

UST GENCYS 2025 CTF WRITEUP

TEAM NAME : GITCYCOOS

**FINAL RANK : 29th

Team Member 1 : KOTTALI ROHAN KARTHIK

Team Member 2 : MOHAMMED SHEDABED

MISC

Challenge name : Welcome

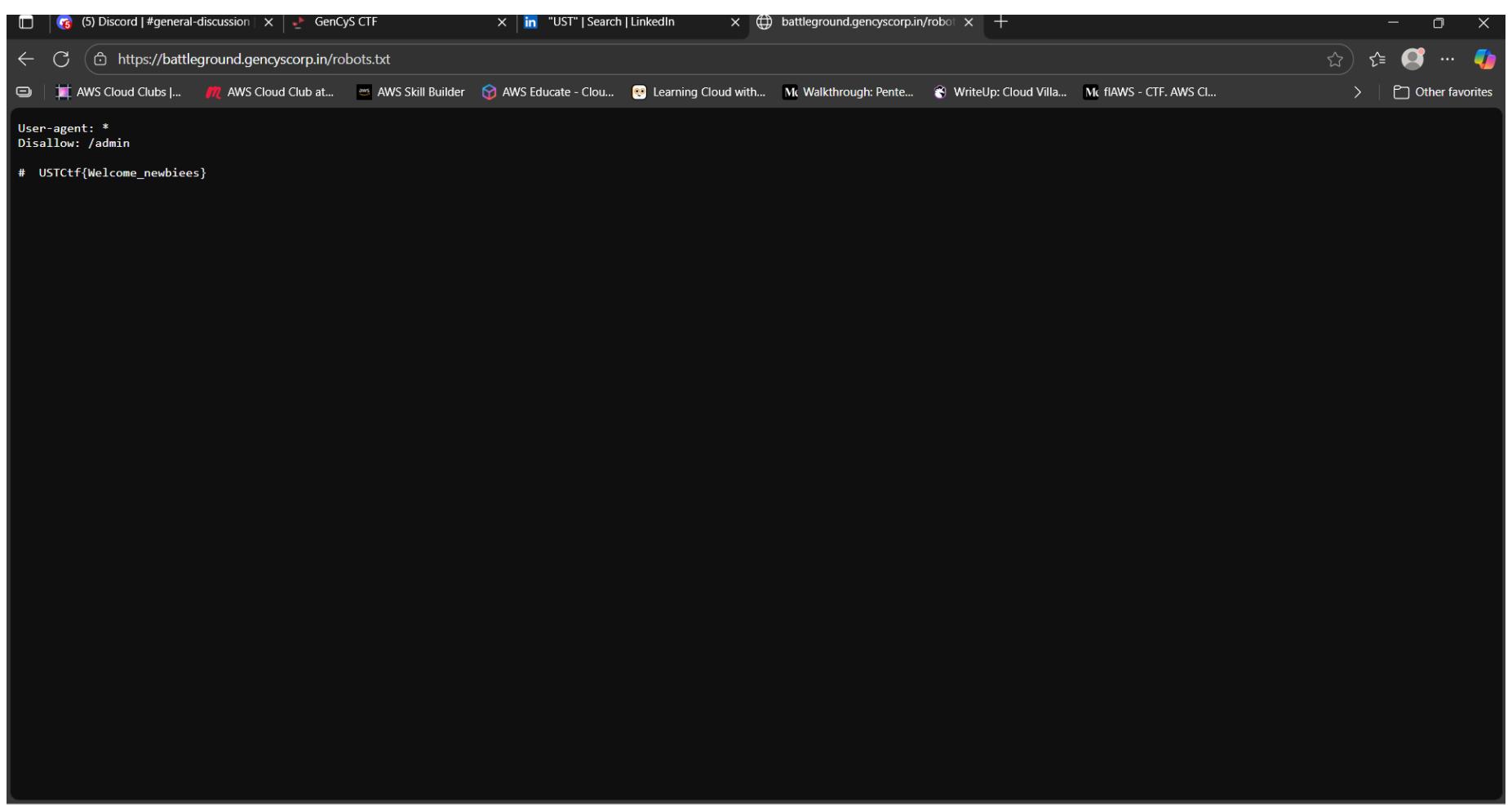
Points - 5

solution: Testing with robots.txt

<https://battleground.gencyscorp.in/robots.txt>

In concept of Web Application Vulnerability Testing, Testing with robots.txt is the Basic Fundamental Concept

Hence when tested with robots.txt were in developers miss to hide the path /robots.txt



The screenshot shows a browser window with three tabs open. The active tab displays the content of the robots.txt file from the URL <https://battleground.gencyscorp.in/robots.txt>. The content of the file is as follows:

```
User-agent: *
Disallow: /admin
#
# USTCtf{Welcome_newbiees}
```

Final Flag : USTCtf{Welcome_newbiees}

OSINT

Challenge Name - Hidden in Plain Sight

Points : 100



Description: Hidden in Plain Sight The mysterious group GenCyS left behind a picture. Hidden within the details is a clue to their last dinner and it is their online identity. Find it — then use that social media, there's something interesting waiting for you. Sometimes, the place you eat becomes the name you wear online. Don't just think of the big social media platforms. There are newer ones rising in the cloud too. they gave me a osint challenge where the image has a letter saying pikbest

solution:

when looking the image in detail we can see a Name highlighting "Pikbest"

A hint given that it is related to last dinner of gencys and to a social media platform but a newer one which is growing in cloud

We tried with other social media platform but could not find a lead , we remember using BlueSky platform in our last ctf

Bluesky is an American **microblogging social media service**

so we checked with keywords

The screenshot shows the Bluesky mobile application interface. The top navigation bar includes links to AWS Cloud Clubs, AWS Cloud Club at..., AWS Skill Builder, AWS Educate - Clou..., Learning Cloud with..., Walkthrough: Pente..., WriteUp: Cloud Villa..., flAWS - CTF, AWS Cl..., and Other favorites. The main screen displays the Discover feed with several posts from different users. The first post is by 'CALLSIGN: SCUTTLEBUTT' (@fack.bsky.social) with the text: 'Okay why is peter thiel doing a lecture series about the antichrist'. The second post is by 'Jesse Walker' (@notjessewalker.bsky.social) with the text: 'It ain't all fancy universities with pretty quads. 43% of undergrads attend community college, and 75% of those are enrolled part-time; 20% of undergrads are parents; 26% of college students take classes exclusively online.' This post includes a photograph of a woman working at a desk while two children play in the background. The third post is by 'Harold Krell' (@haroldkrell.bsky.social) with the text: 'The Typical College Student Is Not Who You Think' and a link to www.nytimes.com. The right side of the screen features a sidebar with sections for 'GETTING STARTED' (Follow 10 accounts, Find people to follow), 'Discover', 'Following', 'More feeds', and 'Trending' (Wes Moore, Supreme Court, Kilmar Abrego Garcia, FEMA, Snoop Dogg, Cal Raleigh). The bottom of the screen shows a weather widget (28°C, Mostly cloudy), a search bar (Search web & PC), and various system icons (ENG IN, 25-08-2025).

Searching with keywords :

gencys

ustglobal

ustctf

The top screenshot shows a search for 'gencys' with the results tab set to 'Top'. The message 'No results found for gencys' is displayed. The bottom screenshot shows a search for 'ustglobal' with the results tab set to 'Top'. A post from 'Techbridge @techbridge.org' is shown, thanking USTGlobal for sponsoring #DreamBigGA #crowdfunding. It includes two images: one of a screen displaying the Dream Big program and another of a stage with speakers.

A screenshot of the Bluesky web interface. The search bar at the top contains 'ustctf'. The results page shows a post from 'cyssssse.bsky.social' (@cyssssse.bsky.social) with the text 'USTCtf{630cf2455c8b02474bf2f245254e2e0b}'. The right sidebar features sections for 'GETTING STARTED', 'Trending', and 'Feedback'.

Bingo! here is the flag for this challenge with the keyword "ustctf"

Final Flag : USTCtf{630cf2455c8b02474bf2f245254e2e0b}

Challenge Name : Where am I ?

Points : 300

Description: Where Am I? osint A data leak by employee has been associated with gencys corporation With a online post , need to check with digital footprints and get the hidden flag

solution:

There was a Hint given with a keyword "gencysstrategy"

When searching this keyword in several social media platform

1.linkedin

2.BlueSky

3.Instagram

With Instagram we got a lead with "gencysstrategy"

The screenshot shows the Instagram search interface. In the search bar at the top left, the query 'gencysst' is entered. Below the search bar, there are five user profile cards: '_lk.b._', 'mdc_gitam', 'suhani_in...', 'srikakulam...', and 'lidiyalv'. To the right of these profiles, a user profile for 'rohan_karthik25' is displayed with the name 'Rohan_karthik'. Below the profiles, a section titled 'Suggested for you' lists several other users with their profile pictures and names: 'ccube.netvix', 'vizagforever', 'meghanavasapalli', '.lrb._', and 'vardhaman.trolls'. At the bottom of the screen, a large image is displayed with the text 'BERLIN WANTS TO HOST 2036 OLYMPICS TO CELEBRATE 100TH ANNIVERSARY' overlaid on a background photo of Berlin at sunset.

The screenshot shows the Instagram profile page for the account 'gencysstrategy'. On the left, a sidebar menu includes options like Home, Search, Explore, Reels, Messages, Notifications, Create, and Profile. The main profile area features a large circular profile picture with the text 'GenCys' in the center. Above the profile picture, the account name 'gencysstrategy' is shown with a 'Follow' button and three dots for more options. Below the account name, it says '3 posts', '3 followers', and '0 following'. A grid of numerous heart-eyes emojis is displayed below this information. The central part of the screen shows a large image of a modern building with 'GenCys' signage. To the right of the image, a green binary code pattern is visible. At the bottom right, there is a message bubble with the text 'ct' and a 'Messages' button.

When seen into the account some suspicious emojis



This screenshot shows an Instagram profile page for the account 'gencysstrategy'. The profile picture is a white circle containing the text 'GenCys'. The bio section is filled with a grid of emoji faces. Below the bio, there is a large, pixelated QR code. The Instagram interface includes a sidebar on the left with links to Home, Search, Explore, Reels, Messages, Notifications, Create, Profile, and More. At the bottom, there is a navigation bar with various icons and a status bar showing the date and time.

This screenshot shows an Instagram post from the account 'echo.spade'. The post features a large, pixelated QR code. The right side of the screen displays the post's details: the account name 'echo.spade', a 'Follow' button, and a message indicating 'No comments yet.' with a 'Start the conversation.' link. Below this, there are like, comment, and share icons, along with a timestamp of 'July 8'. The Instagram interface and navigation bar are visible at the bottom.

In tagged section we can see a QR which can say it is lead
the account was: echo.spade (<https://www.instagram.com/echo.spade/>)

A screenshot of an Instagram profile page for the account 'echo.spade'. The profile picture is a blue circular image with the text 'enCys' on it. The bio says 'CEO @gencysstrategy'. The stats show 4 posts, 0 followers, and 0 following. Below the bio are three large QR codes arranged in a grid. The left sidebar shows navigation options like Home, Search, Explore, Reels, Messages, Notifications, Create, and Profile. The bottom right corner shows a 'Messages' icon.

Yup Hurray we have the qr's and also mentions the other first account of emojis CEO <https://www.instagram.com/gencysstrategy/>

Copng all the qr's into a platform like canva to form a single qr to scan for further extraction

A screenshot of a Canva project titled 'Gold and White Luxury Background Instagram Post'. The project features a white background with gold decorative scrollwork borders at the top and bottom. In the center is a large black and white QR code. The left sidebar of the Canva interface shows various tools and sections like Design, Elements, Text, Brand, Uploads, Tools, Projects, and Apps. The bottom of the screen shows a toolbar with icons for notes and timer, and a status bar indicating 58% zoom, 1/1 page, and the date 26-08-2025.

Upon scanning the qr we can got a link [Morse code with emojis: Encode and decode online - cryptii](https://morsecode-with-emojis.com/)

we can see it is a Morse code with emojis encoder and decoder

we remembered of getting a emojis in the insta account of gencysstratagy account :

[Instagram](#)



trying with these can result us a youtube link : https://youtu.be/vbz2qb_-lny

AWS Cloud Club Presents - Instag... | Gold and White Luxury Backgrou... | Morse code with emojis: Encode... | Instagram

https://cryptii.com/pipes/morse-code-with-emojis

AWS Cloud Clubs | AWS Cloud Club at... | AWS Skill Builder | AWS Educate - Clou... | Learning Cloud with... | Walkthrough: Pente... | WriteUp: Cloud Villa... | flAWS - CTF. AWS Cl... | Other favorites

cryptii Visibility matters

Students and Teachers, save up to 60% on Adobe Creative Cloud.

VIEW Plaintext ▾

ENCODE DECODE

Morse code ▾

VARIANT English

REPRESENTATION Code

SHORT LONG SPACE

Decoded 28 chars

VIEW Cipher emojis ▾

的学生和教师，节省高达60%的Adobe Creative Cloud。

https://youtu.be/vbz2qb_-lny

ENG IN 10:11 26-08-2025

Morse code with emojis: Encode and decode online

See what happens when exchanging the morse code dot, dash and space characters by emojis. Add your own emojis to the mix and translate messages back and forth.

SUNPHARMA -2.48%

Search web & PC

YouTube (116) YouTube

AWS Cloud Clubs | AWS Cloud Club at... | AWS Skill Builder | AWS Educate - Clou... | Learning Cloud with... | Walkthrough: Pente... | WriteUp: Cloud Villa... | flAWS - CTF. AWS Cl... | Other favorites

YouTube

Search

This video isn't available anymore

GO TO HOME

SUNPHARMA -2.48%

Search web & PC

ENG IN 10:12 26-08-2025

Later a Hint was given to a youtube channel : [Petflix vs Magpie](#)

The screenshot shows a YouTube video player for a channel named "Petflix vs Magpie". The video thumbnail features a large white "CTF" logo on a blue whale with a red flag. The video has 271 views and was uploaded 2 months ago. The caption includes "Z=5000". The YouTube interface shows other video suggestions like "4k Relaxing Coding Screensaver" and "Shorts" content.

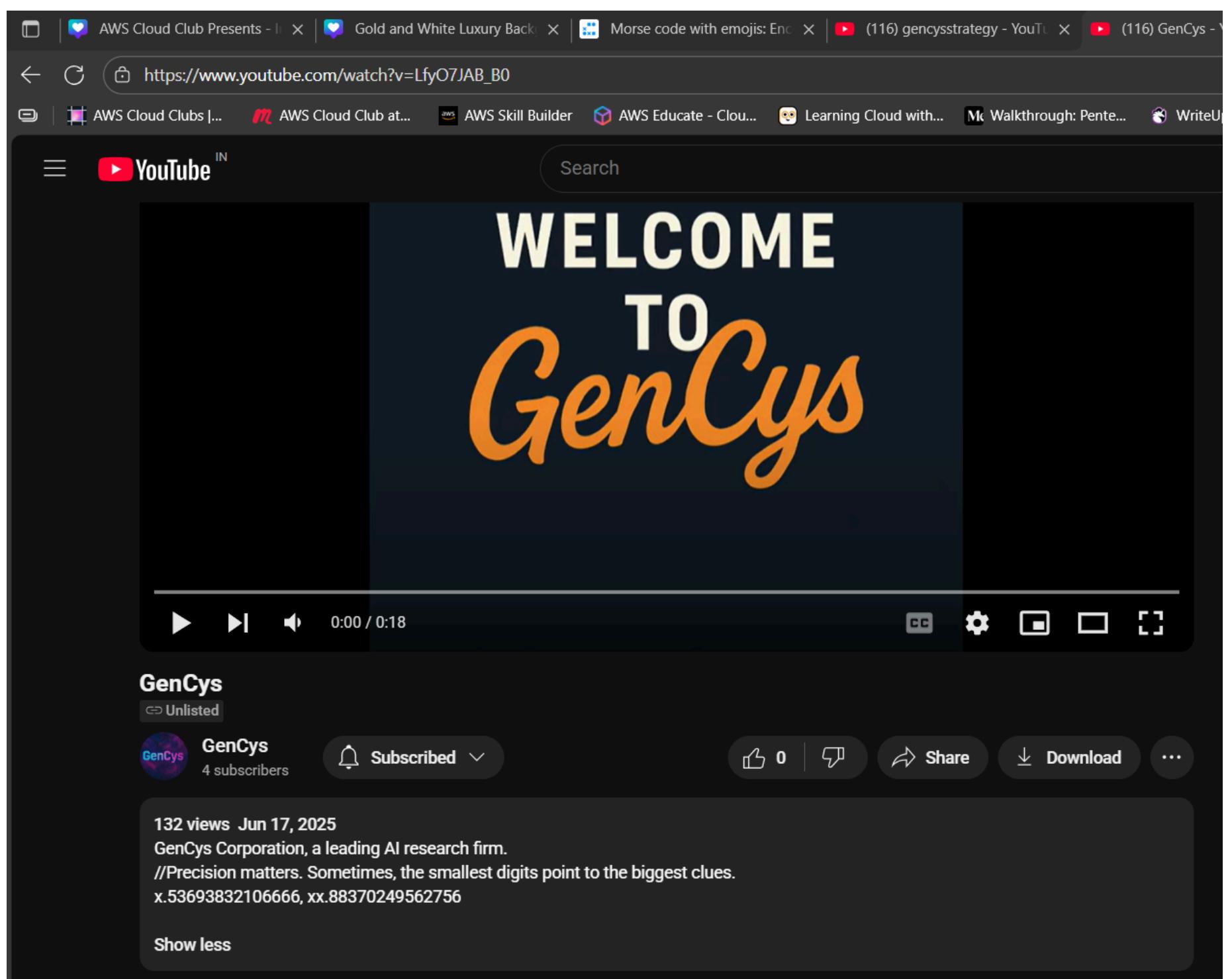
We have seen "Z=5000" in Caption

Upon searching in that channel petflix

The screenshot shows the YouTube channel page for "Petflix". The channel has 2 videos. The first video is "Petflix vs Magpie" and the second is "Petflix vs Whale". The channel page also shows a sidebar for "Subscriptions" and a footer with various links.



In this second video we can see a code in captions : LfyO7JAB_B0
when copying and trying that to check if it a youtube link value we are redirected to below link
[\(116\) GenCys - YouTube](https://www.youtube.com/watch?v=LfyO7JAB_B0)



we can see a small info in the description

GenCys Corporation, a leading AI research firm. //Precision matters. Sometimes, the smallest digits point to the biggest clues.
x.53693832106666, xx.88370249562756

these "x.53693832106666, xx.88370249562756" look like coordinates

The screenshot shows a YouTube channel page for 'GenCys'. The channel has 4 subscribers. A 'Subscribed' button is visible. The channel art features a purple and blue abstract design with the word 'GenCys' in white. The background of the page is dark. On the left, there's a sidebar with links to Home, Shorts, Subscriptions, History, Playlists, Your videos, Your courses, Watch later, and Liked videos. Below that is a 'Subscriptions' section listing 'TIMES NOW', 'ANI News', and 'Hindustan Times'. The URL 'gencyssosintwebpage.web.app' is visible at the top right of the page.

when checking with that channel we got a web link "gencyssosintwebpage.web.app"

The screenshot shows a web browser displaying the website 'gencyssosintwebpage.web.app'. The main content area has a dark grey background with a central white box containing the text 'GenCys Corporation' in large, bold, white font. Below it, in smaller white font, is the tagline 'Pioneering the next frontier in Artificial Intelligence.' and a descriptive paragraph: 'At GenCys, we build AI systems that seamlessly integrate with real-world operations — from healthcare to defense — unlocking intelligence beyond automation.'

```
Line wrap □
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>GenCys Corporation</title>
7   <style>
8     body {
9       margin: 0;
10      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
11      background: url('Flag.png') no-repeat center center fixed;
12      background-size: cover;
13      color: white;
14      text-align: center;
15      height: 100vh;
16      display: flex;
17      flex-direction: column;
18      justify-content: center;
19      backdrop-filter: brightness(0.7);
20    }
21
22   .content {
23     background-color: rgba(0, 0, 0, 0.6);
24     padding: 40px;
25     border-radius: 20px;
26     display: inline-block;
27     max-width: 700px;
28     margin: auto;
29   }
30
31   h1 {
32     font-size: 3em;
33     margin-bottom: 0.5em;
34   }
35
36   p {
37     font-size: 1.2em;
38     line-height: 1.6;
39   }
40 </style>
41 </head>
42 <body>
43   <div class="content">
44     <h1>GenCys Corporation</h1>
```

After checing the page source code

we can see a lead stating that

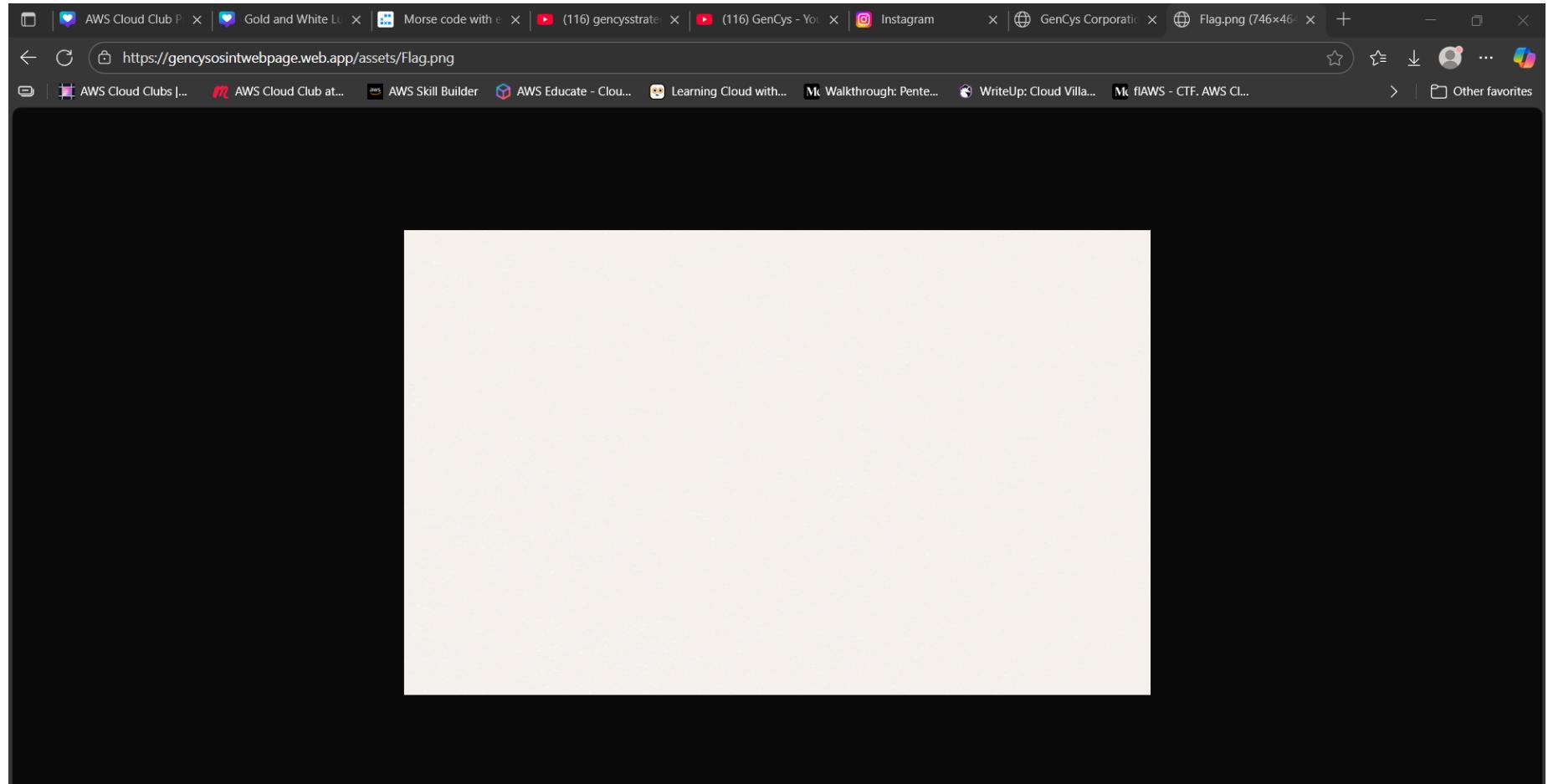
background: url('Flag.png') no-repeat center center fixed;

and

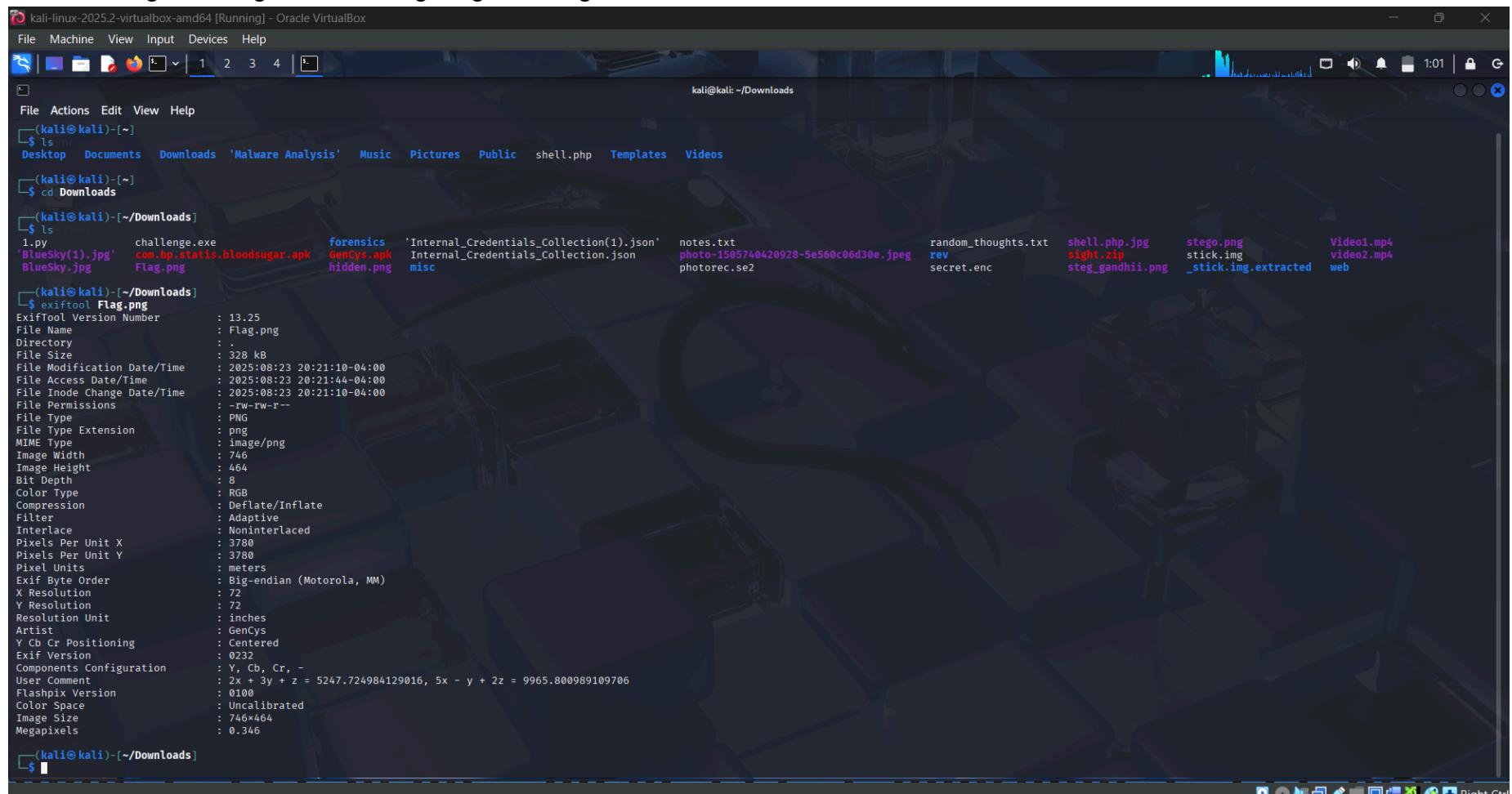
""

```
31   h1 {
32     font-size: 3em;
33     margin-bottom: 0.5em;
34   }
35
36   p {
37     font-size: 1.2em;
38     line-height: 1.6;
39   }
40 </style>
41 </head>
42 <body>
43   <div class="content">
44     <h1>GenCys Corporation</h1>
45     <p>
46       Pioneering the next frontier in Artificial Intelligence.<br />
47       At GenCys, we build AI systems that seamlessly integrate with real-world operations – from healthcare to defense – unlocking intelligence beyond automation.
48     </p>
49     <!--Not all flags wave in plain sight. Some rest quietly in /assets/, waiting for curious minds. Can you see what your eyes can't but your browser can? -->
50   </div>
51 </body>
52 </html>
```

checking in the url :



Downloading the image and investigating the image with exiftool



```
(kali㉿kali)-[~]
$ ls
Desktop Documents Downloads 'Malware Analysis' Music Pictures Public shell.php Templates Videos
(kali㉿kali)-[~]
$ cd Downloads
(kali㉿kali)-[~/Downloads]
$ ls
1.py challenge.exe forensics 'Internal_Credentials_Collection(1).json' notes.txt random_thoughts.txt shell.php.jpg stego.png Video1.mp4
'Bluesky(1).jpg' com.bp.statsis.bloodsugar.apk GenCys.apk Internal_Credentials_Collection.json photo-1505740420928-5e560c06d30e.jpeg rev secret.enc sight.zip stick.img Video2.mp4
BlueSky.jpg Flag.png hidden.png misc photorec.se2 steg_gandhi.png _stick.img.extracted web

(kali㉿kali)-[~/Downloads]
$ exiftool Flag.png
ExifTool Version Number : 13.25
File Name   : Flag.png
Directory  :
File Size    : 328 kB
File Modification Date/Time : 2025:08:23 20:21:10-04:00
File Access Date/Time  : 2025:08:23 20:21:44-04:00
File Inode Change Date/Time : 2025:08:23 20:21:10-04:00
File Permissions : -rw-rw-r--
File Type    : PNG
File Type Extension: png
MIME Type   : image/png
Image Width  : 746
Image Height : 464
Bit Depth   : 8
Color Type  : RGB
Compression : Deflate/Inflate
Filter      : Adaptive
Interlace   : Noninterlaced
Pixels Per Unit X: 3780
Pixels Per Unit Y: 3780
Pixel Units  : meters
Exif Byte Order: Big-endian (Motorola, MM)
X Resolution : 72
Y Resolution : 72
Resolution Unit: inches
Artist      : GenCys
YCbCr Positioning: Centered
Exif Version : 0232
Components Configuration: Y, Cb, Cr, -
User Comment : 2x + 3y + z = 5247.724984129016, 5x - y + 2z = 9965.800989109706
Flashpix Version: 0100
Color Space  : Uncalibrated
Image Size   : 746x464
Megapixels   : 0.346

(kali㉿kali)-[~/Downloads]
$
```

$$2x + 3y + z = 5247.724984129016, 5x - y + 2z = 9965.800989109706$$

```
YCbCr Positioning : Centered
Exif Version : 0232
Components Configuration: Y, Cb, Cr, -
User Comment : 2x + 3y + z = 5247.724984129016, 5x - y + 2z = 9965.800989109706
Flashpix Version: 0100
Color Space  : Uncalibrated
Image Size   : 746x464
```

we remember a value prev Z = 5000 https://www.youtube.com/watch?v=Vbz2qB_-LnY

$$2x + 3y + 5000 = 5247.72498412901$$

$$5x - y + 2(5000) = 9965.800989109706$$

Solving the equation

8.53693832106666

76.88370249562756

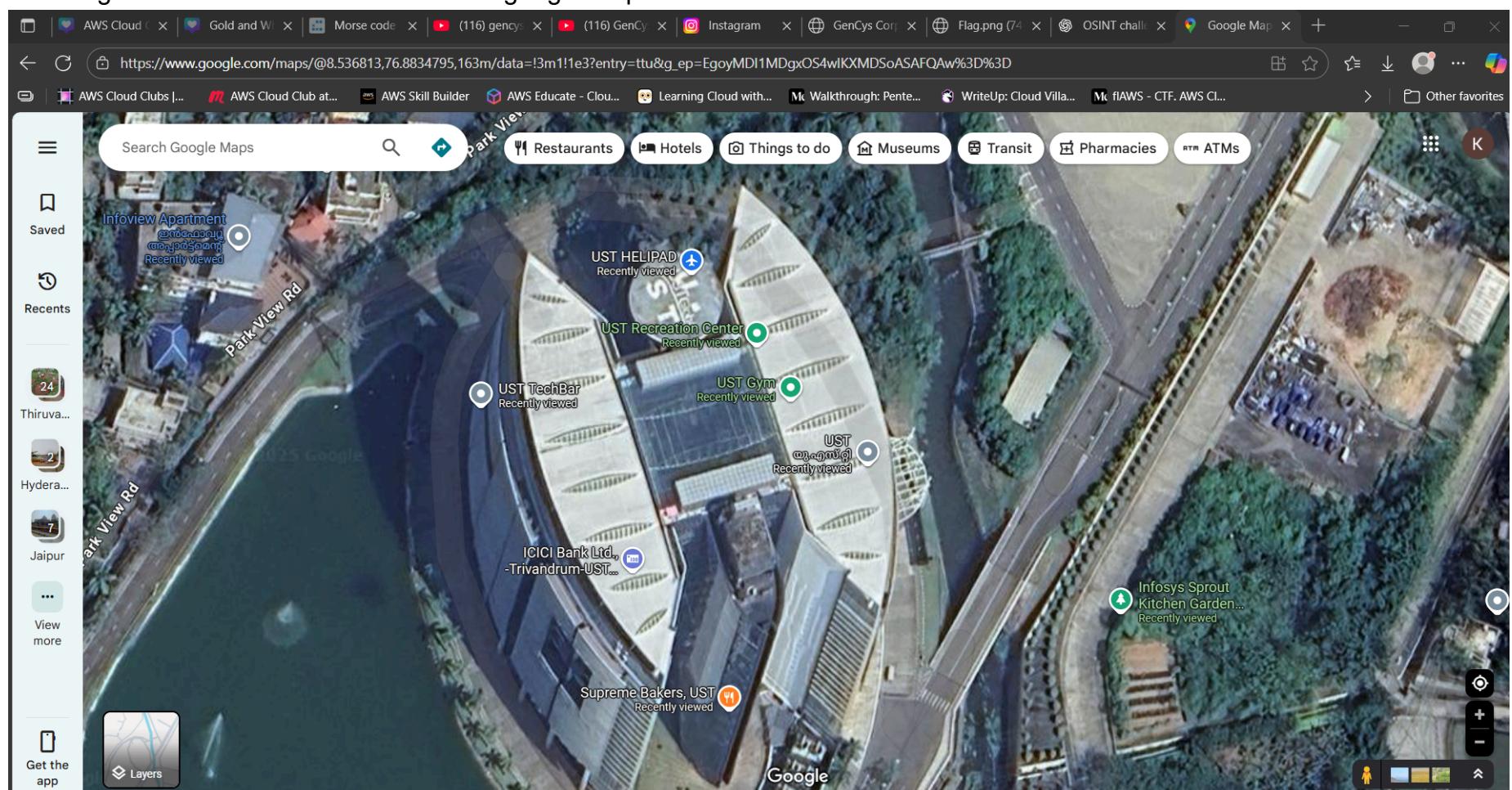
this is the coordinates we got

according to these x.53693832106666, xx.88370249562756

x.53693832106666 is 8.53693832106666

xx.88370249562756 is 76.88370249562756

checking with this directs to UST Global in google maps



II : 9th Floor, UST Campus, TechnoPark Phase II, Electronics Technology Parks SEZ, Kulathoor, Thiruvananthapuram, Kerala 695583, India

4.6 ★★★★★ 3,524 reviews ⓘ



Mohd Ashraf Hisham

5 reviews · 28 photos

⋮

★★★★★ Edited 3 days ago **NEW**

Really loved visiting this place! Peaceful and quiet. Also stumbled across something curious here — felt almost like a mini CTF puzzle: Z2I0aHViLmNvbS9Ob3RTb0lubm9jZW50



upon decoding the this CTF Puzzle Z2I0aHViLmNvbS9Ob3RTb0lubm9jZW50

Decode from Base64 format

Simply enter your data then push the decode button.

Z2I0aHViLmNvbS9Ob3RTb0lubm9jZW50

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

Source character set: **UTF-8**

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

DECODE Decodes your data into the area below.

github.com/NotSoInnocent

we got the github account which is /NotSoInnocent

The screenshot shows the GitHub profile page for the user 'NotSoInnocent'. The profile picture is a large red cross on a white background. The 'Popular repositories' section shows two repositories: 'arc-site' (Public, HTML) and 'AccessDenied' (Public, Python). Below this is a chart titled '11 contributions in the last year' showing activity from September 2024 to August 2025. The 'Contribution activity' section for August 2025 shows no activity. A 'Follow' button is present.

The screenshot shows the GitHub repository page for 'AccessDenied'. The repository is public and contains 7 commits. The commit history includes:

- NotSoInnocent Create deploy.yml (f2ec380, 2 months ago)
- .github/workflows Create deploy.yml (2 months ago)
- ReverseString Create reverse.py (2 months ago)
- oddOrEven Create 1.py (2 months ago)
- saveandreadHiddenMsg.py Create saveandreadHiddenMsg.py (2 months ago)

The repository has 0 stars, 1 fork, and 0 watching. It also has 0 releases published, 0 packages published, and 1 language listed (Python at 100%).

A potential repo [NotSoInnocent/AccessDenied](#)

The screenshot shows the GitHub code editor for the file 'project-build.yml' in the 'AccessDenied' repository. The code is as follows:

```
1 # gencys_project.yaml
2
3 project:
4   name: GenCys AI
5   description: AI-driven solutions for secure intelligence and automation
6   version: 1.0.0
7   active: true
8
9 team:
10  - name: Ahamed
11    role: Lead Developer
12    email: xyz@gencys.ai
13  - name: Priya
14    role: ML Engineer
15    email: priya@gencys.ai
16
17 technologies:
18  backend: Python
19  frontend: React
20  database: PostgreSQL
21  cloud: AWS
22
23 features:
24  - Real-time AI threat detection
25  - Secure data pipelines
26  - Dashboard analytics
27
28 deployment:
29  environment: production
30  region: ap-south
```

[AccessDenied/.github/workflows/project-build.yml at main · NotSoInnocent/AccessDenied](#)

The screenshot shows a GitHub repository page for 'NotSoInnocent/AccessDenied'. The specific file displayed is 'deploy.yml'. The code content is as follows:

```
1 name: Build
2
3 on:
4   workflow_dispatch: # Manually trigger from GitHub UI
5
6 jobs:
7   run-private-script:
8     runs-on: ubuntu-latest
9
10 steps:
11   - name: Checkout public repo
12     uses: actions/checkout@v4
13
14   - name: Clone private repo
15     env:
16       TOKEN: ${{ secrets.WHATS_YOUR_PAT }}
17     run: |
18       git clone https://x-access-token:${TOKEN}@github.com/GenCysCompany/dump.git dump
19
20   - name: Set up Python
21     uses: actions/setup-python@v5
22     with:
23       python-version: '3.10'
24
25   - name: Run Python file
26     run: |
27       python dump/temp.py
```

[AccessDenied/.github/workflows/deploy.yml at main · NotSoInnocent/AccessDenied](https://github.com/NotSoInnocent/AccessDenied/blob/main/.github/workflows/deploy.yml)

we can see in the deploy.yml there are some steps given one among them was :

name: Checkout public repo

uses: actions/checkout@v4

So we checked the actions

[Workflow runs · NotSoInnocent/AccessDenied](https://github.com/NotSoInnocent/AccessDenied/actions)

The screenshot shows the GitHub Actions page for the 'AccessDenied' repository. The left sidebar is collapsed, and the main area displays the 'Actions' tab. A sidebar on the left lists various metrics: All workflows, Build, Management, Caches, Attestations, Usage metrics, and Performance metrics. The main area shows a summary of workflow runs:

All workflows
Showing runs from all workflows

Help us improve GitHub Actions
Tell us how to make GitHub Actions work better for you with three quick questions.

1 workflow run

Event	Status	Branch	Actor
main	4 days ago	8s	...

Build
Build #3: Manually run by NotSoInnocent

The screenshot shows a GitHub Actions build summary for a workflow named 'run-private-script'. The build was triggered manually 4 days ago and completed successfully in 8 seconds. The status bar indicates 'Status: Success' and 'Total duration: 8s'. The artifacts section is empty. On the left, there are links for 'Jobs', 'Run details', 'Usage', and 'Workflow file'. The main panel displays the workflow file 'deploy.yml' with a single step 'run-private-script' that took 6 seconds to run.

The screenshot shows the detailed logs for the 'run-private-script' job. The job succeeded 4 days ago in 6 seconds. The logs are organized into sections: 'Set up job', 'Checkout public repo', and 'Run Python file'. The 'Set up job' section shows the runner version and provisioning details. The 'Checkout public repo' section shows the cloning of a private repository using a GitHub Action. The 'Run Python file' section shows the execution of a Python script named 'dump.py' which resulted in the flag 'USTCtf{All_The_wa\$_a_LiE}'.

Checking the actions further we got the final flag of this challenge

This screenshot provides a more detailed view of the 'run-private-script' job logs. It includes sections for 'Checkout public repo', 'Clone private repo', 'Set up Python', and 'Run Python file'. The 'Checkout public repo' section shows the use of a GitHub Action to sync a repository. The 'Clone private repo' section shows the cloning of a private repository using a GitHub Action. The 'Set up Python' section shows the setup of a Python environment. The 'Run Python file' section shows the execution of a Python script named 'dump.py' which resulted in the flag 'USTCtf{All_The_wa\$_a_LiE}'.

```
Run Python file  
1 ► Run python dump/temp.py  
11 USTCtf{All_Th15_wa$____a_LiE}
```

Final Flag : "USTCtf{All_Th15_wa\$____a_LiE}"

Physical OSINT

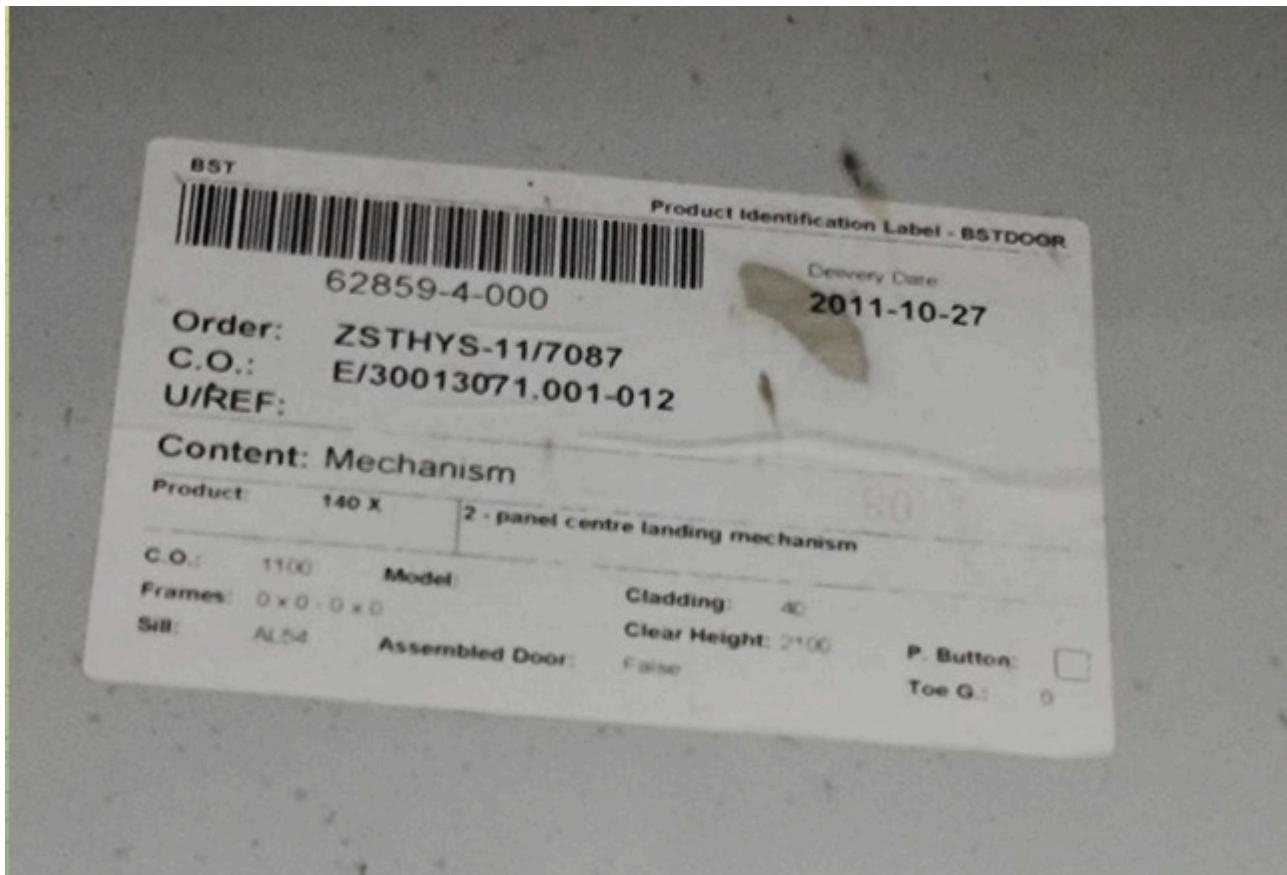
Points : 100

Challenge Name : Operation C.O

Challenge Description: GenCyS Corp is known for hiding things where few bother to look. Certain codes are attached to everyday structures inside their facilities, disguised as ordinary identifiers. Most employees pass by them daily without ever realizing their true purpose. These markings don't align with any public records, yet insiders say they can be "verified" if you know the right place to check. You've managed to note down one such code. Can you discover what it unlocks?

- Approach & Steps:

1. The challenge provided an Order ID (ZSTHYS-11/7087) via an assets page or link, hinting at a verification process requiring a corresponding CO Number.
2. Recognizing this as a physical OSINT task, I explored the event facility for hidden codes or stickers that might match the given Order ID.
3. While heading to the washroom, I noticed a sticker near the lift (elevator), affixed behind the glass—a location easily overlooked by passersby.



4. Inspected the sticker closely and found it contained a CO Number that directly corresponded to the provided Order ID.
5. Returned to the challenge's verification page (likely an online asset portal).

Order Verification

OrderID:	<input type="text" value="ZSTHYS-11/7087"/>
CO Number:	<input type="text"/>
Verify	

© 2025 GenCyS Corporation | Confidential Asset Management System

GenCyS Corporation - Asset Management Portal

Order Verification

OrderID:

ZSTHYS-11/7087

CO Number:**Verify**

Correct! Here is your flag: USTCtf{55fb208100607c4946889d63108312a4f}

Here we got the F14g

6. Entered the discovered CO Number into the input field and submitted the form via the "Verify" button.
7. The system confirmed the match, displaying a success message and revealing the flag.

Final Flag : USTCtf{55fb208100607c4946889d63108312a4f}

**Stegnography

Challenge Name : Silent Shades

Points :

Description: An ordinary PNG image was found on an employee's workstation. It appeared harmless, but accompanying notes indicated the use of "non-visual storage techniques" to embed hidden data. The goal was to extract the concealed secret from the file.

- Approach & Steps:

1. Downloaded the image from: <https://gencysctf.blob.core.windows.net/gencys/stego.png> and saved it locally as ustctf.png.
2. Performed initial analysis using standard Kali Linux tools:
 binwalk to scan for embedded files.
 steghide to check for hidden data requiring a password.
 exiftool to examine metadata.
 strings to extract printable strings. No immediate leads were found.
3. Suspecting a simple embedding method, I viewed the raw file contents using the cat command:
 text

```
cat ustctf.png
```

```
(kali㉿kali)-[~/Desktop]
$ cat stego.png
♦PNG
IHDR♦♦-dPLTE5#8P*Hk1Sx1U♦i♦Ei♦.AW♦♦*B[ k♦0CKu♦#Ry♦[ =a{=c♦a♦b♦-v!-9l♦n♦Ee
vC♦E♦,♦bSTCtf{9861c2da2b27ae4cfb33454e04fd5fb0}♦IDATx♦♦F♦♦(fRLUI
♦nv!c♦mG♦10♦.♦3K♦L♦/♦N♦L>♦♦]♦/♦90♦#♦N♦-♦w7♦;
y♦~♦=♦,♦v♦{e♦,♦q♦_BV♦Dnbl♦wv" C/♦&0♦l♦c♦b.♦%zrA4♦X♦e♦4♦
♦\Xe♦D♦Z~♦+♦76;♦y♦6♦9♦
`♦{♦0♦e♦!♦~♦E♦4♦.8♦}1F♦L♦\L♦
(
```

4. Scrolling through the output revealed a readable string containing the flag, likely appended or hidden in non-image data sections. (Note from hints: "Stegongraphy just save it and Cat" confirms this low-tech approach.)

- Flag Obtained: bSTCtf{9861c2da2b27ae4cfb33454e04fd5fb0}
- But as for the format:-USTCtf{}
- So I inputted the in brackets string in it: USTCtf{9861c2da2b27ae4cfb33454e04fd5fb0}

Final Flag : **USTCtf{9861c2da2b27ae4cfb33454e04fd5fb0}**

Web

Challenge Name : Inference Override – 1

Points : 100

Challenge Description: The challenge involved an e-commerce-like site with unfinished authentication flows. Hints pointed to tier-based access (e.g., "Gold members get special treatment"), suggesting privilege escalation vulnerabilities.

- Approach & Steps:

1. Opened the challenge link, which redirected to: <https://inference.gencyscorp.in/login.php>.
2. Checked robots.txt for disallowed paths, revealing entries like
 - /admin-backup/
 - /dev-notes/
 - /checkout-old/
 - /beta-test/
 - /internal.php
 - /flag.txt # Nope, not here ;) (fake flag)
 - /gold-access-panel/
 - /support.php
3. Tested directories: At /internal.php, discovered a JSON file Internal_Credentials_Collection.json.
4. Now gave that json file to chatgpt it gave me an endpoint
5. API endpoint: <https://inference.gencyscorp.in/api/creds.php>, which returned credentials:

```
Pretty-print □
{
  "username": "johndoe",
  "password": "Summer2025!"
}
```

6. Logged in with these credentials, gaining access as a default-tier user. Inspected source code and cookies but found no escalation paths.
7. Used Burp Suite to intercept requests. Visiting /deals.php showed:
text
GET /deals.php HTTP/2
Response: "Welcome johndoe (Tier: default)".
As we got to know from robot.txt that only gold had access so I tried with tier=gold as Chatgpt gave that suggestion.
8. Attempted parameter tampering: Added ?tier=gold to the GET request, but it failed.

9. Switched to POST method via Burp Repeater:

text

POST /deals.php?tier=gold HTTP/2

```
1 POST /deals.php?tier=gold HTTP/2
2 Host: inference.gencyscorp.in
```

The server accepted this, treating the request as from a Gold-tier user and revealing the flag.

10. Analysis: here search for USTCtf in find menu

```
Show Discount
</button>

<p id="employee-discount" style="display: none; text-align: center; padding-top: 10px;">
    <div class="modal fade" id="discountModal" tabindex="-1" role="dialog" aria-labelledby="discountLabel" aria-hidden="true">
        <div class="modal-dialog" role="document">
            <div class="modal-content">
                <div class="modal-body text-center">
                    <div class="icon text-danger mb-3">
                        <i class="fas fa-gift">
                    </i>
                </div>
                <div class="notice">
                    <h4>
                        Get 50% Discount
                    </h4>
                    <p>
                        For the next 24 hours you can get any product at half-price.
                    </p>
                    <p>
                        Use promo code <code>
                            USTCtf{ecommerce_hpp_G01D_pwned}
                        </code>
                    </p>
                </div>
            </div>
        <div class="modal-footer d-flex justify-content-between">
```

Final Flag : USTCtf{ecommerce_hpp_G01D_pwned}

Cryptography

Challenge name : Silent Sentinel

Points : 100

we got the secret.enc file in the challenge

then i ran this script :

```
lyric.py
1 # Fix the ciphertext slice to exclude the newline before the '#'
2 data = open("secret.enc", "rb").read()
3 hash_pos = data.find(b"#")
4 ct = data[:hash_pos]
5 # Drop any trailing whitespace/newlines from ciphertext
6 ct = ct.rstrip(b"\r\n\t ")
7 print("Ciphertext length after trim:", len(ct))
8
9 from Crypto.Cipher import AES
10
11 def pkcs7_unpad(data, block_size=16):
12     if not data or len(data) % block_size != 0:
13         return None
14     pad_len = data[-1]
15     if pad_len < 1 or pad_len > block_size:
16         return None
17     return data[:-pad_len]
18
19 found = None
20 info = None
21
22 for i in range(10000):
23     pin = f"{i:04d}"
24     dskey = (pin + "0"*12).encode()
25     # ECB
26     if len(ct) % 16 == 0:
27         cipher = AES.new(key, AES.MODE_ECB)
28         pt = cipher.decrypt(ct)
29         unp = pkcs7_unpad(pt, 16) or pt
30         s = unp.decode("utf-8", errors="ignore")
31         if "{" in s and "}" in s and any(tag in s.lower() for tag in ["flag", "ctf"]):
32             found = unp
33             info = ("ECB", pin)
34             break
35     # CBC with zero IV
36     cipher = AES.new(key, AES.MODE_CBC, iv=b"\x00"*16)
37     if len(ct) % 16 == 0:
38         pt = cipher.decrypt(ct)
39         unp = pkcs7_unpad(pt, 16) or pt
40         s = unp.decode("utf-8", errors="ignore")
41         if "{" in s and "}" in s and any(tag in s.lower() for tag in ["flag", "ctf"]):
42             found = unp
43             info = ("CBC-iv0", pin)
44             break
45
~/workspace$ python lyric.py
Ciphertext length after trim: 48
== FLAG FOUND ==
Mode: CBC-iv0 PIN: 7352
```

```
~/workspace$ python lyric.py
Ciphertext length after trim: 48
--- FLAG FOUND ---
Mode: CBC-iv0 PIN: 7352

+-----+
USTCtf{01eb61687e16324487eca30736cf4d6d}
+-----+
File "/home/lyric/lyric.py", line 54, in <module>
    with open(out_path, "w", encoding="utf-8") as f:
FileNotFoundError: [Errno 2] No such file or directory: '/mnt/data/decrypted_flag.txt'
~/workspace$
```

Final flag : USTCtf{01eb61687e16324487eca30736cf4d6d}

silent sentinel

Search for a tool

★ [SEARCH A TOOL ON DCODE](#)

e.g. type 'sudoku'

★ [BROWSE THE FULL DCODE TOOLS' LIST](#)

Results

=abc12xyz89

USTctf{De_flag}

XOR Cipher - [dCode](#)

Tag(s) : Modern Cryptography

Show

XOR CIPHER

Cryptography > Modern Cryptography > XOR Cipher

XOR DECODER

★ TEXT TO BE XORED (MULTIPLIED BY XOR)

Encoding/Format: Decimal [0-127] (Automatic Detection)

52 49 55 114 70 30 2 62 93 102 7 14 2 86 79

ENCRYPTION/DECRYPTION METHOD

- AUTOMATIC (BRUTEFORCE 1 TO 16 BYTES)
- USE THE BINARY KEY X
- USE THE HEXADECIMAL KEY X
- USE THE ASCII KEY X

Final Flag : USTCtf{De_flag}

Forensics

Challenge Name : Out of sight

Points : 100

Firstly there is a zip file given which contains 2 videos one is a 30 min another is 2 min.

So the 30 minute video contains 13 group of different frequencies at different time lines

. So we need to trim the complete video and only leave the 13 groups of freq.

Now we have to convert it to the mp3 file . Now uploading the mp2 file in a DTMF online converter we get this

107#71#66#73#116#73#70#69#116#120#72#66#102#79#86#88#81#104#51#68#77#84#97#73#57#89#86#74#115#112#102#1
09#110#99#110#106#111#48#121#97#43#90#69#108#71#122#100#117#80#111#76#122#72#99#121#111#90#115#102#51#99
#66#57#80

So this is an encrypted text on decrypting this we get a string which is

kGBItlFEtxHBfOVXQh3DMTal9YVJspfmncnjo0ya+ZEIGzduPoLzHcyoZsf3cB9P

Looking into the video carefully it asks us to decrypt using AES256

so we have a long string and we need a 32 bit length key to decrypt

So looking into the video file we need to binwalk both the files we get a yaffs1.bin

so we need to use this command to extract the data dd if=Video1.mp4 of=yaffs1.bin bs=1 skip=1994075

so doing all these further we get a key which looks like this 23872947523978598732495873289321

so it is a 32 bit length one so using it as a key for the AES256 we got half flag USTCtf{445eb8b34

after running the script

```
import re
from Crypto.Cipher import AES
import base64
```

--Read the yaffs1.bin

```
with open("yaffs1.bin", "rb") as f:
    data = f.read()
```

-- Extract ASCII strings that might be keys

```
candidates = re.findall(rb"[A-Za-z0-9+=]{8,}", data)
```

```
print(f"Found {len(candidates)} candidate strings")
```

-- Try each candidate as AES key (16, 24, 32 bytes after padding/trimming)

```
for cand in candidates:
```

```
try:
```

```
    key = cand.decode()
```

```
except:
```

```
continue
```

```
for size in [16, 24, 32]:
    k = key.encode()[:size].ljust(size, b"\0") # normalize key size
    try:
        cipher = AES.new(k, AES.MODE_ECB)
        -- Try decrypting first 32 bytes of data
        decrypted = cipher.decrypt(data[:32])
        if decrypted.isascii():
            print(f"[+] Possible key: {key} (len {len(key)}) → {decrypted}")
    except Exception as e:
        pass
```

Final flag: USTCtf{445eb8b348dbaef887011882b56df69e}

Mobile

Challenge Name : The Untold

Points : 50

We are given an APK file (GenCys.apk) For mobile forensics analysis i used a tool called "apktool" after this i extracted the Main activity.java file over here you can clearly see the links from which the app is calling the data from i started visiting each website i got a morse code audio , i used a morse decoder online, and was able to extract the text SUO1VB0 on reversing the text turns out out to be 0BV1OUS since this is not the actuall flag i tested the images which was given i got this QR code from image 2.png This is what i received upon scanning the QR code it traversed to a website, instantly i viewed the source page of the site where i was able to find a base64 code, which wa eventually decoded.

```
(kali㉿kali)-[~/Desktop]
$ apktool d GenCys.apk -o app_src
```

* *

MainActivity.java

morsecode.world/international/decoder/audio-decoder-adaptive.html

The screenshot shows a web browser window for the 'MorseCode.World' website. The URL in the address bar is 'morsecode.world/international/decoder/audio-decoder-adaptive.html'. The page has a dark background with a red header bar. The header contains the site name 'MorseCode.World' and navigation links for 'Shop', 'International', 'American', 'Labs', 'More', and 'SCPPhillips.com'. Below the header is a large title 'Morse Code Adaptive Audio Decoder'. A secondary navigation bar below the title includes 'International Morse Decoders', 'Audio Decoder', 'Audio Decoder (Expert)', and 'Gaze Decoder'. A central callout box features the text 'T-Shirts, Mugs, Clocks and Cards' and 'Visit the Morse Code World Shop for a unique gift'. The main content area is titled 'Morse Decoder'. It explains that the tool is for listening to, analysing, and decoding International Morse code. It also suggests using a Morse code translator for sounds. A dropdown menu for 'Alphabet to decode into' is set to 'Latin', with a note that other alphabets can be sent in Morse using standard timing. There are two sections for audio input: 'Use the microphone:' with 'Listen' and 'Stop' buttons, and 'Or analyse an audio file containing Morse code:' with 'Upload' and 'Play' buttons. The uploaded file is named '3.mp3'. At the bottom, the decoded text 'SU01VB0' is displayed.

MorseCode.World

Shop International American Labs More SCPPhillips.com

Morse Code Adaptive Audio Decoder

International Morse Decoders

Audio Decoder Audio Decoder (Expert) Gaze Decoder

T-Shirts, Mugs, Clocks and Cards

Visit the Morse Code World Shop for a unique gift

Morse Decoder

This is an experimental tool for listening to, analysing and decoding [International Morse code](#). No information from the microphone is transmitted to the server, but the connection to the server is encrypted nonetheless.

If you cannot produce your own Morse code sounds then try using my [Morse code translator](#) to play or download some.

Alphabet to decode into

Latin

All these alphabets can be sent in Morse using standard timing. The "Latin" alphabet is e.g. "ABC".

Use the microphone:

Or analyse an audio file containing Morse code:

Listen Stop

Upload Play Stop

Filename: "3.mp3"

SU01VB0



AI Mode All Exact matches Products Visual matches About this image Fe

<https://gencysmusicplayer.web.app/>

```

padding: 40px 60px;
border-radius: 16px;
box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1);
text-align: center;
max-width: 400px;
}
h2 {
margin-bottom: 16px;
font-size: 24px;
color: #222;
}
p {
font-size: 18px;
color: #444;
}
code {
background-color: #f4f4f4;
padding: 2px 6px;
border-radius: 4px;
font-family: monospace;
color: #333;
}

```

</style>

head>

ody>

<div class="container">

<h2>Welcome to the Challenge</h2>

<!-- VVNUQ3Rme24wdF9zMF8qKioqKioqXzNuYzBkMW5nfQ== -->

<p>Try searching <code>/flag.txt</code></p>

</div>

body>
VVNUQ3Rme24wdF9zMF8qKioqKioqXzNuYzBkMW5nfQ==

For encoded binaries (like images, documents, etc.) use the file upload form a

Source character set.

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports on

DECODE Decodes your data into the area below.

USTCtf{n0t_s0_*****_3nc0d1ng}

From the mobile SF we submitted the flag

**Final Flag : USTCtf{n0t_s0_0BV1OUS_3nc0d1ng}