

**Nama Tim : USER2**

**Nama Sekolah : Smk Negeri 1 Lahat**

## **AES biasalah**

Apa dampak Key jika tidak dirahasiakan ?

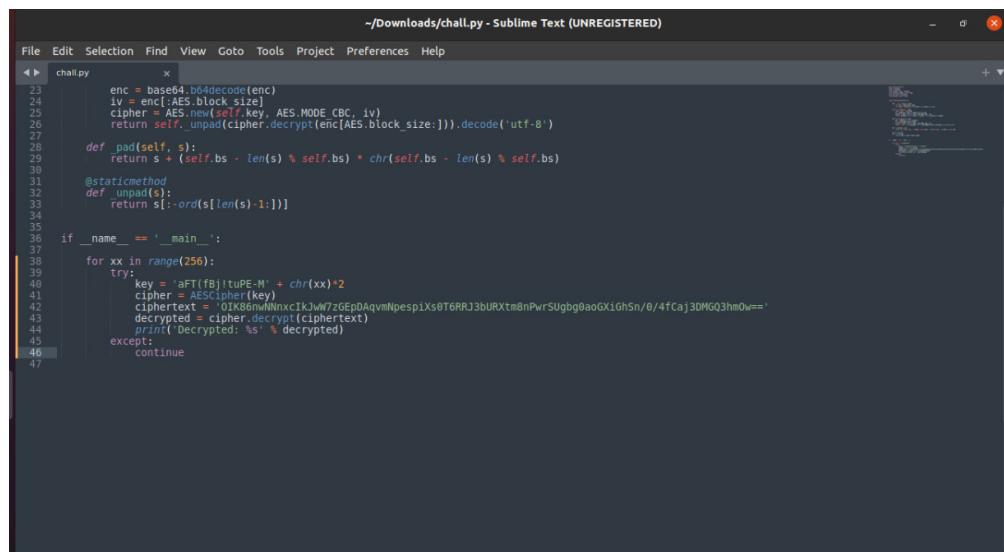
- Dampaknya akan lebih mudah untuk di brute force  
kami menemukan sebuah key di file chall.py

```
if __name__ == '__main__':
    xx = urandom(1)
    key = b'aFT(fBj!tuPE-M' + xx*2
    print(key)
    cipher = AESCipher(key.decode())
```

Key yang dibutuhkan untuk melakukan enkripsi pada soal ini berjumlah 16 byte / 16 Karakter

- Dibawah ini adalah isi dari file chall.py yang di berikan. disana kami menemukan sebuah key dan kami lakukan enskrip yang menghasilkan sebuah ciphertext , dan sesuai dengan ciphertext tersebut kami deskrip , disini kami menggunakan tools pycrypto.

Setelah itu lalu kami ubah File chall.py yang seharusnya berfungsi untuk enskrip kami ubah agar bisa men deskrip ciphertext



The screenshot shows a Sublime Text editor window with the file 'chall.py' open. The code has been modified to include a try-except block that catches a 'ValueError' and continues the loop. The original code from the previous slide is present at the top, followed by the modifications:

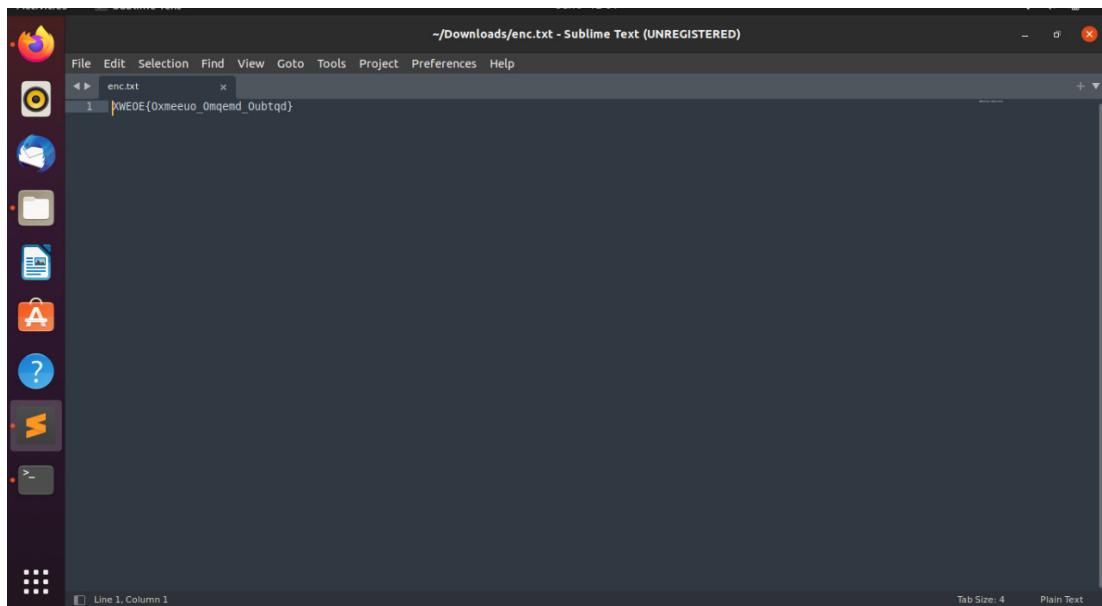
```
if __name__ == '__main__':
    for xx in range(256):
        try:
            key = b'aFT(fBj!tuPE-M' + chr(xx)*2
            cipher = AESCipher(key)
            ciphertext = '0IK86nNNnxCIKjwWzgEp0AqvmNpesp1xs0T6RRJ3bURXtm8nPwrSUgbg9aoGX1ghSn/0/4fCaj3DMGQ3hm0w=='
            decrypted = cipher.decrypt(ciphertext)
            print('Decrypted: ', decrypted)
        except:
            continue
```

Setelah mengedit file lalu kami jalankan Program/tools tersebut di terminal, dengan cara :

Disana ditemukan sebuah flag untuk menjawab Soal di AES biasalah.  
Flag : LKSCS{u\_can\_brute\_2\_byte\_secret\_key}

# **Caesar Chiper**

-Ini adalah isi dari file yang di berikan.



-Setelah itu dideskripsikan menggunakan website

<http://rumkin.com/tools/cipher/caesar.php>

-Berdasarkan soal yang telah dikerjakan ciri khas flag diawali dengan LKSCS. Jadi setelah dicoba pergeseran dari 0 sampai 25 maka ditemukan angka yang cocok dengan ciri khas flag yaitu angka 14 atau N menjadi = 14 , dan paste isi file enc.txt

This is a standard Caesarian Shift cipher encoder, also known as a rot-N encoder and is also a style of substitution cipher. This way, you can add one, two, or any number up to 25 to your string and see how it changes. This is an offshoot of the ROT13 encoder on this web site. To perform this shift by hand, you could just write the alphabet on two strips of paper. Line them up so the top strip's A matches the bottom strip's D (or something) and then you can encode. A simple test to see how this works would be to [insert the alphabet](#) into the encoder and then change the values of N.

This sort of cipher can also be known as a wheel cipher. This is where an inner wheel has the alphabet around the outside, and that is placed upon an outer wheel, also with the alphabet going around it. You can rotate the wheels so that ABC lines up with ABC, or ABC may line up with QRS.

To encode something, just pick an N and type in your message. To decode something, subtract the encryption N from 26 and it should be decoded for you.

N: 14

XWEOE(Xmesue\_Quneed\_Dubted)

This is your encoded or decoded text:

LKSCS(Classic\_Caesar\_Cipher)

INDEX

- Affine
- Atbash
- Baconian
- Base64
- Bifid
- Caesar
  - Keyed
  - ROT13
- Column Trans.
- Double
- Übchi
- Cryptogram
- Gronsfeld
- Morse
- Numbers
- One Time Pad
- Playfair
- Railfence
- Rotate
- Skip
- Substitution
- Vigenere
  - Keyed
  - Autokey
- Crypto Solver
- Frequency Manipulator

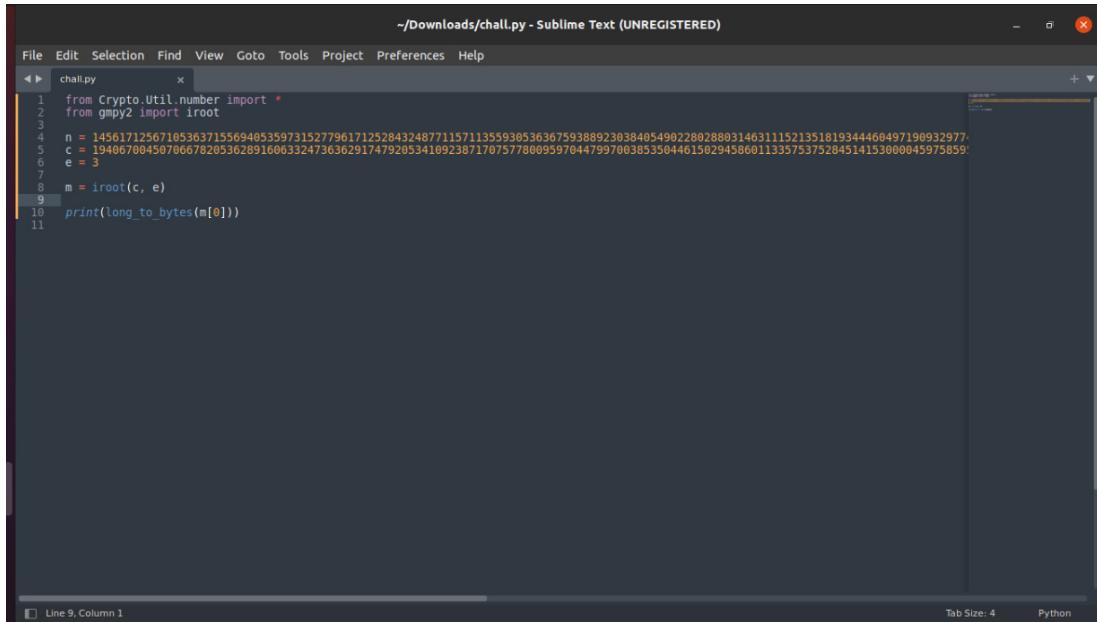
Flag : LKSCS{Classic\_Caesar\_Cipher}

## RSA Small e

-Ini adalah isi file chall.py disana telah diberitahu nilai n dan c.

```
#!/usr/bin/python3
from Crypto.Util.number import *
from secret import flag
m = bytes_to_long(flag)
e = 3
p = getRandomNBitInteger(512)
q = getRandomNBitInteger(512)
n = p*q
c = pow(m, e, n)
print("n : {}".format(n))
print("c : {}".format(c))
# n : 1456171256710536371556940535973152779617125284324877115711355930536367593889230384054902280288031463111521351819344604971909329
# c : 19406700456786678205362891606332473636291747920534109238717075778009597044799700385356446156294586011335753752845141530000459758
```

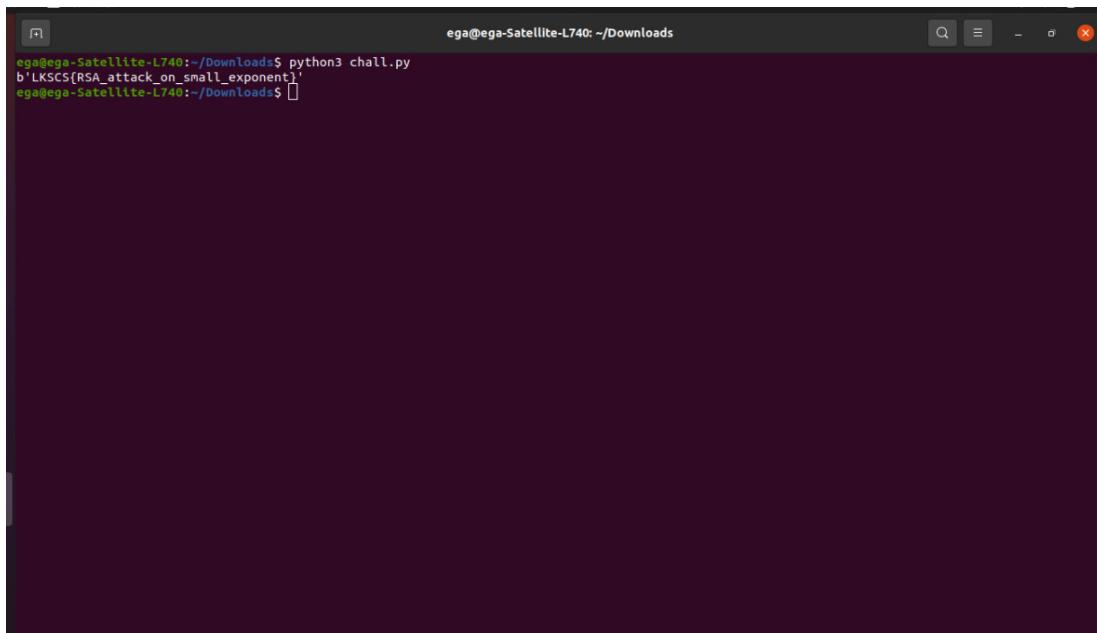
-Setelah nilai n dan c diketahui selanjutnya membuat program untuk mencari flag dengan tools gmpy2 dengan fungsi iroot.



The screenshot shows a Sublime Text window with the title bar "-/Downloads/chall.py - Sublime Text (UNREGISTERED)". The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. A tab bar at the bottom shows "Line 9, Column 1" and "Tab Size: 4". The code editor contains the following Python script:

```
1 from Crypto.Util.number import *
2 from gmpy2 import iroot
3
4 n = 1456171256710536371556940535973152779617125284324877115711355938536367593889230384054902288028803146311152135181934468497190932977
5 c = 194067004507066782053628916063324736362917479205341092387170757780959704479970638535044615029458601133575375284514153000045975859
6 e = 3
7
8 m = iroot(c, e)
9
10 print(long_to_bytes(m[0]))
11
```

-Berikut adalah hasil run dari program yang dibuat mencari flag.



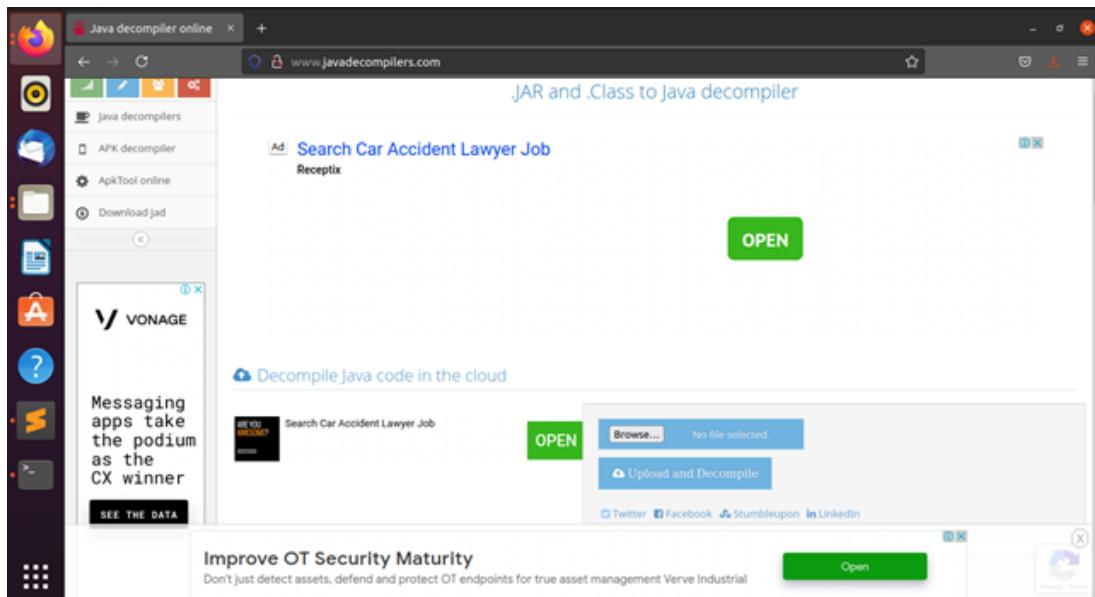
The screenshot shows a terminal window with the title bar "ega@ega-Satellite-L740: ~/Downloads". The command entered is "python3 chall.py". The output is: "b'LKSCS{RSA\_attack\_on\_small\_exponent}'". The terminal prompt is "ega@ega-Satellite-L740: ~/Downloads\$".

```
ega@ega-Satellite-L740:~/Downloads$ python3 chall.py
b'LKSCS{RSA_attack_on_small_exponent}'
ega@ega-Satellite-L740:~/Downloads$
```

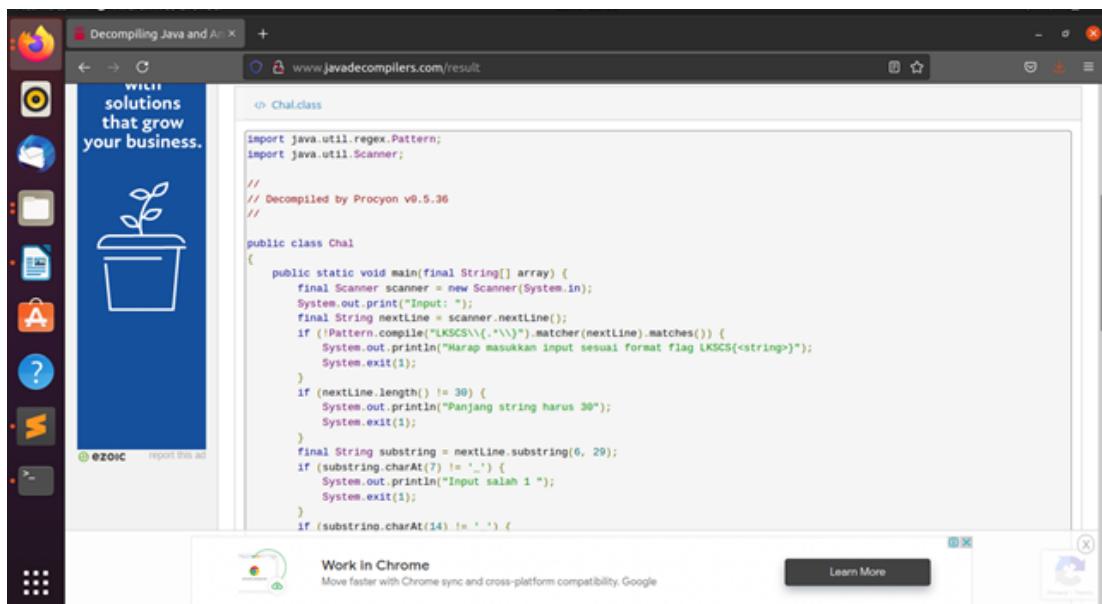
Flag :LKSCS{RSA\_attack\_on\_small\_exponent}

## Jawa

- Setelah mendownload file Chal.class yg diberikan di deskripsi soal selanjutnya buka website [javadecompilers.com](http://javadecompilers.com) untuk decompile file Chal.class
- Kemudian pilih menu browse untuk memasukkan file Chal.class ke website tersebut
- Lalu pilih menu Upload and Decompile untuk melakukan decompile pada file Chal.class

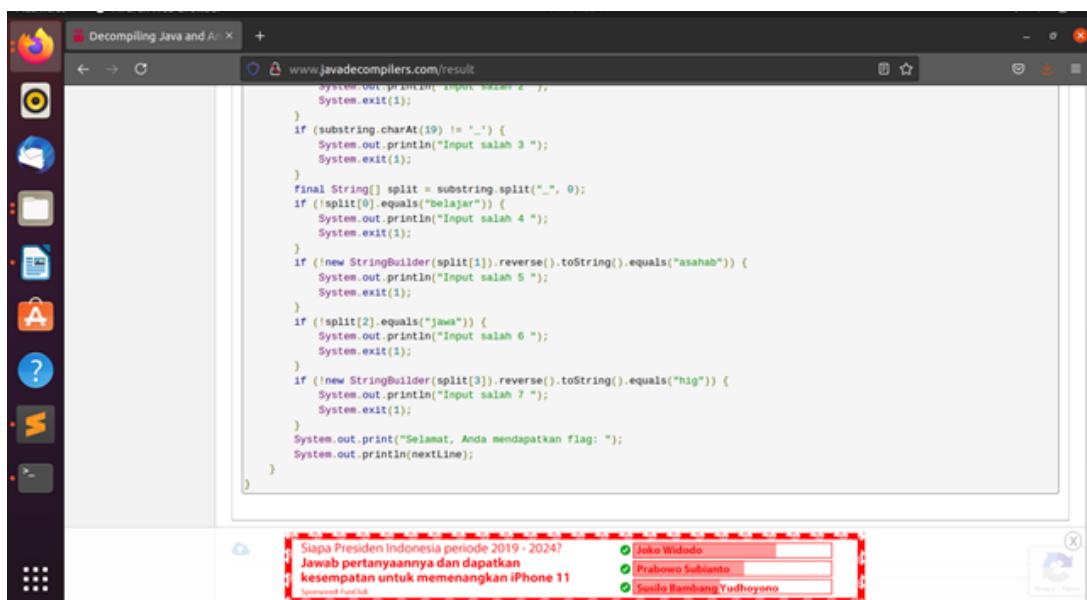


-Setelah di decompile maka akan tampil sebagai berikut:



-Dari hasil decompile tersebut terlihat sesuatu yg mencurigakan seperti terdapat kata "Harap masukkan input sesuai format flag LKSCS{<string>}  
" dan tanda "\_"

-Lalu sesuatu yg mencurigakan juga terdapat di program berikut:



```
Decompiling Java and Asm +  
www.javadecompilers.com/result  
System.out.println("Input salah 2 ");  
System.exit(1);  
}  
if (substring.charAt(19) != '_') {  
    System.out.println("Input salah 3 ");  
    System.exit(1);  
}  
final String[] split = substring.split("_", 0);  
if (!split[0].equals("belajar")) {  
    System.out.println("Input salah 4 ");  
    System.exit(1);  
}  
if (!new StringBuilder(split[1]).reverse().toString().equals("asahab")) {  
    System.out.println("Input salah 5 ");  
    System.exit(1);  
}  
if (!split[2].equals("jawa")) {  
    System.out.println("Input salah 6 ");  
    System.exit(1);  
}  
if (!new StringBuilder(split[3]).reverse().toString().equals("hig")) {  
    System.out.println("Input salah 7 ");  
    System.exit(1);  
}  
System.out.print("Selamat, Anda mendapatkan flag: ");  
System.out.println(nextLine);  
}  
  
Siapa Presiden Indonesia periode 2019 - 2024?  
Jawab pertanyaannya dan dapatkan  
kesempatan untuk memenangkan iPhone 11  
Sumber: https://www.surveymonkey.com/r/2019Presiden
```

-Di program tersebut terdapat kata yg mencurigakan seperti belajar,asahab,jawa,dan hig

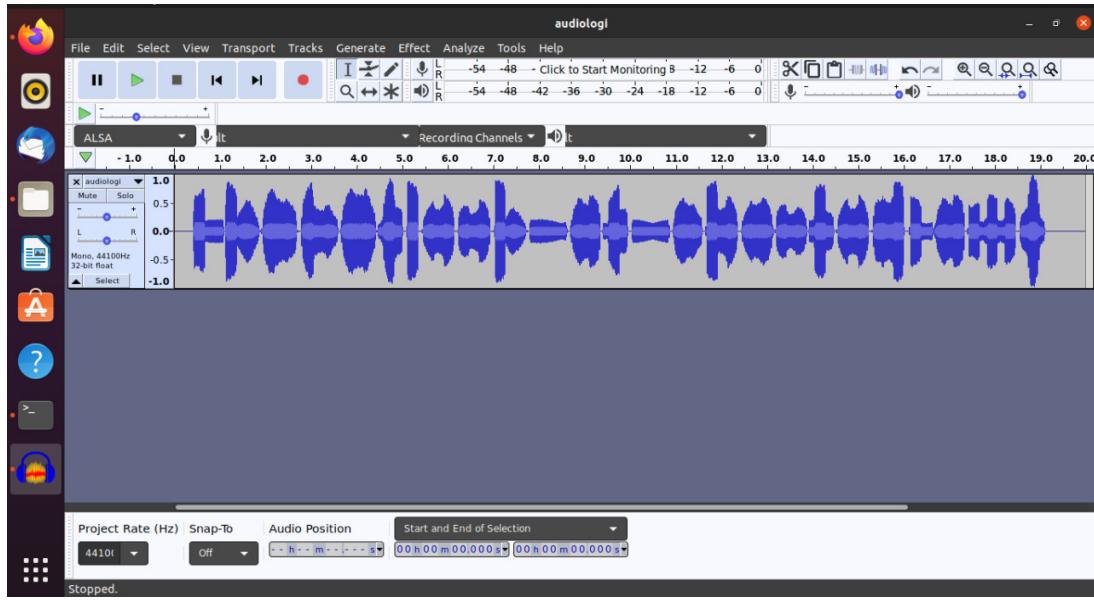
-Seperti yang diketahui kata asahab dan hig itu terbalik jika katanya diurutkan dari belakang maka akan menjadi bahasa dan gih

-Jika semua yang mencurigakan itu kita hubungkan maka akan merangkai sebuah kata LKSCS{belajar\_bahasa\_jawa\_gih}

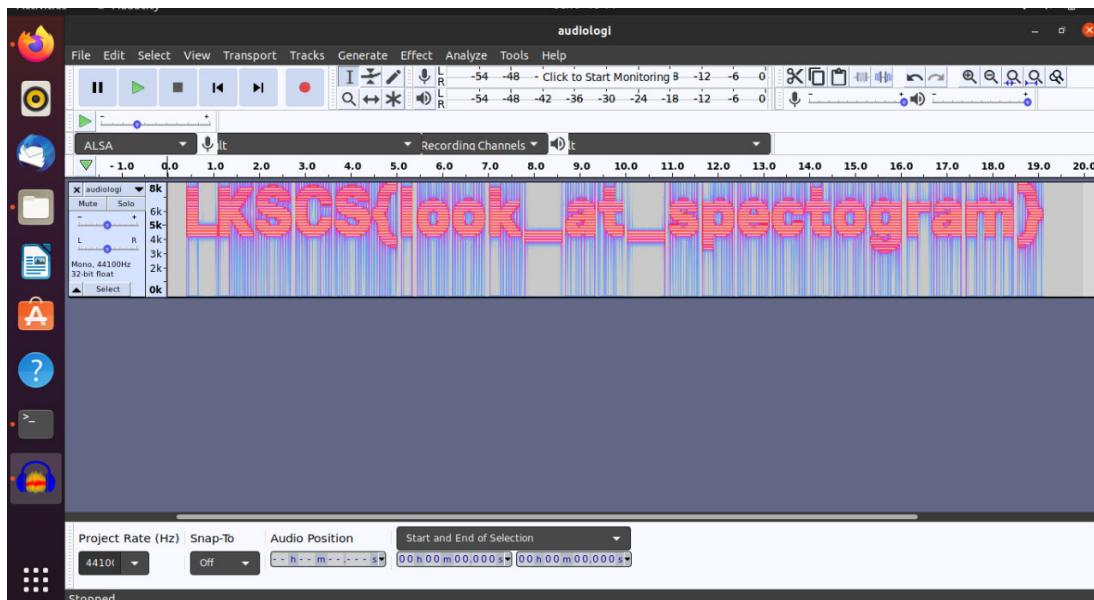
Flag : LKSCS{belajar\_bahasa\_jawa\_gih}

## Audiology

-Buka file audiologi.wav menggunakan tool audacity dan ini adalah tampilan dari file nya.



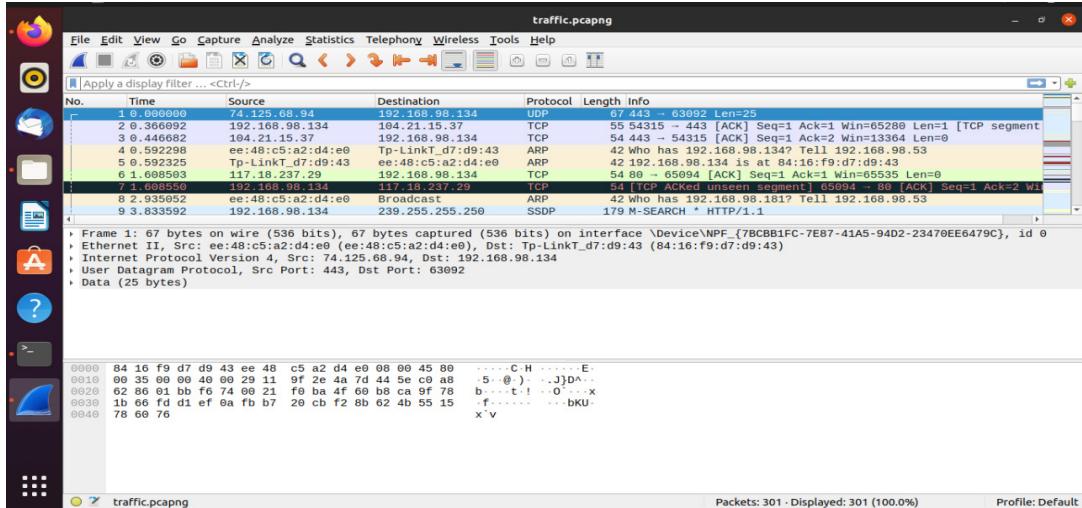
-Setelah file di buka pilih menu audiologi lalu pilih spectrogram. Dan berikut adalah hasilnya.



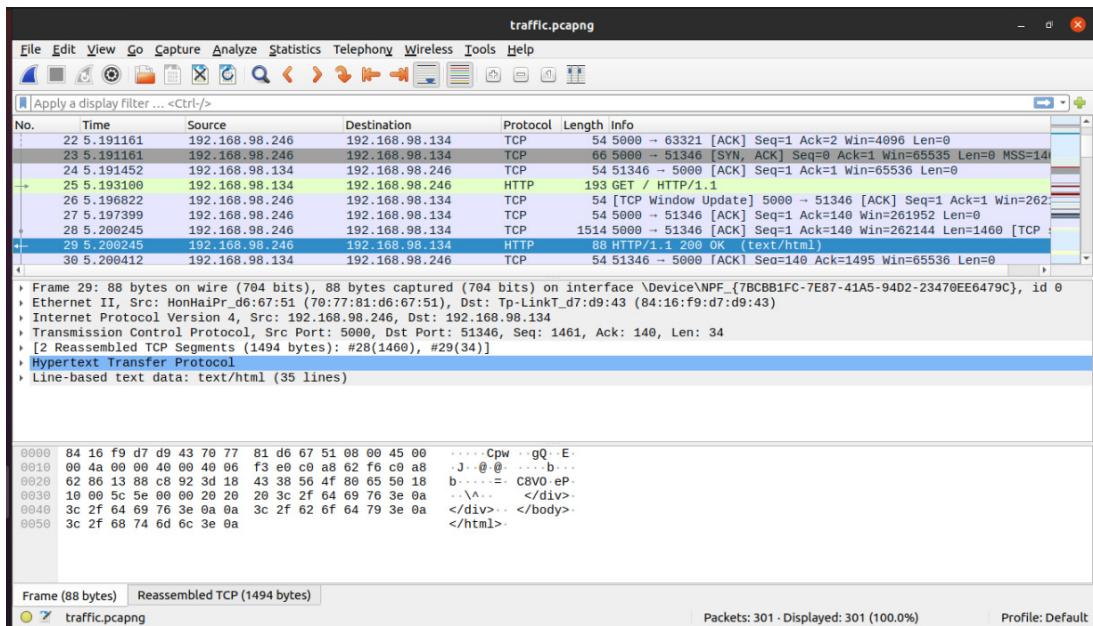
Flag : LKSCS{look\_at\_spectrogram}

## Capture The Traffic

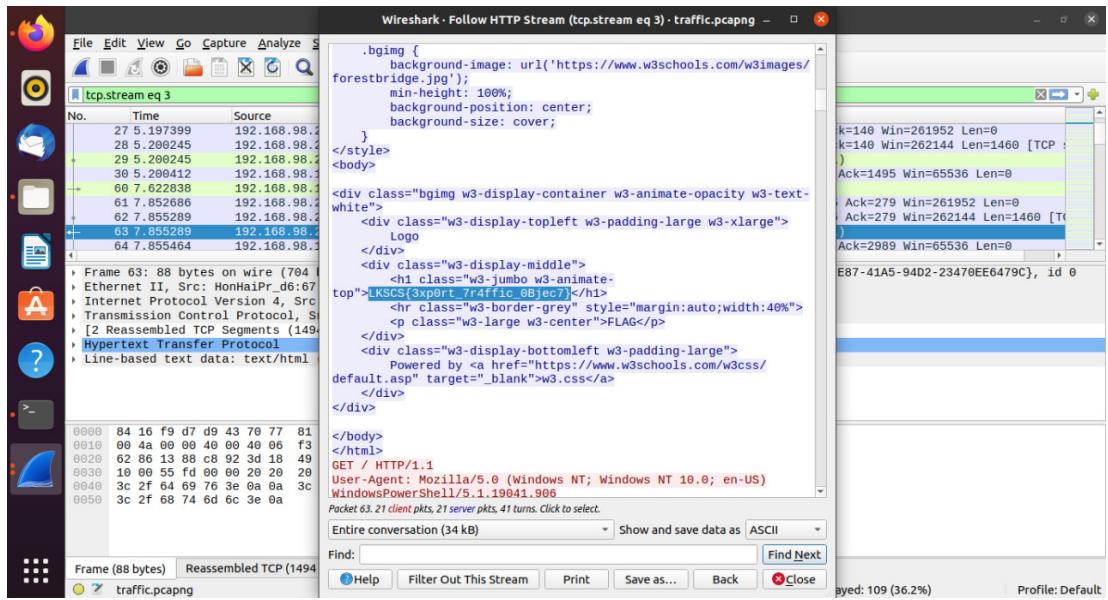
-Buka file traffic.pcapng di tool wireshark



-Cari protocol HTTP lalu klik kanan pilih menu follow lalu pilih menu HTTP stream.



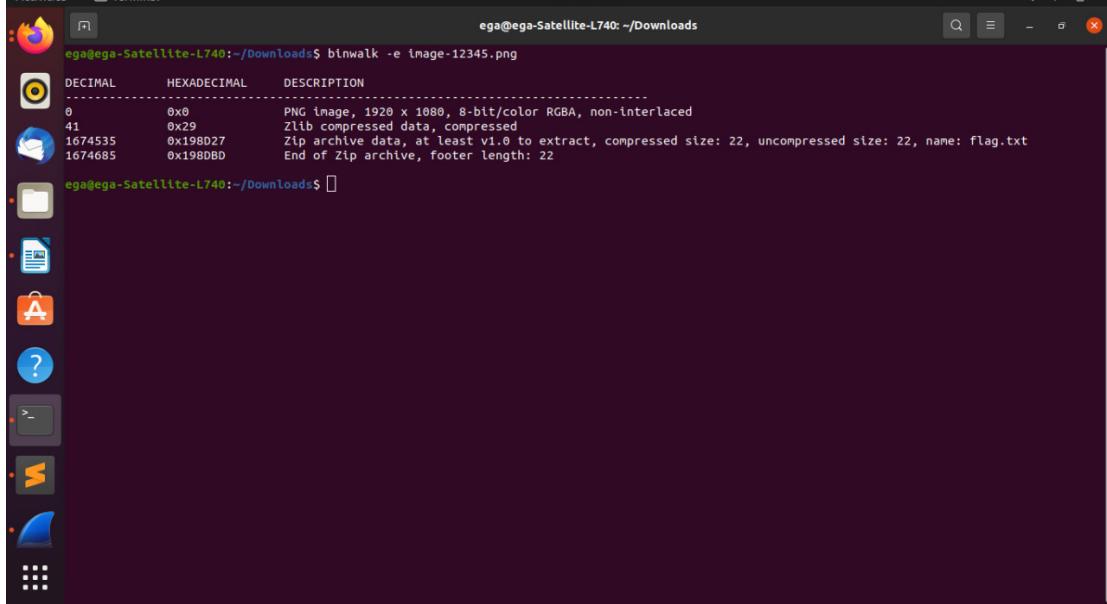
-Setelah muncul seperti digambar maka cari flag nya.



Flag:LKSCS{3xp0rt\_7r4ffic\_0Bjec7}

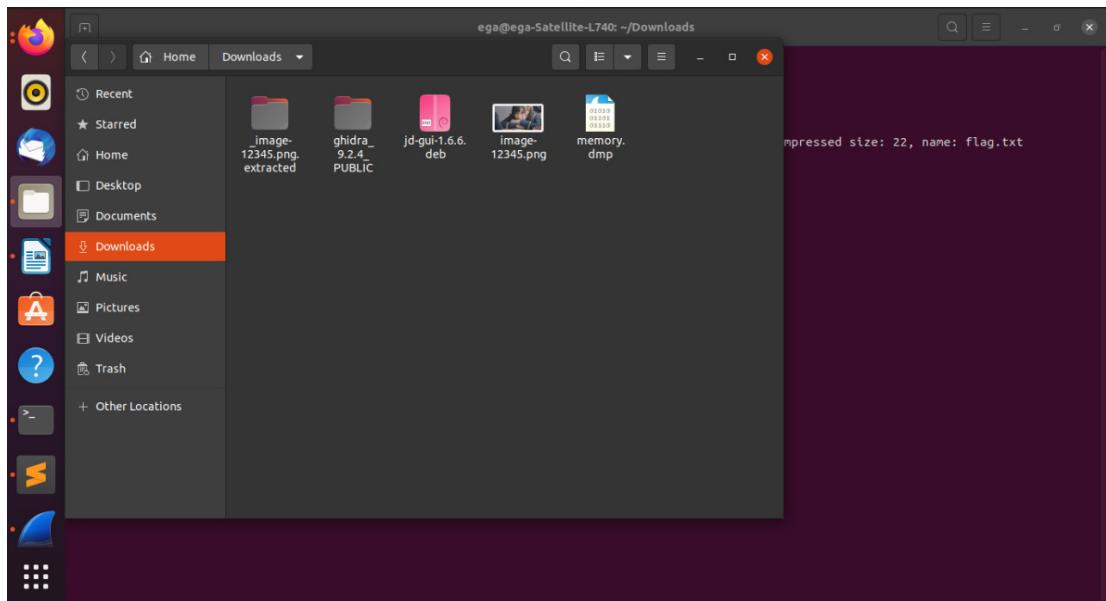
## **Hidden File**

-Ekstrak image -12345.png dengan menggunakan binwalk seperti gambar berikut :

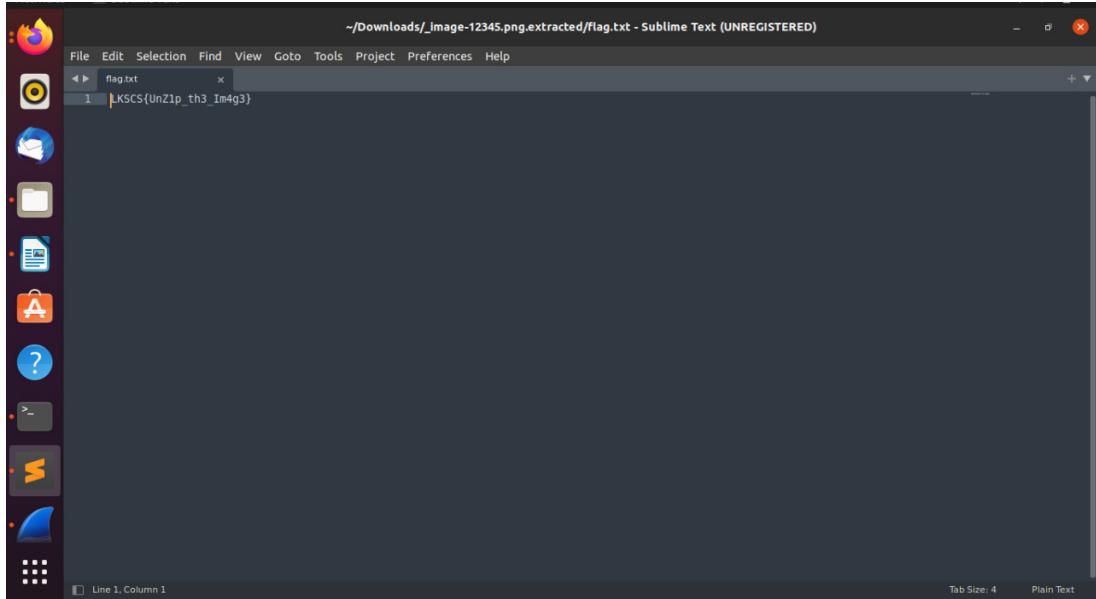


```
ega@ega-Satellite-L740:~/Downloads$ binwalk -e image-12345.png
DECIMAL      HEXADECIMAL      DESCRIPTION
-----      -----      -----
0            0x0      PNG image, 1920 x 1080, 8-bit/color RGBA, non-interlaced
41           0x29      Zlib compressed data, compressed
1674535     0x198027      Zip archive data, at least v1.0 to extract, compressed size: 22, uncompressed size: 22, name: flag.txt
1674685     0x1980BD      End of Zip archive, footer length: 22
ega@ega-Satellite-L740:~/Downloads$
```

-Setelah di ekstrak maka akan muncul folder baru hasil dari ekstrak image-12345.png yang tadi.



-Di dalam folder hasil ekstrak image-12345.png tadi terdapat flag.txt yang mempunyai isi berikut:



The screenshot shows a dark-themed Sublime Text editor window. The title bar reads: '~/Downloads/\_Image-12345.png.extracted/flag.txt - Sublime Text (UNREGISTERED)'. The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. A tab bar at the top shows 'flag.txt'. The main editor area contains a single line of text: '1 JKSC5{UnZ1p\_th3\_1m4g3}'. On the left side, there is a vertical toolbar with various icons: a target, a hand, a document, a file, a gear, a question mark, a terminal, a script, and a gear. At the bottom of the editor, it says 'Line 1, Column 1' and 'Tab Size: 4 Plain Text'.

Flag:{UnZ1p\_th3\_1m4g3}

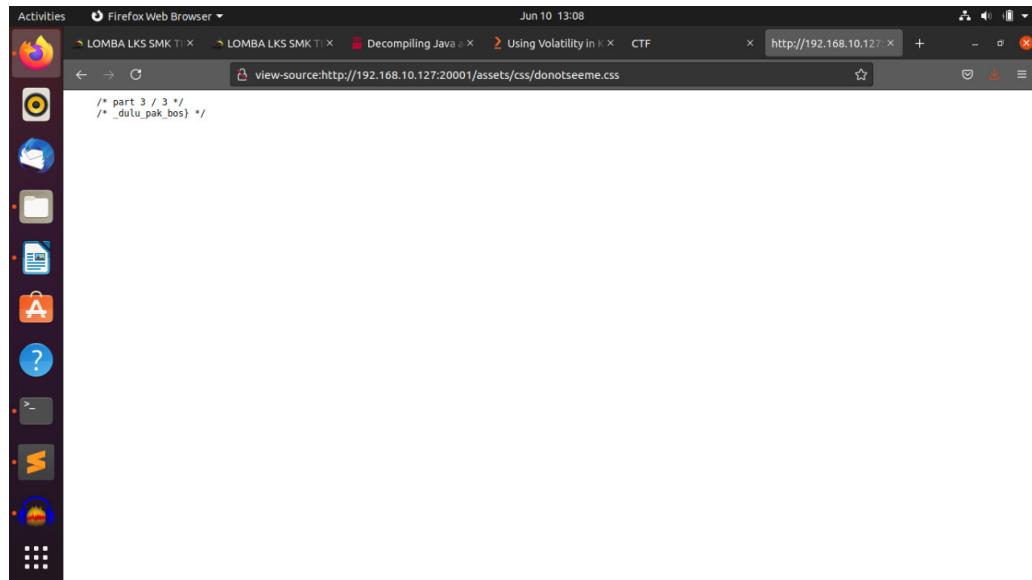
## **Know Your HTML**

-Buka link server yang diberikan lalu inspect web tersebut sehingga menghasilkan tampilan sebagai berikut.

-Jika discroll ke bawah akan ketemu kata-kata seperti "part 1 / 3" dan "LKSCS{ins}" yang artinya itu adalah part 1 dari 3 part flag yg terpisah.kita dapat mencari part lainnya dengan mengklik link yang terdapat pada fungsi href.

```
Activities Firefox Web Browser Jun 10 13:09
LOMBA LKS SMK TI x LOMBA LKS SMK TI x Decompiling Java x Using Volatility in x CTF x http://192.168.10.127:20001 +
LOMBA LKS SMK TI x view-source:http://192.168.10.127:20001/
<div>
  <div class="col"><span class="numberstep"><i class="fa fa-download"></i></span>
    <h3 class="numbertext">Digital Forensics</h3>
    <p>Digital forensics adalah kategori CTF yang bertujuan untuk menguji kemampuan peserta dalam menganalisis file digital untuk mendapatkan informasi tersembunyi dalam file tersebut. CTF bidang forensik ini biasanya melibatkan analisis log, file sistem, dan data jaringan untuk menemukan bukti kriminal atau keamanan.
  </div>
  <div class="col-md-4 editContent">
    <div class="col"><span class="numberstep"><i class="fa fa-download"></i></span>
      <h3 class="numbertext">Cryptography</h3>
      <p>Kriptografi bertujuan untuk menyembunyikan informasi dari pihak yang tidak berkepentingan dengan cara mengenkripsi informasi tersebut. CTF bidang kriptografi ini biasanya melibatkan pemecahan kunci enkripsi, analisis algoritma, dan teknik dekripsi.
    </div>
    <div class="col-md-4 editContent">
      <div class="col"><span class="numberstep"><i class="fa fa-download"></i></span>
        <h3 class="numbertext">Reverse Engineering</h3>
        <p>Reverse engineering adalah usaha untuk membongkar atau membalikkan atau memahami alur kerja suatu program. Program yang dibongkar dapat berupa binary executable atau sumber daya lainnya. Tujuan utamanya adalah untuk menemukan bug, kelemahan, atau kesalahan dalam program.
      </div>
      <div class="col-md-4 editContent">
        <div class="col"><span class="numberstep"><i class="fa fa-download"></i></span>
          <h3 class="numbertext">Miscellaneous</h3>
          <p>Berbagai kategori tambahan dalam CTF seperti OSINT, coding skill, dan lain-lain.</p>
        </div>
      </div>
    </div>
  </div>
  <div class="col text-center">
    <div class="container">
      <p>Keep looking for the hidden thing</p>
      <!-- part 1 / 3 -->
      <!-- LKSCS1ns -->
    </div>
  </div>
</div>
<script src="/assets/s/donotcheckthis.js"></script>
<script src="/assets/s/jquery_min.js"></script>
<script src="/assets/s/bootstrap.js"></script>
<script src="/assets/s/bootstrap_min.js"></script>
<script src="/assets/s/captionshop.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/js/all.min.js" integrity="sha512-RXF+QSDCU05uWkRKAoDxt55jygZznV+wUZduau/Ul/9EGp3f/2KZVahFZBKGH0s7zJLwvHqXnOOGdPmJGg=="></script>
</body>
</html>
```

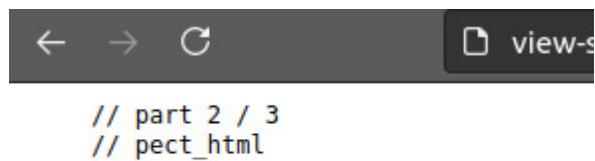
- Setelah diselidiki part 3 flag yang terdapat di href="assets/css/donotseeme.css"
- Berikut adalah tampilan dari href="assets/css/donotseeme.css":



The screenshot shows a Linux desktop environment with a dark theme. A Firefox window is open, displaying the source code of a CSS file. The URL in the address bar is `http://192.168.10.127:20001/assets/css/donotseeme.css`. The code in the window is:

```
/* part 3 / 3 */  
/* _dulu_pak_bos */
```

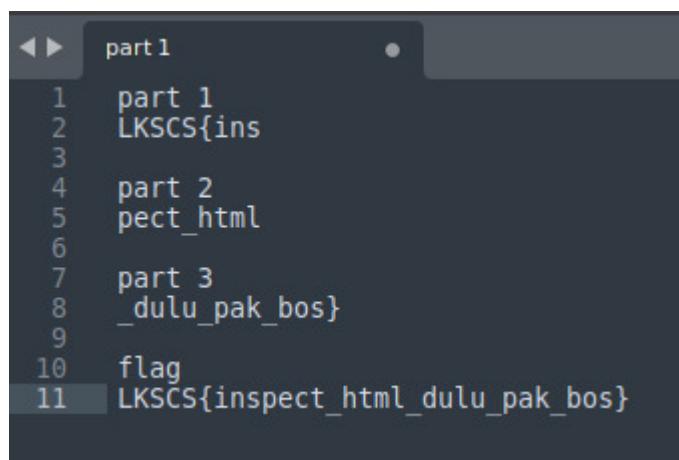
- Part 3 berada di src="assets/js/donotcheckthis.js" dan berikut ini adalah tampilan setelah meng-klik src="assets/js/donotcheckthis.js"



The screenshot shows a browser window with a dark theme. The title bar says "view-source". The content area displays the source code of a JavaScript file. The code is:

```
// part 2 / 3  
// pect_html
```

- Setelah 3 part telah diketahui waktunya kita mengurutkan flagnya sesuai urutan dari part 1 sampai part 3.berikut adalah hasilnya.



The screenshot shows a terminal window with a dark theme. The output is:

```
part1  
1 part 1  
2 LKSCS{ins  
3  
4 part 2  
5 pect_html  
6  
7 part 3  
8 _dulu_pak_bos}  
9  
10 flag  
11 LKSCS{inspect_html_dulu_pak_bos}
```

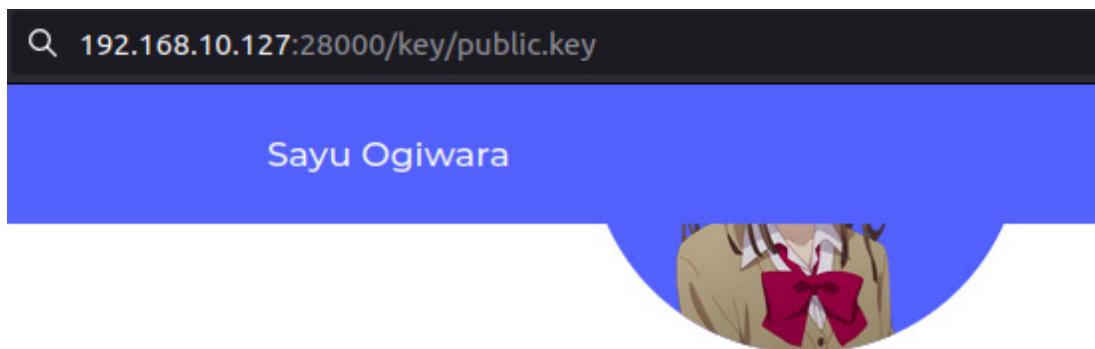
Flag : LKSCS{inspect\_html\_dulu\_pak\_bos}

## Sayu's Blog

-Buka link yg tertera di deskripsi soal lalu inspect element disana ada note mengenai public.key

```
1 <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <link rel="shortcut icon" href="/favicon.ico" type="image/x-icon">
7       <link rel="icon" href="/favicon.ico" type="image/x-icon">
8
9       <!-- ===== CSS ===== -->
10      <link rel="stylesheet" href="assets/css/styles.css">
11
12      <!-- ===== BOX ICONS ===== -->
13      <link href='https://cdn.jsdelivr.net/npm/boxicons@2.0.5/css/boxicons.min.css' rel='stylesheet'>
14
15      <!-- -----TODO: remove this----- -->
16      <!-- <link href="/key/public.key" rel='public-key'> -->
17      <!-- -----TODO: move keys from public----- -->
18
19      <title>Sayur</title>
20    </head>
21    <body>
22      <!--===== HEADER =====-->
23      <header class="l-header">
24        <nav class="nav bd-grid">
25          <div>
26            <a href="#" class="nav__logo">Sayu Ogiwara</a>
27          </div>
28
29          <div class="nav__menu" id="nav-menu">
30            <ul class="nav__list">
31              <li class="nav__item"><a href="#home" class="nav__link active">Home</a></li>
32              <li class="nav__item"><a href="#about" class="nav__link">About</a></li>
33              <li class="nav__item"><a href="#skills" class="nav__link">Skills</a></li>
34              <li class="nav__item"><a href="#portfolio" class="nav__link">Portfolio</a></li>
35              <li class="nav__item"><a href="#contact" class="nav__link">Contact</a></li>
36              <li class="nav__item"><a href="/admin" class="nav__link">Admin</a></li>
37            </ul>
38          </div>
39
40          <div class="nav__toggle" id="nav-toggle">
41
```

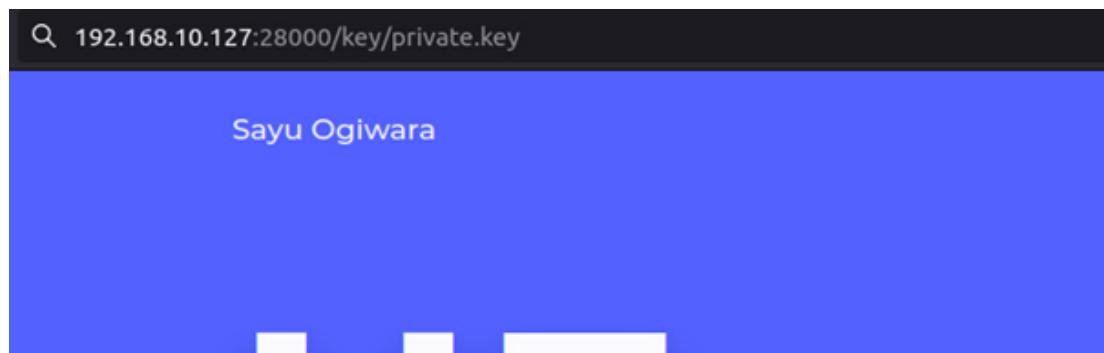
-Tambahkan /key/public.key di link website nya.



-Berikut ini adalah tampilan public.key

```
|-----BEGIN PUBLIC KEY-----  
MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAIXYNieS4rI7+o0m2Bvj9iZPc5tshWDh  
g512hauwgX+5pi5jFg2c/ldSomB8fCoyvW+aVlkkgpNP0jtyA3pnEFMCAwEAAQ==  
-----END PUBLIC KEY-----
```

-Kemudian cari private.key dengan menambahkan /key/private.key di akhir link



-Kemudian decode private key tadi seperti berikut:

The screenshot shows the JUUT.io interface for decoding a JWT token. The token itself is pasted into the 'Encoded' field:

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9eyJ
JuYW1lIjoiZVlc3QzNTUiLCJhZG1pbI6dHJ1Z
SwiaWF0IjoxNjIzMzA50DU2LCJleHAiOjE2MjMz
NTMwNTYsImF1ZCI6Imh0dHBzOi8vam9pbnRzlml
kIiwiXNzIjoiSk9JTlRTMjEiLCJzdWIiOiJjdG
ZAam9pbmRzlmlkIn0.gATYJ7c-
vCosfWdzx450f8TjyPuaNAjs3aFuzAihvSlhort
I36UhkX0iYG10Ufh8xwMhKXNjq45ayX8NX3_4HA
```

The 'Decoded' section shows the token's structure:

```
HEADER: ALGORITHM & TOKEN TYPE
{
  "alg": "RS256",
  "typ": "JWT"
}

PAYLOAD: DATA
{
  "name": "guest355",
  "admin": true,
  "iat": 1623369856,
  "exp": 1623353056,
  "aud": "https://joints.id",
  "iss": "JOINTS21",
  "sub": "ctf@joints.id"
}
```

The 'VERIFY SIGNATURE' section contains the RSA public key used for signing:

```
RSASHA256(
base64UrlEncode(header) + "." +
base64UrlEncode(payload),
-----END PUBLIC KEY-----
-----END RSA PRIVATE KEY-----)
```

**Signature Verified**

**SHARE JWT**

-Lakukan seperti digambar dulu sebelum membuka link  
192.168.10.127:28000/admin

The screenshot shows the Network tab of a browser developer tools window. A cookie named 'token' is listed, which corresponds to the JWT token shown in the previous screenshot.

Domain	Details
192.168.10.127	Domain: 192.168.10.127 First-Party Name: token Value: eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9eyJ         JuYW1lIjoiZVlc3QzNTUiLCJhZG1pbI6dHJ1Z         SwiaWF0IjoxNjIzMzA50DU2LCJleHAiOjE2MjMz         NTMwNTYsImF1ZCI6Imh0dHBzOi8vam9pbnRzlml         kIiwiXNzIjoiSk9JTlRTMjEiLCJzdWIiOiJjdG         ZAam9pbmRzlmlkIn0.gATYJ7c-         vCosfWdzx450f8TjyPuaNAjs3aFuzAihvSlhort         I36UhkX0iYG10Ufh8xwMhKXNjq45ayX8NX3_4HA         ...         -----END RSA PRIVATE KEY-----)

-Terakhir buka link 192.168.10.127:28000/admin seperti berikut:

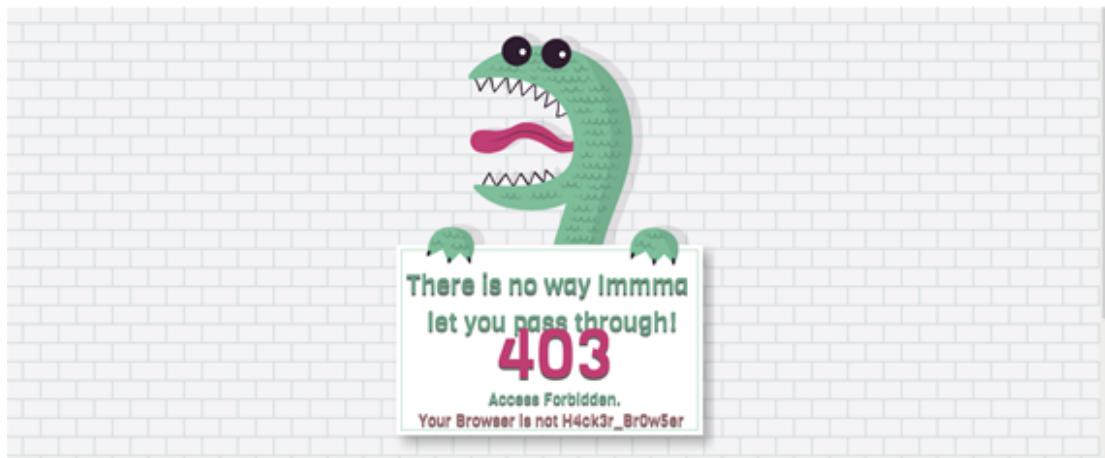


## Flag=LKSCS{Don't\_share\_your\_JWT\_key}

Flag : LKSCS{Don't\_share\_your\_JWT\_key}

### ***Invalid browser***

-Berikut adalah tampilan dari http://192.168.10.127:27900/rsvp



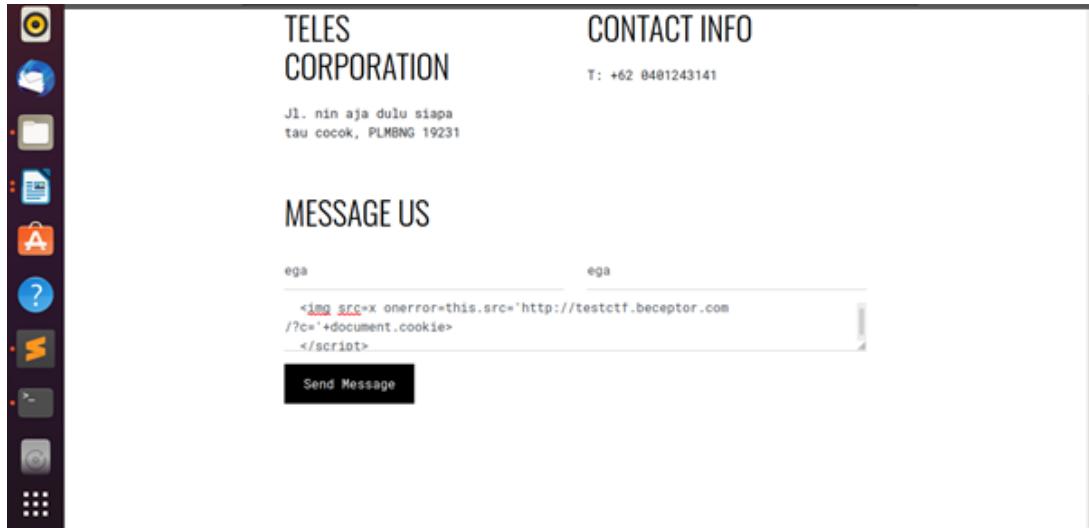
-Untuk mencari flag cukup jalankan “curl --user-agent "H4ck3r\_Br0w\$er" 192.168.10.127:27900/rsvp” di terminal seperti berikut:

```
mega@mega-Satellite-L740:~/Downloads$ curl --user-agent "H4ck3r_Br0w$er" 192.168.10.127:27900/rsvp
<!-- Flag: LKSCS{u53r_4gen7_wh1tel15t}--><!DOCTYPE html><title>PC</title><meta charset="UTF-8"><meta name="viewport" content="width=device-width, initial-scale=1"><link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css"><link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Raleway"><style>body,h1,h2{font-family: "Raleway", sans-serif}body, html {height: 100%}</style><!-- Header / Home--><div id="l001" style="display: block" class="w3-modal"><div class="w3-modal-content w3-card-4 w3-animate-zoom"><div class="w3-container w3-white w3-center"><h1>CAN YOU COME?</h1><p>We really hope you can make it..</p><h2>Flag: LKSCS{u53r_4gen7_wh1tel15t}</h2><p><small>Sincerely, Tatsuya & Miyuki</small></p></div></div></div>
```

Flag : LKSCS{u53r\_4gen7\_wh1tel15t}

## Form Feedback

-Buka link yg berikan di deskripsi.pada bagian message isi script untuk mencuri cookie seperti gambar berikut.



-Berikut ini adalah tampilan setelah berhasil mendapatkan cookie.

https://testctf.free.beeceptor.com → [nowhere]

4 requests Mocking Rules (2) Proxy Set

GET /?c=LKSCS(x55\_e4zy\_kan\_y4)

200 0.0s a minute ago

Create Mo

Request Body: View Headers { } Response Body: View Headers { }

Hey ya! Great to see you here. Btw, nothing is configured for this request path. Create a rule and start building a mock API.

## Virus Program

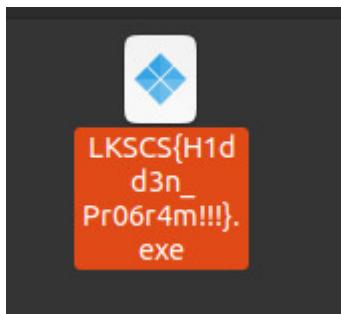
-Pertama scan file memory.dmp

```
ega@ega-Satellite-L740:~$ cd Downloads
ega@ega-Satellite-L740:~/Downloads$ volatility --profile=Win7SP1x86 pstrace -f memory.dmp
Volatility Foundation Volatility Framework 2.6.1
Name          Pid  PPid  Thds  Hnds Time
-----+-----+-----+-----+-----+
0x90030030:wininit.exe      460   392    4     88 2021-06-03 17:51:35 UTC+0000
.. 0x9005030:services.exe    556   460    7    191 2021-06-03 17:51:35 UTC+0000
... 0x903c2d8:svchost.exe    1280  556   17    361 2021-06-03 17:51:47 UTC+0000
... 0x901ebc70:svchost.exe   912   556   21    397 2021-06-03 17:51:47 UTC+0000
.... 0x9032dd20:audlodg.exe 1092   912   5     131 2021-06-03 17:51:47 UTC+0000
... 0x903c89e0:spoolsv.exe   1432  556   15    309 2021-06-03 17:51:47 UTC+0000
... 0x9c08fd20:taskhost.exe  1744  556   10    174 2021-06-03 17:51:48 UTC+0000
... 0x9c36030:svchost.exe   1576  556   9     221 2021-06-03 17:51:48 UTC+0000
... 0x901be770:svchost.exe   812   556   7     247 2021-06-03 17:51:47 UTC+0000
... 0x903da1f8:svchost.exe   1460  556   20    323 2021-06-03 17:51:47 UTC+0000
... 0x90307030:svchost.exe   948   556   24    495 2021-06-03 17:51:47 UTC+0000
... 0x9c0c0030:dwm.exe      1888  948   5     100 2021-06-03 17:51:48 UTC+0000
... 0x90313030:svchost.exe   976   556   15    321 2021-06-03 17:51:47 UTC+0000
... 0x901ac030:VBoxService.exe 744   556   12    152 2021-06-03 17:51:47 UTC+0000
... 0x90347940:svchost.exe   1148  556   6     129 2021-06-03 17:51:47 UTC+0000
... 0x901a2030:svchost.exe   680   556   13    369 2021-06-03 17:51:46 UTC+0000
... 0x90311330:svchost.exe   1016  556   29    706 2021-06-03 17:51:47 UTC+0000
... 0x84f36d20:SearchIndexer. 2324  556   13    611 2021-06-03 17:51:56 UTC+0000
... 0x9c253d20:SearchProtocol 2408  2324   7    278 2021-06-03 17:51:56 UTC+0000
... 0x9c25ed20:SearchFilterHo 2428  2324   6     88 2021-06-03 17:51:56 UTC+0000
... 0x900ac1a0:lsm.exe       580   460   10    152 2021-06-03 17:51:36 UTC+0000
... 0x9009a770:lsass.exe     564   460   10    481 2021-06-03 17:51:36 UTC+0000
0x867eb030:crsss.exe      400   392   12    362 2021-06-03 17:51:34 UTC+0000
0x83f2f4a0:System          4     0   113    474 2021-06-03 17:51:32 UTC+0000
... 0x8508c020:smss.exe     316   4     2     34 2021-06-03 17:51:32 UTC+0000
0x90024030:crss.exe       452   444   12    231 2021-06-03 17:51:35 UTC+0000
... 0x9c15bcf8:conhost.exe   2536  452   2     53 2021-06-03 17:51:58 UTC+0000
... 0x9c32e4f8:conhost.exe   3116  452   2     53 2021-06-03 17:52:21 UTC+0000
0x9004bd20:winlogon.exe   508   444   5     123 2021-06-03 17:51:35 UTC+0000
0x9c0b7d20:explorer.exe   1964  1872   34    849 2021-06-03 17:51:48 UTC+0000
... 0x9c185030:VBoxTray.exe  708   1964   15    156 2021-06-03 17:51:50 UTC+0000
... 0x9c0b4d20:livekd.exe    3108  1964   5     65 2021-06-03 17:52:21 UTC+0000
... 0x9c303030:kd.exe       3144  3108   2     65 2021-06-03 17:52:21 UTC+0000
... 0x9c303030:kd.exe       3144  3108   2     28 2021-06-03 17:52:25 UTC+0000
... 0x9c303030:kd.exe       3144  3108   2     28 2021-06-03 17:52:25 UTC+0000
... 0x9c279d20:LKSCS{H1dd3n_P} 2560  2524   5     101 2021-06-03 17:51:59 UTC+0000
... 0x9c279d20:LKSCS{H1dd3n_P} 2560  2524   5     101 2021-06-03 17:51:59 UTC+0000
```

-Di bagian bawah terdapat file dengan awalan "LKSCS{H1dd3n\_P}"

```
. 0x9c185030:VBoxTray.exe      708   1964   15    156 2021-06-03 17:51:50 UTC+0000
. 0x9c0b4d20:livekd.exe      3108  1964   5     65 2021-06-03 17:52:21 UTC+0000
.. 0x9c303030:kd.exe        3144  3108   2     28 2021-06-03 17:52:25 UTC+0000
. 0x9c27e1e8:LKSCS{H1dd3n_P} 2524  1964   4     38 2021-06-03 17:51:57 UTC+0000
.. 0x9c279d20:LKSCS{H1dd3n_P} 2560  2524   5     101 2021-06-03 17:51:59 UTC+0000
```

-Jika kita ekstrak maka akan menghasilkan file sebagai berikut.



Flag :LKSCS{H1dd3n\_P}.exe