

NETSA SKILL CHALLENGE CTF

24 MAY 2025



The poster features a dark background with a central image of a person in a hoodie with a glowing blue skull mask. At the top, the UTeM 25 logo is visible. The main title 'NETSA SKILL CHALLENGE CTF' is in large, bold, white and pink letters. Below the title, the date '24 MAY 2025 (SATURDAY)' and the location 'FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY, UNIVERSITI TEKNIKAL MALAYSIA MELAKA.' are listed. A red banner on the right indicates 'RM150/TEAM (3 person)'. A central box titled 'PRIZE POOL' lists the prizes: '1ST PRIZE : RM 600', '2ND PRIZE : RM 400', and '3RD PRIZE : RM 200'. On the left, a QR code is labeled 'LIMITED TO 20 SLOTS! REGISTER HERE'. At the bottom left, contact information is provided: 'For more information: 01151930927 (WAFI)'. The bottom of the poster includes logos for FTAK, UTeM, and other sponsors, along with a '4 QUALITY EDUCATION' logo.

UTeM 25

NETSA SKILL CHALLENGE CTF

24 MAY 2025 (SATURDAY)
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY,
UNIVERSITI TEKNIKAL MALAYSIA MELAKA.

RM150/TEAM
(3 person)

PRIZE POOL

- 1ST PRIZE : RM 600
- 2ND PRIZE : RM 400
- 3RD PRIZE : RM 200

LIMITED TO 20 SLOTS!
REGISTER HERE

For more information:
01151930927 (WAFI)

Sponsored by:

FTAK UTeM UTM

4 QUALITY EDUCATION

Writeup by: Triple A Battery

Team members:

1. Ammar Saifuddin
2. Aida Sakinah
3. Eusoff Aminurrashid

Contents

Steganography.....	3
Welcome to CTF.....	3
Rasengram.....	5
Cryptography.....	6
Baby RSA.....	6
Forensics	7
Phantom Protocol	7
Phantom Protocol II	8
Phantom Protocol III	10
IOT.....	11
4 Bytes to Freedom.....	11
Firmware Reverse.....	12
Trace.....	13
Blink Logic	14
EEPROM Secret	15
OSINT.....	16
McD.....	16
Train to Hospital.....	17
Final Debug.....	18

Steganography

Welcome to CTF


Welcome to CTF


50

We Intercept images ,its looks ordinary but intel says something is hidden deep inside.

The message is rumored to be encoded in the red channel, but not just anywhere. Look closely at the bits.

Can you extract the hidden flag?

 [netsa.png](#)



```
# Function to extract and decode message from a specific bit position in the red
channel
```

```
def extract_message_from_bit(bit_position):
```

```
bits = []
for pixel in img.getdata():
    red = pixel[0]
    bit = (red >> bit_position) & 1
    bits.append(str(bit))
```

```
# Convert bits to characters
```

```
message = "  
for i in range(0, len(bits), 8):  
    byte = bits[i:i+8]  
    if len(byte) < 8:  
        break  
    char = chr(int("".join(byte), 2))  
    message += char
```

return message

```
# Try bits 1, 2, and 3 of red channel
```

```
messages = {bit: extract_message_from_bit(bit)[:500] for bit in range(1, 4)}
messages
```

Result

 $\{1:$ [illegible]

[illegible]

The message hidden in the second least significant bit (bit 1) of the red channel is:

Flag: netsa{Fl@g!sHidd3n}

Rasengram

Rasengram
175

Hear it carefully. Does it have something sounds suspicious in it? Maybe the question title may give a hint. Flag format is UTeM{flag}

[rasengram.wav](#)

(epic Naruto flute song with morse code)

The image shows two screenshots. The left screenshot is from 'International Morse Decoders' and displays a decoded audio waveform with the text 'UTeM{n4ru70'. The right screenshot is from a CTF challenge titled 'Rasengram' with a value of 500. It shows the same hint and audio file link as the challenge card. Below the challenge card, a larger screenshot shows the full decoded audio waveform with the text 'UTeM{n4ru70_P14Y_flu73e'.

Song is around 1 minutes 30 seconds. Listen or look at the audio wave to determine where each morse code starts and end. In this audio, there is 2 parts.

Put the audio file in morse decoder and it will show in the chart.

Flag: **UTeM{n4ru70_P14Y_flu7e}**

Cryptography

Baby RSA

Baby RSA

175

escription: We've implemented a secure communication system using RSA encryption, but something seems off about the size of our encryption key. Can you help us recover the original message from the ciphertext?

You are given the following RSA parameters:

n(the modulus): 1057169

e (the public exponent): 65537

c (the ciphertext): 586132

Your task is to:

Compute the private key d

Flag Format netsa{Decimal}

```
1 # RSA Decryption with small n, no sympy
2
3 def factor(n):
4     for i in range(2, int(n ** 0.5) + 1):
5         if n % i == 0:
6             return i, n // i
7     return None, None
8
9 def inverse(e, phi):
10     # Extended Euclidean Algorithm
11     old_r, r = e, phi
12     old_s, s = 1, 0
```

Ln: 39, Col: 1

Run Share Command Line Arguments

```
[+] Factors: p = 337, q = 3137
[+] Private key d = 231425
[+] Decrypted message (m) = 828365
[+] Flag: netsa{828365}
```

** Process exited - Return Code: 0 **
Press Enter to exit terminal

A simple python code was made to calculate the value of m

Flag: **netsa{828365}**

Forensics

Phantom Protocol


Phantom Protocol

175

Your team has intercepted network traffic from a compromised internal server (phantom_web.pcap). Upon investigation, it was discovered that an attacker exploited a WebSocket interface to bypass authentication and exfiltrate sensitive data.

1-Reconnaissance What Websocket endpoint did the attacker connect do ?

Flag Format = netsa{ws://10.100.0.1:1234/ws}

 phantom_soc...

Forgor

Flag: **forgor**

Phantom Protocol II

Phantom Protocol II

212

From Question Phantom Protocol ,What Credentials did the attacker use to authenticate ?

Flag Format : netsa{username:password}

Same Wireshark file as the predecessor.

```
}
{"cmd": "users"}
{
  "active_connections": 1,
  "your_ip": "192.168.244.132",
  "username": "anonymous",
  "note": "Admin can see more details"
}
{"auth": "YWRtaW46cGFzc3dvcmQ="}
{
  "status": "failed",
  "hint": "Invalid credentials"
}
{"auth": "YWRtaW46cGFzc3dvcmQxMjM="}
{
  "status": "failed",
  "hint": "Invalid credentials"
}
{"auth": "YWRtaW46UEBzc3dvcmQ="}
{
  "status": "failed",
  "hint": "Invalid credentials"
}
{"auth": "YWRtaW46UEBzc3cwcmQ="}
{
  "status": "success",
  "role": "admin",
  "token": "RkxBRzogaWV0c2Fze3diM19TMGNrM3RfUzRuZGJveDF9",
  "message": "Try exploring commands to find the real flag"
}
{"cmd": "debug_logs"}
```

Authentication token.

Decode from Base64 format

Simply enter your data then push the decode button.

YWRtaW46UEBzc3cwcwQ=

ⓘ

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

UTF-8

▼

Source character set.

☐

Decode each line separately (useful for when you have multiple lines of data)

☒

Live mode OFF

Decodes in real-time as you type or paste

<

DECODE

>

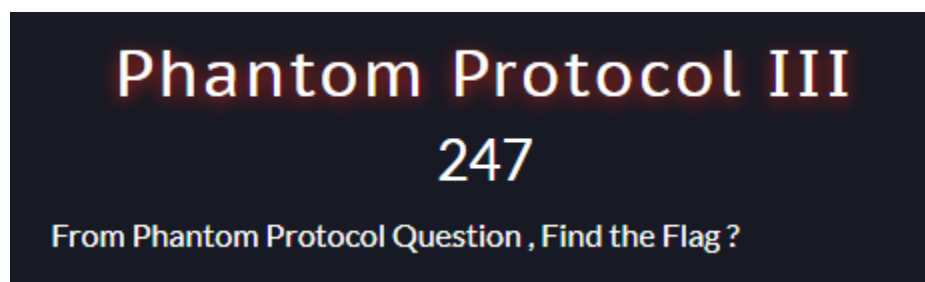
Decodes your data into the area below.

admin:P@ssw0rd

Decode Base64 format.

Flag: `net{admin:P@ssw0rd}`

Phantom Protocol III



Search for WebSocket protocol

No.	Time	Source	Destination	Protocol	Length	Info	Flags	Mask
2557	282.560881	192.168.244.132	192.168.244.129	WebSoc...	82	WebSocket Text	[FIN]	[MASKED]
2559	285.562555	192.168.244.129	192.168.244.132	WebSoc...	172	WebSocket Text	[FIN]	
2560	285.566358	192.168.244.129	192.168.244.132	WebSoc...	72	WebSocket Ping	[FIN]	
2562	285.566673	192.168.244.132	192.168.244.129	WebSoc...	76	WebSocket Pong	[FIN]	[MASKED]

Frame 2559: 172 bytes on wire (1376 bits), 172 bytes captured (1376 bits)

Ethernet II, Src: VMware_c7:c5:17 (00:0c:29:c7:c5:17), Dst: VMware_71:d3:a3 (00:0c:29:71:d3:a3)

Internet Protocol Version 4, Src: 192.168.244.129, Dst: 192.168.244.132

Transmission Control Protocol, Src Port: 8000, Dst Port: 39518, Seq: 869, Ack: 753, Len: 106

WebSocket

- 1... .. = Fin: True
- .100 = Reserved: 0x4
- .1.. = Per-Message Compressed: True
- 0001 = Opcode: Text (1)
- 0... = Mask: False
- .110 1000 = Payload length: 104

Payload

Line-based text data (5 lines)

```
{\n  "response": "pong",\n  "debug": "Look deeper: bmV0c2F7dzNiX3MwY2szdF9mMHJlbnMxY19wNGNrMzd9",\n  "timestamp": "2025-05-17T03:38:40.789132"\n}
```

bmV0c2F7dzNiX3MwY2szdF9mMHJlbnMxY19wNGNrMzd9

For encoded binaries (like images, documents, etc.) use the file up

UTF-8 Source character set.

☐ Decode each line separately (useful for when you have multiple er

☒ Live mode OFF Decodes in real-time as you type or paste (:

< DECODE > Decodes your data into the area below.

netsa{w3b_s0ck3t_f0rens1c_p4ck37}

Base64.

Flag: **netsa{w3b_s0ck3t_f0rens1c_p4ck37}**

IOT

4 Bytes to Freedom

4 Bytes to Freedom

136

Access logs have been wiped. No tags left behind. The only clue? Four strange bytes buried deep in memory. The flag is the obfuscated UID.

Flag format: NETSA{FLAG}

```
#include <EEPROM.h>
byte key[4] = {0xAA, 0xBB, 0xCC, 0xDD};
byte obfuscatedUID[4] = {0x74, 0x16, 0x0C, 0x03};

void setup() {
  Serial.begin(9600);
  for (int i = 0; i < 4; i++) {
    EEPROM.write(i, obfuscatedUID[i]);
  }
  Serial.println("Obfuscated UID written to EEPROM.");
}
void loop() {
```

Author forgot

Flag: **netsa{??}** cant do anything if he forgot


Firmware Reverse

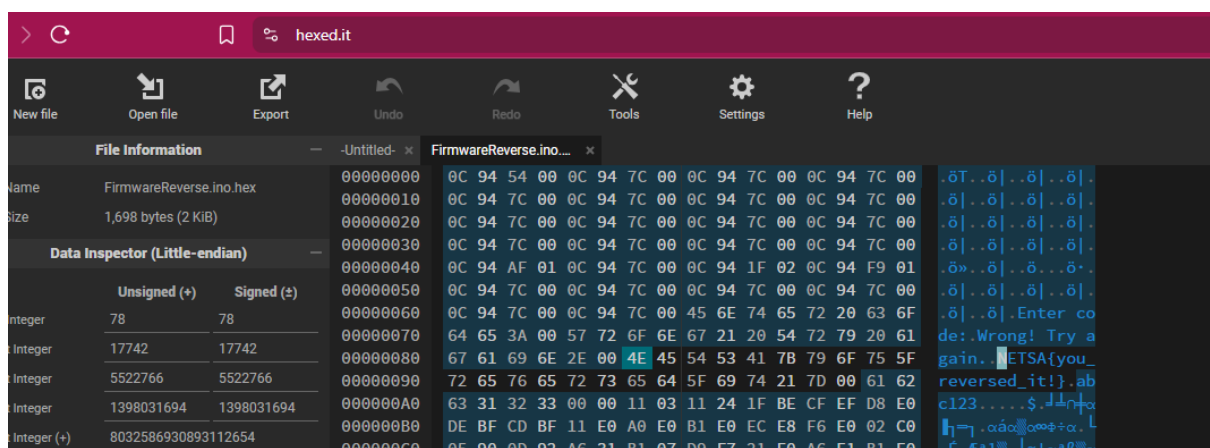
Firmware Reverse

136

Pretty easy. Figure out what's hidden in the firmware.

Flag Format = NETSA{flag}

 FirmwareRev...



Hex editor. Answer straight forward.

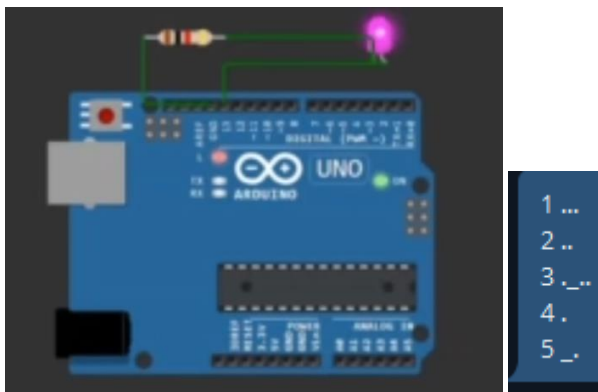
Flag: NETSA{you_reversed_it!}

Trace

Trace 247

The old ways never die. It doesn't speak, it doesn't print, it only blinks. FLAG FORMAT = NETSA{flag}

 trace.mp4



A video of Arduino module blinking. Blinking = morse code.

Flag: NETSA{SILENT}

Blink Logic

Blink Logic

364

You've accessed a locked embedded device. Only one 4-Key sequence to unlock the system

How to Run:

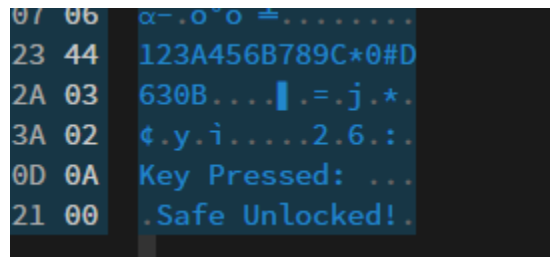
1. Go to <https://wokwi.com/>
2. Create New Project, then click the folder icon
3. Replace the 'diagram.json' with the one provided
4. Upload the 'hex' file by clicking F1 and type "upload firmware"
5. Press Play Button to simulate
6. Contact Sarah if you encounter problems ^_^

Flag Format :NETSA{1234}

[diagram.json](#) [BlinkLogic.in...](#)

Simulation

Key Pressed: 0
Key Pressed: B
Key Pressed: 6
Key Pressed: 3
Key Pressed: 0
Key Pressed: B
Safe Unlocked!



Follow the instructions.

Check hex editor last digits.

Flag: NETSA{630B}

EEPROM Secret

The challenge card has a dark blue background. At the top, the title 'EEPROM Secret' is written in a large, white, serif font with a red glow effect, followed by the number '460' in a smaller white serif font. Below this, a paragraph in a white monospace font reads: 'A binary EEPROM image contains a scrambled flag. Your task is to reverse the obfuscation and extract the secret.' At the bottom left, there is a light blue rectangular button containing a white download icon and the text 'eeprom.bin'.

Author forgor

Flag: `net{sa{??}}` D:

OSINT

McD

McD
212

Lisa has travel to somewhere in Malaysia and take a picture of mcd. Lisa don't want to disclose the location in the picture to her friend. Can you help Lisa's friend find the mcd location?

Flag Format: netsa{location}

 mcd.png



This McD is in Farlim, Penang.

Flag: netsa{farlim}

Train to Hospital

Train to Hospital

247

Your friend Jeam Wong just became a proud father in a hospital somewhere in Singapore. He sent you a quick photo from the hospital and invited you to visit. Locate the nearest MRT station to the hospital so you can visit the happy family !

Flag format: netsa{Location_Of_The_Station}

021d911e-b...



Gongcha in a hospital. :D

Flag: netsa{Little_India}

Final Debug

Final Debug 364

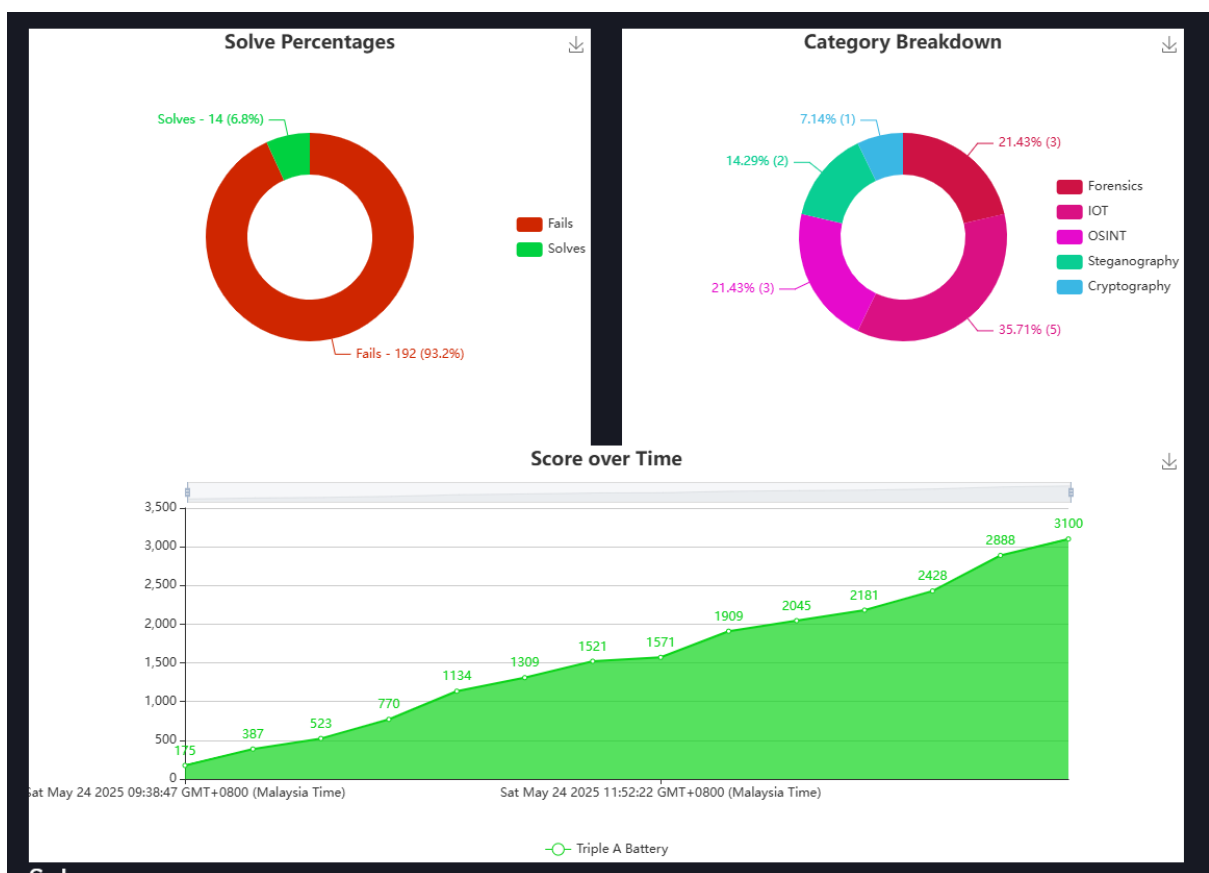
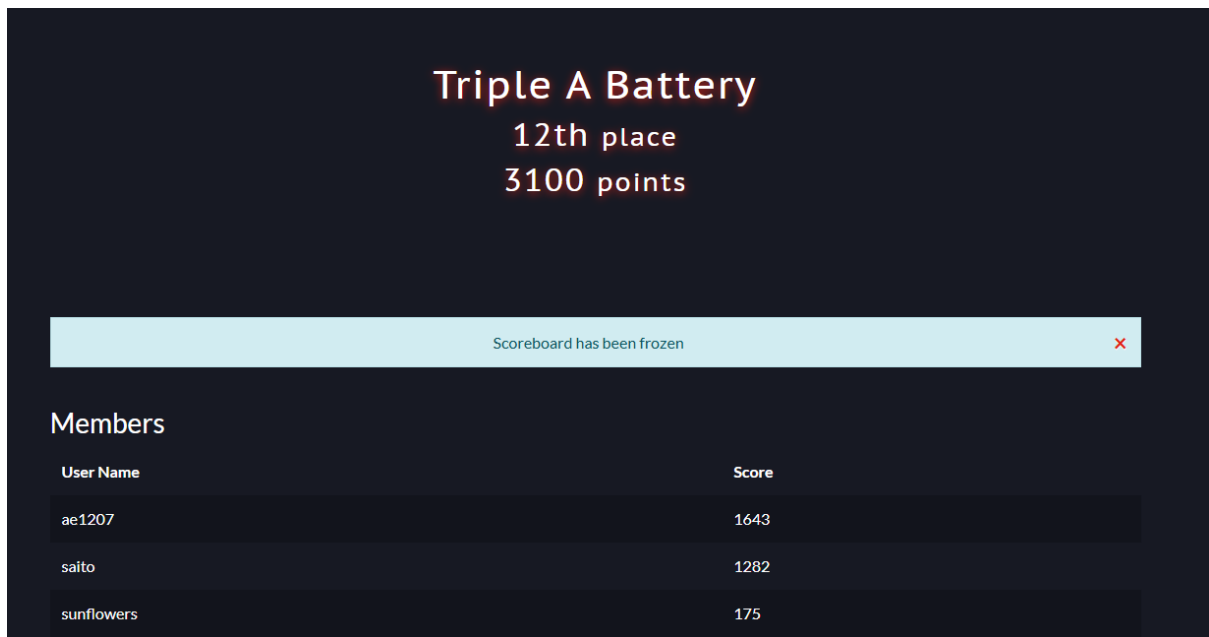
One of our former developers did not leave peacefully. After we fired her, she disappeared and took something valuable with her. Recently, she has been active online under the name pyth0n_I0v3rz, posting strange messages and small clues. We need your help to find and recover what she took — maybe it contains your flag.

Flag Format : netsa{}

Find at GitHub. Author said it

Flag: netsa{??} (he forgot)

Triple A Battery as a Team



Solves

Challenge	Category	Value	Time
Phantom Protocol II	Forensics	212	May 24th, 2:55:11 PM
EEPROM Secret	IOT	460	May 24th, 2:54:54 PM
Phantom Protocol III	Forensics	247	May 24th, 2:45:45 PM
4 Bytes to Freedom	IOT	136	May 24th, 2:41:05 PM
Phantom Protocol	Forensics	136	May 24th, 2:10:09 PM
Final Debug	OSINT	338	May 24th, 11:53:55 AM
Welcome to CTF	Steganography	50	May 24th, 11:52:22 AM
Train to Hospital	OSINT	212	May 24th, 11:36:59 AM
Baby RSA	Cryptography	175	May 24th, 10:33:22 AM
Blink Logic	IOT	364	May 24th, 10:29:53 AM
Trace	IOT	247	May 24th, 10:23:07 AM
Firmware Reverse	IOT	136	May 24th, 10:11:04 AM
McD	OSINT	212	May 24th, 10:10:01 AM
Rasengram	Steganography	175	May 24th, 9:38:47 AM