CUHK Cybersecurity Capture The Flag Competition 2025

Write-up Sharing - “The Betrayal” series

p3n9uin

$ whoami

CTF Handle: p3n9uin

Major:

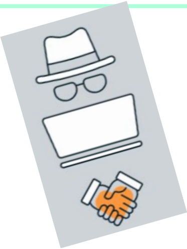
Information Engineering, Year 4

My CTF categories:

OSINT (mainly), rev, forens

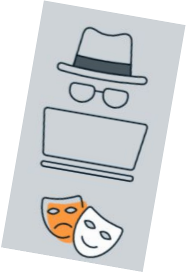
Worked as a SOC junior analyst in one of my summer internships

WARNING:The ethics of hacking in CTFs



Ethical hacking:
AUTHORIZED and **APPROVED** practice of hacking into computer system to identify potential vulnerabilities in the computer systems

Purpose:
Investigate vulnerabilities so system administrators can fix it!



Only hack within the scope of assessment and plan
□ **In CTFs: DO NOT attack the CTF platform itself nor the players!**

Keep the learned vulnerabilities **CONFIDENTIAL**.
NEVER utilize the vulnerabilities in a way detrimental to the owner of the system

WARNING:The ethics of hacking in CTFs

DISCLAIMER:

WE ASSUME NO RESPONSIBILITY FOR ANY ACTIONS PERFORMED OUTSIDE THE WORKSHOP.

ALL VULNERABILITIES AND WRITE-UPS SHARED IN THE WORKSHOPS ARE FOR EDUCATIONAL PURPOSES ONLY.

WARNING:The ethics of hacking in CTFs

The exercises in this session should be attempted ONLY INSIDE THE SECLUDED LAB ENVIRONMENT documented or provided. Please note that most of the attacks described in the slides would be ILLEGAL if attempted on machines that you do not explicit permission to test and attack. You should bear responsibilities for any actions performed outside the secluded lab.

The challenge server should be regarded as a hostile environment. You should not use your real information when attempting challenges.

Do not intentionally disrupt other students who are working on the challenges or disclose private information you found on the challenge server (e.g. IP address of other students). Please let us know if you accidentally broke the challenge.

“The Betrayal” series - Background

The challenges in this series aims to give participants a small taste into some parts of the defense side (Blue Team) of cyber security.

You are not just here to exploit a software or system, you are here to **investigate what happened** .

The three challenges each serves a part in the whole attack chain:

- “Copyright Infringement” (misc/OSINT) - Why attack?
- “Remote Intrusion” (forens) - How can adversaries break in?
- “Layer by Layer” (rev) - What is the potential impact?

Each challenge is inspired by what I *personally* learned or heard about in the past year.

“The Betrayal” series - “Remote Intrusion”

Challenge Description:

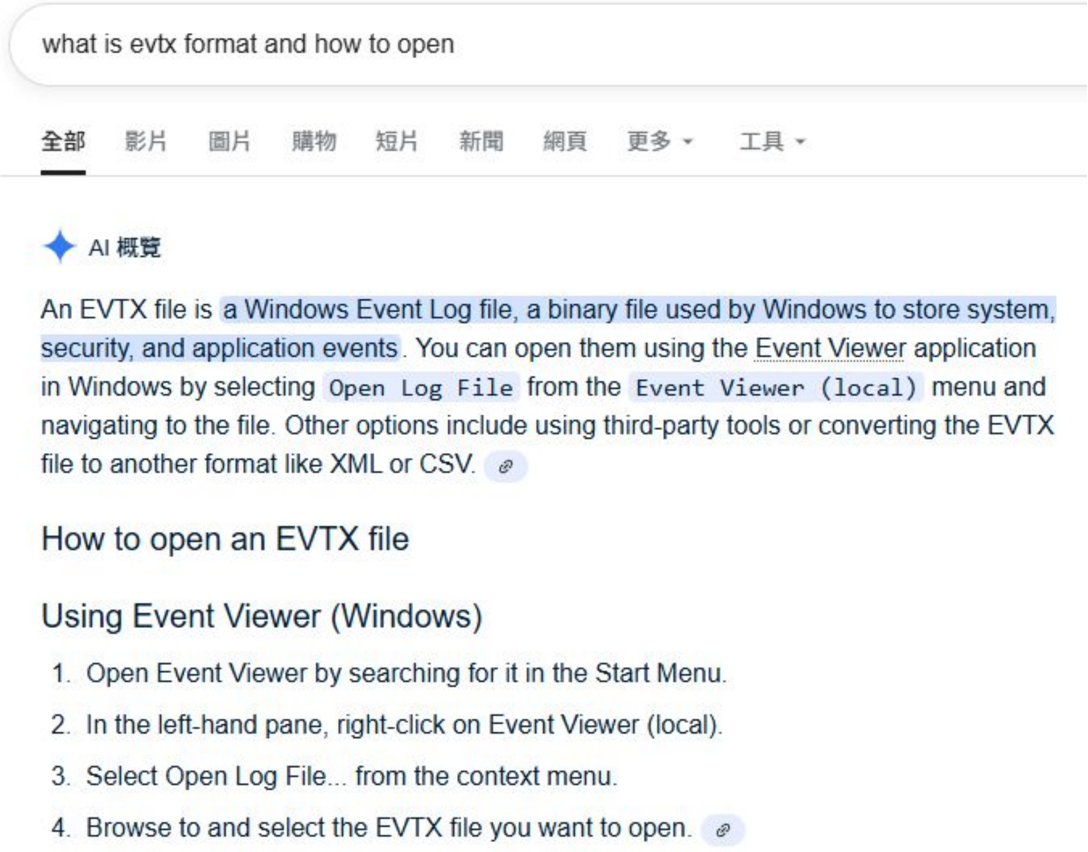
It seems like an attacker is attempting to remotely access the computer of the CEO of Icey Penguin Marketing Agency, the Incident Response team has extracted the login events for you. However since remoting into the device using RDP is quite common in Icey Penguin Marketing Agency, it is quite hard to trace down the attack. Can you discover the attacker's identity?

Given file: 04_chall.evtx (what is this file format…)

CUHK CTF 2025

“The Betrayal” series - “Remote Intrusion”

What is .evtx file format:



what is evtx format and how to open

An EVTX file is a Windows Event Log file, a binary file used by Windows to store system, security, and application events. You can open them using the Event Viewer application in Windows by selecting `Open Log File` from the `Event Viewer (local)` menu and navigating to the file. Other options include using third-party tools or converting the EVTX file to another format like XML or CSV.

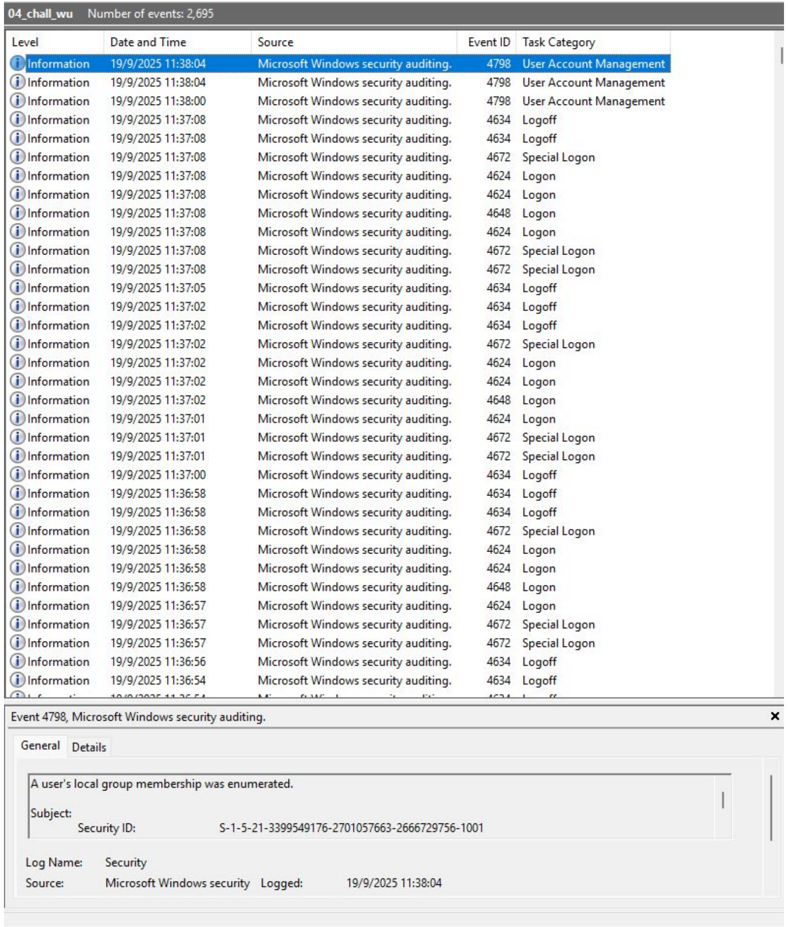
How to open an EVTX file

Using Event Viewer (Windows)

1. Open Event Viewer by searching for it in the Start Menu.
2. In the left-hand pane, right-click on Event Viewer (local).
3. Select Open Log File... from the context menu.
4. Browse to and select the EVTX file you want to open.

“The Betrayal” series - “Remote Intrusion”

Too many logs...
how to filter them out?



04_chall_wu Number of events: 2,695

Level	Date and Time	Source	Event ID	Task Category
Information	19/9/2025 11:38:04	Microsoft Windows security auditing.	4798	User Account Management
Information	19/9/2025 11:38:04	Microsoft Windows security auditing.	4798	User Account Management
Information	19/9/2025 11:38:00	Microsoft Windows security auditing.	4798	User Account Management
Information	19/9/2025 11:37:08	Microsoft Windows security auditing.	4634	Logoff
Information	19/9/2025 11:37:08	Microsoft Windows security auditing.	4634	Logoff
Information	19/9/2025 11:37:08	Microsoft Windows security auditing.	4672	Special Logon
Information	19/9/2025 11:37:08	Microsoft Windows security auditing.	4624	Logon
Information	19/9/2025 11:37:08	Microsoft Windows security auditing.	4624	Logon
Information	19/9/2025 11:37:08	Microsoft Windows security auditing.	4648	Logon
Information	19/9/2025 11:37:08	Microsoft Windows security auditing.	4624	Logon
Information	19/9/2025 11:37:08	Microsoft Windows security auditing.	4672	Special Logon
Information	19/9/2025 11:37:08	Microsoft Windows security auditing.	4672	Special Logon
Information	19/9/2025 11:37:05	Microsoft Windows security auditing.	4634	Logoff
Information	19/9/2025 11:37:02	Microsoft Windows security auditing.	4634	Logoff
Information	19/9/2025 11:37:02	Microsoft Windows security auditing.	4634	Logoff
Information	19/9/2025 11:37:02	Microsoft Windows security auditing.	4672	Special Logon
Information	19/9/2025 11:37:02	Microsoft Windows security auditing.	4624	Logon
Information	19/9/2025 11:37:02	Microsoft Windows security auditing.	4624	Logon
Information	19/9/2025 11:37:02	Microsoft Windows security auditing.	4648	Logon
Information	19/9/2025 11:37:01	Microsoft Windows security auditing.	4624	Logon
Information	19/9/2025 11:37:01	Microsoft Windows security auditing.	4672	Special Logon
Information	19/9/2025 11:37:01	Microsoft Windows security auditing.	4672	Special Logon
Information	19/9/2025 11:37:00	Microsoft Windows security auditing.	4634	Logoff
Information	19/9/2025 11:36:58	Microsoft Windows security auditing.	4634	Logoff
Information	19/9/2025 11:36:58	Microsoft Windows security auditing.	4634	Logoff
Information	19/9/2025 11:36:58	Microsoft Windows security auditing.	4672	Special Logon
Information	19/9/2025 11:36:58	Microsoft Windows security auditing.	4624	Logon
Information	19/9/2025 11:36:58	Microsoft Windows security auditing.	4624	Logon
Information	19/9/2025 11:36:58	Microsoft Windows security auditing.	4648	Logon
Information	19/9/2025 11:36:57	Microsoft Windows security auditing.	4624	Logon
Information	19/9/2025 11:36:57	Microsoft Windows security auditing.	4672	Special Logon
Information	19/9/2025 11:36:57	Microsoft Windows security auditing.	4672	Special Logon
Information	19/9/2025 11:36:56	Microsoft Windows security auditing.	4634	Logoff
Information	19/9/2025 11:36:54	Microsoft Windows security auditing.	4634	Logoff

Event 4798, Microsoft Windows security auditing.

A user's local group membership was enumerated.

Subject:
Security ID: S-1-5-21-3399549176-2701057663-2666729756-1001

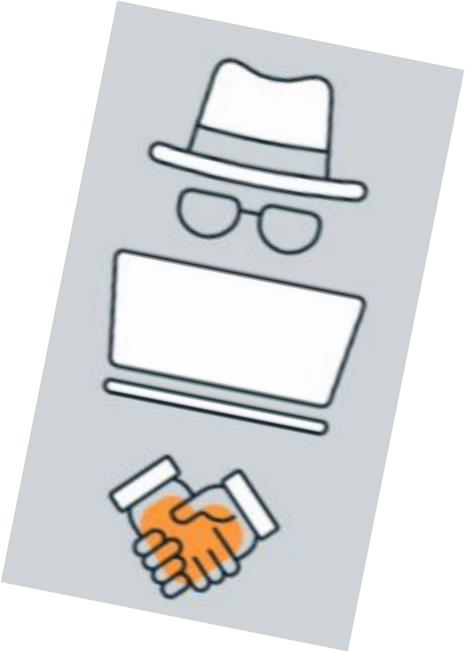
Log Name: Security
Source: Microsoft Windows security
Logged: 19/9/2025 11:38:04

"The Betrayal" series - "Remote Intrusion"

One of the first lines of detection:
Suspicious logins

What happens when an attacker don't know your password, but still want to access your account?

They try to brute-force password, or cause a password reset (sometimes back to a default password) which they can possibly have more control of the process
In the meantime, there will be records of failed logins.

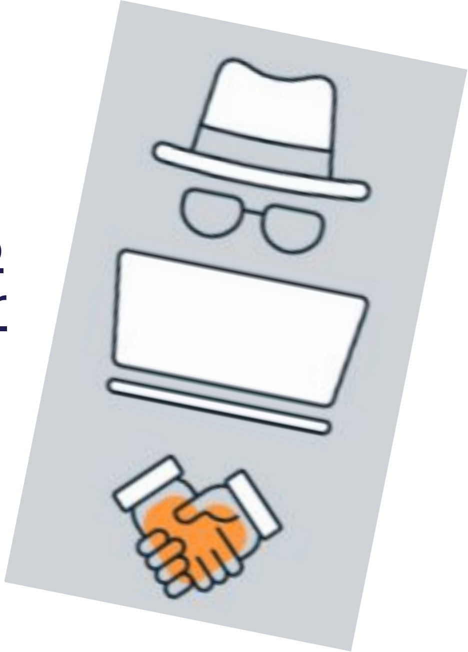


“The Betrayal” series - “Remote Intrusion”

Windows Event Logs shows different event IDs, which ones are related to failed logins?

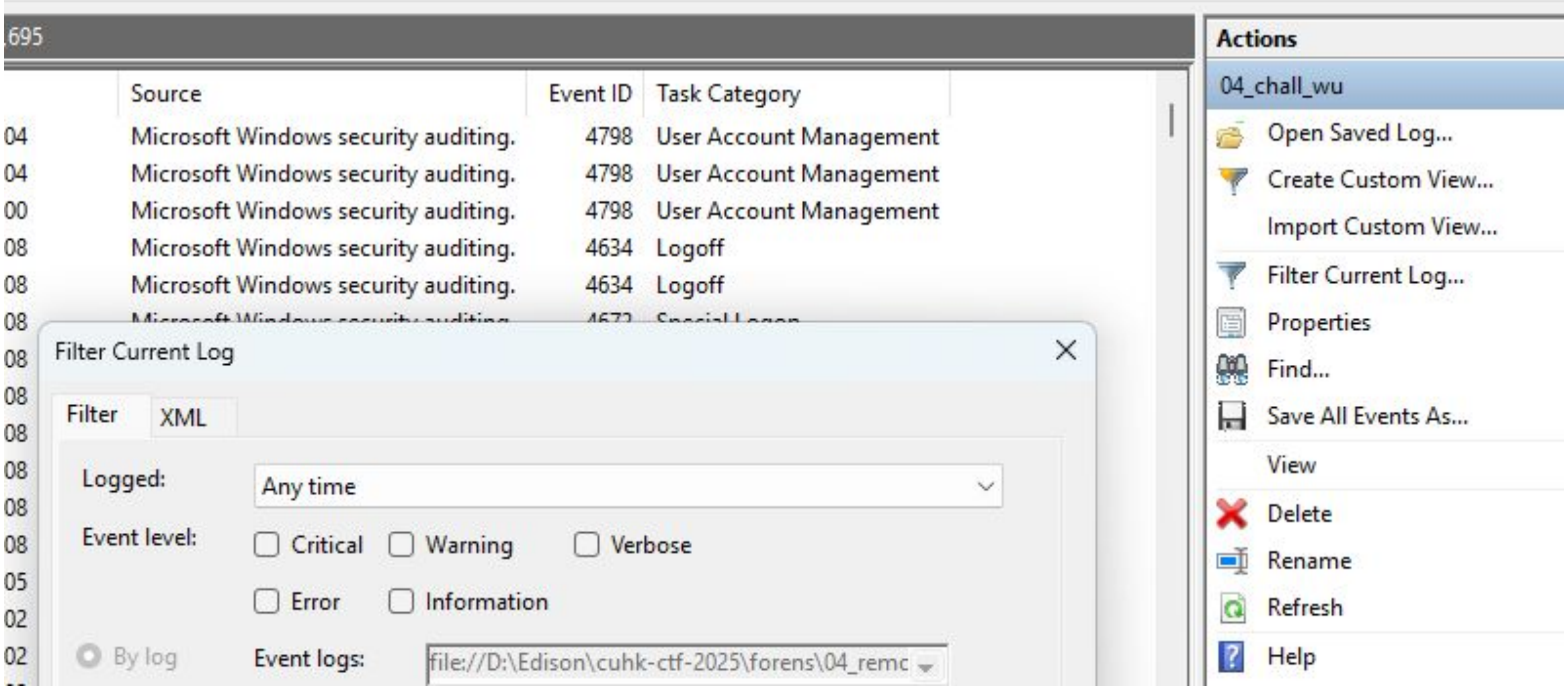
Event 4625 – An account failed to log on

https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-10/security/threat-protection/auditing/event-4625



“The Betrayal” series - “Remote Intrusion”

In Windows Event Viewer, click into “Filter Current Log”:



	Source	Event ID	Task Category
04	Microsoft Windows security auditing.	4798	User Account Management
04	Microsoft Windows security auditing.	4798	User Account Management
00	Microsoft Windows security auditing.	4798	User Account Management
08	Microsoft Windows security auditing.	4634	Logoff
08	Microsoft Windows security auditing.	4634	Logoff
08	Microsoft Windows security auditing.	4672	Special Logon
08			
08			
08			
08			
08			
08			
05			
02			
02			

Filter Current Log

Filter | XML

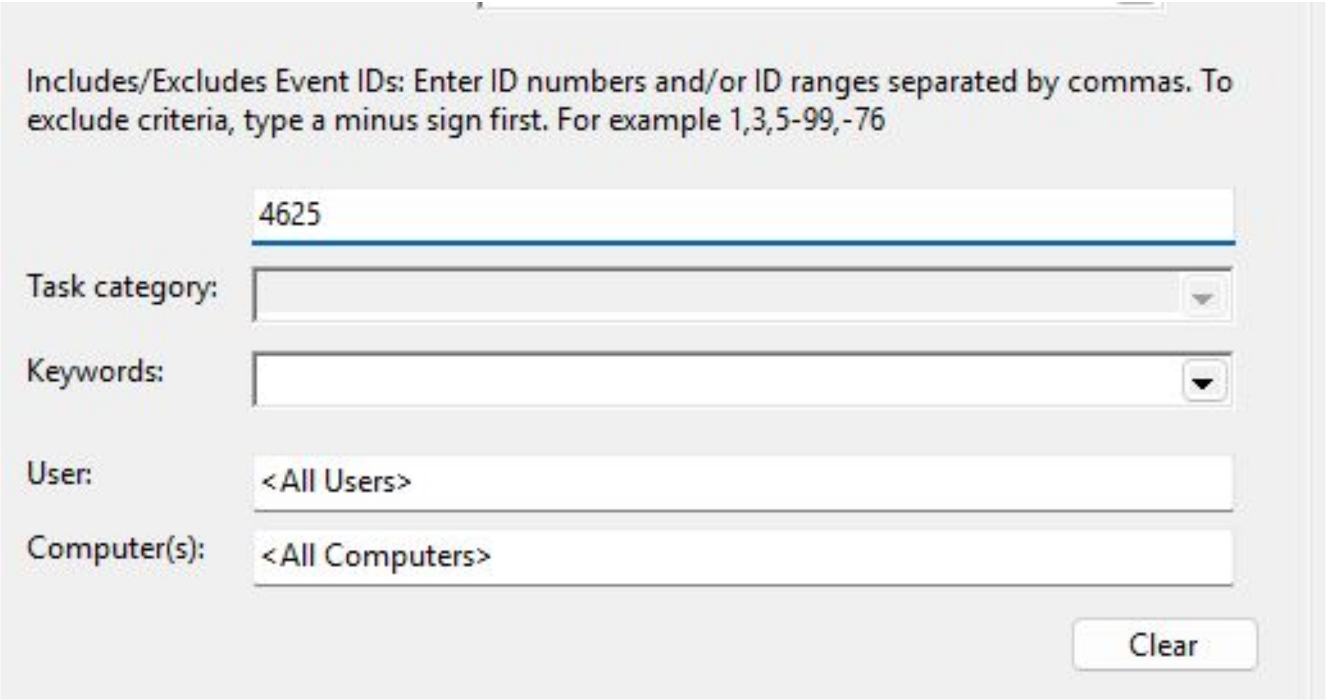
Logged: Any time

Event level: ☐ Critical ☐ Warning ☐ Verbose ☐ Error ☐ Information

○ By log Event logs: file://D:\Edison\cuhk-ctf-2025\forens\04_remc

“The Betrayal” series - “Remote Intrusion”

Filter only 4625 events



Includes/Excludes Event IDs: Enter ID numbers and/or ID ranges separated by commas. To exclude criteria, type a minus sign first. For example 1,3,5-99,-76

4625

Task category:

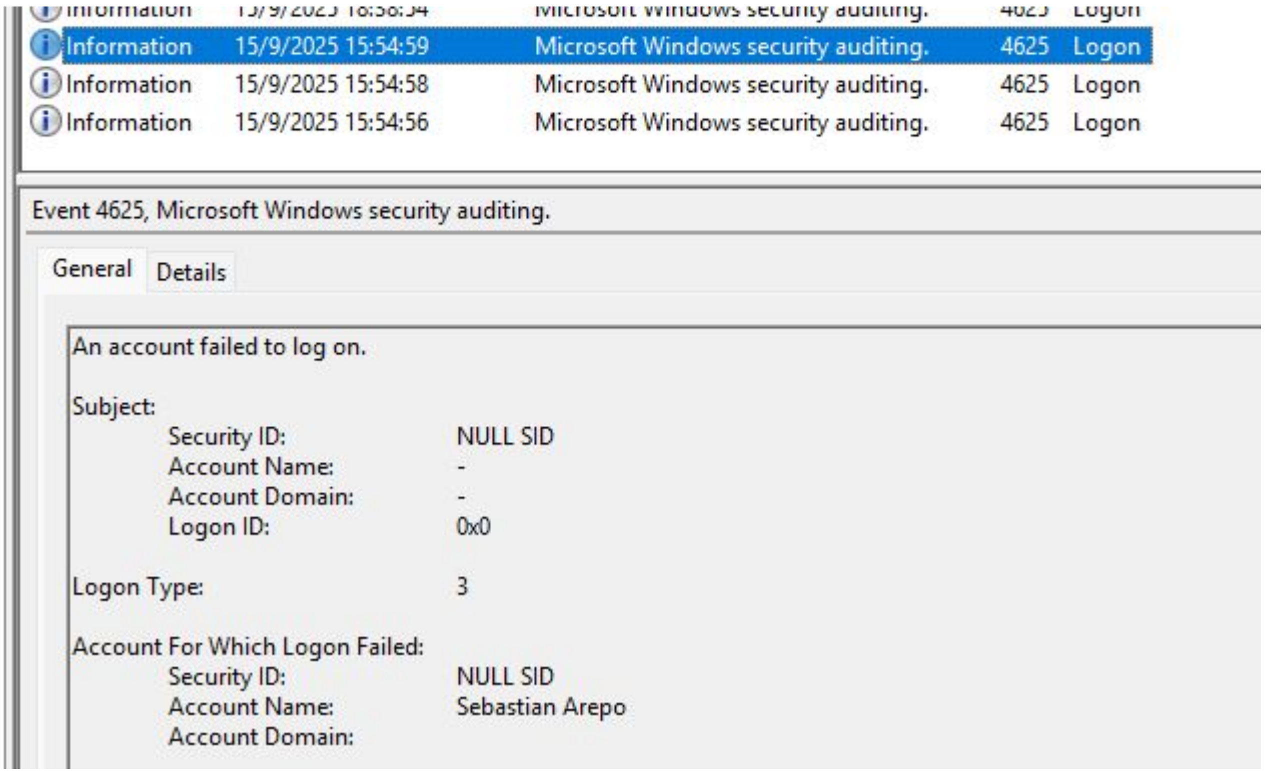
Keywords:

User: <All Users>

Computer(s): <All Computers>

“The Betrayal” series - “Remote Intrusion”

Notice that Account Name is displayed in plaintext… this seems related…



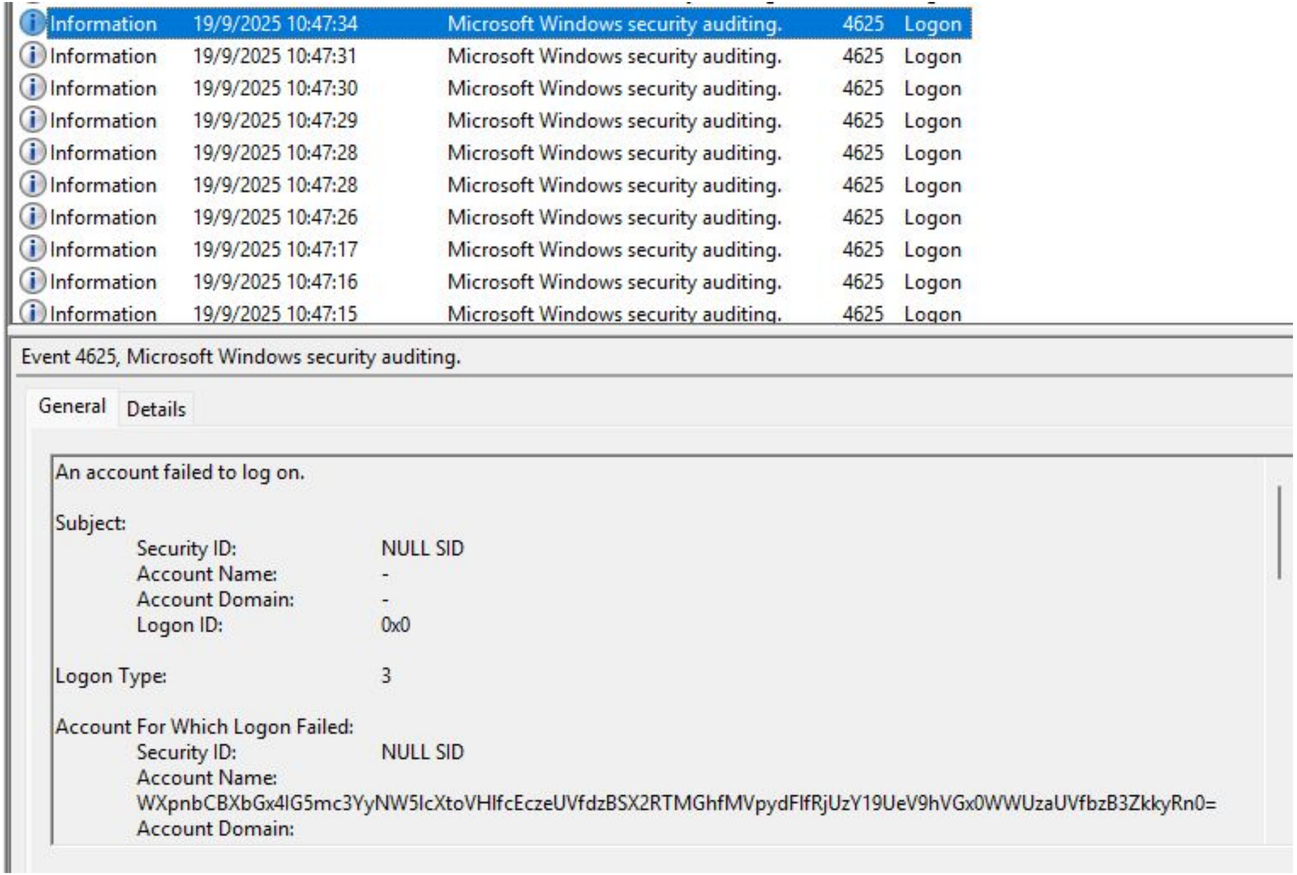
```
Information   15/9/2025 15:54:59   Microsoft Windows security auditing.   4625   Logon
Information   15/9/2025 15:54:58   Microsoft Windows security auditing.   4625   Logon
Information   15/9/2025 15:54:56   Microsoft Windows security auditing.   4625   Logon
```

Event 4625, Microsoft Windows security auditing.

```
An account failed to log on.

Subject:
	Security ID:		NULL SID
	Account Name:		-
	Account Domain:		-
	Logon ID:		0x0

Logon Type:			3

Account For Which Logon Failed:
	Security ID:		NULL SID
	Account Name:		Sebastian Arepo
	Account Domain:
```

“The Betrayal” series - “Remote Intrusion”

Find one of the records that has a Base64 encoded Account Name for some reason…



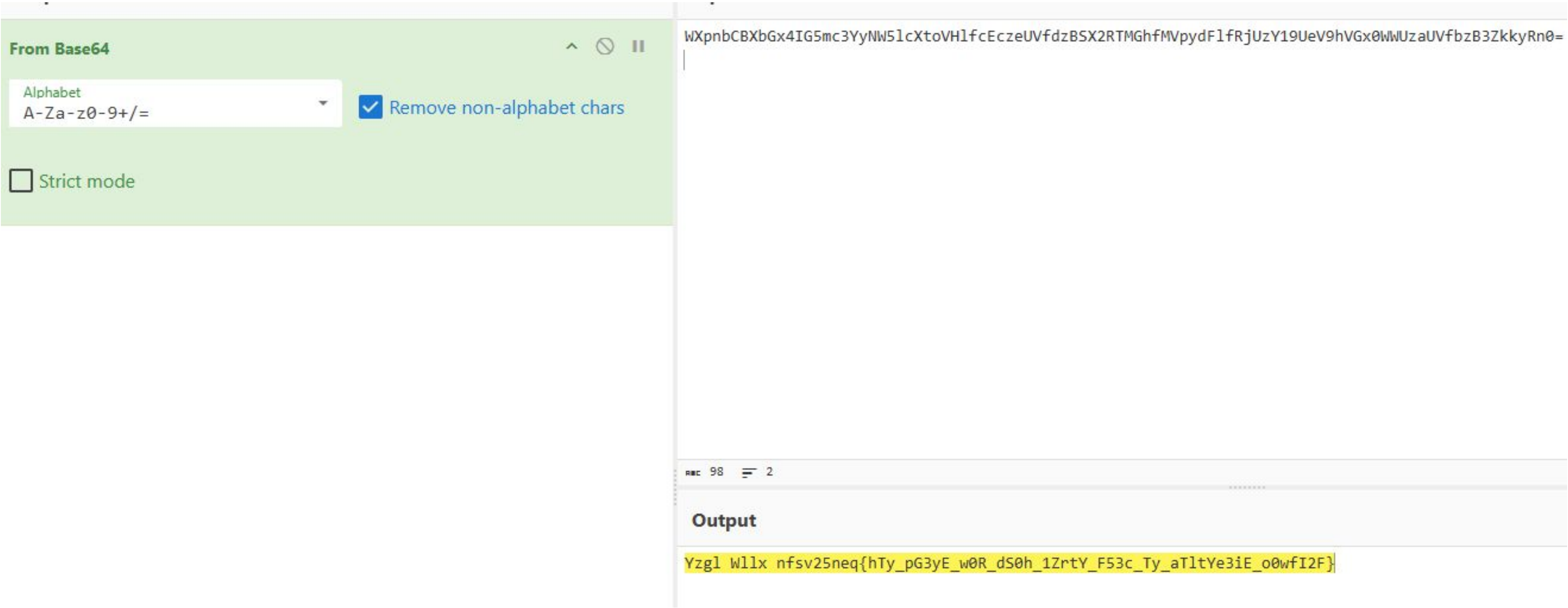
```
Information   19/9/2025 10:47:34   Microsoft Windows security auditing.   4625   Logon
Information   19/9/2025 10:47:31   Microsoft Windows security auditing.   4625   Logon
Information   19/9/2025 10:47:30   Microsoft Windows security auditing.   4625   Logon
Information   19/9/2025 10:47:29   Microsoft Windows security auditing.   4625   Logon
Information   19/9/2025 10:47:28   Microsoft Windows security auditing.   4625   Logon
Information   19/9/2025 10:47:28   Microsoft Windows security auditing.   4625   Logon
Information   19/9/2025 10:47:26   Microsoft Windows security auditing.   4625   Logon
Information   19/9/2025 10:47:17   Microsoft Windows security auditing.   4625   Logon
Information   19/9/2025 10:47:16   Microsoft Windows security auditing.   4625   Logon
Information   19/9/2025 10:47:15   Microsoft Windows security auditing.   4625   Logon
```

Event 4625, Microsoft Windows security auditing.

```
An account failed to log on.

Subject:
	Security ID:		NULL SID
	Account Name:		-
	Account Domain:		-
	Logon ID:		0x0

Logon Type:			3

Account For Which Logon Failed:
	Security ID:		NULL SID
	Account Name:
	WXpnbCBXbGx4IG5mc3YyNW5lcXtoVHlfcEczeUVfdzBSX2RTMGhfMVpydFlfRjUzY19UeV9hVGx0WWUzaUVfbzB3ZkkyRn0=
	Account Domain:
```

“The Betrayal” series - “Remote Intrusion”

Decode it to find something flag-like... maybe try to ROT it?

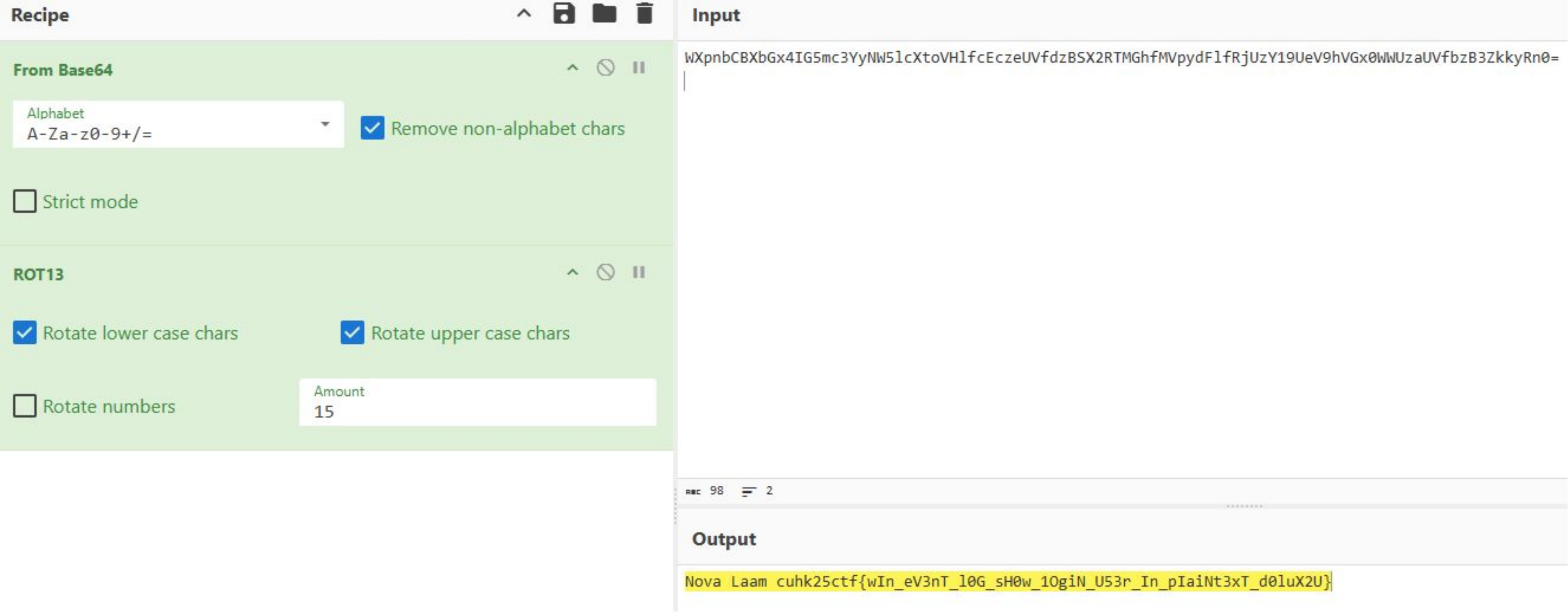


```
WXpnbCBXbGx4IG5mc3YyNW5lcXtoVHlfcEczeUVfdzBSX2RTMGhfMVpydFlfRjUzY19UeV9hVGx0WWUzaUVfbzB3ZkkyRn0=
```

```
Yzgl Wllx nfsv25neq{hTy_pG3yE_w0R_dS0h_1ZrtY_F53c_Ty_aTltYe3iE_o0wfI2F}
```

“The Betrayal” series - “Remote Intrusion”

There is your flag:
cuhk25ctf{wIn_eV3nT_l0G_sH0w_1OgiN_U53r_In_pIaiNt3xT_d0luX2U}



From Base64

Alphabet: A-Za-z0-9+/=

☑ Remove non-alphabet chars

☐ Strict mode

ROT13

☑ Rotate lower case chars

☑ Rotate upper case chars

☐ Rotate numbers

Amount: 15

Input

```
WXpnbCBXbGx4IG5mc3YyNW5lcXtoVHlfcEczeUVfdzBSX2RTMGhfMVpydFlfRjUzY19UeV9hVGx0WWUzaUVfbzB3ZkkyRn0=
```

Output

```
Nova Laam cuhk25ctf{wIn_eV3nT_l0G_sH0w_1OgiN_U53r_In_pIaiNt3xT_d0luX2U}
```

(P.S. there is an unintended fake flag, see if you can find it!)

“The Betrayal” series - “Layer by Layer”

Challenge Description:

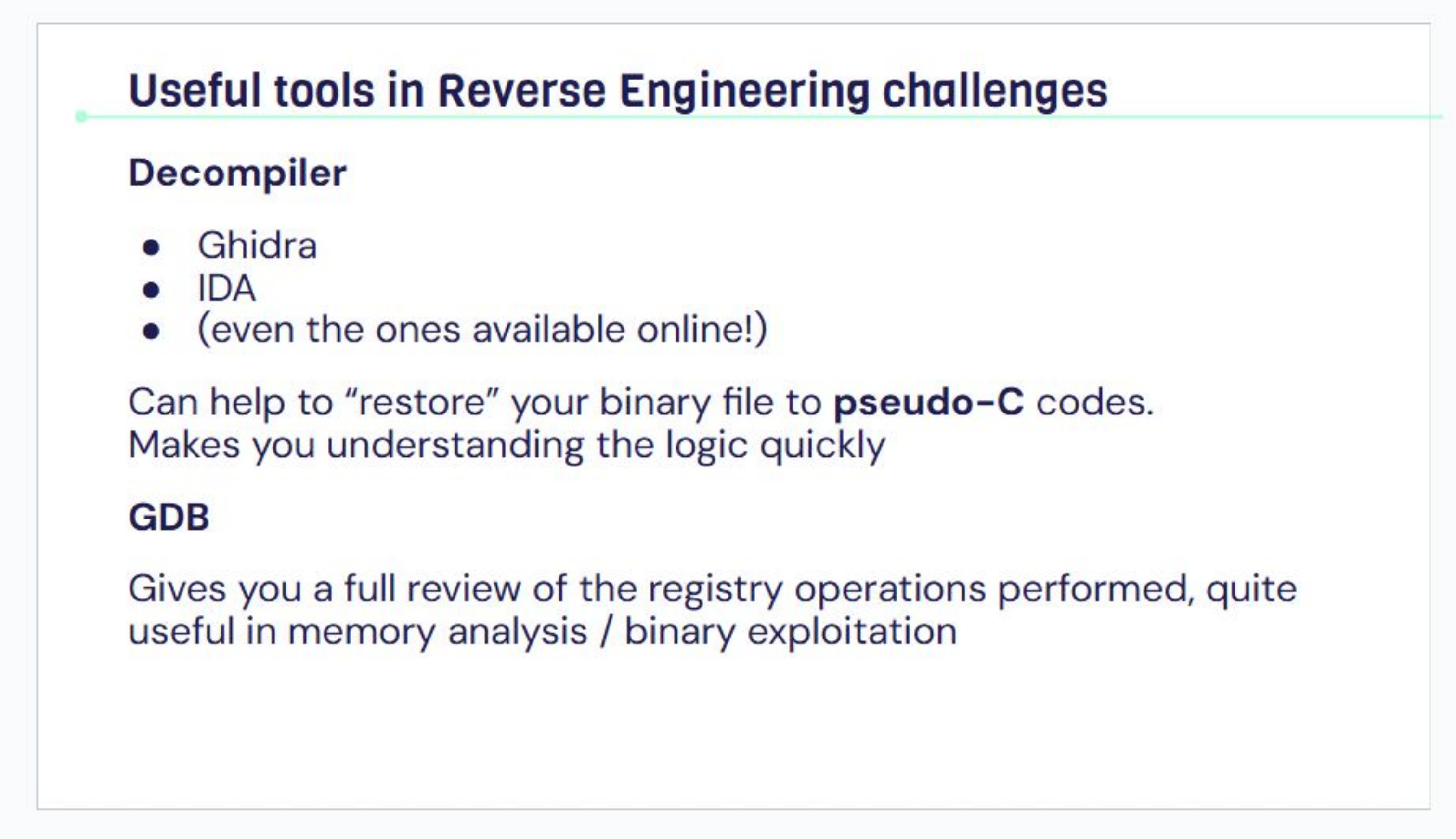
After entering the company's network, the attacker left a potential malware behind. However the anti-virus and endpoint detection software that the company used did not flagged this as a malware. Can you discover the trick that attacker used to bypass the software's detection?

Given file: AccountLedgerPro.jar

(This is a malware analysis challenge!)

CUHK CTF 2025

Remember this slide from the Training Workshop?



Useful tools in Reverse Engineering challenges

Decompiler

- Ghidra
- IDA
- (even the ones available online!)

Can help to “restore” your binary file to **pseudo-C** codes.
Makes you understanding the logic quickly

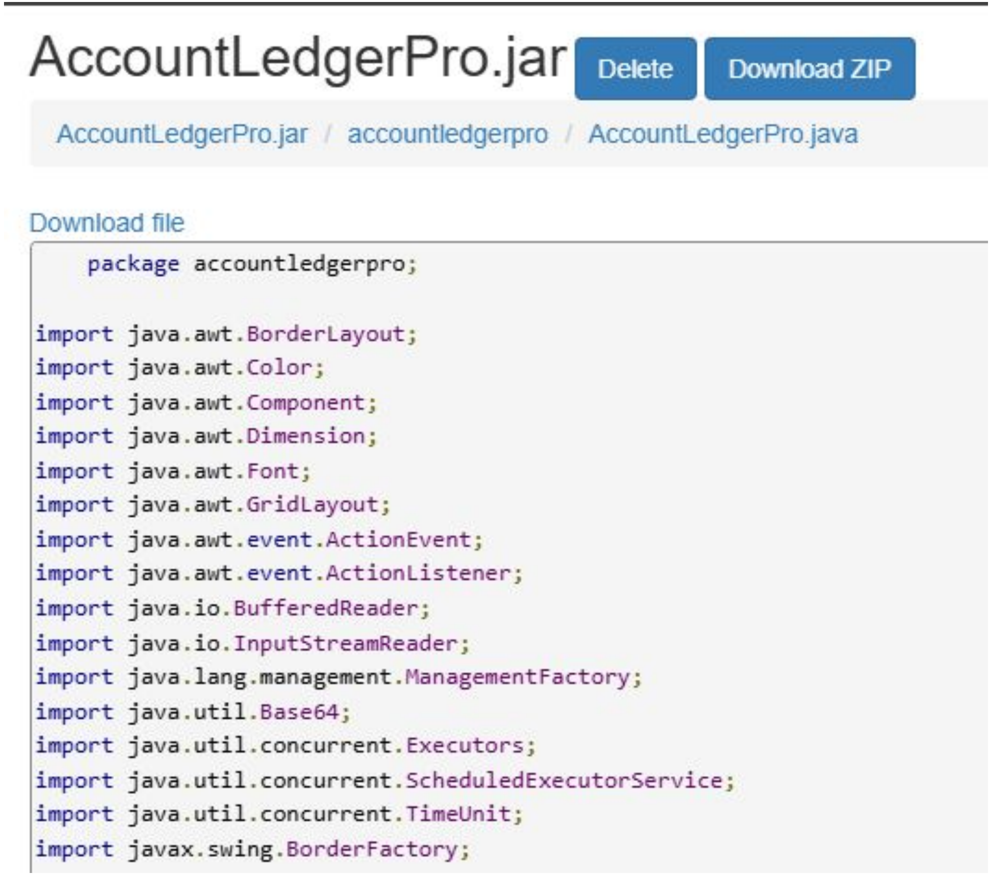
GDB

Gives you a full review of the registry operations performed, quite useful in memory analysis / binary exploitation

“The Betrayal” series - “Layer by Layer”

Using jadx or any online Java decompiler, you will get this code:

(I used Decompiler.com here)

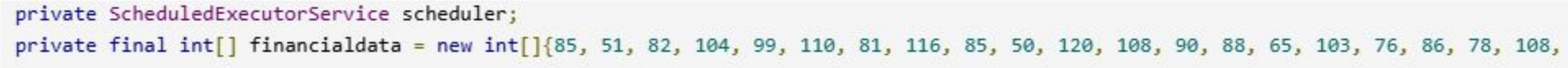

AccountLedgerPro.jar
Delete
Download ZIP
AccountLedgerPro.jar / accountledgerpro / AccountLedgerPro.java
Download file

```
    package accountledgerpro;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Component;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.lang.management.ManagementFactory;
import java.util.Base64;
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.TimeUnit;
import javax.swing.BorderFactory;
```

“The Betrayal” series - “Layer by Layer”

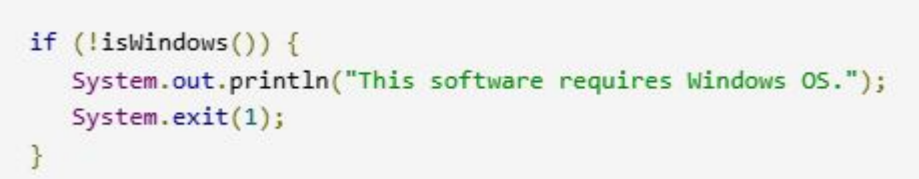
Findings in program inspection:

Unusually long integer array called “financialdata”...



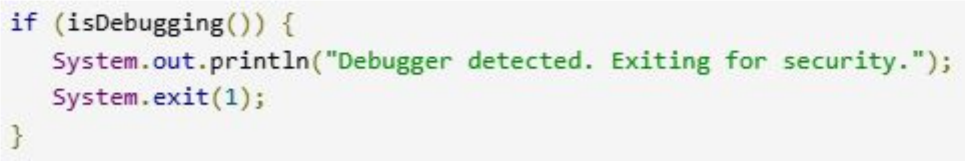
```
private ScheduledExecutorService scheduler;
private final int[] financialdata = new int[]{85, 51, 82, 104, 99, 110, 81, 116, 85, 50, 120, 108, 90, 88, 65, 103, 76, 86, 78, 108,
```

OS check...



```
if (!isWindows()) {
    System.out.println("This software requires Windows OS.");
    System.exit(1);
}
```

Debugger check...



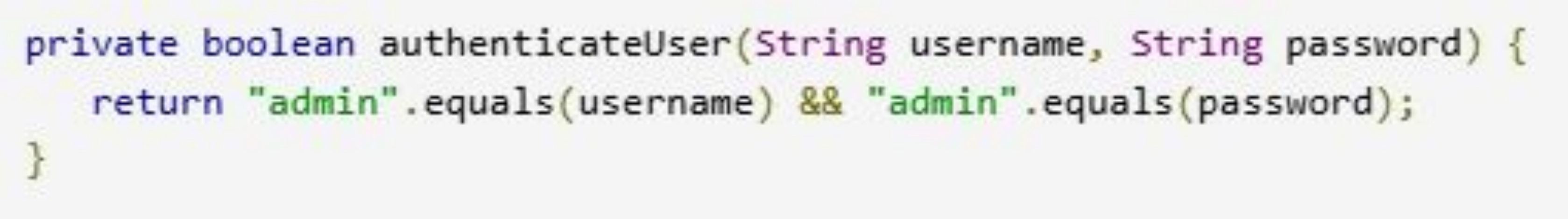
```
if (isDebugging()) {
    System.out.println("Debugger detected. Exiting for security.");
    System.exit(1);
}
```

“The Betrayal” series - “Layer by Layer”

Findings in program inspection:

Plaintext login credentials…

(Bad development practice, don’t do this in real-life software and website development!)

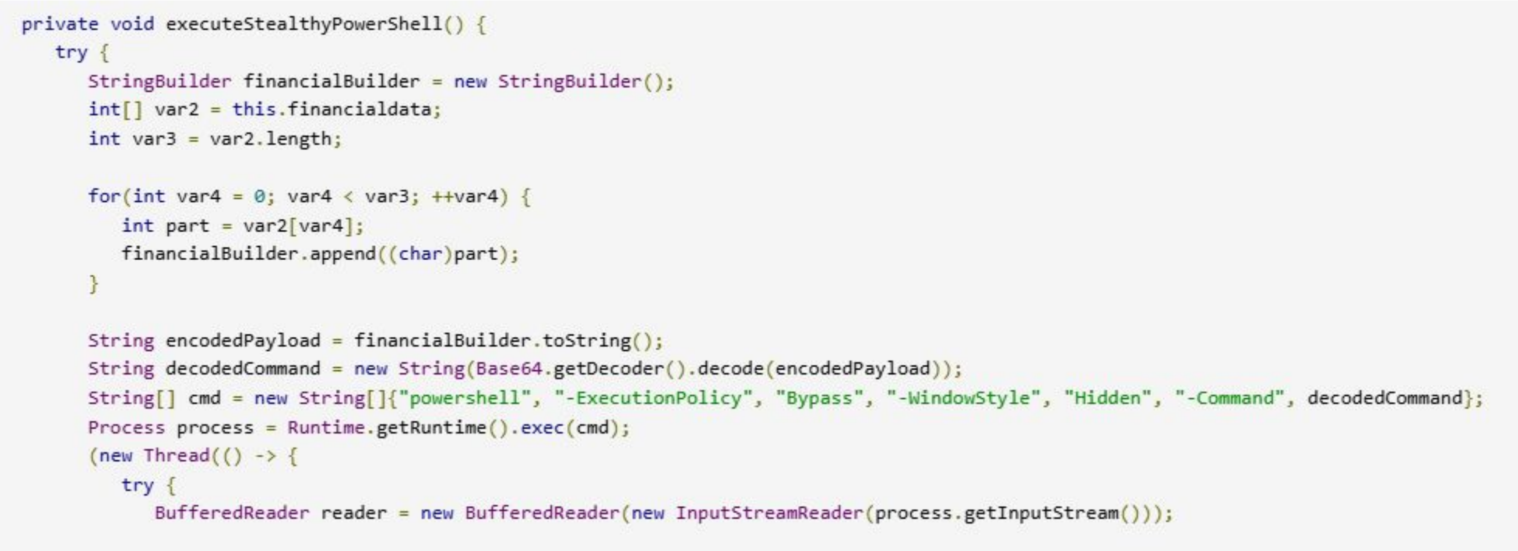


```
private boolean authenticateUser(String username, String password) {
    return "admin".equals(username) && "admin".equals(password);
}
```

“The Betrayal” series - “Layer by Layer”

Findings in program inspection:

Hidden execution of PowerShell commands...

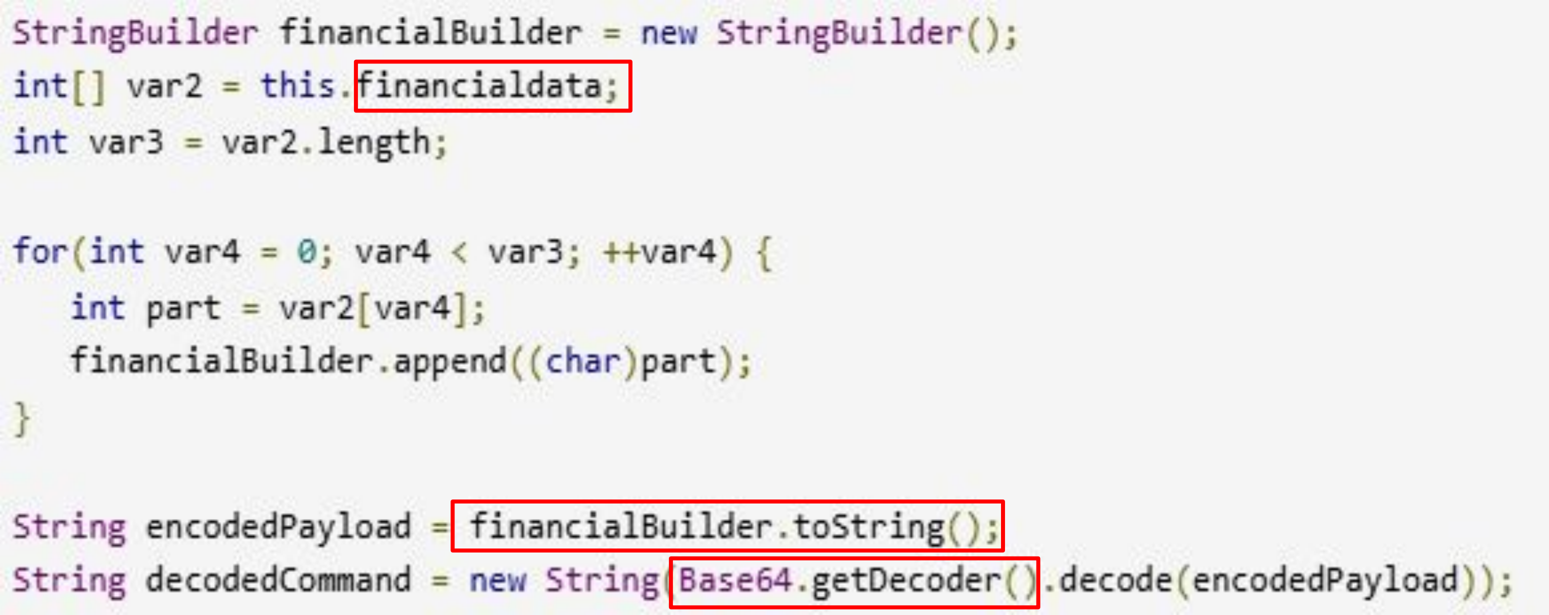


```
private void executeStealthyPowerShell() {
    try {
        StringBuilder financialBuilder = new StringBuilder();
        int[] var2 = this.financialdata;
        int var3 = var2.length;

        for(int var4 = 0; var4 < var3; ++var4) {
            int part = var2[var4];
            financialBuilder.append((char)part);
        }

        String encodedPayload = financialBuilder.toString();
        String decodedCommand = new String(Base64.getDecoder().decode(encodedPayload));
        String[] cmd = new String[]{"powershell", "-ExecutionPolicy", "Bypass", "-WindowStyle", "Hidden", "-Command", decodedCommand};
        Process process = Runtime.getRuntime().exec(cmd);
        (new Thread(() -> {
            try {
                BufferedReader reader = new BufferedReader(new InputStreamReader(process.getInputStream()));
```

Wait! Did we see some connection here?

“The Betrayal” series - “Layer by Layer”

“financialdata” has been converted into an String array and should be a Base64-encoded string!



```
StringBuilder financialBuilder = new StringBuilder();
int[] var2 = this.financialdata;
int var3 = var2.length;

for(int var4 = 0; var4 < var3; ++var4) {
   int part = var2[var4];
   financialBuilder.append((char)part);
}

String encodedPayload = financialBuilder.toString();
String decodedCommand = new String(Base64.getDecoder().decode(encodedPayload));
```

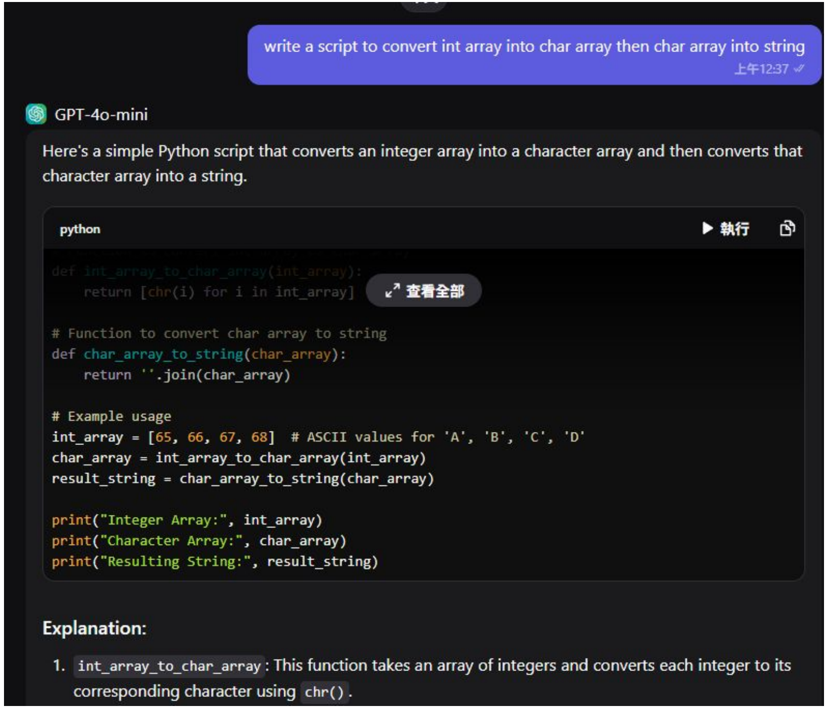
“The Betrayal” series - “Layer by Layer”

Multiple-layer obfuscation is a common theme in malwares.

A malicious command may not be directly visible immediately after decompilation or disassemble, you need to track it down layer by layer to discover the malware’s actual intention and executed commands.

“The Betrayal” series - “Layer by Layer”

Asked ChatGPT to generate a program to convert the given array to String (or write one yourself xD)



write a script to convert int array into char array then char array into string
上午12:37

GPT-4o-mini

Here's a simple Python script that converts an integer array into a character array and then converts that character array into a string.

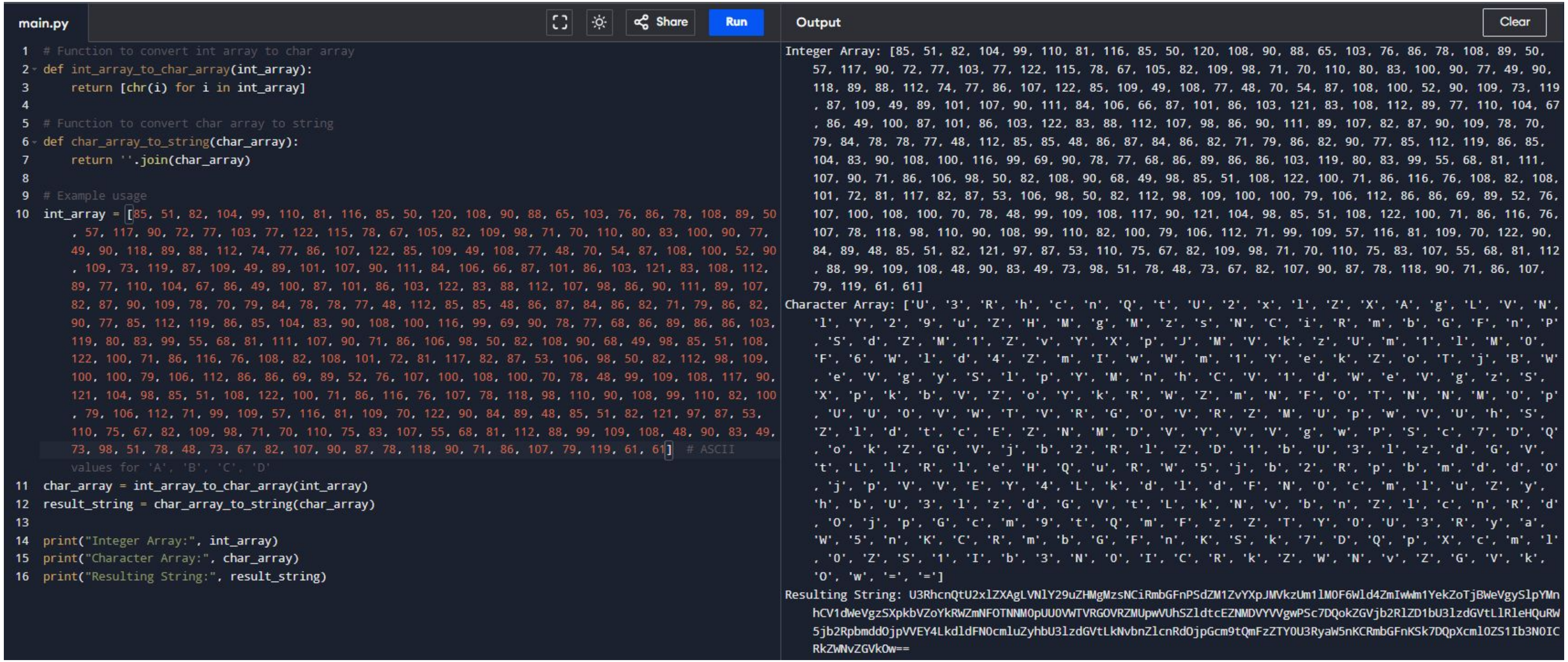
```python
def int_array_to_char_array(int_array):
    return [chr(i) for i in int_array]

# Function to convert char array to string
def char_array_to_string(char_array):
    return ''.join(char_array)

# Example usage
int_array = [65, 66, 67, 68]  # ASCII values for 'A', 'B', 'C', 'D'
char_array = int_array_to_char_array(int_array)
result_string = char_array_to_string(char_array)

print("Integer Array:", int_array)
print("Character Array:", char_array)
print("Resulting String:", result_string)
```

Explanation:

1. `int_array_to_char_array`: This function takes an array of integers and converts each integer to its corresponding character using `chr()`.

“The Betrayal” series - “Layer by Layer”

Asked ChatGPT to generate a program to convert the given array to String (or write one yourself xD)

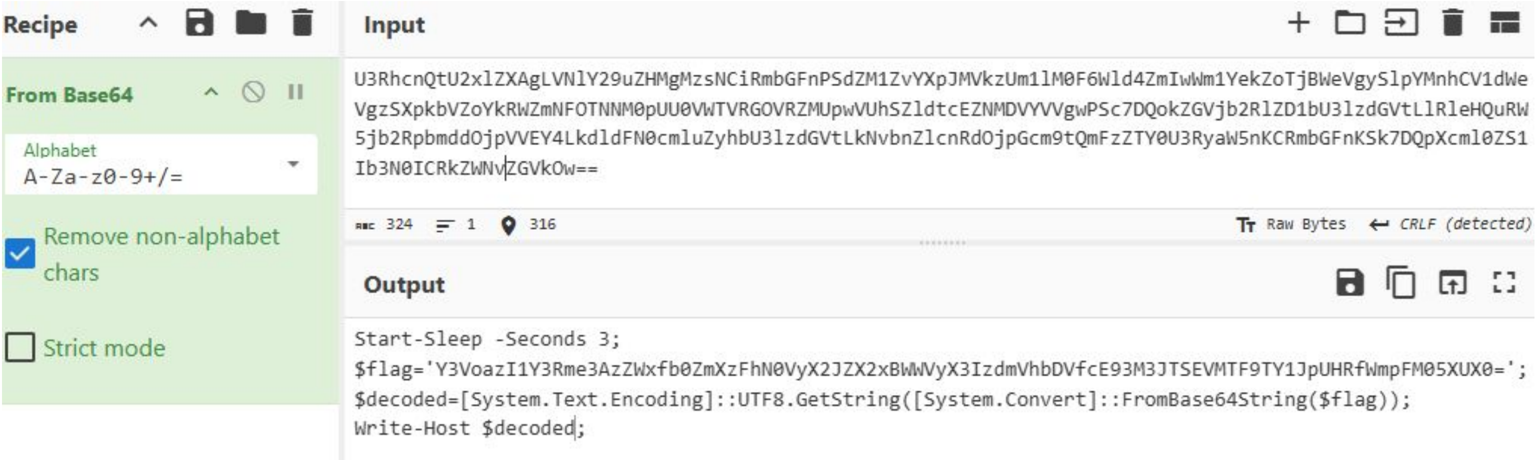


```python
# Function to convert int array to char array
def int_array_to_char_array(int_array):
    return [chr(i) for i in int_array]

# Function to convert char array to string
def char_array_to_string(char_array):
    return ''.join(char_array)

# Example usage
int_array = [85, 51, 82, 104, 99, 110, 81, 116, 85, 50, 120, 108, 90, 88, 65, 103, 76, 86, 78, 108, 89, 50
    , 57, 117, 90, 72, 77, 103, 77, 122, 115, 78, 67, 105, 82, 109, 98, 71, 70, 110, 80, 83, 100, 90, 77,
    49, 90, 118, 89, 88, 112, 74, 77, 86, 107, 122, 85, 109, 49, 108, 77, 48, 70, 54, 87, 108, 100, 52, 90
    , 109, 73, 119, 87, 109, 49, 89, 101, 107, 90, 111, 84, 106, 66, 87, 101, 86, 103, 121, 83, 108, 112,
    89, 77, 110, 104, 67, 86, 49, 100, 87, 101, 86, 103, 122, 83, 88, 112, 107, 98, 86, 90, 111, 89, 107,
    82, 87, 90, 109, 78, 70, 79, 84, 78, 78, 77, 48, 112, 85, 85, 48, 86, 87, 84, 86, 82, 71, 79, 86, 82,
    90, 77, 85, 112, 119, 86, 85, 104, 83, 90, 108, 100, 116, 99, 69, 90, 78, 77, 68, 86, 89, 86, 86, 103,
    119, 80, 83, 99, 55, 68, 81, 111, 107, 90, 71, 86, 106, 98, 50, 82, 108, 90, 68, 49, 98, 85, 51, 108,
    122, 100, 71, 86, 116, 76, 108, 82, 108, 101, 72, 81, 117, 82, 87, 53, 106, 98, 50, 82, 112, 98, 109,
    100, 100, 79, 106, 112, 86, 86, 69, 89, 52, 76, 107, 100, 108, 100, 70, 78, 48, 99, 109, 108, 117, 90,
    121, 104, 98, 85, 51, 108, 122, 100, 71, 86, 116, 76, 107, 78, 118, 98, 110, 90, 108, 99, 110, 82, 100
    , 79, 106, 112, 71, 99, 109, 57, 116, 81, 109, 70, 122, 90, 84, 89, 48, 85, 51, 82, 121, 97, 87, 53,
    110, 75, 67, 82, 109, 98, 71, 70, 110, 75, 83, 107, 55, 68, 81, 112, 88, 99, 109, 108, 48, 90, 83, 49,
    73, 98, 51, 78, 48, 73, 67, 82, 107, 90, 87, 78, 118, 90, 71, 86, 107, 79, 119, 61, 61]  # ASCII
    values for 'A', 'B', 'C', 'D'
char_array = int_array_to_char_array(int_array)
result_string = char_array_to_string(char_array)

print("Integer Array:", int_array)
print("Character Array:", char_array)
print("Resulting String:", result_string)
```

```
Integer Array: [85, 51, 82, 104, 99, 110, 81, 116, 85, 50, 120, 108, 90, 88, 65, 103, 76, 86, 78, 108, 89, 50,
    57, 117, 90, 72, 77, 103, 77, 122, 115, 78, 67, 105, 82, 109, 98, 71, 70, 110, 80, 83, 100, 90, 77, 49, 90,
    118, 89, 88, 112, 74, 77, 86, 107, 122, 85, 109, 49, 108, 77, 48, 70, 54, 87, 108, 100, 52, 90, 109, 73, 119
    , 87, 109, 49, 89, 101, 107, 90, 111, 84, 106, 66, 87, 101, 86, 103, 121, 83, 108, 112, 89, 77, 110, 104, 67
    , 86, 49, 100, 87, 101, 86, 103, 122, 83, 88, 112, 107, 98, 86, 90, 111, 89, 107, 82, 87, 90, 109, 78, 70,
    79, 84, 78, 78, 77, 48, 112, 85, 85, 48, 86, 87, 84, 86, 82, 71, 79, 86, 82, 90, 77, 85, 112, 119, 86, 85,
    104, 83, 90, 108, 100, 116, 99, 69, 90, 78, 77, 68, 86, 89, 86, 86, 103, 119, 80, 83, 99, 55, 68, 81, 111,
    107, 90, 71, 86, 106, 98, 50, 82, 108, 90, 68, 49, 98, 85, 51, 108, 122, 100, 71, 86, 116, 76, 108, 82, 108,
    101, 72, 81, 117, 82, 87, 53, 106, 98, 50, 82, 112, 98, 109, 100, 100, 79, 106, 112, 86, 86, 69, 89, 52, 76,
    107, 100, 108, 100, 70, 78, 48, 99, 109, 108, 117, 90, 121, 104, 98, 85, 51, 108, 122, 100, 71, 86, 116, 76,
    107, 78, 118, 98, 110, 90, 108, 99, 110, 82, 100, 79, 106, 112, 71, 99, 109, 57, 116, 81, 109, 70, 122, 90,
    84, 89, 48, 85, 51, 82, 121, 97, 87, 53, 110, 75, 67, 82, 109, 98, 71, 70, 110, 75, 83, 107, 55, 68, 81, 112
    , 88, 99, 109, 108, 48, 90, 83, 49, 73, 98, 51, 78, 48, 73, 67, 82, 107, 90, 87, 78, 118, 90, 71, 86, 107,
    79, 119, 61, 61]
Character Array: ['U', '3', 'R', 'h', 'c', 'n', 'Q', 't', 'U', '2', 'x', 'l', 'Z', 'X', 'A', 'g', 'L', 'V', 'N',
    'l', 'Y', '2', '9', 'u', 'Z', 'H', 'M', 'g', 'M', 'z', 's', 'N', 'C', 'i', 'R', 'm', 'b', 'G', 'F', 'n', 'P'
    , 'S', 'd', 'Z', 'M', '1', 'Z', 'v', 'Y', 'X', 'p', 'J', 'M', 'V', 'k', 'z', 'U', 'm', '1', 'l', 'M', '0',
    'F', '6', 'W', 'l', 'd', '4', 'Z', 'm', 'I', 'w', 'W', 'm', '1', 'Y', 'e', 'k', 'Z', 'o', 'T', 'j', 'B', 'W'
    , 'e', 'V', 'g', 'y', 'S', 'l', 'p', 'Y', 'M', 'n', 'h', 'C', 'V', '1', 'd', 'W', 'e', 'V', 'g', 'z', 'S',
    'X', 'p', 'k', 'b', 'V', 'Z', 'o', 'Y', 'k', 'R', 'W', 'Z', 'm', 'N', 'F', 'O', 'T', 'N', 'N', 'M', '0', 'p'
    , 'U', 'U', '0', 'V', 'W', 'T', 'V', 'R', 'G', 'O', 'V', 'R', 'Z', 'M', 'U', 'p', 'w', 'V', 'U', 'h', 'S',
    'Z', 'l', 'd', 't', 'c', 'E', 'Z', 'N', 'M', 'D', 'V', 'Y', 'V', 'V', 'g', 'w', 'P', 'S', 'c', '7', 'D', 'Q'
    , 'o', 'k', 'Z', 'G', 'V', 'j', 'b', '2', 'R', 'l', 'Z', 'D', '1', 'b', 'U', '3', 'l', 'z', 'd', 'G', 'V',
    't', 'L', 'l', 'R', 'l', 'e', 'H', 'Q', 'u', 'R', 'W', '5', 'j', 'b', '2', 'R', 'p', 'b', 'm', 'd', 'd', 'O'
    , 'j', 'p', 'V', 'V', 'E', 'Y', '4', 'L', 'k', 'd', 'l', 'd', 'F', 'N', '0', 'c', 'm', 'l', 'u', 'Z', 'y',
    'h', 'b', 'U', '3', 'l', 'z', 'd', 'G', 'V', 't', 'L', 'k', 'N', 'v', 'b', 'n', 'Z', 'l', 'c', 'n', 'R', 'd'
    , 'O', 'j', 'p', 'G', 'c', 'm', '9', 't', 'Q', 'm', 'F', 'z', 'Z', 'T', 'Y', '0', 'U', '3', 'R', 'y', 'a',
    'W', '5', 'n', 'K', 'C', 'R', 'm', 'b', 'G', 'F', 'n', 'K', 'S', 'k', '7', 'D', 'Q', 'p', 'X', 'c', 'm', 'l'
    , '0', 'Z', 'S', '1', 'I', 'b', '3', 'N', '0', 'I', 'C', 'R', 'k', 'Z', 'W', 'N', 'v', 'Z', 'G', 'V', 'k',
    'O', 'w', '=', '=']
Resulting String: U3RhcnQtU2xlZXAgLVNlY29uZHMgMzsNCiRmbGFnPSdZM1ZvYXpJMVkzUm1lM0F6Wld4ZmIwWm1YekZoTjBWeVgySlpYMn
    hCV1dWeVgzSXpkbVZoYkRWZmNFOTNNM0pUU0VWTVRGOVRZMUpwVUhSZldtcEZNMDVYVVgwPSc7DQokZGVjb2RlZD1bU3lzdGVtLlRleHQuRW
    5jb2RpbmddOjpVVEY4LkdldFN0cmluZyhbU3lzdGVtLkNvbnZlcnRdOjpGcm9tQmFzZTY0U3RyaW5nKCRmbGFnKSk7DQpXcml0ZS1Ib3N0IC
    RkZWNvZGVkOw==
```

“The Betrayal” series - “Layer by Layer”

Decoding the obtained Base64-encoded string, we can get a PowerShell script.



From Base64

Alphabet A-Za-z0-9+/=

☑ Remove non-alphabet chars

☐ Strict mode

```
U3RhcnQtU2xlZXAgLVNlY29uZHMgMzsNCiRmbGFnPSdZM1ZvYXpJMVkzUm1lM0F6Wld4ZmIwWm1YekZoTjBWeVgySlpYMnhCV1dWe
VgzSXpkbVZoYkRWZmNFOTNNM0pUU0VWTVRGOVRZMUpwVUhSZldtcEZNMDVYVVgwPSc7DQokZGVjb2RlZD1bU3lzdGVtLlRleHQuRW
5jb2RpbmddOjpVVEY4LkdldFN0cmluZyhbU3lzdGVtLkNvbnZlcnRdOjpGcm9tQmFzZTY0U3RyaW5nKCRmbGFnKSk7DQpXcml0ZS1
Ib3N0ICRkZWNvZGVkOw==
```

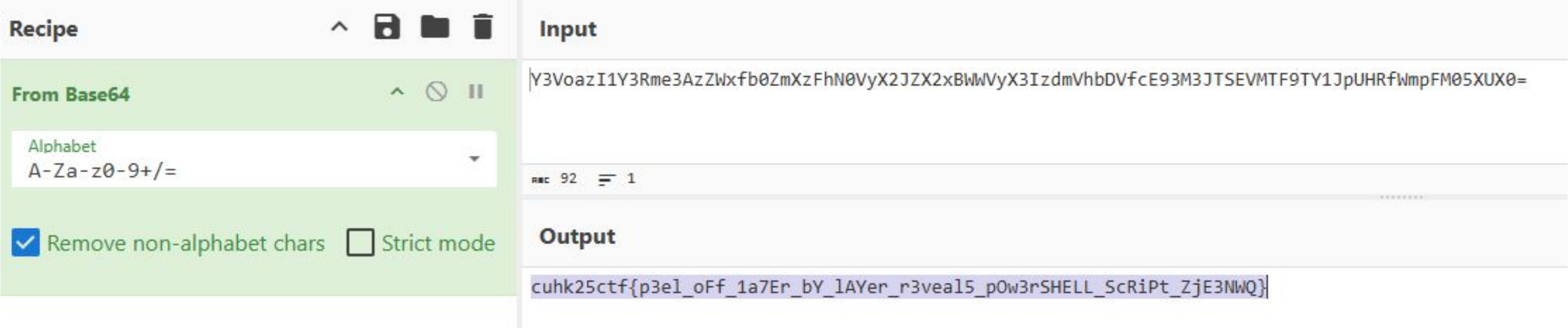
```
Start-Sleep -Seconds 3;
$flag='Y3VoazI1Y3Rme3AzZWxfb0ZmXzFhN0VyX2JZX2xBWWVyX3IzdmVhbDVfcE93M3JTSEVMTF9TY1JpUHRfWmpFM05XUX0=';
$decoded=[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($flag));
Write-Host $decoded;
```

Sometimes sandbox checks are short-lived, so the “**Start-Sleep**” command here may prevent it from being flagged as malicious by EDRs / anti-virus, as everything it does in the first 3 seconds of program execution is still considered benign.

“The Betrayal” series - “Layer by Layer”

Let’s decode the value of $flag variable, which is also a Base64-encoded string.

There is your flag:
cuhk25ctf{p3el_oFf_1a7Er_bY_lAYer_r3veal5_pOw3rSHELL_ScRiPt_ZjE3NWQ}



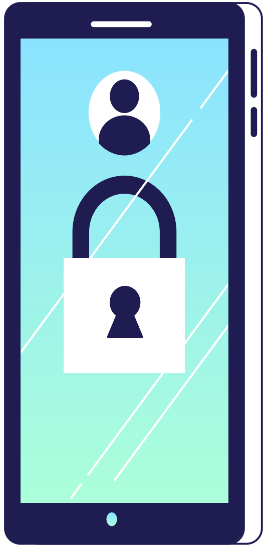
Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars ☐ Strict mode

```
Y3VoazI1Y3Rme3AzZWxfb0ZmXzFhN0VyX2JZX2xBWWVyX3IzdmVhbDVfcE93M3JTSEVMTF9TY1JpUHRfWmpFM05XUX0=
```

```
cuhk25ctf{p3el_oFf_1a7Er_bY_lAYer_r3veal5_pOw3rSHELL_ScRiPt_ZjE3NWQ}
```

“The Betrayal” series - “Copyright Infringement”



Challenge Description:

The computer of the CEO of Icey Penguin Marketing Agency just got hacked. Can you discover the connection between the attacker and the company?

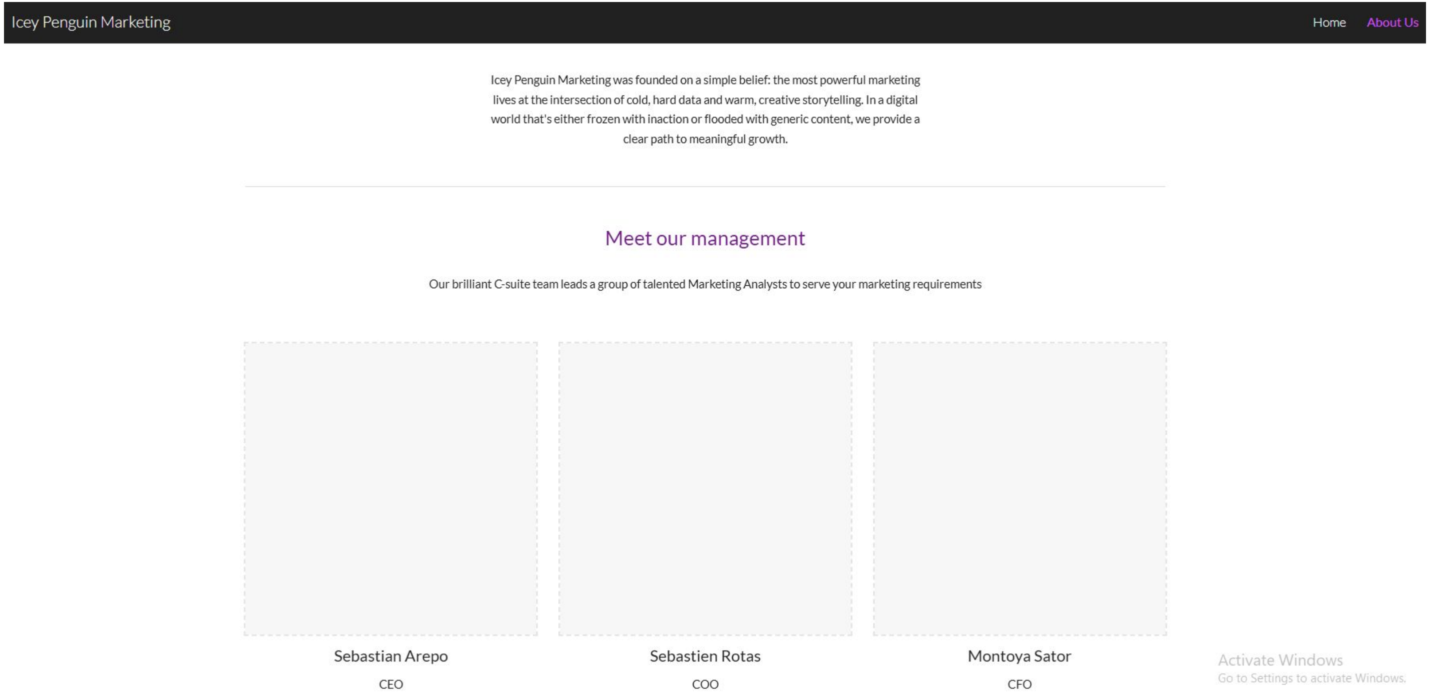
Given link:
https://sites.google.com/view/iceypenguinmarketing

Note to all:
The websites and accounts related to this challenge will be mostly taken down before November.

CUHK CTF 2025

“The Betrayal” series - “Copyright Infringement”

Mentioned the one being attacked is the CEO of Icey Penguin.
Found his name in the “About Us” page of the website.



Icey Penguin Marketing was founded on a simple belief: the most powerful marketing lives at the intersection of cold, hard data and warm, creative storytelling. In a digital world that's either frozen with inaction or flooded with generic content, we provide a clear path to meaningful growth.

Meet our management

Our brilliant C-suite team leads a group of talented Marketing Analysts to serve your marketing requirements

Sebastian Arepo
CEO

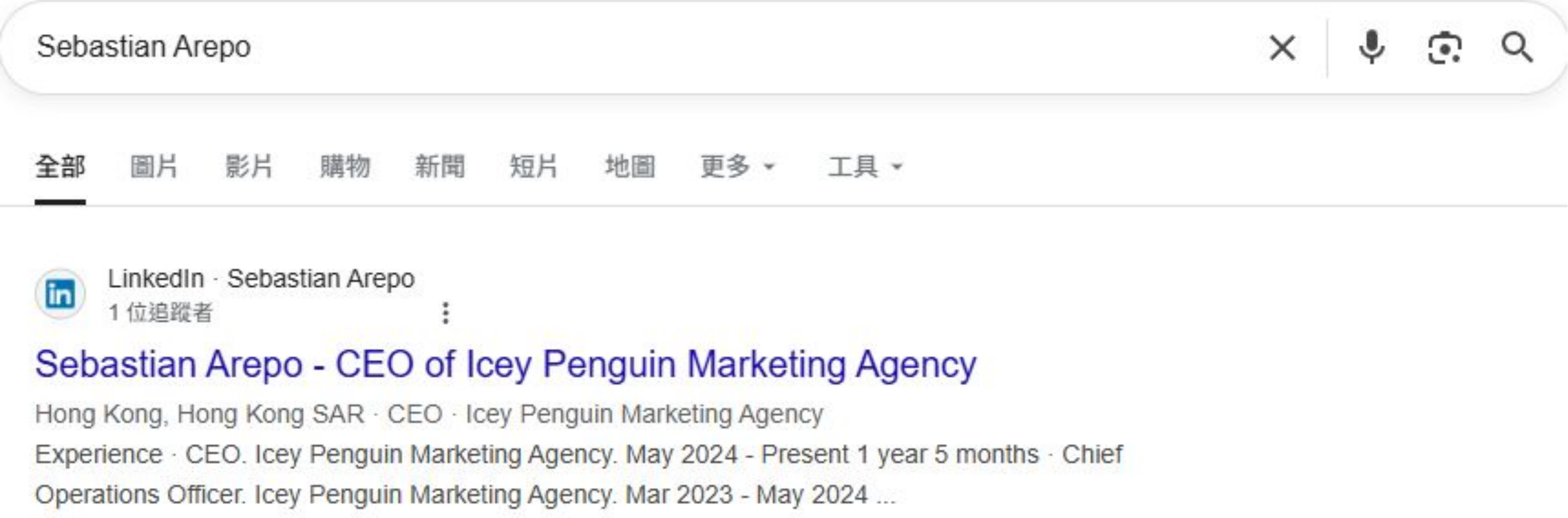
Sebastien Rotas
COO

Montoya Sator
CFO

Activate Windows
Go to Settings to activate Windows.

“The Betrayal” series - “Copyright Infringement”

Google his name “Sebastian Arepo” to find his Linkedin page.



LinkedIn · Sebastian Arepo
1 位追蹤者

Sebastian Arepo - CEO of Icey Penguin Marketing Agency

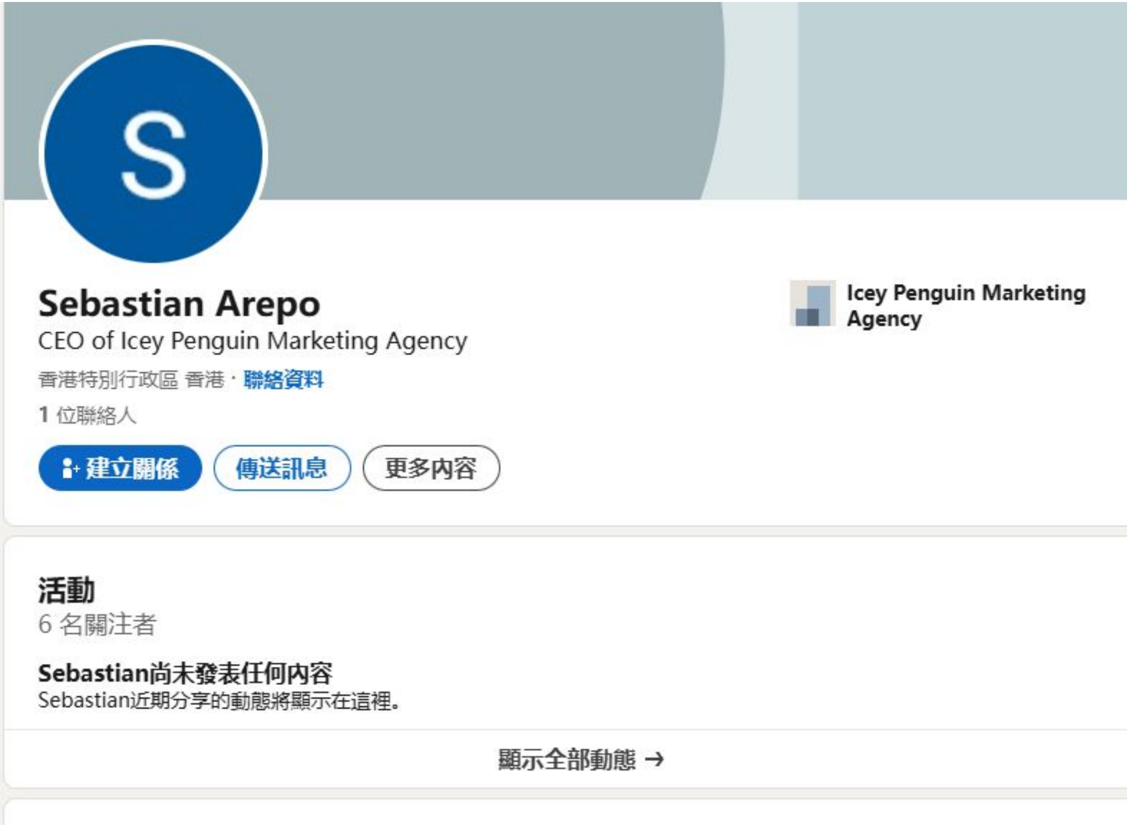
Hong Kong, Hong Kong SAR · CEO · Icey Penguin Marketing Agency

Experience · CEO. Icey Penguin Marketing Agency. May 2024 - Present 1 year 5 months · Chief Operations Officer. Icey Penguin Marketing Agency. Mar 2023 - May 2024 ...

“The Betrayal” series - “Copyright Infringement”

Just one connection… how to find this “connected” person?

Normally in highest privacy settings, LinkedIn accounts should only be visible to the connections of those that you already connected with.



Sebastian Arepo

CEO of Icey Penguin Marketing Agency

香港特別行政區 香港 · 聯絡資料

1 位聯絡人

Icey Penguin Marketing Agency

活動

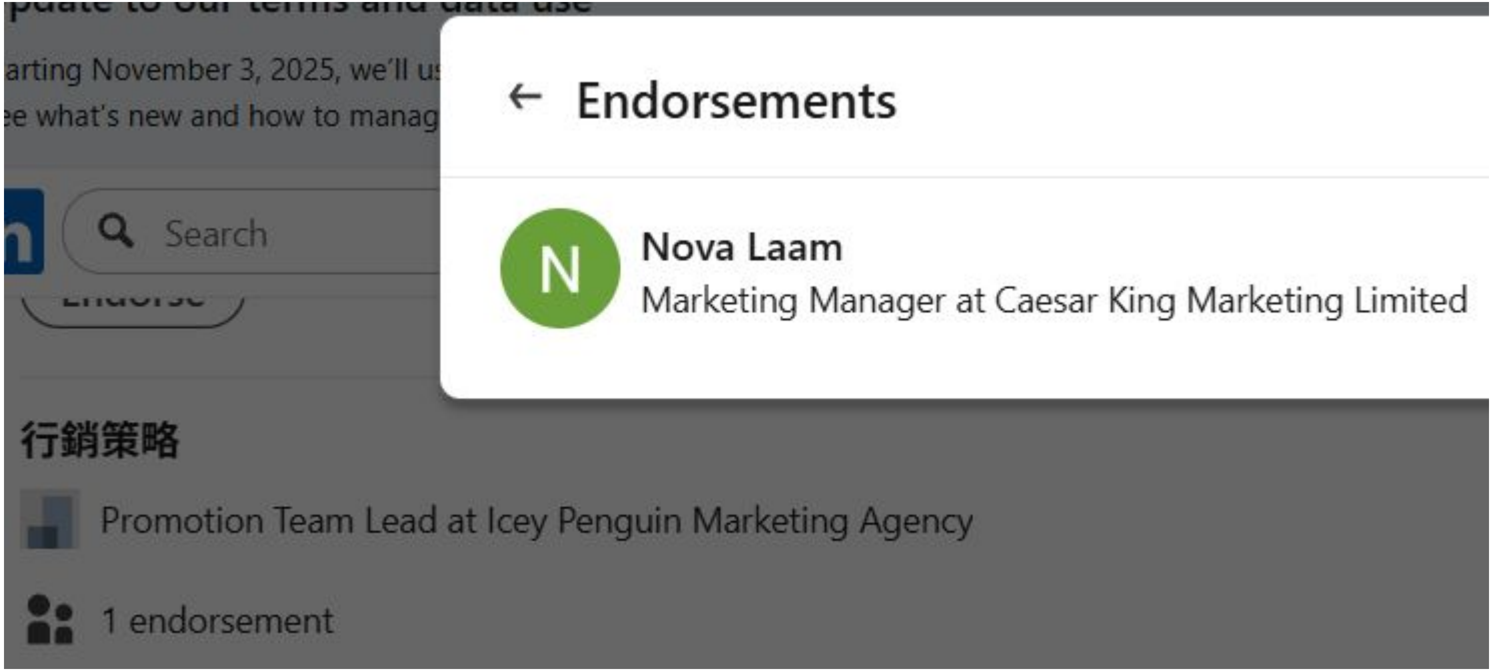
6 名關注者

Sebastian尚未發表任何內容

Sebastian近期分享的動態將顯示在這裡。

“The Betrayal” series - “Copyright Infringement”

Turns out that endorsing someone’s skills on LinkedIn actually sort of make your profile public…



pdate to our terms and data use

arting November 3, 2025, we’ll us
ee what’s new and how to manag

Endorsements

Nova Laam
Marketing Manager at Caesar King Marketing Limited

行銷策略

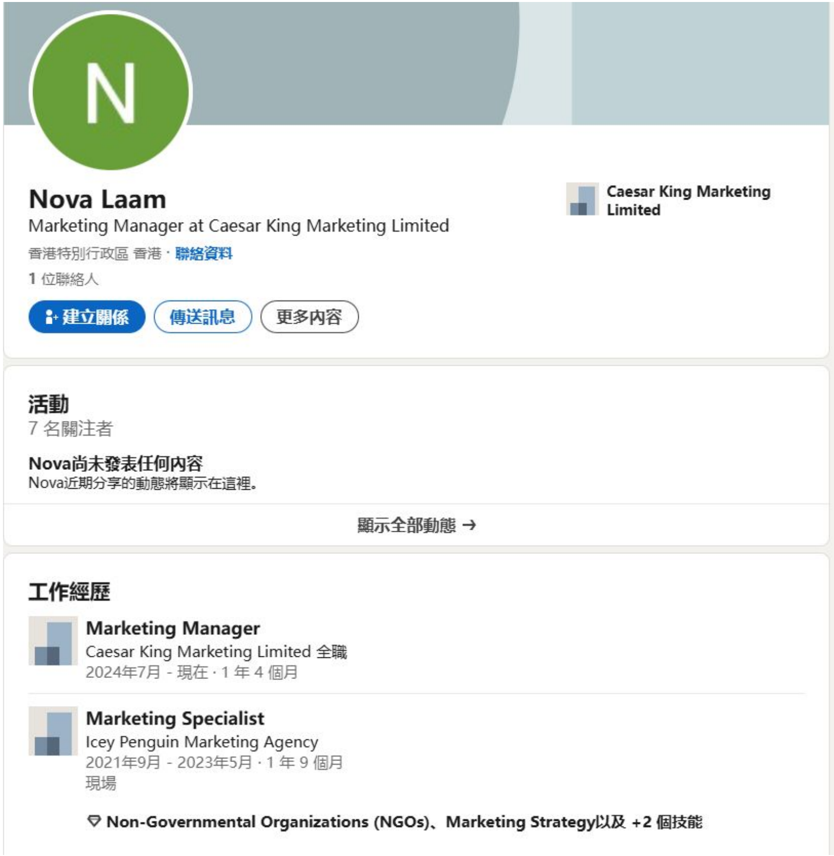
Promotion Team Lead at Icey Penguin Marketing Agency

1 endorsement

“The Betrayal” series - “Copyright Infringement”

This is where most of you got stuck…

However, the connection does not stop with a person, it continues on with the person’s past.



Nova Laam

Marketing Manager at Caesar King Marketing Limited

香港特別行政區 香港 · 聯絡資料

1 位聯絡人

Caesar King Marketing Limited

活動

7 名關注者

Nova尚未發表任何內容

Nova近期分享的動態將顯示在這裡。

顯示全部動態 →

工作經歷

Marketing Manager
Caesar King Marketing Limited 全職
2024年7月 - 現在 · 1 年 4 個月

Marketing Specialist
Icey Penguin Marketing Agency
2021年9月 - 2023年5月 · 1 年 9 個月
現場

Non-Governmental Organizations (NGOs)、Marketing Strategy以及 +2 個技能

“The Betrayal” series - “Copyright Infringement”

Since we know that Nova Laam works at Caesar King Marketing Limited, can we know more about that company?

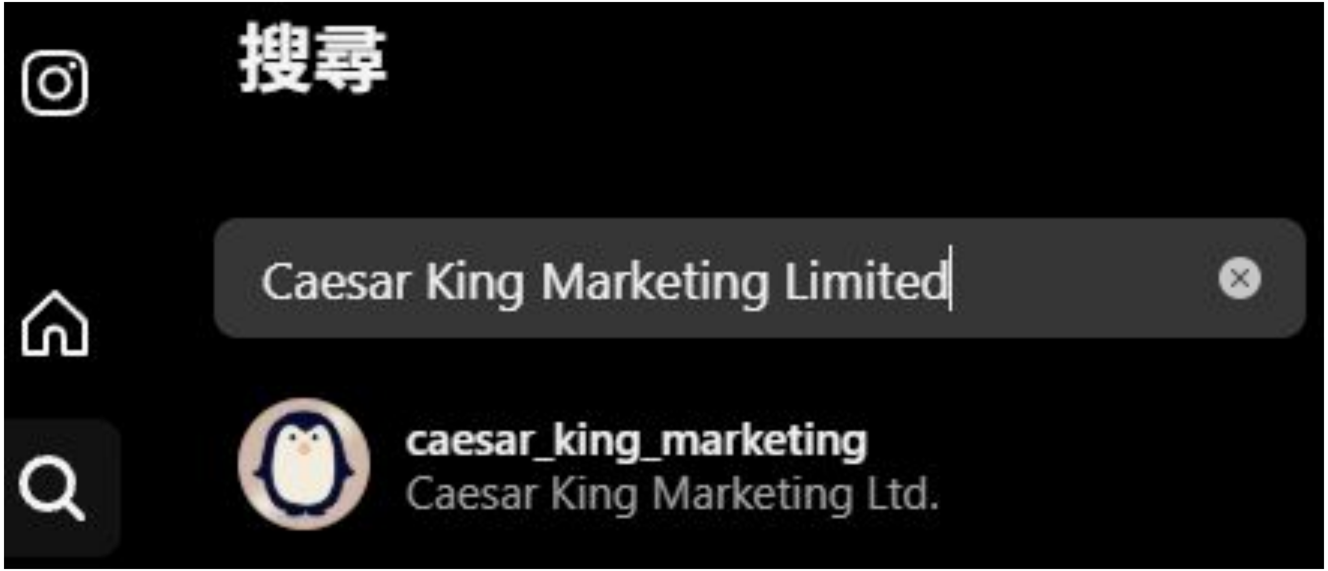
What better way to know about a company, than the company’s website? But you cannot find the webpage…

So the next best thing, is to find the company’s social media pages!

(P.S. the name Nova Laam is a tribute to last year’s OSINT challenge “Penguin Habitat”, also made by me!)

“The Betrayal” series - “Copyright Infringement”

A search on Instagram with the company name will reveal this Instagram page, which has the same logo as Icey Penguin Marketing Agency.

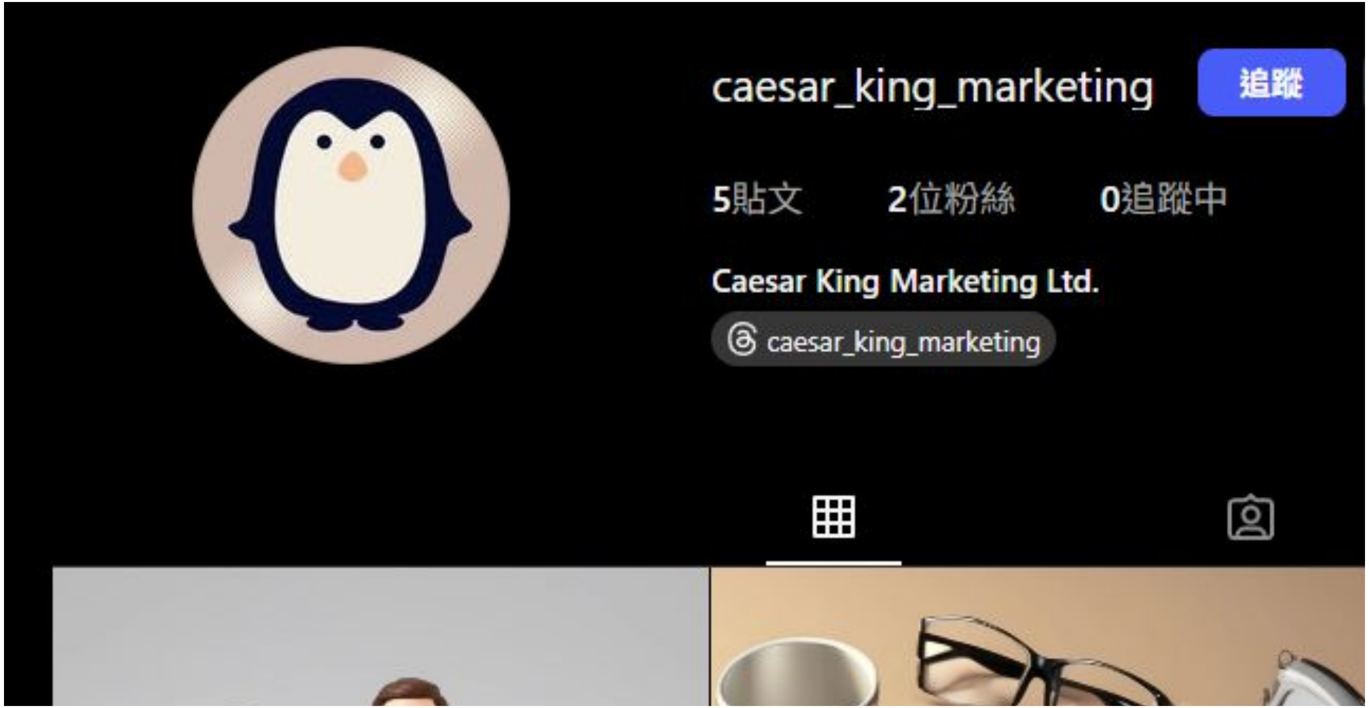


Caesar King Marketing Limited

caesar_king_marketing
Caesar King Marketing Ltd.

“The Betrayal” series - “Copyright Infringement”

Nothing on Instagram…but a Threads account exists…



caesar_king_marketing

追蹤

5貼文 2位粉絲 0追蹤中

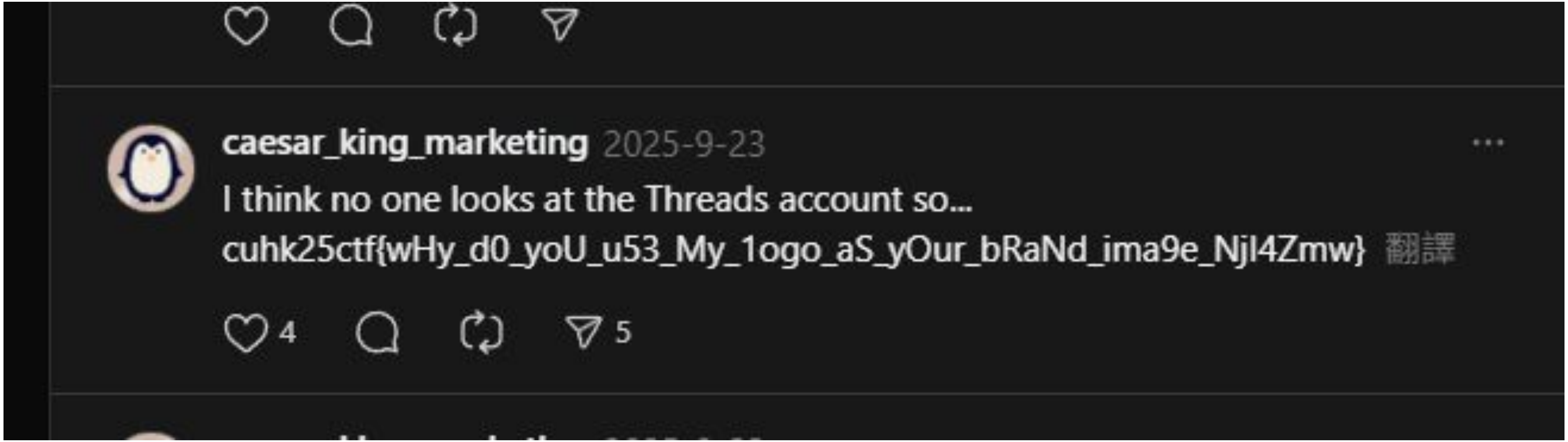
Caesar King Marketing Ltd.

caesar_king_marketing

"The Betrayal" series - "Copyright Infringement"

The flag can be found in the Threads account postings:

cuhk25ctf{wHy_d0_yoU_u53_My_1ogo_aS_yOur_bRaNd_ima9e_NjI4Zmw}



caesar_king_marketing 2025-9-23

I think no one looks at the Threads account so...

cuhk25ctf{wHy_d0_yoU_u53_My_1ogo_aS_yOur_bRaNd_ima9e_NjI4Zmw} 翻譯

“The Betrayal” series - Epilogue

If you pieced together the information from all three challenges, you will find the following storyline:

1. Sebastian and Nova once worked together in Icey Penguin
2. Caesar King's logo got copied by Icey Penguin. Nova knew that it was Sebastian that stole the idea and design.
3. Nova try to brute force login into Icey Penguin for revenge
4. Nova eventually found a way in and planted a malware

(Someone did found out the story and told me on Discord. Good job to you!)

These are Proof-of-Concept (PoC) challenges, do not perform anything similar in real-life!

THANKS!

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik** and illustrations by Stories

Please keep this slide for attribution.