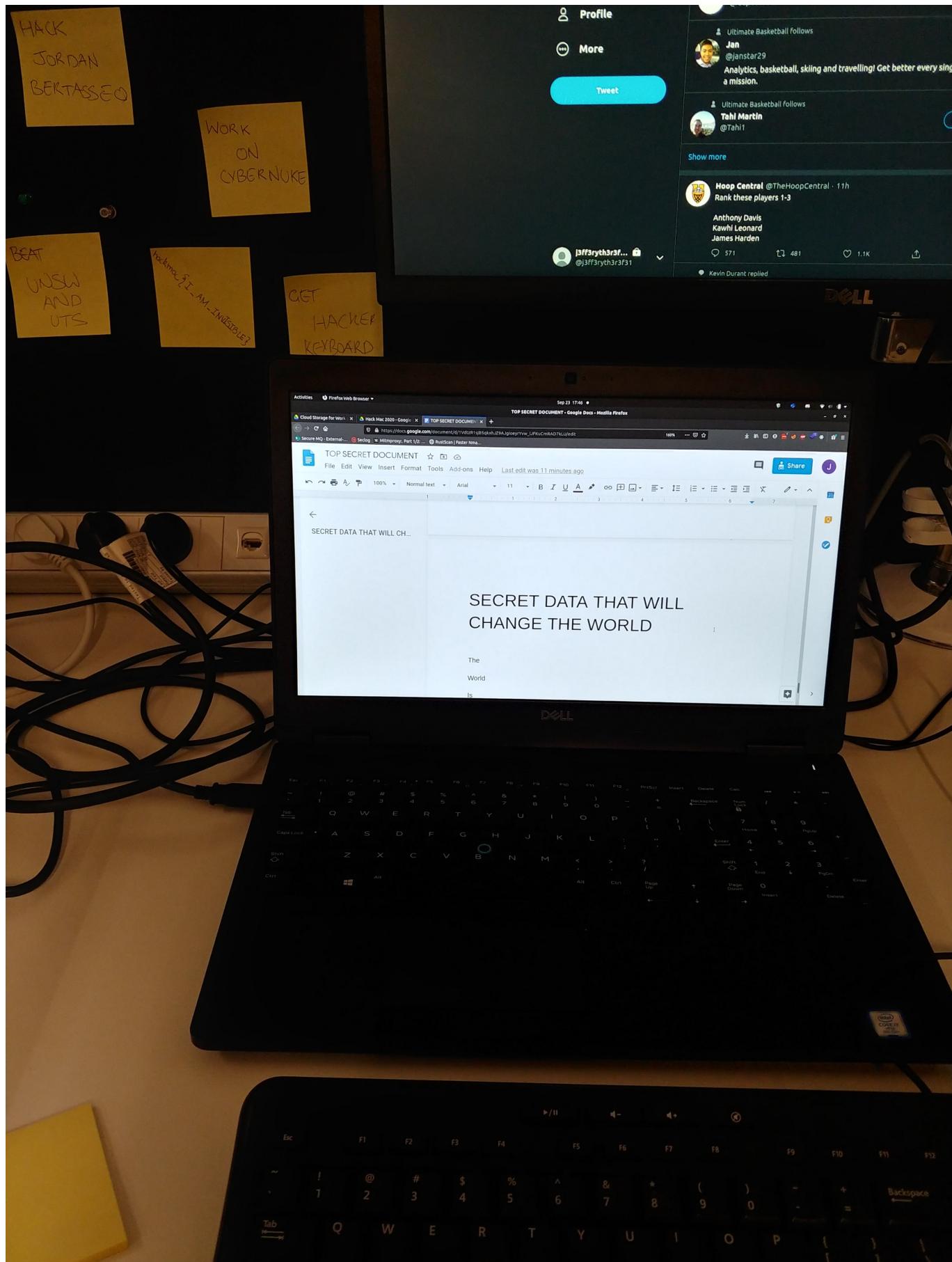
```
hackmac{zmodo19820816}
```

Keep Me Posted

Originally, we could not see many of Jeffery's twitter posts, but eventually, we were able to see this post:



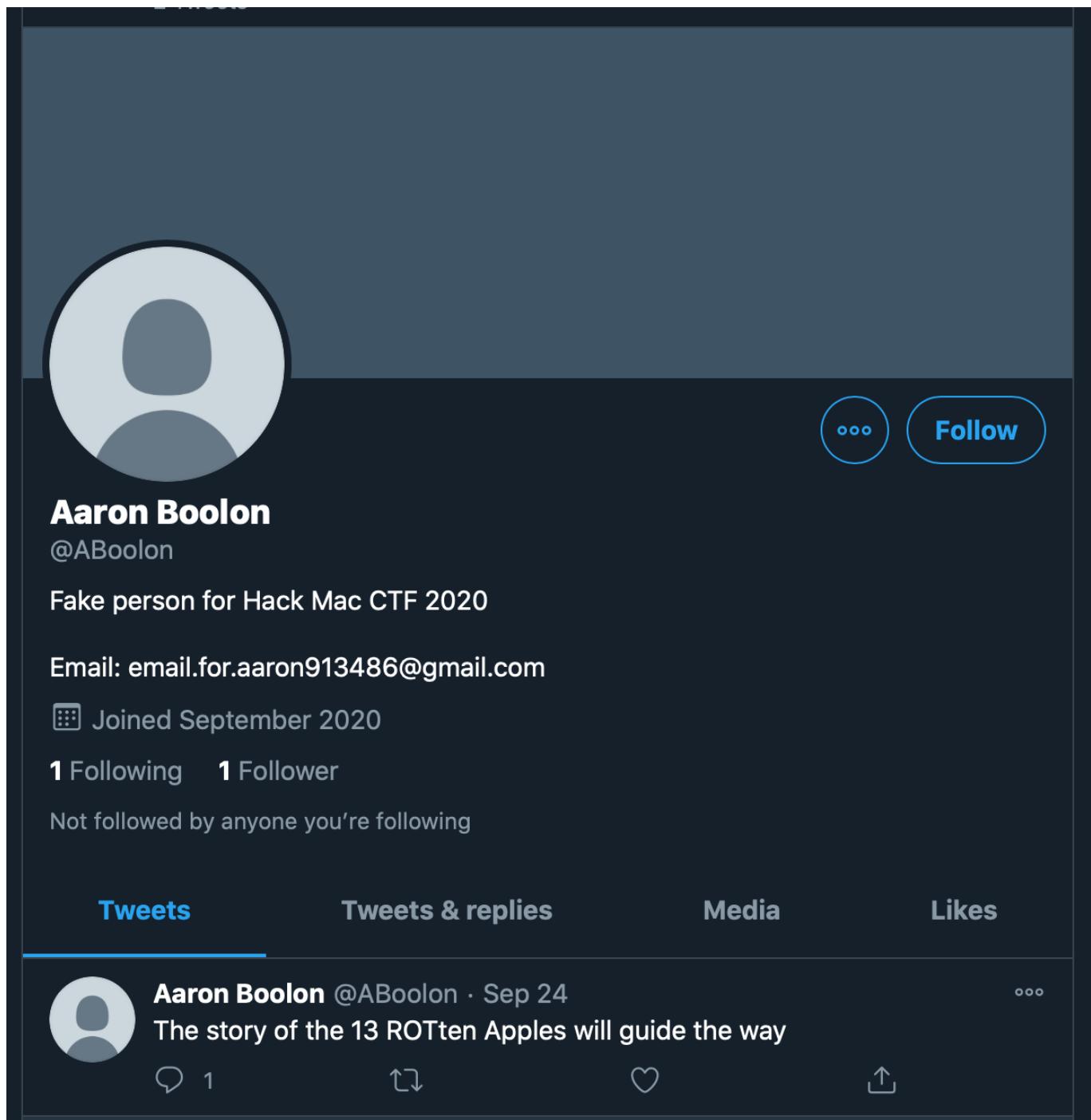
We see a wall of post-it notes, where one has a flag in the hackmac format:



```
hackmac{I_AM_INVISIBLE}
```

Postman Pat knows everything about 13 Rotten Apples

On Jeffery's Twitter we can see he has a follower called "Aaron Boolon" who is also a fake account created for Hackmac:



A screenshot of a Twitter profile for a user named "Aaron Boolon". The profile picture is a placeholder user icon. The name "Aaron Boolon" is displayed in bold black text, followed by the handle "@ABoolon". A bio below the handle reads "Fake person for Hack Mac CTF 2020". The "Email" field shows "Email: email.for.aaron913486@gmail.com". The "Joined" section indicates the user joined in September 2020. Below the bio, it says "1 Following" and "1 Follower", with a note that the user is "Not followed by anyone you're following". At the bottom, there are four tabs: "Tweets" (which is highlighted in blue), "Tweets & replies", "Media", and "Likes". A single tweet from "Aaron Boolon" (@ABoolon · Sep 24) is shown, reading "The story of the 13 ROTten Apples will guide the way". The tweet has 1 reply, 0 retweets, and 0 likes.

We could see his first post referred to "["13 ROTten Apples"](#)", so we knew it was likely to be related to this challenge.

Aaron Boolon (@ABoolon)

The story of the 13 ROTten Apples will guide the way

1:03 PM · Sep 24, 2020 · Twitter Web App

j3ff3ryth3r3f3r33 @j3ff3ryth3r3f3r33 · Sep 24

Replying to [@ABoolon](#)

What was the site again sir? I forgot

Aaron Boolon @ABoolon · Sep 24

you idiot. Don't show anyone this secret code that will lead the way to the gate.

uggc://onqoblf.unpxznp.klm

We could see an encrypted message in the form of a hyperlink (due to the ://)

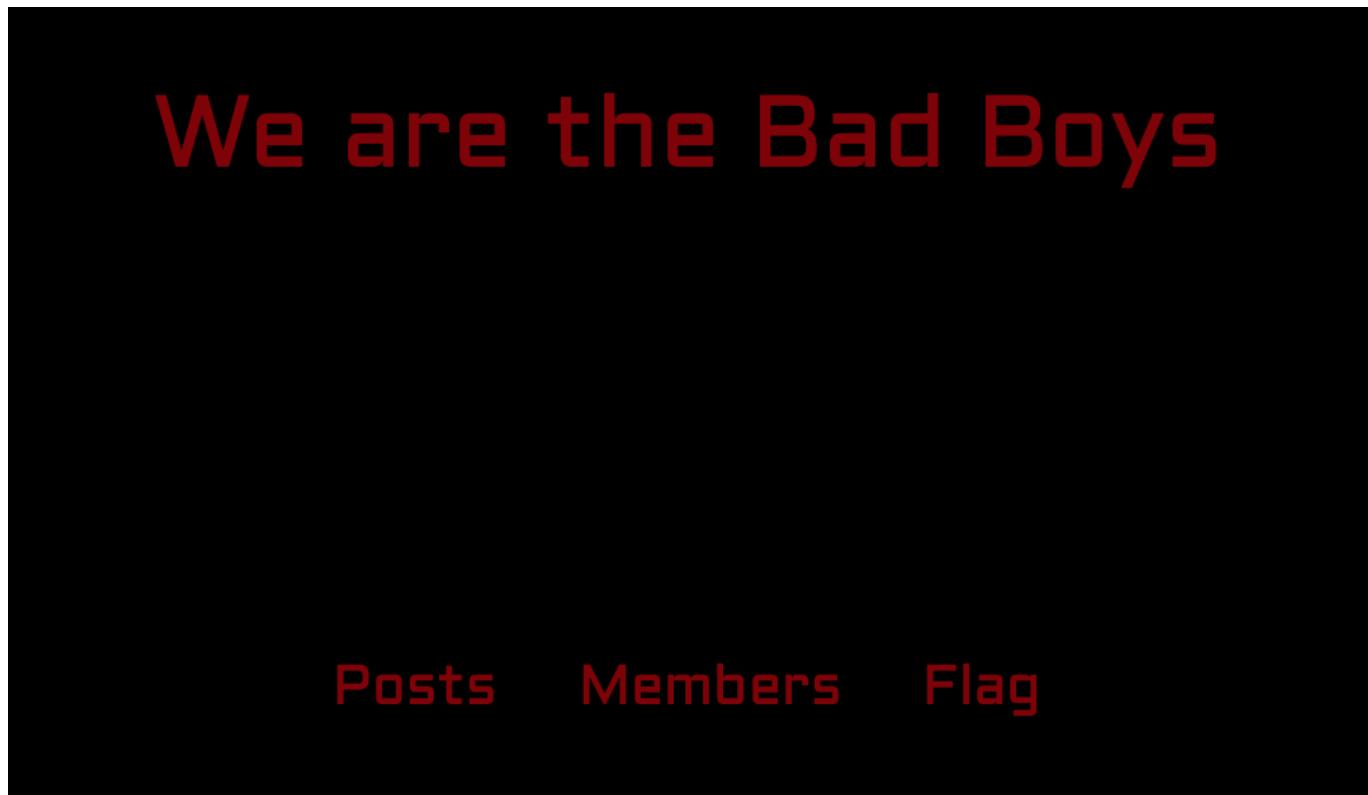
Using Cyberchef, and decoding ROT13:

Recipe	Input	Output
ROT13	uggc://onqoblf.unpxznp.klm start: 0 end: 26 length: 26 lines: 1	http://badboys.hackmac.xyz start: 0 time: 4ms end: 26 length: 26 lines: 1
Amount: 13		
Rotate lower case chars		
Rotate upper case chars		

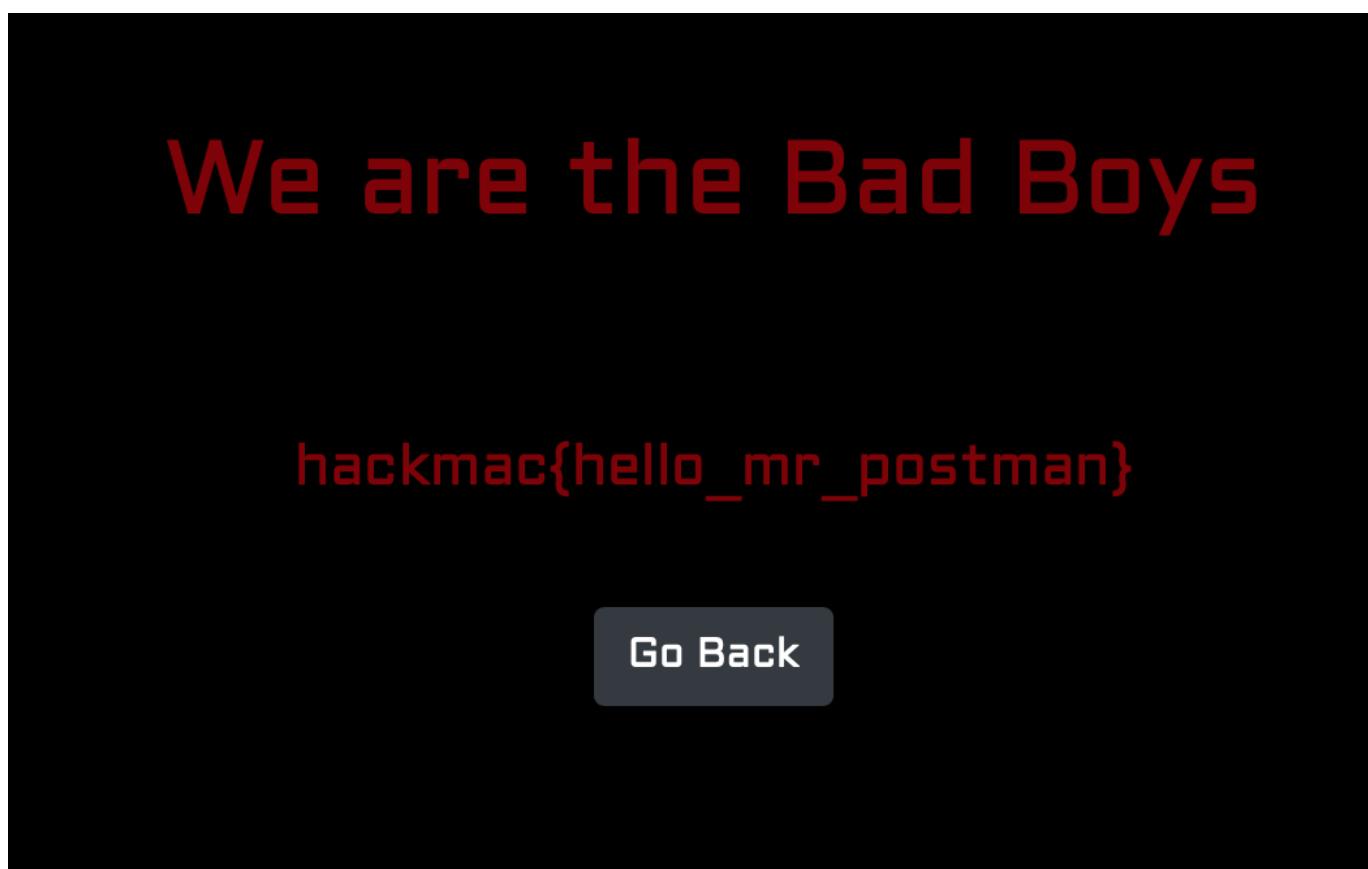
We get a new hackmac website:

http://badboys.hackmac.xyz

On the website, we see an option that says [flag](#):



Of course, we click on this and we get something in the hackmac flag format:



```
hackmac{hello_mr_postman}
```

Remote Access 1

Both remote access challenges were solved after hints were already given, but with techniques learned after the event I will explain what I believe is the intended solution.

Solve with hint

The hint suggests that we look up a wikipedia list for the 10,000 most common passwords, and gives us a number (2211). By looking at the 2211th most common password, **trance**, we can log into the RDP server.

In the standard format, the flag is:

```
hackmac{trance}
```

Intended solution

The intended solution was likely a dictionary brute-force attack on the target host using Wikipedia's most common passwords list as the dictionary.

Remote Access 2

Similar to [Remote Access 1](#), this challenge was solved after the hints were released. The same method is used in this challenge, but now the hint says **2213**. The corresponding password in the Wikipedia list is **playtime**, so the flag is:

```
hackmac{playtime}
```

RTSP That Thing

From the challenge description, we can find some useful pieces of information:

- We're looking for an RTSP stream
- The stream is unencrypted
- The brand of camera is Zmodo

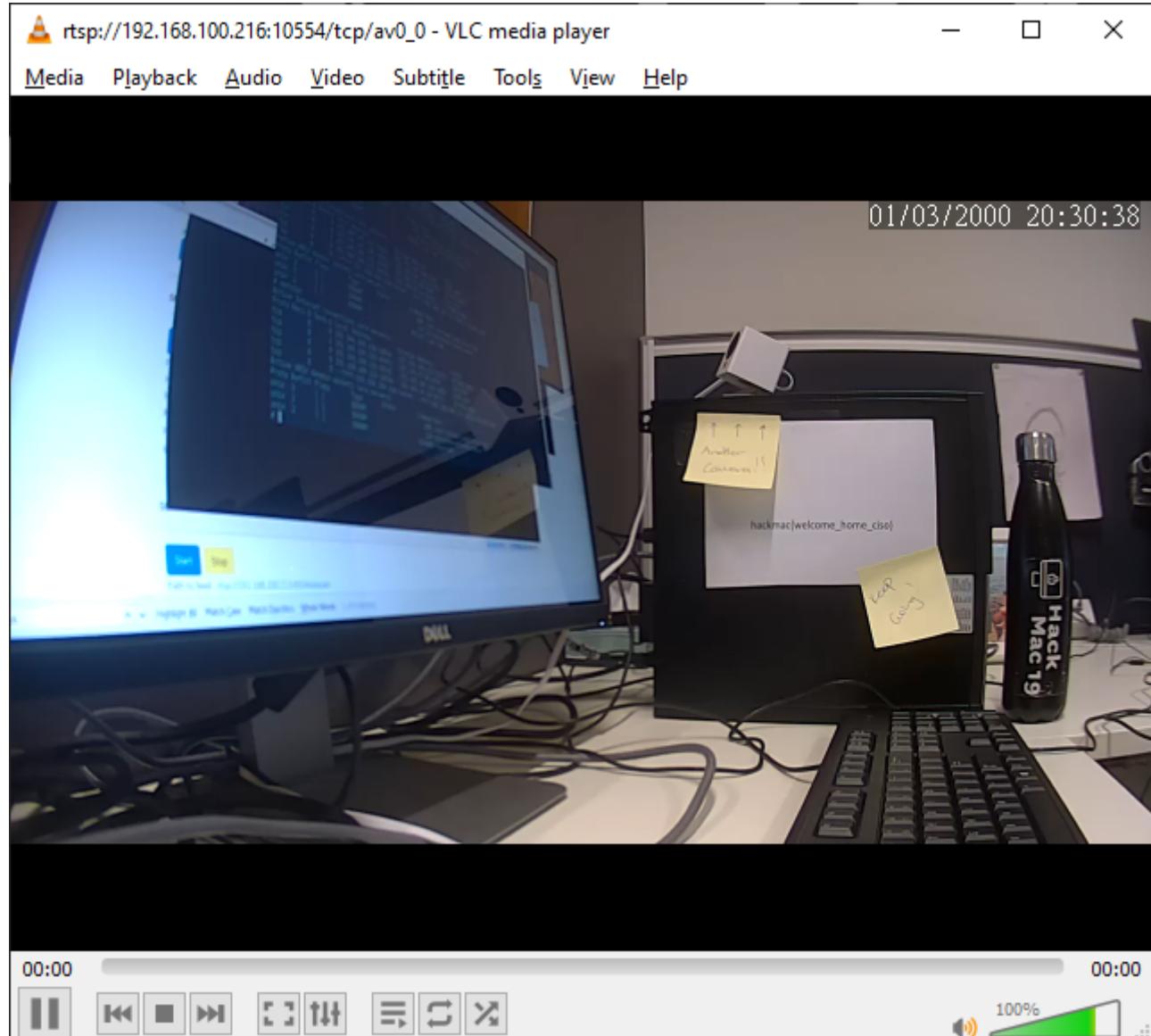
Let's first have a look at what ports we should be scanning. A quick google search for **Zmodo RTSP** shows some common URLs used by Zmodo cameras to display their RTSP feeds. They appear to use the port 10554.

We can do an nmap scan of the network to find hosts with this port open.

Now that we have the host address, we can try some of the common URLs in VLC to get the RTSP stream.

```
rtsp://192.168.100.210:10554/udp/av0_0
```

This gives us a video stream, and the flag is written on a piece of paper.



Screenshot Credit: David Sanders

```
hackmac{welcome_home_ciso}
```

Smart Camera Secret File

This challenge is a continuation of [Intelli-telnet](#). Firstly, we need to connect to the camera with Telnet using the password from the previous challenge.

Once we are in, we can start looking for the hidden file mentioned in the challenge description.

```
$ ls -a
```

Nothing of note is found in this directory, so we can navigate up a directory and list the files

```
$ cd ..  
$ ls -a
```

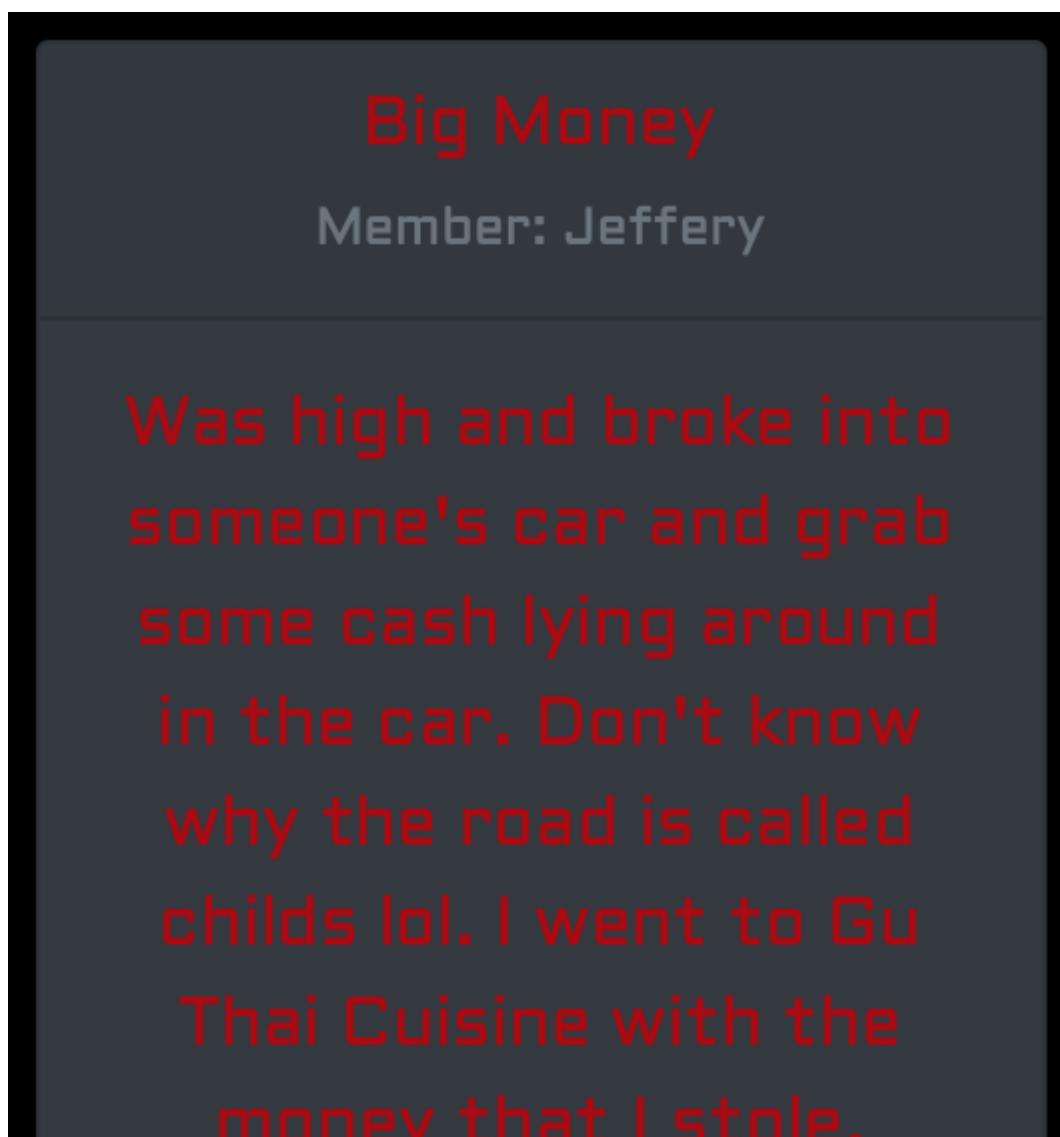
Here, we can see a file of note. We can access the contents of the file:

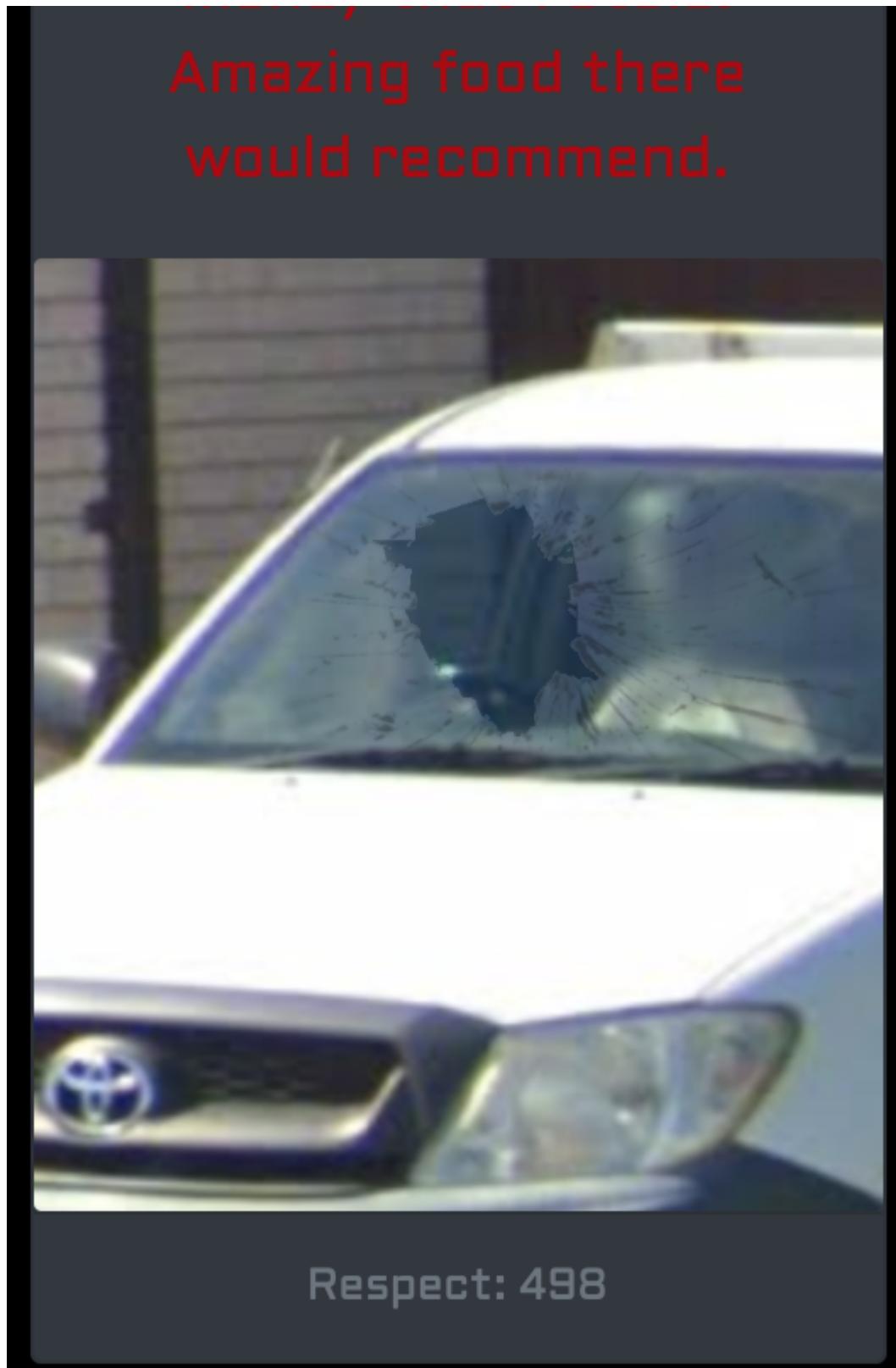
```
$ cat flag.txt
```

And the flag is printed to the terminal!

The Reality Deep Down in our Hearts

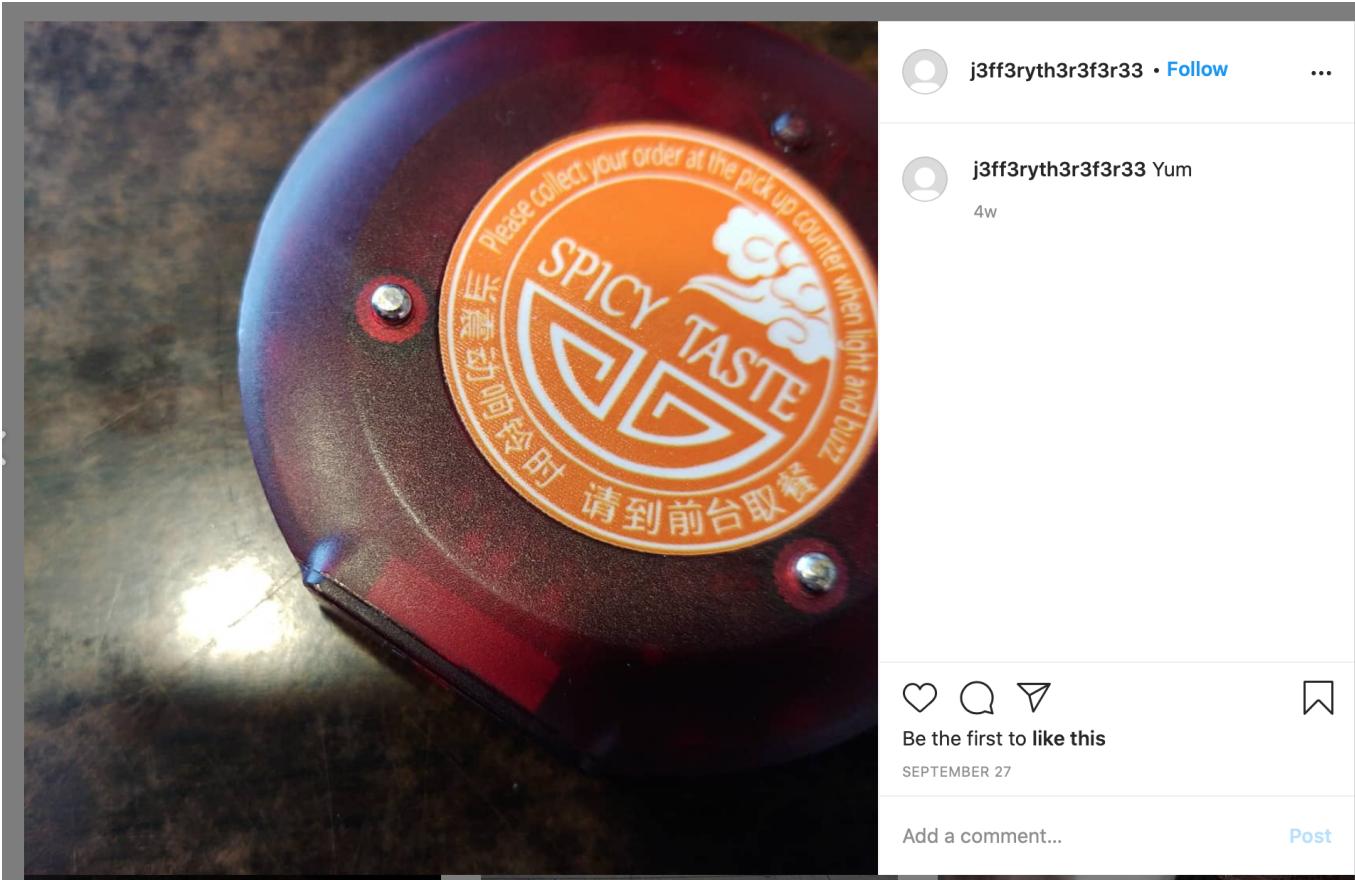
The [posts](#) page on the badboys website shows us this image and description:





It makes reference to a restaurant called [Gu Thai Cuisine](#). Although we don't see that name in the Instagram page, we know Jeffery went to two places that serve Asian food.

As one of the posts says [Sushi Hotaru](#), by elimination, we figure out it is most likely referring to the other post made on the 27th of September:



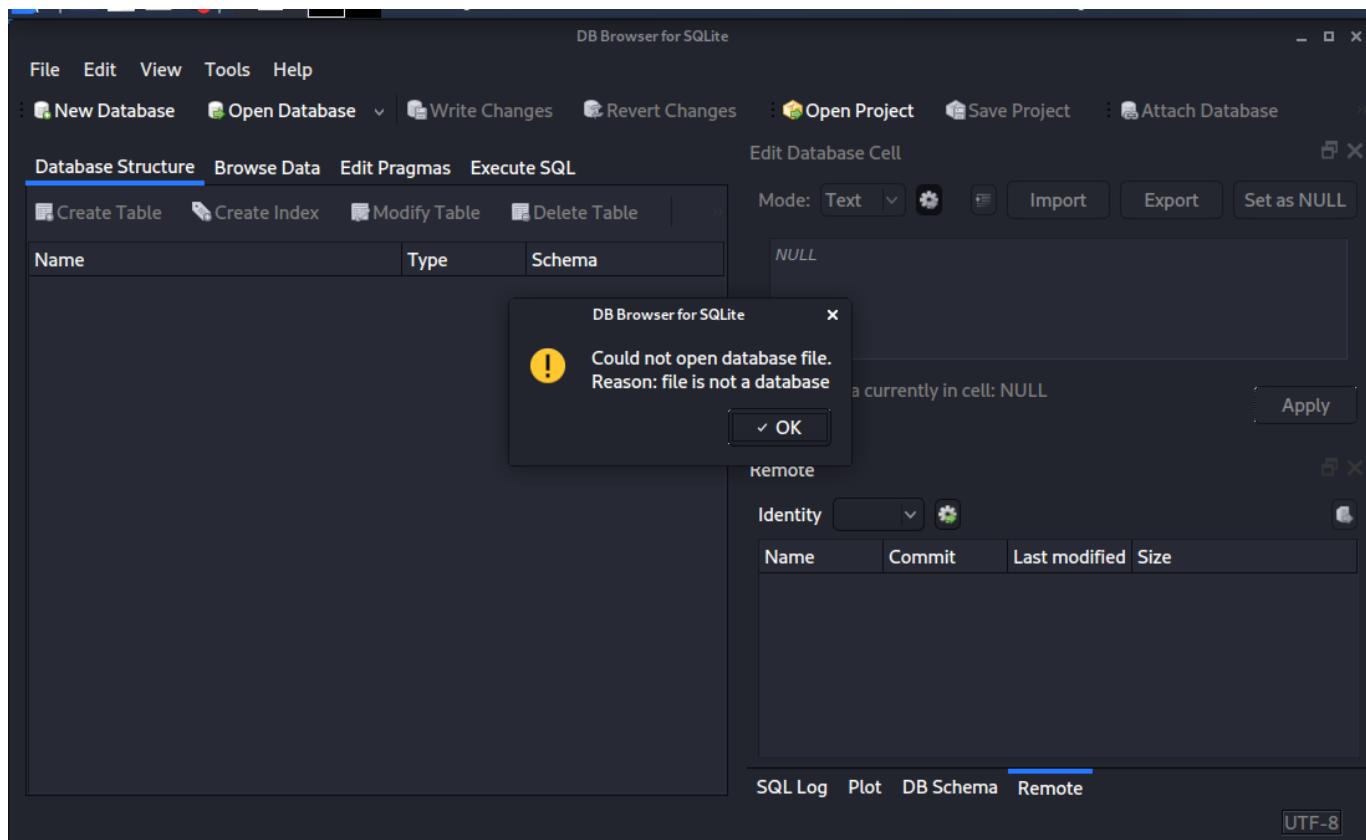
We inspect element and check this would still be on the same date in AEST time:

```
<!DOCTYPE html>
<html lang="en" class="js logged-in client-root js-focus-visible sDN5V">
  <head></head>
  <body>
    <div id="react-root">
      <section class="9eogI E3X2T">
        <main class="SCxLW og64R" role="main">
          <div class="Kj7h1 ylx9G">
            <div class="ltekP">
              <article class="QBxJ M9sTE h0YNM SgTZl" role="presentation" tabindex="-1">
                <header class="Ppjfr UE9AK wd0qh"></header>
                <div class="MEAGS"></div>
                <div class="97aPb wKK0Q"></div>
                <div class="eo2Aa ">
                  <section class="ltpmP Slqrh"></section>
                  <section class="EDffK ygqzN"></section>
                  <div class="EtawK"></div>
                  <div class="k_QBX NnvRN">
                    <a class="c-Y17" href="/CFoVG0115t3/" tabindex="0">September 27, 2020</a>
                  </div>
                </div>
                <div class="IgwOE IwRSH eGOV_ _4EzTm IM32b "></div>
                <div class="Z666a"></div>
              </article>
            </div>
            </main>
            <nav class="NXc7H jLuN9 "></nav>
            <footer class=" _8Rna9 _3Laht " role="contentinfo"></footer>
          </div>
          <link rel="stylesheet" href="/static/bundles/es6/ConsumerUICommons.css/8f7fcbb0c148.cs" type="text/css" crossorigin="anonymous">
          <link rel="stylesheet" href="/static/bundles/es6/Consumer.css/9w551ba5d97e.css" type="text/css" crossorigin="anonymous">
          <script type="text/javascript"></script>
          <script type="text/javascript">window._initialDataLoaded(window._sharedData);</script>
          <script type="text/javascript"></script>
          <script type="text/javascript" src="/static/bundles/es6/Vendor.js/c911f5848b78.js" crossorigin="anonymous"></script>
          <script type="text/javascript" src="/static/bundles/es6/en-US.js/7hB3afB80f0h.js" crossorigin="anonymous"></script>
```

hackmac{27_September}

Time for a Little Magic Trick

Trying to open Joker.db in database software reveals it's not actually a database.



To see what is actually in the `.db` file, we can run

```
$ binwalk Joker.db
```

This will reveal the true contents of the file

A screenshot of a terminal window on a Kali Linux system. The prompt is 'mungo@kali: ~/shared\$'. The user runs the command 'binwalk Joker.db'. The output shows the following table:

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	SQLite 3.x database,, user version 185342226
16	0x10	JPEG image data, JFIF standard 1.01

As we can see, the file has a JPEG image hidden within. To see the actual image, we need to extract it from the carrier file

```
$ binwalk -D='.*' Joker.db
```

This will put the contents of the file in a new folder

The terminal window shows two runs of the `binwalk` command:

```
mungo@kali:~/shared$ binwalk joker.db
Rubbish
File System
DECIMAL      HEXADECIMAL      DESCRIPTION
-----      -----      -----
0            0x0              SQLite 3.x database,, user version 185342226
16           0x10             JPEG image data, JFIF standard 1.01

mungo@kali:~/shared$ binwalk -D='.*' joker.db
File System
DECIMAL      HEXADECIMAL      DESCRIPTION
-----      -----      -----
0            0x0              SQLite 3.x database,, user version 185342226
16           0x10             JPEG image data, JFIF standard 1.01
```

The file manager window shows the directory `_Joker.db.extracted` containing a file system device and two files, one of which is a JPEG image.

From here, we can simply open the file with an image viewer of choice



And we have our flag!

hackmac{did_i_trick_ya}

To Kill a Blue Bird

We knew it was likely that Jeffery wouldn't only have an instagram page, and so we checked twitter right away. We tried:

<https://twitter.com/j3ff3ryth3r3f3r33/>

which gave us no results.

However, by looking up **j3ff3ryth3r3f3r33** in the Twitter search bar, we got this page:



A screenshot of a Twitter profile page. The header bar is dark blue with white text. On the left is a blue arrow pointing left, followed by the username "j3ff3ryth3r3f3r33" in large white font, and "11 Tweets" in smaller white font below it. Below the header is a large, circular placeholder icon for a profile picture, showing a stylized person's head and shoulders. To the right of the icon are two buttons: a blue-outlined circle with three white dots ("More options") and a blue-outlined rounded rectangle with the word "Follow" in blue text. The main profile area has a dark background. The username "j3ff3ryth3r3f3r33" is displayed in large white bold font at the top, followed by the handle "@j3ff3ryth3r3f31" in smaller white font. Below this is a bio in white font: "Fake person for HackMac CTF 2020". An email address "Email: jeffery.is.the.best913486@gmail.com" is shown in white. A location and joining date are listed: "Sydney NSW" with a location pin icon, and "Joined September 2020". Below these are statistics: "13 Following" and "1 Follower". A note below says "Not followed by anyone you're following". At the bottom of the profile section are four tabs: "Tweets" (underlined in blue), "Tweets & replies", "Media", and "Likes".

A quick look at his followers shows the majority of pages are related to basketball:

← **j3ff3ryth3r3f3r33**
@j3ff3ryth3r3f31

Followers **Following**

 **Stephen Curry** ✓
@StephenCurry30 [Follow](#)

Believer. Husband to [@ayeshacurry](#), father to Riley, Ryan and Canon, son, brother. Golden State Warriors guard. Davidson Wildcat. Philippians 4:13 #IWILL

 **Hoop Central**
@TheHoopCentral [Follow](#)

Providing accurate analysis, news, rumors, trades, signings, stats, highlights, and throwbacks around the NBA. Turn Post Notifications ON.

 **ESPN** ✓
@espn [Follow](#)

Serving sports fans. Anytime. Anywhere. Download the ESPN App [Download](#)

 **Kevin Durant** ✓
@KDTrey5 [Follow](#)

IM ME, I DO ME, AND I CHILL. [@35ventures](#) [@boardroom](#)

 **LeBron James** ✓
@KingJames [Follow](#)

EST. AKRON - ST.V/M Class of '03 [LeBronJamesFamilyFoundation.org](#)
#IPROMISE

 **Los Angeles Lakers** ✓
@Lakers [Follow](#)

Welcome to the [#LakeShow](#) 🏀 | 🏆 17x Champions

 **Miami HEAT** ✓
@MiamiHEAT [Follow](#)

#UnitedInBlack #BlackLivesMatter 🤝

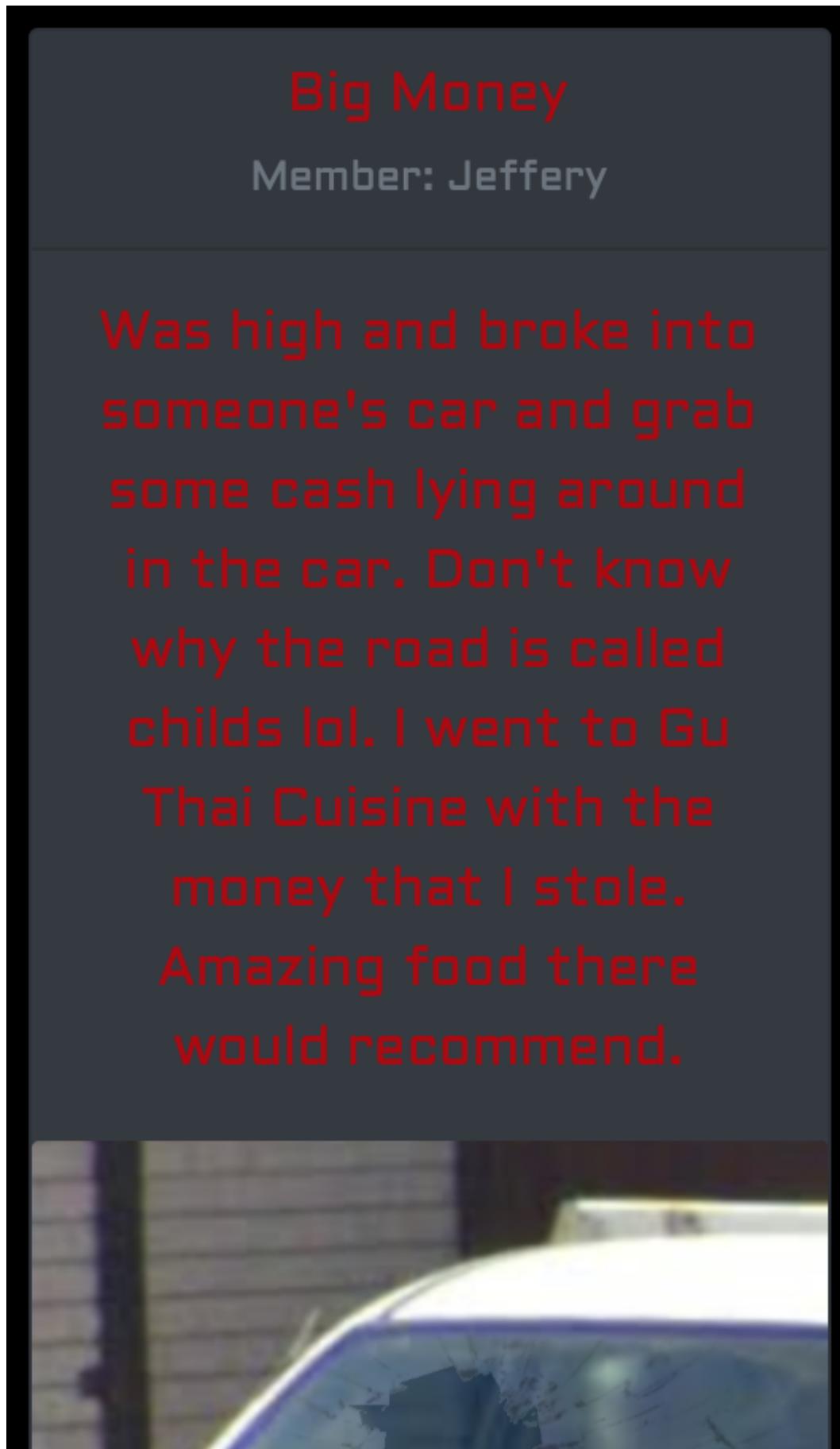
Hence we presumed it is likely he plays basketball.

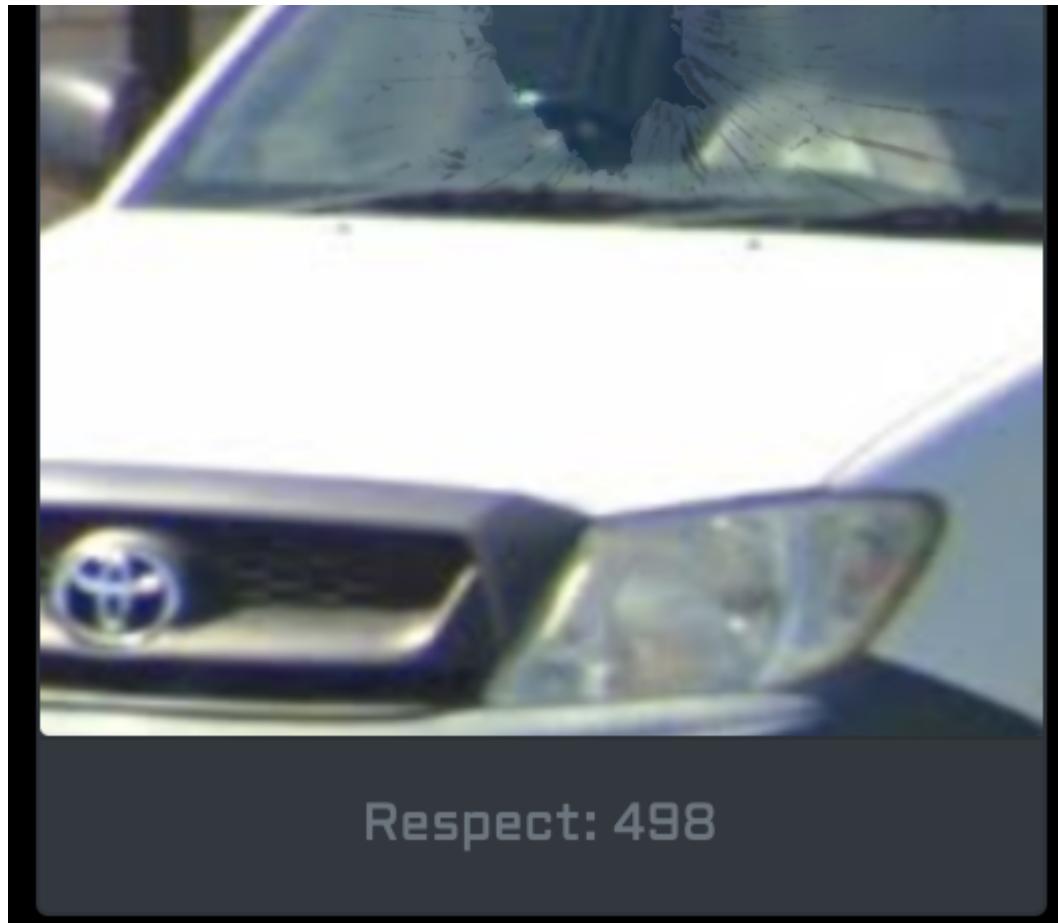
NOTE: A google search for 'Blue Bird' didn't reveal anything particularly insightful, and to this day, I'm not too sure on its relation to the challenge (although I believe it is possible it could have been a hint).

hackmac{basketball}

Uh Oh

Checking the same image from the challenge 'The Reality Deep Down in our Hearts':





We can see this photo has obviously been photoshopped, and it is likely we will be looking for a car that looks like the one in the photo (without a broken windshield), presumably in Google maps street view.

A Google maps search for a place called **Gu Thai Cuisine** on **childs**:

Gu Thai Cuisine childs

All Maps Images Shopping News More Settings Tools

About 2,390,000 results (0.70 seconds)

www.guthaicuisine.com.au › contact ▾

Contact Gu Thai Cuisine

Our restaurant is located in a nice area in Chipping Norton, and we made sure you'll fall in love with ... 2/94 Childs Rd., Chipping Norton NSW 2170, Australia.

www.guthaicuisine.com.au › takeaway-food-delivery ▾

Takeaway Food & Delivery - Gu Thai Cuisine

See what's cooking and come pick it up from: 2/94 Childs Rd., Chipping Norton NSW 2170. We can bring the food to your doorstep (within our delivery areas).

www.guthaicuisine.com.au ▾

Gu Thai Cuisine - Food delivery - Chipping Norton - Order online

Order Online for Takeaway / Delivery. Here at Gu Thai Cuisine - Chipping Norton you'll experience delicious Thai cuisine. Try our mouth-watering dishes, ...

Missing: childs | Must include: childs

www.facebook.com › ... › Fast Food Restaurant ▾

Gu Thai Cuisine - Home - Chipping Norton, New South Wales ...

Gu Thai Cuisine, Chipping Norton, New South Wales, Australia. 597 likes. Enjoy our authentic

See photos

Georges River Epsom Rd Epsom St Ashfordby St Charnbury St Valley Cres Cream & Co

Gu Thai Cuisine at Chipping Norton

Website Directions Save

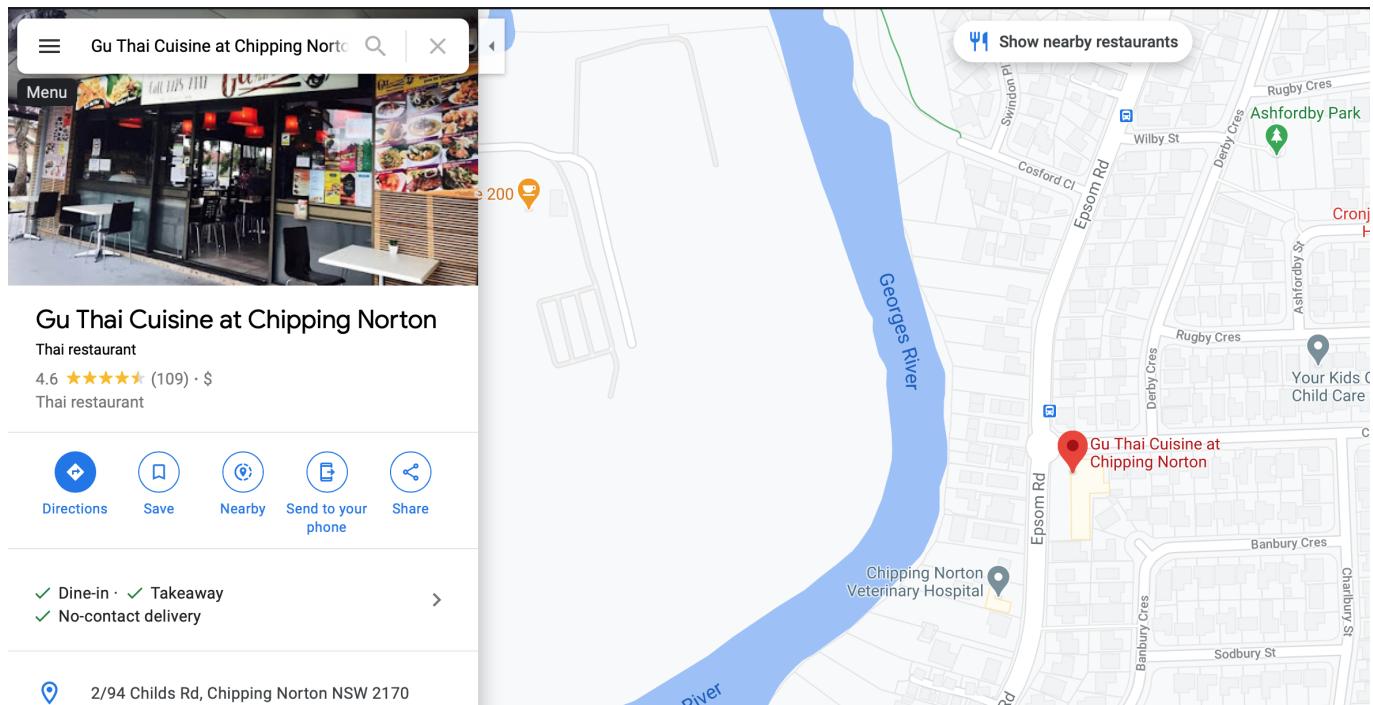
4.6 ★★★★★ 109 Google reviews

\$ · Thai restaurant

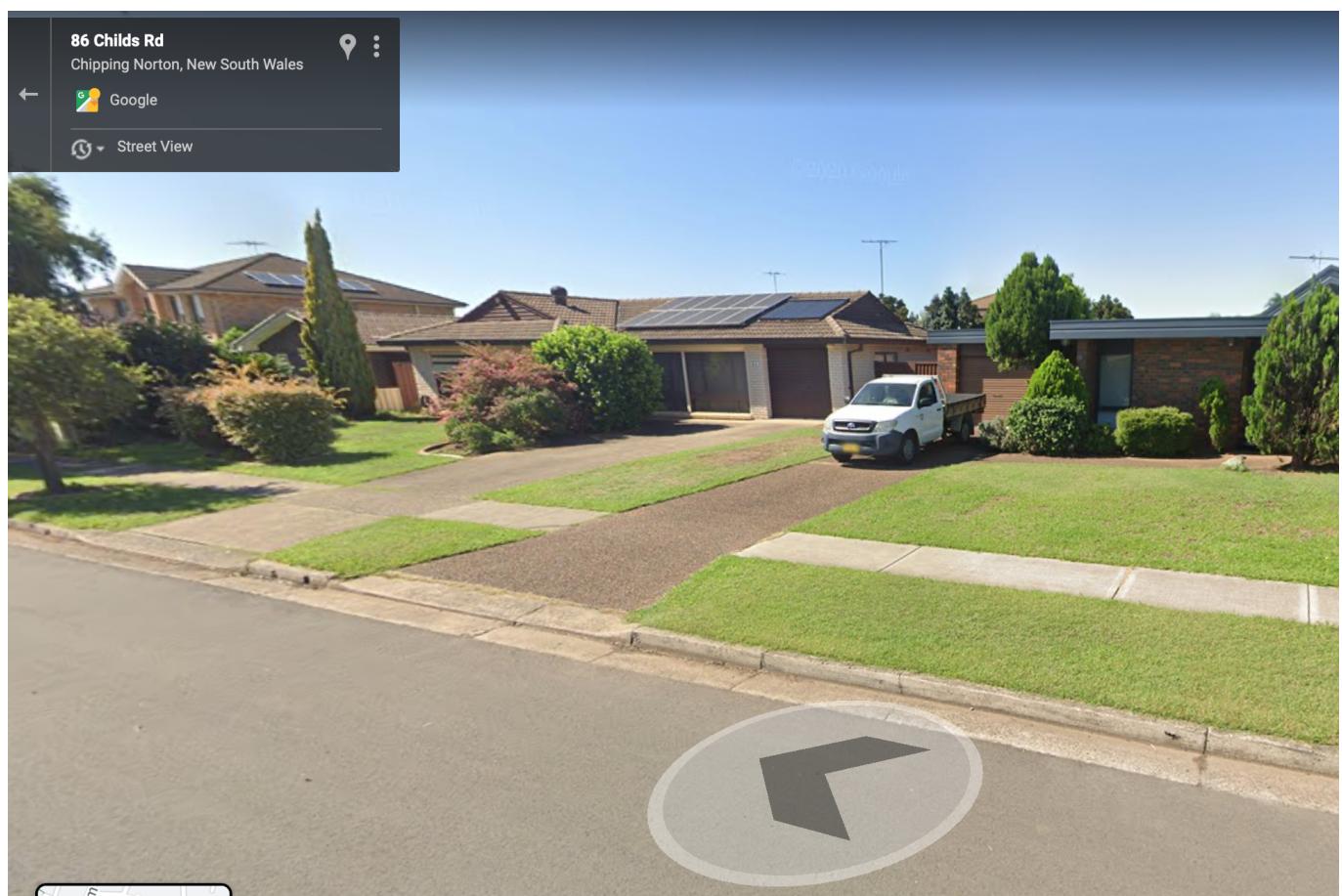
✓ Dine-in · ✓ Takeaway · ✓ No-contact delivery

Address: 2/94 Childs Rd, Chipping Norton NSW 2170
Hours: Open · Closes 9:30PM ·
Menu: guthaicuisine.com.au

Reveals there is a restaurant in Chipping Norton on Childs Rd:



We realise the location is likely to be on this street, and so we look for a vehicle resembling the one in the picture. We eventually find a vehicle in front of a house made from the same material as shown in the picture, at **86 Childs Rd, Chipping Norton**:



We rearrange this information to fit the flag format provided.

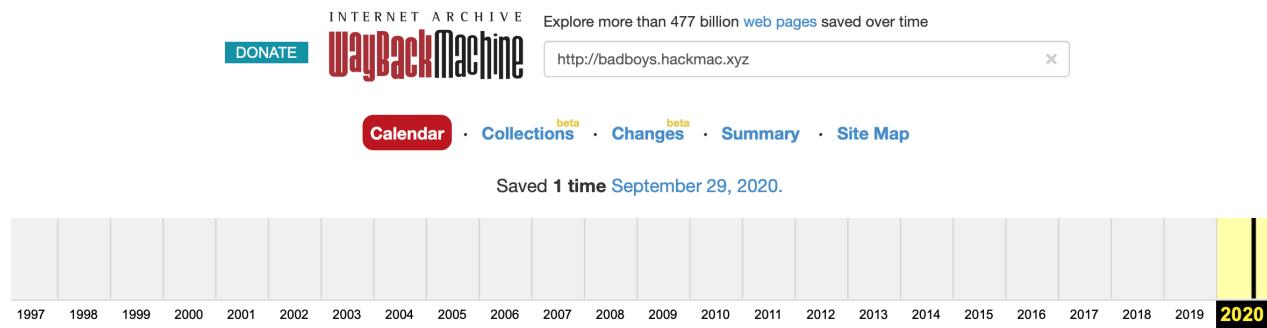
```
hackmac{86_childs_rd_chipping_norton}
```

Way Back Home

In this challenge, we knew the title probably referred to using the Wayback Machine:

```
https://web.archive.org
```

We tried checking Jeffery's Twitter, Instagram, LinkedIn, Aaron's Twitter (and just about every other thing we had found up until this point), until we checked the badboys website:



Checking this result immediately gave us a flag in the hackmac format:

We are the Bad Boys

This website is still in development

List of founding gang members:

Aaron Boolon

Mark Hefton

Bob Smith

Jordani Bertasseo

hackmac{jordan_has_my_family_send_help}

```
hackmac{jordan_has_my_family_send_help}
```

Where's Jeffery?

From one of the instagram posts, we knew Jeffery had replied to a message with a reference to a Snapchat account:



j3ff3ryth3r3f3r33 • [Follow](#)

...

5w



jordanbertasso Run Jeffery run!



3w 2 likes Reply

— Hide replies



j3ff3ryth3r3f3r33

@jordanbertasso Yo I'm

trying to get the most
friends on snapchat! Add
me and I'll add you
straight away.

https://drive.google.com/file/d/1KxEopREjLyZf9MHpcUtvzaBjEPu_xLR/view?usp=sharing

2w 3 likes Reply

From here, we went to:

https://drive.google.com/file/d/1KxEopREjLyZf9MHpcUtvzaBjEPu_xLR/view

And had access to his snapcode:



We created a snapchat account, scanned the snapcode on mobile, and did get added right away.

Originally we tried sending Jeffery messages, but received a sticker telling us to "go away" as he only talks to his real friends.

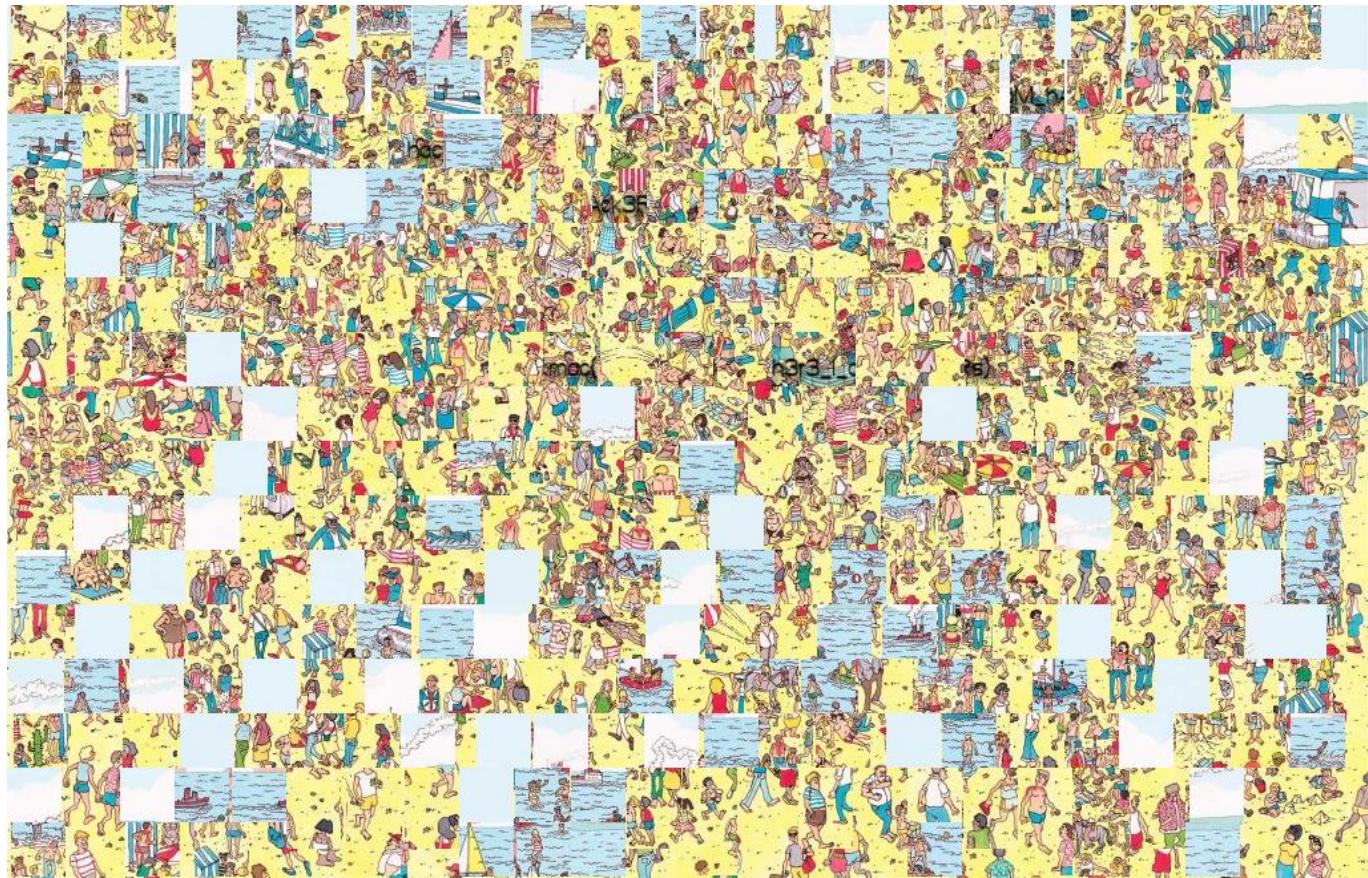
However, when we realised Snapchat had a map with friends' locations, we could see he was in Dee Why.

NOTE: Unfortunately, we did not get screenshots on mobile (and could not retrospectively access the snap map).

```
hackmac{deewhy}
```

Where's Wally

We are given an image file. Upon opening the image, it looks like a jumbled page from a Where's Wally book.



Upon closer inspection, we can see that there is some text that appears to have been written on the original image before it was jumbled up.



By rearranging these tiles with a graphic editor, we can find the flag



```
hackmac{h3r3_I_aM_hAck3Rs}
```

Work work work work work

This flag asks us to find out where Jeffery works. From the instagram page, we know that Jeffery's full name is **Jeffery Emmanuel Lee**:

A screenshot of a fake Instagram profile for a user named **j3ff3ryth3r3f3r33**. The profile picture is a placeholder gray circle. The bio reads: **Jeffery Emmanuel Lee** Fake person for HackMac 2020 CTF. The stats show 12 posts, 2 followers, and 0 following. There is a blue 'Follow' button and a three-dot menu icon.

It is likely that he would put his occupation on either a Facebook or LinkedIn page.

The first result we get when searching **Jeffery Lee** on LinkedIn gives us a fake person for hackmac:

[Message](#)[More...](#)

Jeffery Lee

Fake person for HackMac CTF 2020

Eastwood, New South Wales, Australia · [Contact info](#)



hackmac{yes_i_sell_stuff_...

Experience



Retail Salesperson

hackmac{yes_i_sell_stuff_hehe} · Internship

May 2020 – Present · 6 mos

We see a flag in the hackmac format.

hackmac{yes_i_sell_stuff_hehe}