



# MMU Workshop CTF

**Prepared By:**

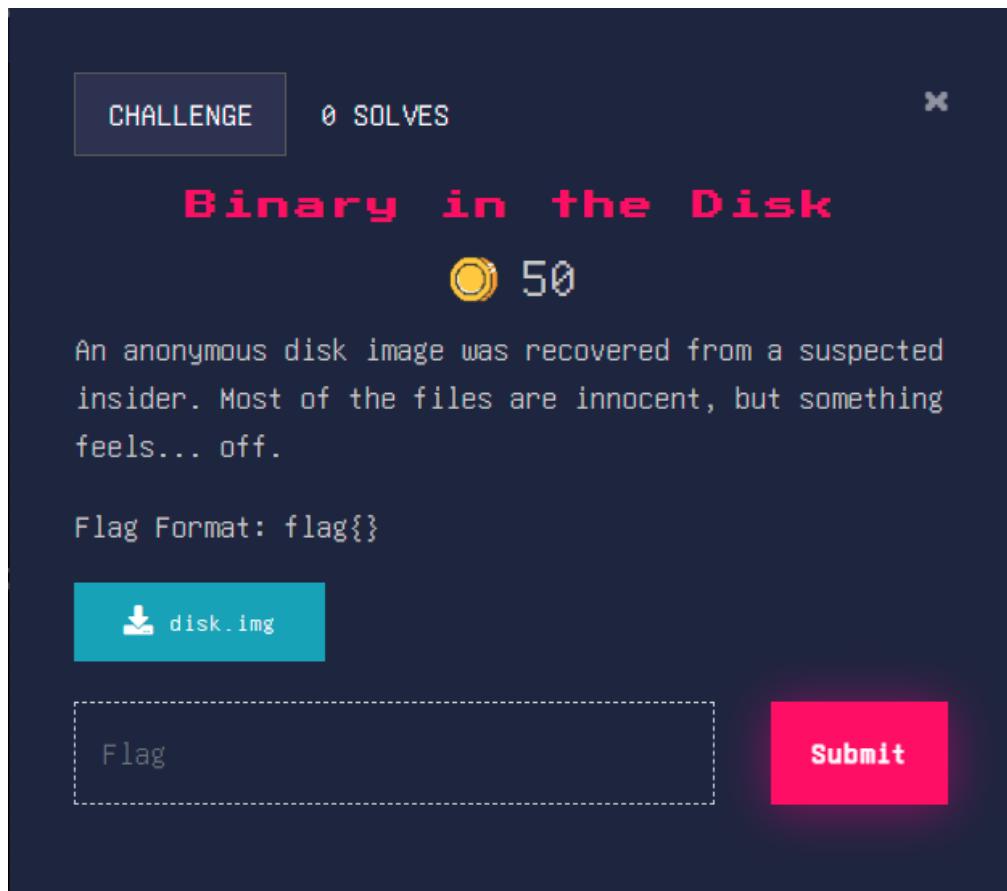
conan a.k.a [Nawfal](#)

## **TABLE OF CONTENTS**

[Forens] Binary in the Disk .....	3
[Forensics] Ghost in the Network 1.....	7
[Forensics] Ghost in the Network 2.....	12
[Forensics] Ghost in the Network 3.....	17
[Forensics] Encrypted mischief.....	21
[Forensics] Suspicious File .....	26

## [Forens] Binary in the Disk

Here is the question:



### Description

An anonymous disk image was recovered from a suspected insider. Most of the files are innocent, but something feels... off.

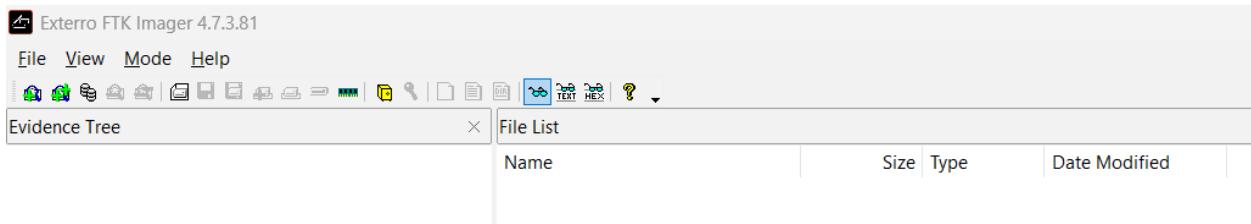
Flag Format: flag{}

### ⭐ Walkthrough

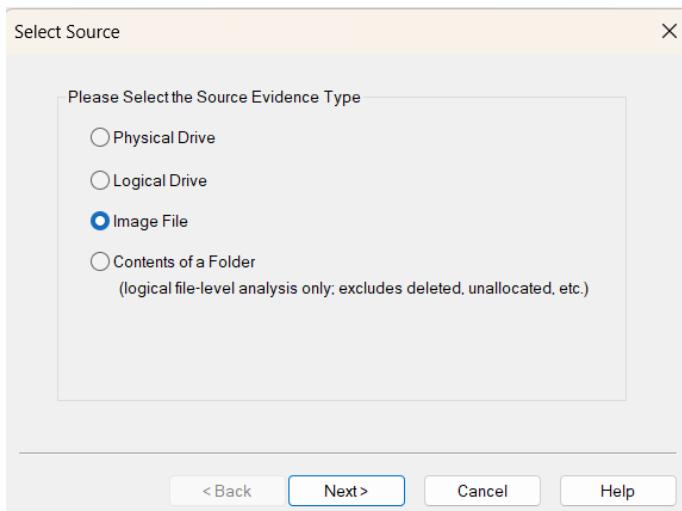
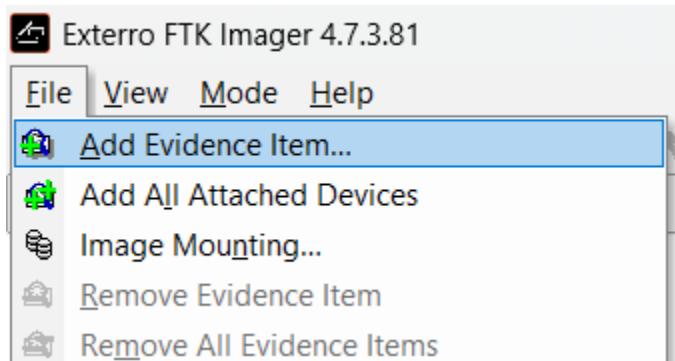
The challenge has given us a disk image file (.dd) to analyze.

```
(kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/Downloads/chal/Disk Forensics]$ file disk.img
disk.img: Linux rev 1.0 ext4 filesystem data, UUID=536d5374-45a0-4d8d-9b9b-517d6ea6c3bc (extents) (64bit) (large files)
(huge files)
```

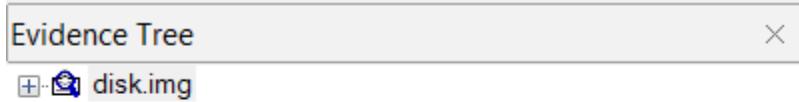
To analyze the disk image, we will utilize a forensics tool called FTK Imager.



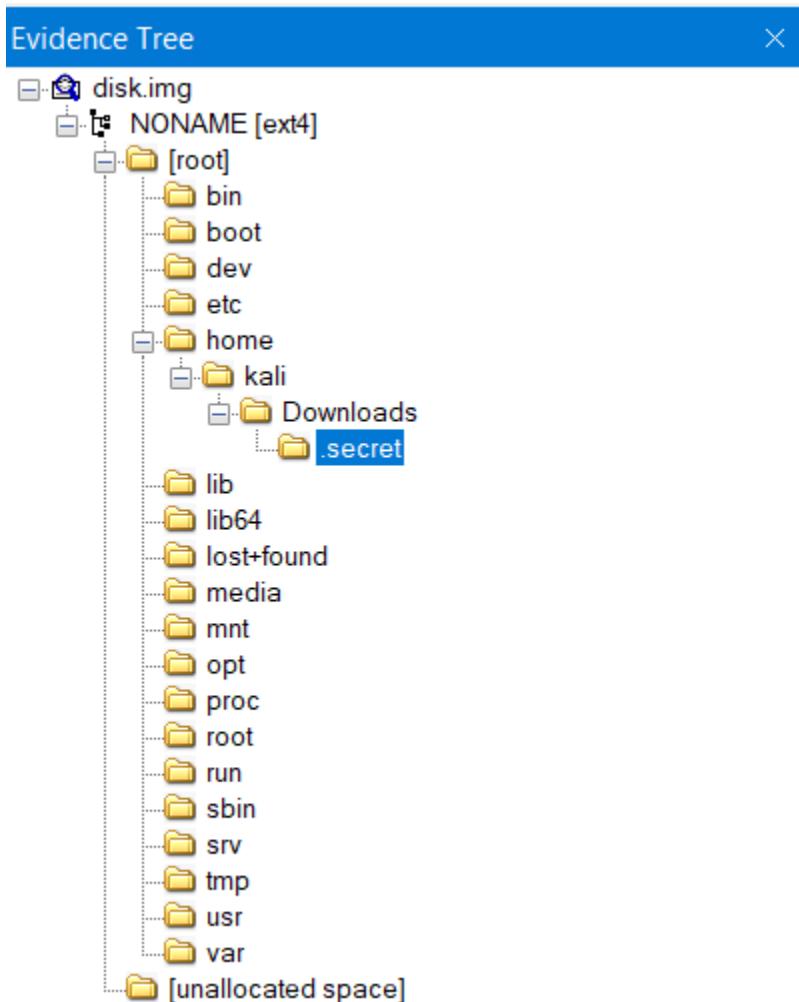
Let's import the disk image as evidence inside the FTK Imager.



After importing the disk image, it will appear in the **Evidence Tree** section.



We can now view the structure inside the disk image to go through all the folders and directories. Usually, the creator of the challenge will hide something inside the home directory. LOOK! There is a secret folder holding a binary file inside 🐸.



## File List

Name	Size	Type	Date Modified
update.bin	46 (1 KB)	Regular File	22/8/2025 3:31:37 ...

```
#!/bin/bash
echo "flag{w45_1t_H4rd_to_L00k?}"
```

Flag- flag{w45\_1t\_H4rd\_to\_L00k?}

## [Forensics] Ghost in the Network 1

Here is the question:

CHALLENGE      16 SOLVES      X

### Ghost in the Network 1

100

The ABC Company IT department recently noticed some unusual activity and possible security problems in their internal network. They captured a snapshot of the network traffic to investigate.

Your task is to carefully examine the provided PCAP file and find the initial access – a hidden email that contains important clues.

Flag Format: flag{...}

 chal.pcap

Flag

Submit

## Description

The ABC Company IT department recently noticed some unusual activity and possible security problems in their internal network. They captured a snapshot of the network traffic to investigate.

Your task is to carefully examine the provided PCAP file and find the initial access — a hidden email that contains important clues.

Flag Format: flag{...}

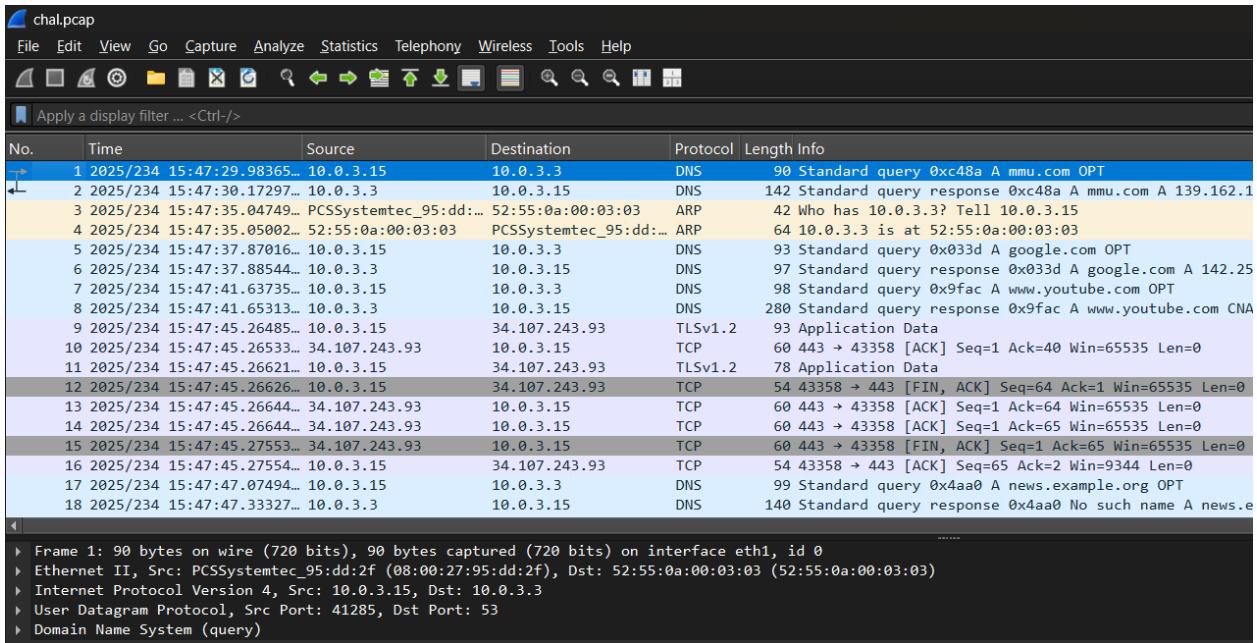
## Walkthrough

Going through the file given, we'll see a pcap file called "**chal.pcap**"

```
└─(kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/Downloads/chal/Network Forensics]
  └─$ ls
  chal.pcap

└─(kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/Downloads/chal/Network Forensics]
  └─$ file chal.pcap
  chal.pcap: pcapng capture file - version 1.0
```

By utilizing **Wireshark**, we can see the network packets captured for a period of time inside the pcap file



chal.pcap

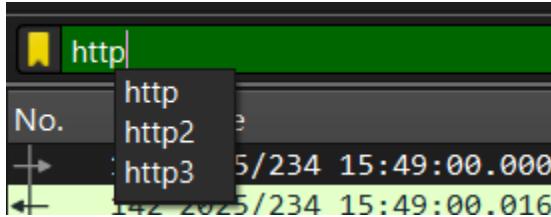
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length Info
1	2025/234 15:47:29.98365...	10.0.3.15	10.0.3.3	DNS	90 Standard query 0xc48a A mmu.com OPT
2	2025/234 15:47:30.17297...	10.0.3.3	10.0.3.15	DNS	142 Standard query response 0xc48a A mmu.com A 139.162.1
3	2025/234 15:47:35.04749...	PCSSystemtec_95:dd:...	52:55:0a:00:03:03	ARP	42 Who has 10.0.3.3? Tell 10.0.3.15
4	2025/234 15:47:35.05002...	52:55:0a:00:03:03	PCSSystemtec_95:dd:...	ARP	64 10.0.3.3 is at 52:55:0a:00:03:03
5	2025/234 15:47:37.87016...	10.0.3.15	10.0.3.3	DNS	93 Standard query 0x033d A google.com OPT
6	2025/234 15:47:37.88544...	10.0.3.3	10.0.3.15	DNS	97 Standard query response 0x033d A google.com A 142.25
7	2025/234 15:47:41.63735...	10.0.3.15	10.0.3.3	DNS	98 Standard query 0x9fac A www.youtube.com OPT
8	2025/234 15:47:41.65431...	10.0.3.3	10.0.3.15	DNS	280 Standard query response 0x9fac A www.youtube.com CNA
9	2025/234 15:47:45.26485...	10.0.3.15	34.107.243.93	TLSv1.2	93 Application Data
10	2025/234 15:47:45.26533...	34.107.243.93	10.0.3.15	TCP	60 443 → 43358 [ACK] Seq=1 Ack=40 Win=65535 Len=0
11	2025/234 15:47:45.26621...	10.0.3.15	34.107.243.93	TLSv1.2	78 Application Data
12	2025/234 15:47:45.26626...	10.0.3.15	34.107.243.93	TCP	54 43358 → 443 [FIN, ACK] Seq=64 Ack=1 Win=65535 Len=0
13	2025/234 15:47:45.26644...	34.107.243.93	10.0.3.15	TCP	60 443 → 43358 [ACK] Seq=1 Ack=64 Win=65535 Len=0
14	2025/234 15:47:45.26644...	34.107.243.93	10.0.3.15	TCP	60 443 → 43358 [ACK] Seq=1 Ack=65 Win=65535 Len=0
15	2025/234 15:47:45.27553...	34.107.243.93	10.0.3.15	TCP	60 443 → 43358 [FIN, ACK] Seq=1 Ack=65 Win=65535 Len=0
16	2025/234 15:47:45.27554...	10.0.3.15	34.107.243.93	TCP	54 43358 → 443 [ACK] Seq=65 Ack=2 Win=9344 Len=0
17	2025/234 15:47:47.07494...	10.0.3.15	10.0.3.3	DNS	99 Standard query 0x4aa0 A news.example.org OPT
18	2025/234 15:47:47.33327...	10.0.3.3	10.0.3.15	DNS	140 Standard query response 0x4aa0 No such name A news.e

Frame 1: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface eth1, id 0  
Ethernet II, Src: PCSSystemtec\_95:dd:2f (08:00:27:95:dd:2f), Dst: 52:55:0a:00:03:03 (52:55:0a:00:03:03)  
Internet Protocol Version 4, Src: 10.0.3.15, Dst: 10.0.3.3  
User Datagram Protocol, Src Port: 41285, Dst Port: 53  
Domain Name System (query)

Not going go through all of it 😅, let's just focus on one protocol that I found interesting which is **http** Protocol.



Owhh, look at these. It seems like during the network capturing, they caught a website.

And...we found a **suspicious email** among the network packets

No.	Time	Source	Destination	Protocol	Length Info
132	2025/234 15:49:00.00056...	10.0.3.15	152.42.198.138	HTTP	418 GET /security_audit_email.html HTTP/1.1
142	2025/234 15:49:00.01691...	152.42.198.138	10.0.3.15	HTTP	2771 HTTP/1.0 200 OK (text/html)
179	2025/234 15:49:34.25472...	10.0.3.15	152.42.198.138	HTTP	351 GET /database.json HTTP/1.1
181	2025/234 15:49:34.27021...	152.42.198.138	10.0.3.15	HTTP/J...	870 HTTP/1.0 200 OK , JSON (application/json)
189	2025/234 15:49:34.30381...	10.0.3.15	152.42.198.138	HTTP	445 GET /dashboard.html HTTP/1.1
199	2025/234 15:49:34.33354...	152.42.198.138	10.0.3.15	HTTP	2629 HTTP/1.0 200 OK (text/html)
211	2025/234 15:49:34.39420...	10.0.3.15	152.42.198.138	HTTP	384 GET /company-style.css HTTP/1.1
212	2025/234 15:49:34.39433...	10.0.3.15	152.42.198.138	HTTP	349 GET /dashboard.js HTTP/1.1
217	2025/234 15:49:34.39841...	10.0.3.15	152.42.198.138	HTTP	382 GET /fontawesome.css HTTP/1.1
229	2025/234 15:49:34.41023...	152.42.198.138	10.0.3.15	HTTP	1475 HTTP/1.0 200 OK (application/javascript)
232	2025/234 15:49:34.41058...	152.42.198.138	10.0.3.15	HTTP	2252 HTTP/1.0 200 OK (text/css)
264	2025/234 15:49:34.44247...	152.42.198.138	10.0.3.15	HTTP	2874 HTTP/1.0 200 OK (text/css)
271	2025/234 15:49:34.48066...	10.0.3.15	152.42.198.138	HTTP	423 GET /webfonts/fa-solid-900.woff2 HTTP/1.1
305	2025/234 15:49:34.54231...	152.42.198.138	10.0.3.15	HTTP	13122 HTTP/1.0 200 OK (font/woff2)
411	2025/234 15:49:59.57095...	10.0.3.15	192.124.249.22	OCSP	482 Request
423	2025/234 15:49:59.57729...	10.0.3.15	192.124.249.22	OCSP	482 Request
429	2025/234 15:49:59.69462...	10.0.3.15	192.124.249.22	OCSP	482 Request
433	2025/234 15:49:59.83867...	10.0.3.15	192.124.249.22	OCSP	482 Request

To access the website, we need to know the ip of the website and the port used to host it.

From clicking the packet reveal **what we need!**

: 1. The destination IP (Website's IP)

2. The destination Port (Website's Port)

```
▶ Internet Protocol Version 4, Src: 10.0.3.15, Dst: 152.42.198.138
▶ Transmission Control Protocol, Src Port: 59782, Dst Port: 8000, Seq: 1, Ack: 1, Len: 364
```

Opening the email file at the website reveals us with a **remainder email** from the **security team** to the **developing team** of the ABC Company to delete a testing user which could compromise the website's security.

**From:** security@abccompany.com  
**To:** dev-team@abccompany.com  
**CC:** it-admin@abccompany.com  
**Subject:** URGENT: Remove Test Credentials from Production System - Security Risk  
**Date:** August 20, 2025, 3:47 PM  
**Priority:** High

Dear Development Team,

I hope this email finds you well. During our routine security audit yesterday, we discovered that several test user accounts from our recent website testing phase are still active in our production system. This poses a significant security risk to our company's confidentiality and must be addressed immediately.

**⚠ CRITICAL ACTION REQUIRED:**

Please remove the following test accounts that are still accessible on our corporate portal:

**⚠ Test Account Details:**

- Username: test
- Password: letmein
- Access Level: Employee Dashboard
- Created: July 15, 2025 (Testing Phase)
- Status: SHOULD BE DELETED

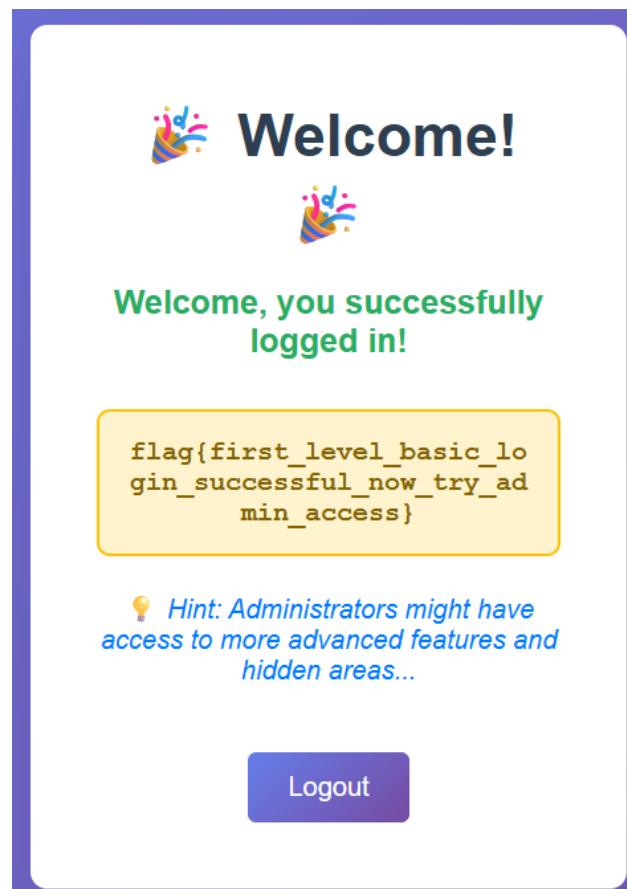
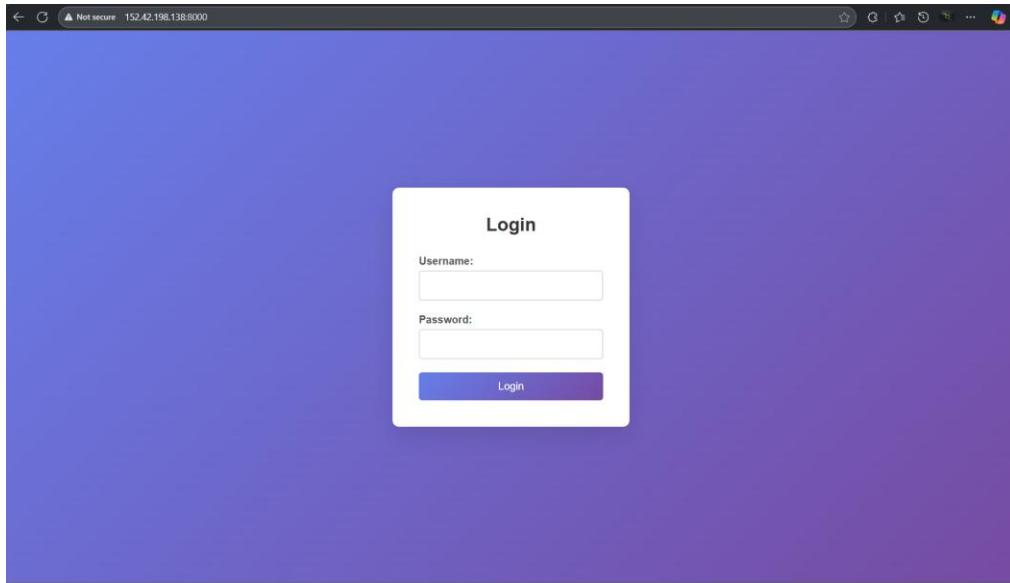
Now we got what we need we can try and **login** into the website using this credential.

**⚠ CRITICAL ACTION REQUIRED:**

Please remove the following test accounts that are still accessible on our corporate portal:

**⚠ Test Account Details:**

- Username: test
- Password: letmein
- Access Level: Employee Dashboard
- Created: July 15, 2025 (Testing Phase)
- Status: SHOULD BE DELETED



Flag - flag{first\_level\_basic\_login\_successful\_now\_try\_admin\_access}

## [Forensics] Ghost in the Network 2

CHALLENGE

16 SOLVES



### Ghost in the Network 2

🟡 100

What else is there in the PCAP file? Look closely for any database or file that could contain the credentials for a higher-level account. Then, start exploring the website to find the second flag.

Flag Format: flag{...}

Hint: Focus on the information a company would most want to keep secure.

Flag

Submit

#### Description

What else is there in the PCAP file? Look closely for any database or file that could contain the credentials for a higher-level account. Then, start exploring the website to find the second flag.

Flag Format: flag{...}

Hint: Focus on the information a company would most want to keep secure.

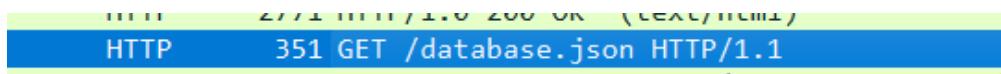
## ⭐ Walkthrough

After retrieving the first flag, we get a hint from the challenge stated “**Administrators might have access to more advanced features and hidden areas**”. This means that we must gain a higher level of access (admin) to get the second flag.

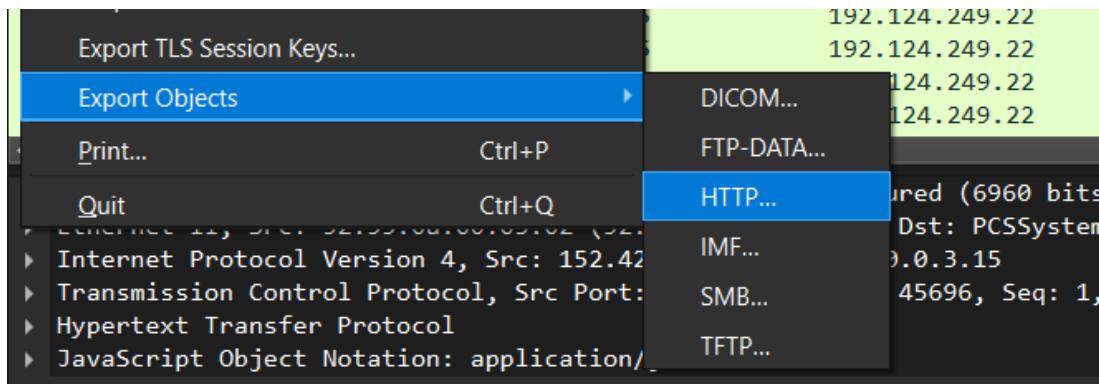
💡 Hint: Administrators might have access to more advanced features and hidden areas...

Logout

Looking back again at the **pcap file** reveals us with a **database.json** file.



We can now use a built-in tool inside **Wireshark** to download the file from the pcap.



Wireshark - Export - HTTP object list

Text Filter: Content Type: All Content-Types

Packet	Hostname	Content Type	Size	Filename
142	152.42.198.138:8000	text/html	9579 bytes	security_audit_email.html
181	152.42.198.138:8000	application/json	472 bytes	database.json
199	152.42.198.138:8000	text/html	15 kB	dashboard.html
229	152.42.198.138:8000	application/javascript	6817 bytes	dashboard.js
232	152.42.198.138:8000	text/css	10 kB	company-style.css
264	152.42.198.138:8000	text/css	89 kB	fontawesome.css
305	152.42.198.138:8000	font/woff2	126 kB	fa-solid-900.woff2
411	ocsp.starfieldtech.com	application/ocsp-request	75 bytes	\
423	ocsp.starfieldtech.com	application/ocsp-request	75 bytes	\
429	ocsp.starfieldtech.com	application/ocsp-request	75 bytes	\
433	ocsp.starfieldtech.com	application/ocsp-request	75 bytes	\
435	ocsp.starfieldtech.com	application/ocsp-response	2149 bytes	\
443	ocsp.starfieldtech.com	application/ocsp-response	2149 bytes	\
461	ocsp.starfieldtech.com	application/ocsp-response	2149 bytes	\
625	o.pki.goog	application/ocsp-request	83 bytes	wr2
632	o.pki.goog	application/ocsp-response	471 bytes	wr2
691	o.pki.goog	application/ocsp-request	83 bytes	wr2
709	o.pki.goog	application/ocsp-response	471 bytes	wr2
711	o.pki.goog	application/ocsp-request	83 bytes	wr2
735	o.pki.goog	application/ocsp-response	471 bytes	wr2
775	o.pki.goog	application/ocsp-request	83 bytes	wr2
778	o.pki.goog	application/ocsp-response	471 bytes	wr2
994	o.pki.goog	application/ocsp-request	83 bytes	wr2
1014	o.pki.goog	application/ocsp-response	471 bytes	wr2
1017	o.pki.goog	application/ocsp-request	83 bytes	wr2
1034	o.pki.goog	application/ocsp-response	471 bytes	wr2

Save Save All Preview Close Help

WOAHHH! It is a database file containing the credentials to access admin account.

```
@ database.json X
C: > Users > jacob > Downloads > chal > Network Forensics >
1 [ {
2   "users": [
3     {
4       "id": 1,
5       "username": "test",
6       "password": "letmein",
7       "role": "employee",
8       "access_level": "basic"
9     },
10    {
11      "id": 2,
12      "username": "admin",
13      "password": "password123",
14      "role": "administrator",
15      "access_level": "dashboard"
16    },
17    {
18      "id": 3,
19      "username": "user",
20      "password": "secret",
21      "role": "guest",
22      "access_level": "basic"
23    }
24  ]
25 }
26 }
```

Log into the website using admin **credentials** we can now access the ABC Company Dashboard.

The screenshot shows the ABC Company Dashboard. At the top, there's a header with a 'Not secure' warning, the URL '152.42.198.138:8000/dashboard.html', and a logo for 'ABC Company'. Below the header is a navigation bar with links for 'Dashboard', 'About', 'Services', 'Statistics', 'Contact', and 'Logout'. The main content area has a title 'Welcome to ABC Company Dashboard' and a message 'Hello admin! Welcome to our corporate portal.' Below this are four cards: 'Revenue' (\$2.4M, +12.5%), 'Clients' (1,247, +8.2%), 'Projects' (89, +15.3%), and 'Uptime' (99.9%, Stable). A 'Recent Activity' section follows, listing three items: 'New project proposal submitted by Marketing Team' (2 hours ago), 'New client onboarded: TechCorp Industries' (5 hours ago), and 'Project Alpha deployment completed successfully' (1 day ago).

Let's go through all the pages inside the website... Based on the challenge again I suspect that the hint given about the thing company want to protect is the **company's data**. We can download the statistics file to view the content.

The screenshot shows the 'Company Statistics' page. The top navigation bar includes 'ABC Company', 'Dashboard', 'About', 'Services', 'Statistics' (which is highlighted in blue), 'Contact', and 'Logout'. Below the navigation is a title 'Company Statistics' and a subtitle 'Comprehensive performance metrics and analytics for ABC Company'. There are two main sections: 'Revenue Growth (Last 5 Years)' (a bar chart showing revenue increasing from 2019 to 2023) and 'Key Performance Indicators' (a section with four horizontal progress bars: Client Retention Rate at 95%, Project Success Rate at 98%, Team Satisfaction at 92%, and Market Share at 78%). A link 'Download Latest Report (CSV)' is located above the KPI section.

Look at the flag!

A1	B	C	D	E	F	G	H	I
1 Company Statistics Report - ABC Company								
2 Generated: August 2025								
3 Report Type: Quarterly Performance Analysis								
4								
5 Quarter	Revenue (L)	Clients	Projects	Success Rate	Market Share (%)			
6 Q1 2024	1800000	1150	72	96.5	74.2			
7 Q2 2024	2100000	1200	78	97.1	75.8			
8 Q3 2024	2300000	1230	84	97.8	76.9			
9 Q4 2024	2200000	1247	89	98.2	78.1			
10								
11 Department Performance Metrics								
12 Department	Budget Util	Goals Ach	Employee	Satisfaction				
13 Engineering	92.5	15/16	4.2/5.0					
14 Marketing	88.3	Dec-14	4.0/5.0					
15 Sales	95.7	18/19	4.4/5.0					
16 HR	76.2	8-Oct	4.1/5.0					
17 Finance	91.1	11-Dec	3.9/5.0					
18								
19 Client Satisfaction Metrics								
20 Category	Score (1-10)	Response Rate (%)						
21 Product Quality	8.7	89.2						
22 Customer Support	9.1	92.5						
23 Technical Support	8.9	87.3						
24 Value for Money	8.4	91.8						
25								
26 flag{congratulations_you_found_the_csv_download_feature_and_analyzed_company_data}								
27								
28 Security Audit Summary								
29 Last Audit Date: July 2025								
30 Critical Issues: 0								
31 High Priority: 2								
< >	ABC_Company_Statistics_Report_2					+		



flag{congratulations\_you\_found\_the\_csv\_download\_feature\_and\_analyzed\_company\_data}

## [Forensics] Ghost in the Network 3

CHALLENGE      13 SOLVES      X

### Ghost in the Network 3

100

Somewhere inside the website, a hidden message is waiting to be found. To uncover it, you will need patience and focus... and perhaps a little creativity.

Flag Format: flag{...}

Hint:

- Look for clues in the website's scripts.
- Sometimes, secret codes or special sequences reveal hidden content.

Submit

### Description

*Somewhere inside the website, a hidden message is waiting to be found. To uncover it, you will need patience and focus... and perhaps a little creativity.*

Flag Format: flag{...}

Hint:

- Look for clues in the website's scripts.
- Sometimes, secret codes or special sequences reveal hidden content.

## ⭐ Walkthrough

To find the Last flag, let's go back to website and now we will view the source code of the website.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>ABC Company - Dashboard</title>
7     <link rel="stylesheet" href="company-style.css">
8     <link rel="stylesheet" href="fontawesome.css">
9 </head>
10 <body>
11     <!-- Navigation -->
12     <nav class="navbar">
13         <div class="nav-container">
14             <div class="nav-logo">
15                 
16                 <span>ABC Company</span>
17             </div>
18             <ul class="nav-menu">
19                 <li><a href="#dashboard" class="nav-link active">Dashboard</a></li>
20                 <li><a href="#about" class="nav-link">About</a></li>
21                 <li><a href="#services" class="nav-link">Services</a></li>
22                 <li><a href="#statistics" class="nav-link">Statistics</a></li>
23                 <li><a href="#contact" class="nav-link">Contact</a></li>
24                 <li><a href="#" class="nav-link" id="logout">Logout</a></li>
25             </ul>
26         </div>
27     </nav>
28
29     <!-- Main Content -->
30     <main class="main-content">
31         <!-- Dashboard Section -->
32         <section id="dashboard" class="section active">
33             <div class="container">
34                 <h1>Welcome to ABC Company Dashboard</h1>
35                 <p class="welcome-text">Hello <span id="username-display"></span>! Welcome to our corporate portal.</p>
36
37                 <div class="dashboard-grid">
38                     <div class="card">
39                         <div class="card-header">
40                             <i class="fas fa-chart-line"></i>
41                             <h3>Revenue</h3>
42                         </div>
43                         <div class="card-body">
44                             <div class="stat-number">$2.4M</div>
45                             <div class="stat-change positive">+12.5%</div>
46                         </div>
47                     </div>
48
49                     <div class="card">
50                         <div class="card-header">
51                             <i class="fas fa-users"></i>
52                             <h3>Clients</h3>
53                         </div>
54                     </div>
55                 </div>
56             </div>
57         </section>
58     </main>
59 
```

Tips from me: Always check the **javascript** files, there might be an interesting things there!

```
303                     </form>
304                 </div>
305             </div>
306         </div>
307     </section>
308 </main>
309
310     <script src="dashboard.js"></script>
311 </body>
312 </html>
313
```

**Volla!** The flag is there! But I'm curious about something “The Konami cheat code?”

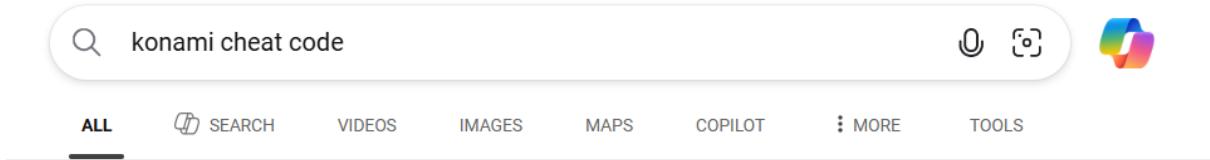
```
// Secret konami code easter egg (optional extra challenge)
let konamiCode = [];
const konami = [38, 38, 40, 40, 37, 39, 37, 39, 66, 65]; // à†‘à†‘à†‘à†“à†®à†‘à†®à†‘BA

document.addEventListener('keydown', function(e) {
    konamiCode.push(e.keyCode);
    if (konamiCode.length > konami.length) {
        konamiCode.shift();
    }

    if (konamiCode.length === konami.length &&
        konamiCode.every((code, index) => code === konami[index])) {

        // Super secret flag for advanced users
        alert(`YOU Konami Code detected! Super secret flag: flag{konami_code_master_found_the_ultimate_secret}`);
        konamiCode = [];
    }
});
```

I did some **googling** and found out that the combination of the Konami cheat code is:



### Copilot Answer

↑↑↓↓←→←→BA

The **Konami Code** is a famous cheat code that has appeared in many Konami video games. The sequence is ↑↑↓↓←→←→BA. It was popularized by the game Contra, where it granted players 30 extra lives. The code has since become a part of popular culture and can be used in various games beyond Konami titles. [Wikipedia](#) +2

For a complete list of games that support the Konami Code, you can refer to sources like DMarket and IGN. [DMarket](#) +1

By clicking the buttons on the keyboard with the Konami cheat code, we triggered the last flag itself!

A screenshot of a web-based dashboard for "ABC Company". The top navigation bar is blue with the company name. A central modal window is open, displaying a message from "152.42.198.138:8000 says": "Konami Code detected! Super secret flag: flag{konami\_code\_master\_found\_the\_ultimate\_secret}" with an "OK" button. Below the modal, the dashboard header reads "Welcome to ABC Company Dashboard". The main content area shows a greeting "Hello admin! Welcome to our corporate portal." and four key performance indicators: Revenue (\$2.4M, +12.5%), Clients (1,247, +8.2%), Projects (89, +15.3%), and Uptime (99.9%, Stable).

Flag - flag{konami\_code\_master\_found\_the\_ultimate\_secret}

## [Forensics] Encrypted mischief

The screenshot shows a challenge card with a dark blue background. At the top left is a button labeled "CHALLENGE". To its right is the text "2 SOLVES" and a close button (an "X"). Below this is the challenge title "Encrypted mischief" in a large, bold, white font. Underneath the title is a circular icon containing a yellow play button symbol, followed by the text "100". A descriptive paragraph follows: "Something mischievous has been hidden in the files you've been given. Your task is to dig through the files, spot hidden patterns, and uncover the password." Below the text is a teal button with a download icon and the file name "chal.zip". To the right of the download button is a dashed rectangular input field containing the word "Flag". To the right of the input field is a pink button labeled "Submit".

### Description

Something mischievous has been hidden in the files you've been given. Your task is to dig through the files, spot hidden patterns, and uncover the password.

### Walkthrough

Given a zip file. Let's unzip it first to view its content.

```
└─(kali㉿NawfalMatebook)-[~/mnt/c/Users/  
└─$ ls  
chal.zip
```

```
└─(kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/  
└─$ unzip chal.zip |
```

We found 2 files inside, “**1\_1.png**” and “**secrets.zip**” file. Let’s try to unzip the secrets.zip file first. **OH NO!!** it need a password to unzip it.

```
└─(kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/Downloads/chal/File Ana  
└─$ file 1_1.png  
1_1.png: PNG image data, 900 x 506, 8-bit/color RGB, non-interlaced
```

```
└─(kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/  
└─$ ls  
1_1.png  secrets.zip  
└─(kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/  
└─$ unzip secrets.zip  
Archive: secrets.zip  
[secrets.zip] secrets.mp4 password: |
```

I'm quite curious with why the challenge give us a png file, so I analyze the metadata using **exiftool**, then I found something interesting "**THE COMMENT**". MMU123 seems like a password to me 😊.

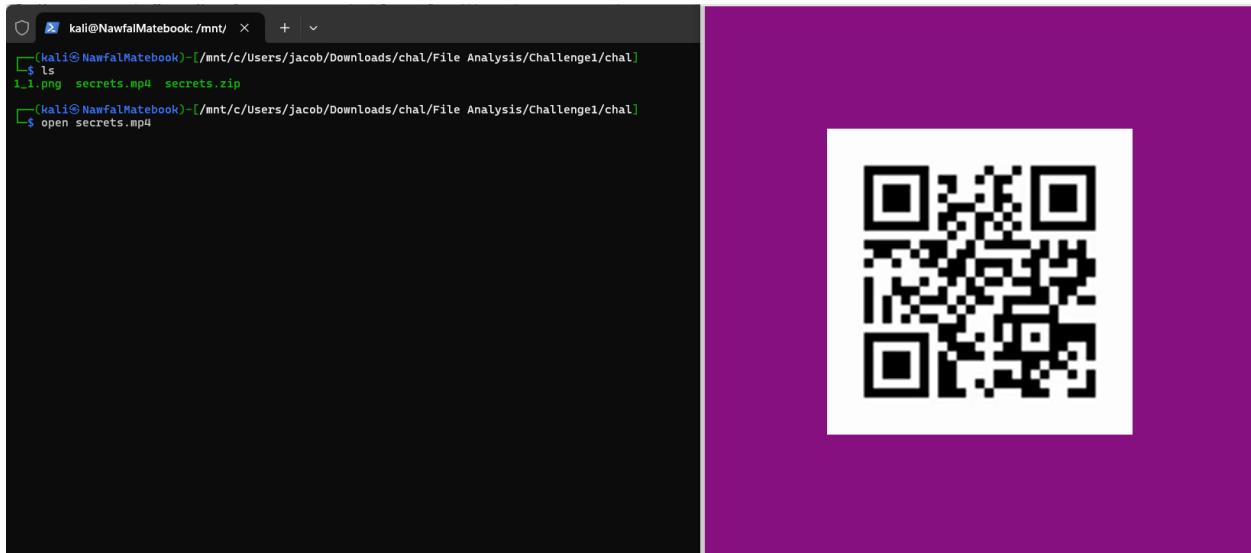
```
(kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/Downloads/chal/File Analysis/Challenge1]
$ exiftool 1_1.png
ExifTool Version Number      : 13.25
File Name                   : 1_1.png
Directory                   :
File Size                    : 467 kB
File Modification Date/Time : 2025:04:16 17:52:08+08:00
File Access Date/Time       : 2025:08:24 17:54:09+08:00
File Inode Change Date/Time: 2025:08:24 17:54:09+08:00
File Permissions            : -rwxrwxrwx
File Type                   : PNG
File Type Extension         : png
MIME Type                   : image/png
Image Width                 : 900
Image Height                : 506
Bit Depth                   : 8
Color Type                  : RGB
Compression                 : Deflate/Inflate
Filter                      : Adaptive
Interlace                   : Noninterlaced
Orientation                 : Horizontal (normal)
X Resolution                : 71.983
Y Resolution                : 71.983
Resolution Unit             : inches
Y Cb Cr Positioning        : Centered
Exif Version                : 0210
Components Configuration    : Y, Cb, Cr, -
Flashpix Version            : 0100
Color Space                 : Uncalibrated
Exif Image Width            : 900
Exif Image Height           : 506
Pixels Per Unit X          : 2834
Pixels Per Unit Y          : 2834
Pixel Units                 : meters
Exif Byte Order              : Little-endian (Intel, II)
Warning                     : IFD0 pointer references previous IFD0 directory
Comment                     : MMU123
Image Size                  : 900x506
Megapixels                  : 0.455
```

Letssss goo!! We can unzip the zip file. The content seems to be a mp4 file, Lets try opening it.

```
[kali㉿NawfalMatebook)-[~/mnt/c/Users/]
$ unzip secrets.zip
Archive: secrets.zip
[secrets.zip] secrets.mp4 password:
  inflating: secrets.mp4

[kali㉿NawfalMatebook)-[~/mnt/c/Users/]
$ 
[kali㉿NawfalMatebook)-[~/mnt/c/Users/]
$ ls
1_1.png  secrets.mp4  secrets.zip
```

A QR code?!



Using online QR Code Scanner, we decode it into a string, “**Base64**”?

The screenshot shows a web-based QR code scanner. At the top, there are navigation links for Home, QR Code Generator, History, Tools, and Language (English). Below the header is the main title "QR Code Scanner". On the left, there's a file upload interface with "Upload" and "Scan QR Code" buttons, and a note to "Upload Another Image". In the center, under "Scanned Items(1)", a QR code is displayed with a purple border. To its right, the text "TXT: ZmxhZ3tibDFua180bmRfbTE1NV8xdH0=" is shown, along with "RAW" and "List 50" buttons. At the bottom left, a note states: "Note: We do not sell or share your data with any third parties. For more details, check out our privacy policy."

Let's decode it in kali using the “**base64 -d**” function, and I was correct here is the flag!

```
(kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/Downloads/challenger]$ echo "ZmxhZ3tibDFua180bmRfbTE1NV8xdH0=" | base64 -d  
flag{bl1nk_4nd_m155_1t}
```

Flag - flag{bl1nk\_4nd\_m155\_1t}

## [Forensics] Suspicious File

CHALLENGE      8 SOLVES     

### Suspicious File

200

The SOC team recovered a suspicious file from a compromised workstation.

It seems harmless when opened, but we suspect it hides something unusual.

Can you find out what it contains?

Flag Format: flag{...}

[chall](#)

Flag

Submit

### Description

The SOC team recovered a suspicious file from a compromised workstation.  
It seems harmless when opened, but we suspect it hides something unusual.  
Can you find out what it contains?

Flag Format: flag{...}

## ⭐ Walkthrough

### Solution

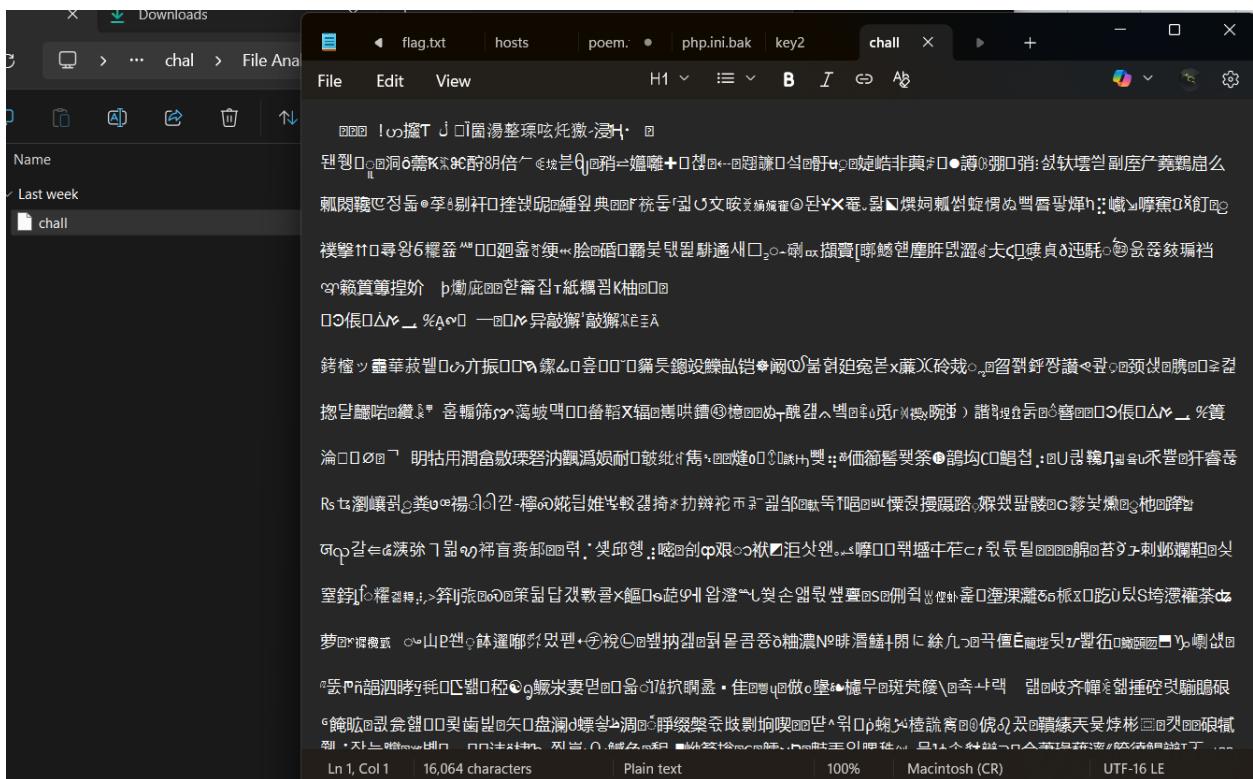
Given a file with no extension, **What?!! No file format?** 🤯

```
(kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/chall]$ ls
```

Let's utilize the file command to view the type of file it is, we now know it is a zip file (probably)

```
(kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/Downloads/chal/File Analysis/Challenge2]$ file chall
chall: Zip archive data, made by v4.5, extract using at least v2.0, last modified Jan 01 1980 00:00:00, uncompressed size 1718, method=deflate
```

Ok now I'm trying to read its content using notepad.... LOST HOPE 😞



Ok now we know it is a zip file, there is a command to view the embedded file inside it, **binwalk command**, odd... There is a lot of word and xml files inside, now I suspect it is actually a **word file**.

```
(kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/Downloads/chal/File Analysis/Challenge2]
$ binwalk chall

DECIMAL      HEXADECIMAL      DESCRIPTION
-----+-----+-----+
997        0x3E5          Zip archive data, at least v2.0 to extract, compressed size: 590, name: _rels/.rels
1797       0x705          Zip archive data, at least v2.0 to extract, compressed size: 5667, name: word/document.xml
3121       0xC31          Zip archive data, at least v2.0 to extract, compressed size: 1214, name: word/_rels/document.xml.rels
3757       0xEAD          Zip archive data, at least v2.0 to extract, compressed size: 12288, name: word/vbaProject.bin
7763       0x1E53         Zip archive data, at least v2.0 to extract, compressed size: 7643, name: word/theme/theme1.xml
9448       0x24E8         Zip archive data, at least v2.0 to extract, compressed size: 277, name: word/_rels/vbaProject.bin.rels
9699       0x25E3         Zip archive data, at least v2.0 to extract, compressed size: 2784, name: word/vbaData.xml
10414      0x28AE         Zip archive data, at least v2.0 to extract, compressed size: 3741, name: word/settings.xml
11707      0x2DBB         Zip archive data, at least v2.0 to extract, compressed size: 260, name: customXml/item1.xml
11966      0x2EBE         Zip archive data, at least v2.0 to extract, compressed size: 341, name: customXml/itemProps1.xml
12285      0x2FFD         Zip archive data, at least v2.0 to extract, compressed size:
```

Using the **xxd command** I need to view the header of the file to view the **magic bytes** (the important thing that declare the format of the file) , the first 8 bytes of the files reveal that the magic bytes is **0000 0000**, now that's why it changes it format before!

```
(env)(kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/Downloads/chal/File
$ xxd chall | head -10
00000000: 0000 0000 1400 0600 0800 0000 2100 1f10  .....!...
00000010: 2865 ac01 0000 b606 0000 1300 0802 5b43  (e.....[c
00000020: 6f6e 7465 6e74 5f54 7970 6573 5d2e 786d  ontent_Types].xm
00000030: 6c20 a204 0228 a000 0200 0000 0000 0000  l ...(. .....
00000040: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000050: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000060: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000080: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000090: 0000 0000 0000 0000 0000 0000 0000 0000  .....
```

By doing some research, we found out that the magic bytes of a word file is **50 4B 03 04**.

A screenshot of a search results page from a search engine. The search query is "magic bytes for docm". The results show approximately 60,100 results. The top result is a link to the Wikipedia page "List of file signatures". The page content describes file signatures as data used to identify or verify file content, mentioning "magic numbers" or "magic bytes". It states that many file formats are not intended to be read as text. A "See more" link is visible at the end of the snippet.

		4D 5A	MZ	0	cpl ocx ax iec ime rs tsp fon efi	descendants (including NE and PE)	1/1	Text	Font	Color	Background	Border	hide
53 4D 53 4E 46 32 30 30		SMSNF200		0	ssp	SmartSniff Packets File [22]		<input type="radio"/> Small	<input checked="" type="radio"/> Standard	<input type="radio"/> Large			
5A 4D		ZM		0	exe	DOS ZM executable and its descendants (rare)		<input checked="" type="radio"/> Standard	<input type="radio"/> Wide				
50 4B 03 04 50 4B 05 06 (empty archive) 50 4B 07 08 (spanned archive)		PKETXEOF PKENACK PKBELBS		0	zip aar apk docx epub ipa jar kmz maff msix odp ods odt pk3 pk4	zip file format and formats based on it, such as EPUB, JAR, ODF, OOXML		<input type="radio"/> Automatic	<input checked="" type="radio"/> Light	<input type="radio"/> Dark			

Now we utilize a tool called **hexedit** to edit the magic bytes of the file

```
[env](kali㉿NawfalMatebook)-[~/mnt/c/Users  
$ hexedit chall
```

Before changes:

```
00000000  00 00 00 00 14 00 06 00 08 00 00 00 21 00 1F 10 28 65 AC 01 .....!...!...  
00000014  00 00 B6 06 00 00 13 00 08 02 5B 43 6F 6E 74 65 6E 74 5F 54 .....[Content_T  
00000028  79 70 65 73 5D 2E 78 6D 6C 20 A2 04 02 28 A0 00 02 00 00 00 ypes].xml ...(.  
0000003C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000064  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000078  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0000008C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000000B4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000000C8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000000DC  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000000F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000104  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000118  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0000012C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000154  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000168  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0000017C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000001A4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000001B8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000001CC  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000001E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000001F4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
--- chall --0x0/0x7D81--0%-----
```

After changes:

```
00000000  50 4B 03 04 |14 00 06 00 08 00 00 00 21 00 1F 10 28 65 AC 01 PK.....!...!..  
00000014  00 00 B6 06 00 00 13 00 08 02 5B 43 6F 6E 74 65 6E 74 5F 54 .....[Content_T  
00000028  79 70 65 73 5D 2E 78 6D 6C 20 A2 04 02 28 A0 00 02 00 00 00 ypes].xml ...(.  
0000003C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000064  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000078  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0000008C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000000B4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000000C8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000000DC  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000000F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000104  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000118  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0000012C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000154  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000168  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0000017C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00000190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000001A4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000001B8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000001CC  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000001E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
000001F4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
** chall --0x4/0x7D81--0%-----
```

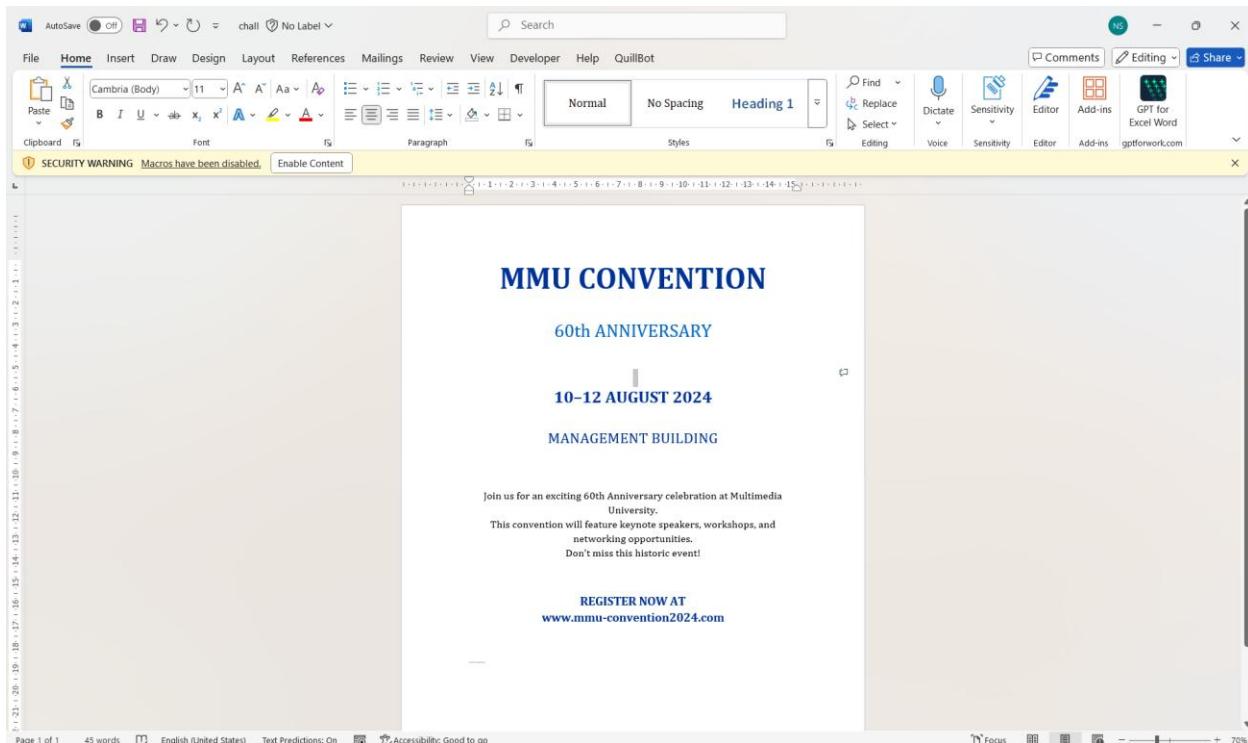
Now it is revealed that the file is actually a **Microsoft Word Files** !

```
[env](kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/  
└ $ file chall  
chall: Microsoft Word 2007+
```

Lets **rename** the file to open the it in Microsoft word.

```
[env](kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/  
└ $ mv chall chall.docm]
```

Opening the file gets us an invitation to a **MMU Convention**, which I don't find anything odd except.....



A macros warning!, someone must have tempered with the macros of the word file



SECURITY WARNING Macros have been disabled.

[Enable Content](#)

How to view the file macros? There is actually a tool for that called olevba in oletools

#### Guide to installation:

- 1) Kali cannot directly install the tool using pip3 because of security issues so we must create a virtual environment using the command below

```
└─(kali㉿NawfalMatebook)-[~/mnt/
└─$ python3 -m venv env

└─(kali㉿NawfalMatebook)-[~/mnt/
└─$ source env/bin/activate
```

- 2) Now we can safely install the oletools inside our kali

```
└─(env)(kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/Downloads/chal/File An
└─$ pip3 install oletools
Collecting oletools
  Using cached oletools-0.60.2-py2.py3-none-any.whl.metadata (16 kB)
Collecting pyparsing<4,>=2.1.0 (from oletools)
  Using cached pyparsing-3.2.3-py3-none-any.whl.metadata (5.0 kB)
Collecting olefile>=0.46 (from oletools)
  Using cached olefile-0.47-py2.py3-none-any.whl.metadata (9.7 kB)
Collecting easygui (from oletools)
  Using cached easygui-0.98.3-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting colorclass (from oletools)
  Using cached colorclass-2.2.2-py2.py3-none-any.whl.metadata (5.2 kB)
Collecting pcodedmp>=1.2.5 (from oletools)
  Using cached pcodedmp-1.2.6-py2.py3-none-any.whl.metadata (11 kB)
Collecting mssofcrypto-tool (from oletools)
  Using cached mssofcrypto_tool-5.4.2-py3-none-any.whl.metadata (10 kB)
Collecting cryptography>=39.0 (from mssofcrypto-tool->oletools)
  Using cached cryptography-45.0.6-cp311abi3-manylinux_2_34_x86_64.whl.m
Collecting cffi>=1.14 (from cryptography>=39.0->mssofcrypto-tool->oletool
  Using cached cffi-1.17.1-cp313-cp313-manylinux_2_17_x86_64.manylinux201
Collecting pycparser (from cffi>=1.14->cryptography>=39.0->mssofcrypto-to
  Using cached pycparser-2.22-py3-none-any.whl.metadata (943 bytes)
Using cached oletools-0.60.2-py2.py3-none-any.whl (989 kB)
Using cached pyparsing-3.2.3-py3-none-any.whl (111 kB)
Using cached olefile-0.47-py2.py3-none-any.whl (114 kB)
Using cached pcodedmp-1.2.6-py2.py3-none-any.whl (30 kB)
```

### 3) Check if the olevba is now useable inside our kali.

```
[env](kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/Downloads/chal/File Analysis]$ olevba
olevba 0.60.2 on Python 2.7.18 - http://decalage.info/python/oletools

olevba.py

olevba is a script to parse OLE and OpenXML files such as MS Office documents
(e.g. Word, Excel), to extract VBA Macro code in clear text, deobfuscate
and analyze malicious macros.
XLM/Excel 4 Macros are also supported in Excel and SLK files.

Supported formats:
- Word 97-2003 (.doc, .dot), Word 2007+ (.docm, .dotm)
- Excel 97-2003 (.xls), Excel 2007+ (.xlsm, .xlsb)
- PowerPoint 97-2003 (.ppt), PowerPoint 2007+ (.pptm, .ppsm)
- Word/PowerPoint 2007+ XML (aka Flat OPC)
- Word 2003 XML (.xml)
- Word/Excel Single File Web Page / MHTML (.mht)
- Publisher (.pub)
- SYLK/SLK files (.slk)
- Text file containing VBA or VBScript source code
- Password-protected Zip archive containing any of the above
- raises an error if run with files encrypted using MS Crypto API RC4

Author: Philippe Lagadec - http://www.decalage.info
License: BSD, see source code or documentation
```

Ok now we can use the **olevba** to view the macros of the word file. **What is that? A suspicious string?!** Look like encryption to me

```
* (env) (kali㉿NawfalMatebook)-[~/mnt/c/Users/jacob/Downloads/chal/File Analysis/Challenge2]
$ olevba chall.docm
olevba 0.60.2 on Python 2.7.18 - http://decalage.info/python/oletools
=====
FILE: chall.docm
Type: OpenXML
WARNING For now, VBA stomping cannot be detected for files in memory
-----
VBA MACRO ThisDocument.cls
in file: word/vbaProject.bin - OLE stream: u'VBA/ThisDocument'
-----
(empty macro)
-----
VBA MACRO Module1.bas
in file: word/vbaProject.bin - OLE stream: u'VBA/Module1'
-----
Sub AutoOpen()
    ' suspicious string
    Dim secret As String
    secret = "++++++[>>++++>++++++<<<-]>>>+++++++.+++++.-----.
<,>+++++++.-----.<--->.+++++++.-----."
End Sub
+-----+
|Type      |Keyword          |Description           |
+-----+-----+-----+
|AutoExec |AutoOpen        |Runs when the Word document is opened |
+-----+-----+-----+
WARNING /home/kali/.local/lib/python2.7/site-packages/msoffcrypto/method/ecma376_agile.py:8: C
m. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography.hazmat.backends import default_backend
```

Using my favorite tool to decode, [Decrypt a Message - Cipher Identifier - Online Code Recognizer](#), we can now analyze the encryption.

The screenshot shows the dCode website interface. At the top, there's a search bar with 'dcode' typed in, and a Copilot Answer section. Below the search bar are navigation links: ALL, SEARCH, IMAGES, VIDEOS, MAPS, COPilot, MORE, and TOOLS. A user icon with '5582' notifications is also present.

**Copilot Answer:**

- dCode** (<https://www.dcode.fr/en>)  
dCode - Online Ciphers, Solvers, Decoders,...
- What are dCode tools?**  
(Definition) dCode calls tool all forms (solver, generator, calculators) and online applications designed to solve specific problems, such as mathematical...
- A must-have product!** DCODE 1755PLUS
- Cryptography and Ciphers**  
dCode, as the name implies, automatically decodes a large variety of encryptions. With its cipher identifier (that recognizes automatically more than 200 ciphers),...
- Calculators and Mathematics**  
dCode make homeworks! Need an equation solver, make boolean calculations, compute prime numbers decomposition or need a cryptarithm solver? On dCode, it's...

At the bottom of the main content area, there are three buttons: 'Explore advanced cipher techniques?', 'What are dCode's unique tools?', and 'How to decode timestamps effectively?'.

**Left sidebar:**

- Search for a tool**: Includes a search input field ('e.g. type "random") and a 'BROWSE THE FULL DCODE TOOLS' LIST' link.
- dCode.fr**: A brief description of the site as a universal site for deciphering coded messages, solving letter games, puzzles, geocaches, and treasure hunts.
- Share**: Links for sharing the site on various platforms.
- dCode and more**: Information about the site being free and useful for various games, math, geocaching, puzzles, and solving every day. It also includes a suggestion for feedback.

**Main content area:**

- DCODE.FR**: A brief description of the site as a collection of over 900 tools for solving games, riddles, ciphers, mathematics, puzzles, etc.
- WORD GAME SOLVERS**: Tools for searching/finding words, solving crosswords, and using anagram generators.
- CRYPTOGRAPHY AND CIPHERS**: Tools for decoding various encryptions using a cipher identifier.
- Summary**: A sidebar listing categories like Word Game Solvers, Cryptography and Ciphers, Codes and Alphabets, Calculators and Mathematics, Computer Science and Algorithms, Puzzle Game Solvers, Miscellaneous Tools and Data, and various how-to guides.
- Similar pages**: Links to About dCode, Cipher Identifier, dCode Mobile App, and DCODE'S TOOLS LIST.
- Support**: Links to Paypal, Patreon, and More.

Oh it is a **brainf\*\*k encryption**, interpreted the encryption reveal us with another encryption.

The image contains two screenshots of the dCode website interface. Both screenshots feature a decorative banner at the top with the text "dCode is preparing a new interface. Come test and give your feedback on the new page: Cipher Identifier!" and "dCode is preparing a new interface. Come test and give your feedback on the new page: Brainfuck!".

**Screenshot 1: Cipher Identifier**

- Left Panel (Search):** A search bar with placeholder "e.g. type 'caesar'" and a "SEARCH A TOOL ON dCODE" button. Below it is a link to "BROWSE THE FULL dCODE TOOLS' LIST".
- Results:** A list titled "dCode's analyzer suggests to investigate:" with several items:
  - Brainfuck (highlighted)
  - Substitution Cipher
  - ReverseFuck
  - Shift Cipher
  - Homophonic Cipher
  - Pig Latin
- Bottom:** "Cipher Identifier - dCode", "Tags(s) : Cryptography, Cryptanalysis, dCode", and a "Share" button.

**Screenshot 2: Brainfuck**

- Left Panel (Search):** A search bar with placeholder "e.g. type 'caesar'" and a "SEARCH A TOOL ON dCODE" button. Below it is a link to "BROWSE THE FULL dCODE TOOLS' LIST".
- Results:** A list titled "dCode's analyzer suggests to investigate:" with one item:
  - Brainfuck
- Input:** A text input field containing "Input: ++++++[>...]. Arg: Output: synt{z4pe0\_4a4ylf1f}"
- Memory Dump:** A table showing memory dump values:
 

Index	Value
[0]	(0)
[1]	(10)
[2]	(30)
[3]	1 (49)
[4]	] (125)

pointer = 4
- Bottom:** "Brainfuck - dCode", "Tags(s) : Programming Language", and a "Share" button.

Using the same method as before we can analyze the encryption again and now we got a **ROT Cipher instead!**

Among the decode is the **FLAG!**

Flag - flag{m4cr0\_4n4lys1s}