Writeup IDN Bootcamp CTF 2025

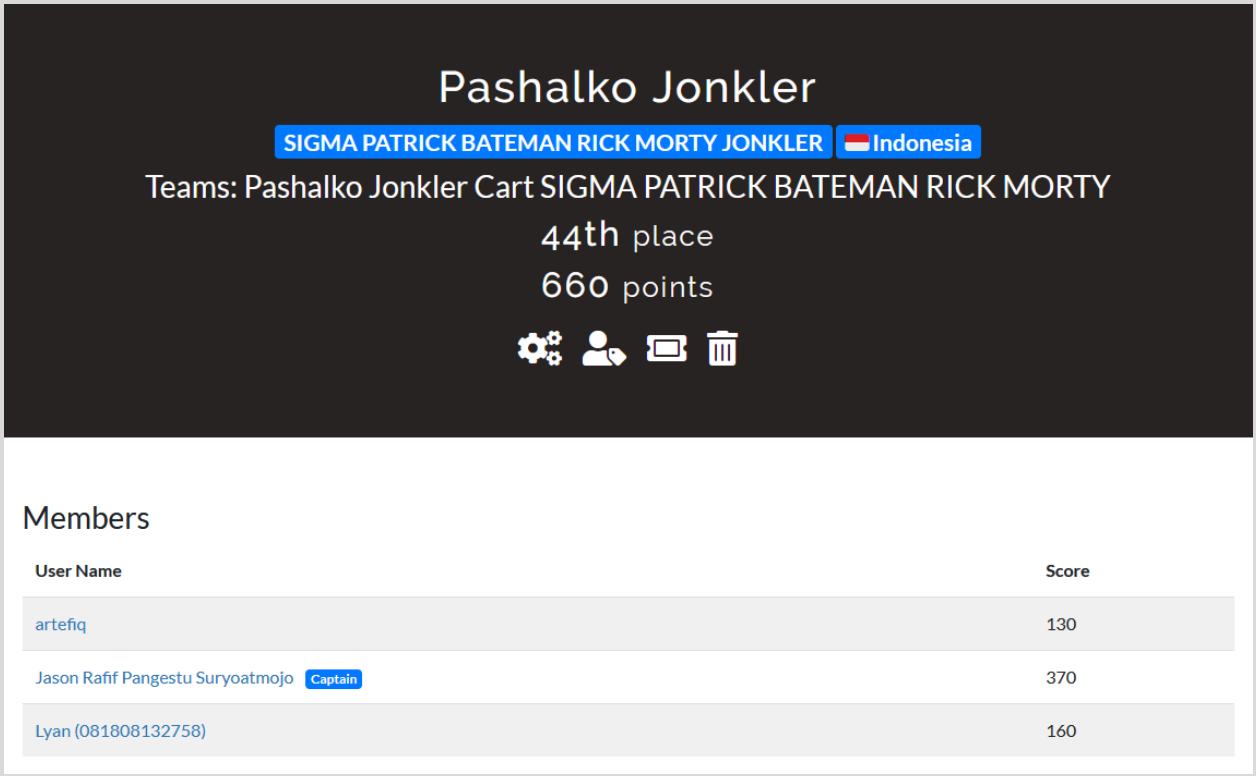
ID-Networkers
Indonesian IT Expert Factory

Team Pashalko Jonkler

Jason Rafif Pangestu Suryoatmojo
Ahmad Taufiq Hidayat
Lyan Nazhabil Dzuquwwa

Discord:
@shisones @artefiq @antonc2872

Challenges Solved



Categories	Solve	Score
Other	1/2	10
Cryptography	6/6	60
USB Forensic	7/8	70
Windows Forensic	11/15	110
Browser Forensic	10/10	100
Web Exploitation	13/13	130
Web 303	7/7	70
Log Analysis	8/9	80
Forensic	2/2	20
Welcome Flag	1/1	10
Total	66	660

Table of Contents

Other

User Guide

FLAG.

Simply check the User Guide pdf file for the flag, we just used strings, like any other forensic challenge.

```
Shisones@Arch-Thinkpad [~/Documents/IDN - Cybersecurity Bootcamp]
 ➜ strings User\ Guide\ CTFd\ IDN\ Cyber\ Security.pdf | grep IDN
<</Title (User Guide CTFd IDN Cyber Security)
<</Title (User Guide CTFd : CTF IDN Cyber Security  )
<</Title (IDN_FLAG{makasih_sudah_baca_guide} )
```

Flag : IDN_FLAG{makasih_sudah_baca_guide}

Cryptography

Jadi Gini Lagi

mau coba-coba aja terus, coba maen dino
Author: Aditya Firman Nugroho

Given an archive file jhlzhy.zip, containing a jpg file jhlzhy.jpg.

Seems like another stego challenge. provided that there's nothing wrong with the zip file size

```
Shisones@Arch-Thinkpad [Documents/CTF/IDN_Bootcamp]
➜ du jhlzhy.jpg jhlzhy.zip
248     jhlzhy.jpg
248     jhlzhy.zip
```

bang plis bang udh pake stegsolve, udh pake jsteg, udh pake stegsolve udh ngotak ngatik binary hex pake okteta tetep gadapeeeeet.

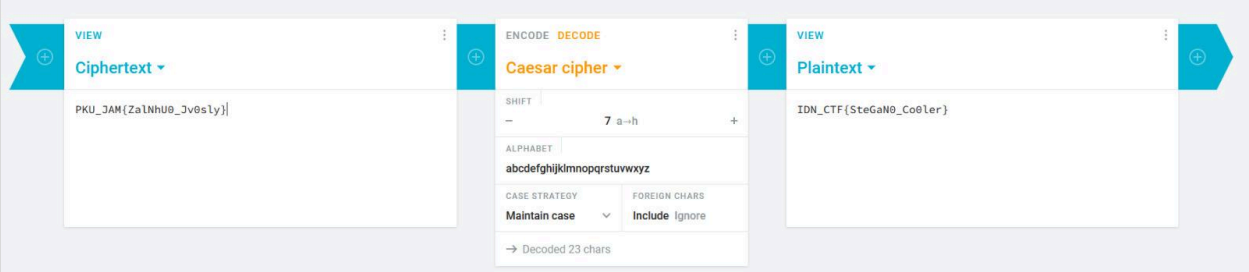
^ that's my thought before i found stegseek

```
Shisones@Arch-Thinkpad  [~/Documents/IDN - Cybersecurity Bootcamp]
 ➜ stegseek -sf jhlzhy.jpg -wl ~ ~/Downloads/rockyou.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Found passphrase: "jhlzhy"
[i] Original filename: "flag.txt".
[i] Extracting to "jhlzhy.jpg.out".
```

You kinda have to "bruteforce" it to get the secret hidden file, but when shown, it's another cipher

```
Shisones@Arch-Thinkpad  [~/Documents/IDN - Cybersecurity Bootcamp]
 ➜ cat jhlzhy.jpg.out
       File: jhlzhy.jpg.out
   1   PKU_JAM{ZalNhU0_Jv0sly}
```

Turns out its a simple caesar cipher with key=7.



Flag : IDN_CTF{SteGaN0_Co0ler}

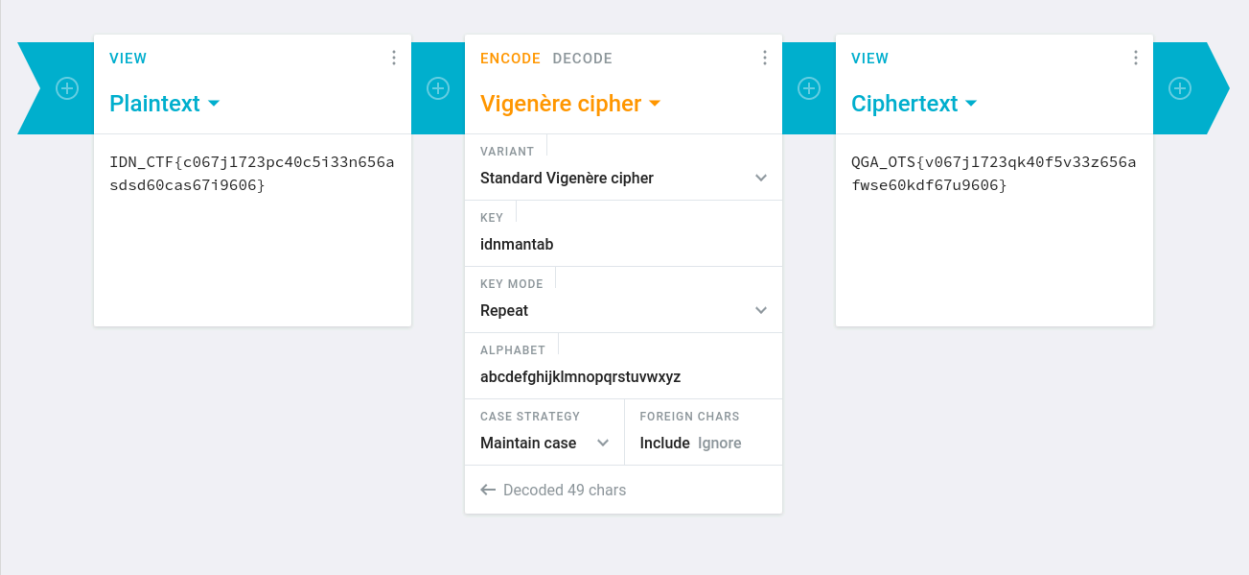
Might Guy's Secret

Suatu hari, Might Guy mengirimkan sebuah pesan rahasia ke Konoha HQ. Namun, pesan tersebut dicegat di tengah jalan.
Ini isi pesannya: QGA_OTS{v067j1723qk40f5v33z656afwse60kdf67u9606}
Bersama dengan pesan itu, kamu menemukan secarik kertas bertuliskan: "Giovan Battista Bellaso: 1553M: idnmantab"
Tampaknya Might Guy menggunakan teknik enkripsi klasik namun ampuh
Author: Nur Cholis Majid

From the description, it seems to be a ciphertext with the key "idnmantab". honestly, just put two and two together and get the flag.

- Batista Belasso, 1553M created and perfected Vigenere Cipher

So, just use a vigenere cipher decryptor



Flag : IDN_CTF{c067j1723pc40c5i33n656asdsd60cas67i9606}

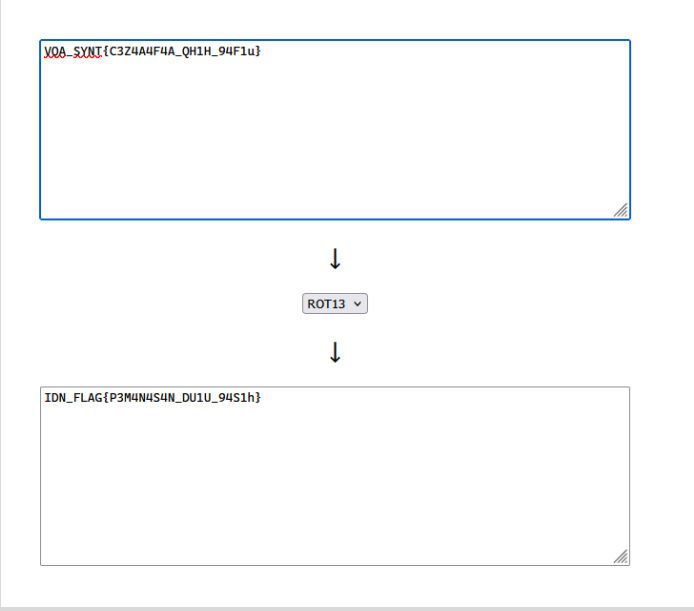
Rot1Aoka

Clue nya udah jelas kan?

VQA_SYNT{C3Z4A4F4A_QH1H_94F1u}

Author : Mohamad Fattyr

By seeing the clue, we know that it is ciphered in rotation 13, so we deciphered it and got the plaintext

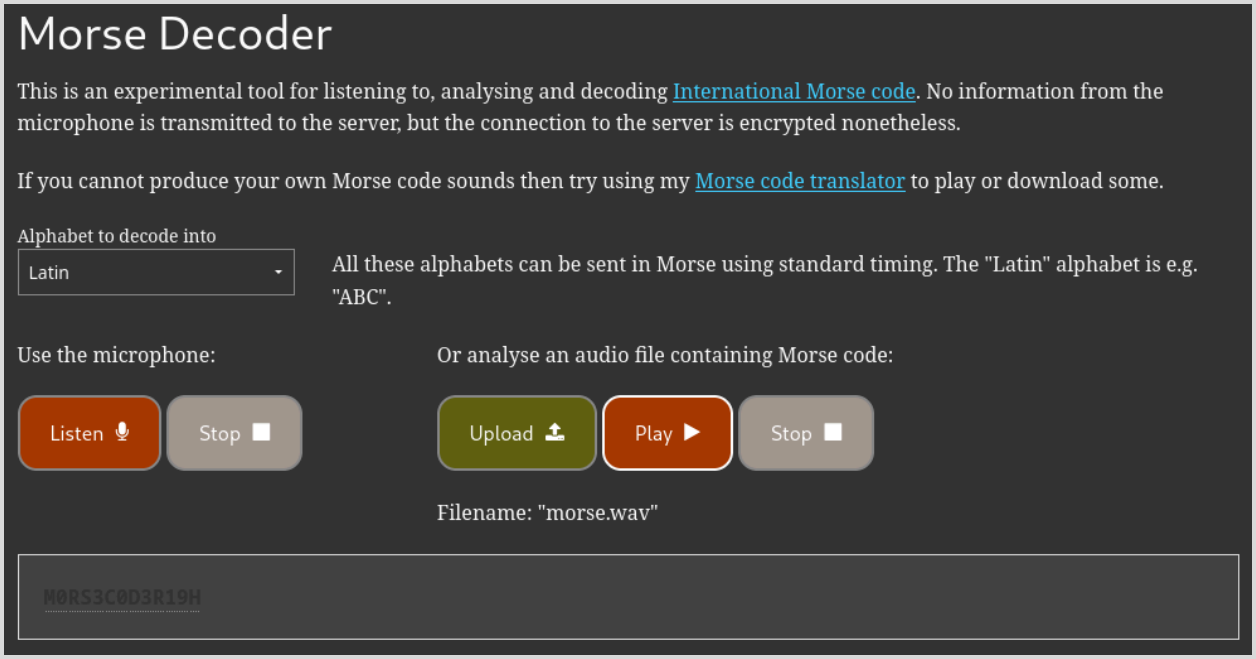


Flag : IDN_FLAG{P3M4N4S4N_DU1U_94S1h}

Pramuka

*terjemahan kan pesan tersebut. Format Flag IDN_CTF{****}*
Author : Mohamad Fattyr

Given a wav file containing a morse code, I hated boy scout class back in middle school, so I just put it in an online decoder.



Tried submitting it, and got a wrong answer, after spacing the words using underscore, it's correct

Flag : IDN_FLAG{M0RS3_C0D3_R19HT}

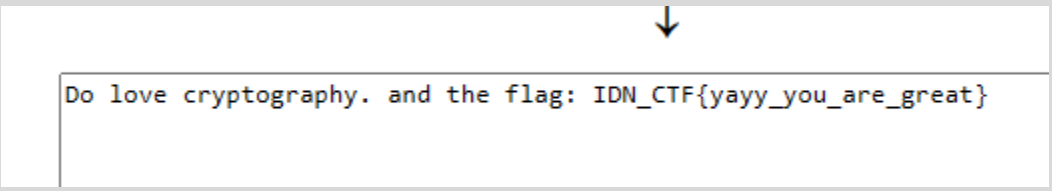
Classic Cryptography

Cn knud bqxosnfqzogx. zmc sgd ekzf:
HCM_BSE{xzxx_xnt_zqd_fqdzs}

Author: Rafly Permana

By seeing the question, we know that it is ciphered in rotation 1, so we deciphered it and got the plaintext:

ROT-0: Cn knud bqxosnfqzogx. zmc sgd ekzf: HCM_BSE{xzxx_xnt_zqd_fqdzs}
ROT-1: Do love cryptography. and the flag: IDN_CTF{yayy_you_are_great}
ROT-2: Ep mpwf dszquphsbqiz. boe uif gmbh: JEO_DUG{zbzz_zpv_bsf_hsfbu}
ROT-3: Fq nqxg etarvqitcrja. cpf vjg hnci: KFP_EVH{acaa_aqw_ctg_itgcv}
ROT-4: Gr oryh fubswrjudskb. dqg wkh iodj: LGQ_FWI{bdbb_brx_duh_juhdw}
ROT-5: Hs pszi gvctxskvetlc. erh xli jpek: MHR_GXJ{cecc_csy_evi_kviex}
ROT-6: It qtaj hwduytlwfumd. fsi ymj kqfl: NIS_HYK{dfdd_dtz_fwj_lwjfy}
ROT-7: Ju rubk ixevzumxgvne. gtj znk lrgm: OJT_IZL{egee_eua_gxk_mxkgz}
ROT-8: Kv svcl jyfwavnyhwof. huk aol mshn: PKU_JAM{fhff_fvb_hyl_nylha}
ROT-9: Lw twdm kzgxbwozixpg. ivl bpm ntio: QLV_KBN{gigg_gwc_izm_ozmib}
ROT-10: Mx uxen lahycxpajyqh. jwm cqn oujp: RMW_LCO{hjhh_hxd_jan_panjc}
ROT-11: Ny vyfo mbizdyqbkzri. kxn dro pvkq: SNX_MDP{ikii_iye_kbo_qbokd}
ROT-12: Oz wzgp ncjaezrclasj. lyo esp qwlr: TOY_NEQ{jljj_jzf_lcp_rcple}
ROT-13: Pa xahq odkbfasdmbtk. mzp ftq rxms: UPZ_OFR{kmkk_kag_mdq_sdqmf}
ROT-14: Qb ybir pelcgbtencul. naq gur synt: VQA_PGS{lnll_lbh_ner_terng}
ROT-15: Rc zcjs qfmdhcufodvm. obr hvs tzou: WRB_QHT{momm_mci_ofs_ufsoh}
ROT-16: Sd adkt rgneidvgpewn. pcs iwt uapv: XSC_RIU{npnn_ndj_pgt_vgtpi}
ROT-17: Te belu shofjewhqfxo. qdt jxu vbqw: YTD_SJV{oqoo_oek_qhu_whuqj}
ROT-18: Uf cfmv tipgkfxirgyp. reu kyv wcrx: ZUE_TKW{prpp_pfl_riv_xivrk}
ROT-19: Vg dgnw ujqhlgyjshzq. sfv lzw xdsy: AVF_ULX{qsqq_qgm_sjw_yjwsl}
ROT-20: Wh ehox vkrimhzktiar. tgw max yetz: BWG_VMY{rtrr_rhn_tkx_zkxtm}
ROT-21: Xi fipy wlsjnialujbs. uhx nby zfua: CXH_WNZ{suss_sio_uly_alyun}
ROT-22: Yj gjqz xmtkojbmvkct. viy ocz agvb: DYI_XOA{tvtt_tjp_vmz_bmzvo}
ROT-23: Zk hkra ynulpkcnwldu. wjz pda bhwc: EZJ_YPB{uwuu_ukq_wna_cnawp}
ROT-24: Al ilsb zovmqldoxmev. xka qeb cixd: FAK_ZQC{vxvv_vlr_xob_dobxq}
ROT-25: Bm jmtc apwnrmepynfw. ylb rfc djye: GBL_ARD{wyww_wms_ypc_epcyr}

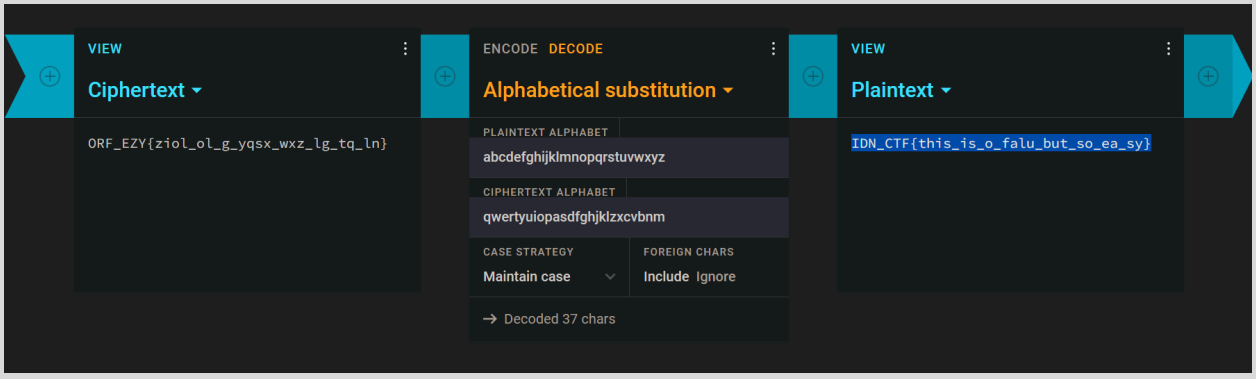


Flag : IDN_CTF{yayy_you_are_great}

Simple Substitution Cipher

ORF_EZY{ziol_ol_g_yqsx_wxz_lg_tq_ln}
Author: Rafly Permana

What even is this problem, the mappings are as follows:
abcdefghijklmnopqrstuvwxyz
To
qwertyuiopasdfghjklzxcvbnm
With that we can instantly get the flag by using an online mapping tool



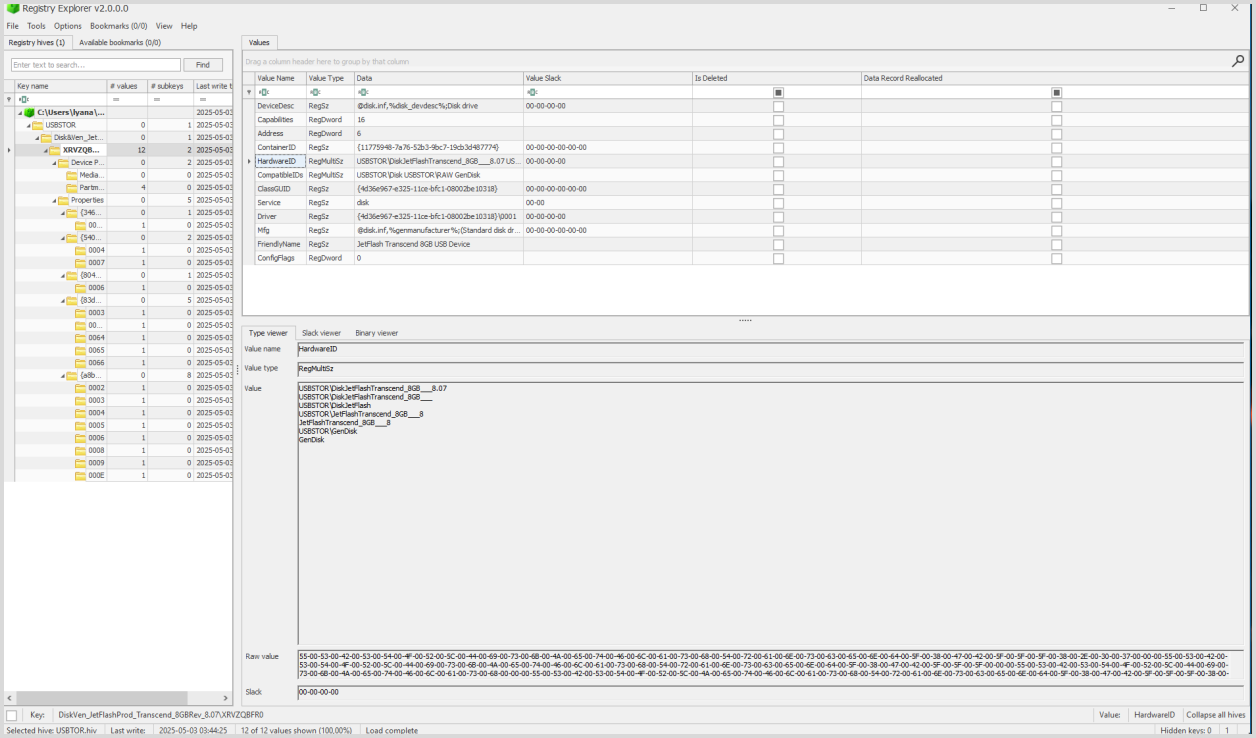
Flag : IDN_CTF{this_is_o_falu_but_so_ea_sy}

USB Forensics

USB Forensics 1

ada hacker, physical acces ke laptop.. bantuin dong !
Merek usb apa yang dipakai oleh hacker untuk delivery file nya ?
format flag : IDN_FLAG{Nama_Device_Ukuran_USB_Device}
Author: Aditya Firman Nugroho

Looking at the given zip file, there is an Acquisition folder. And since it looks like it has registry files, we just need to open it using this tool called Registry Explorer made by Eric Zimmerman. Link: https://ericzimmerman.github.io/#!index.md
After that, we load the relevant hive which is USBTOR.hiv from the folder and began navigating our way until we find something of value:



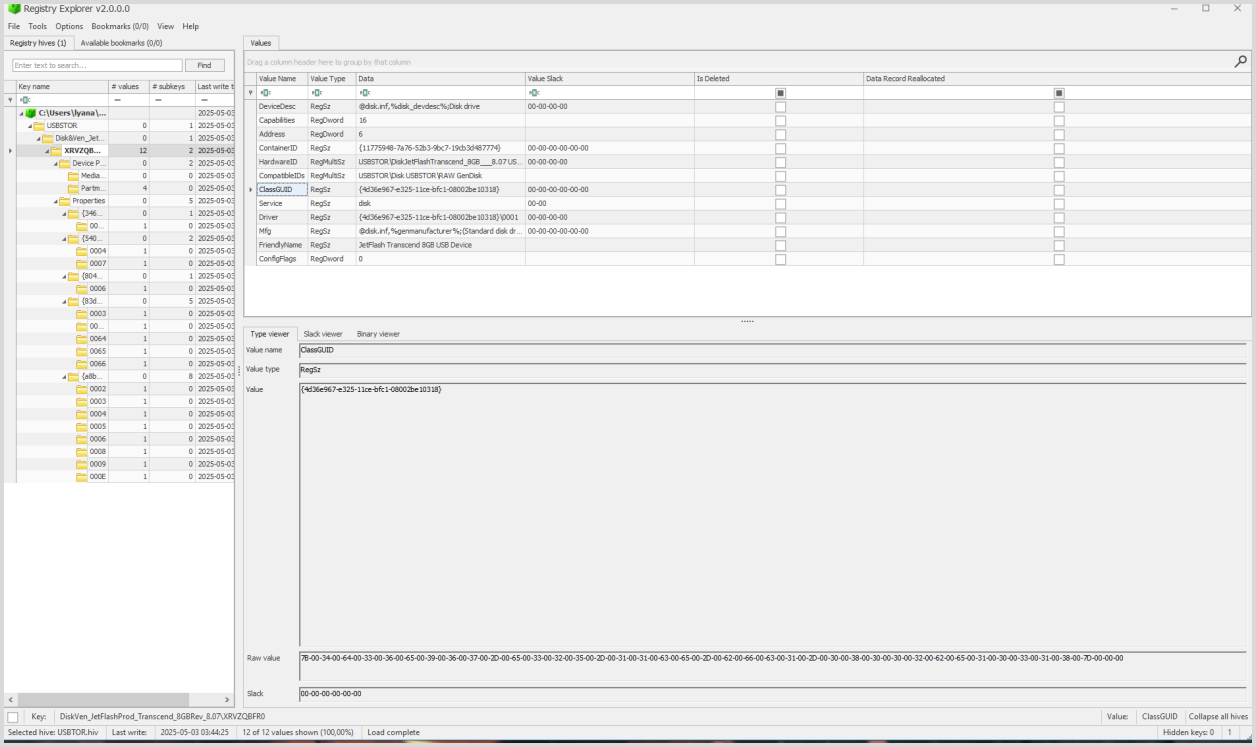
We then found out that in that particular registry, there is a HardwareID that gives us the description of the USB being used, which is Flash Transcend 8GB.

Flag : IDN_FLAG{Flash_Transcend_8GB_USB_Device}

USB Forensics 2

ada hacker, physical acces ke laptop.. bantuin dong !
ClassGUID Pada USB Hacker ?
format flag : IDN_FLAG{Jawaban yang disoal}
Author: Aditya Firman Nugroho

Back to our beloved Registry Explorer, we can actually see the classGUID in the same Key.

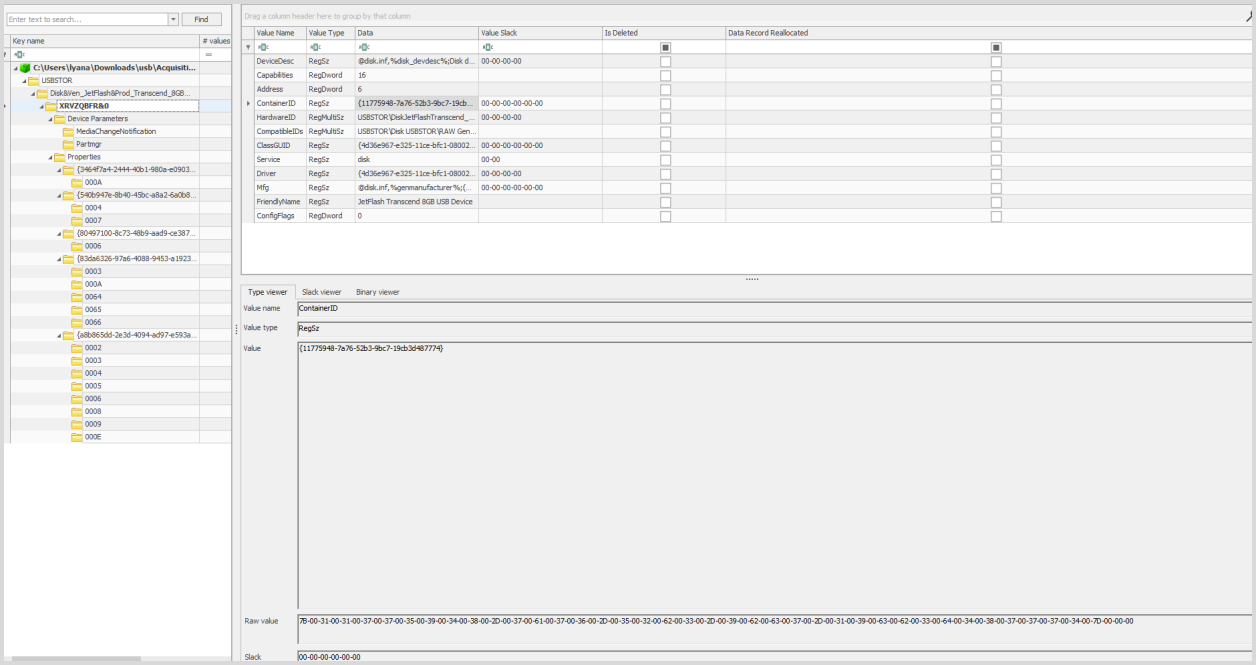


Flag : IDN_FLAG{4d36e967-e325-11ce-bfc1-08002be10318}

USB Forensics 3

ada hacker, physical acces ke laptop.. bantuin dong !
Apa Containder ID USB Yang dipakai Hacker ?
format flag : IDN_FLAG{Jawaban yang disoal}
Author: Aditya Firman Nugroho

Hope you don't miss our registry explorer because we're gonna keep using it. For the containerID its still in the same Key

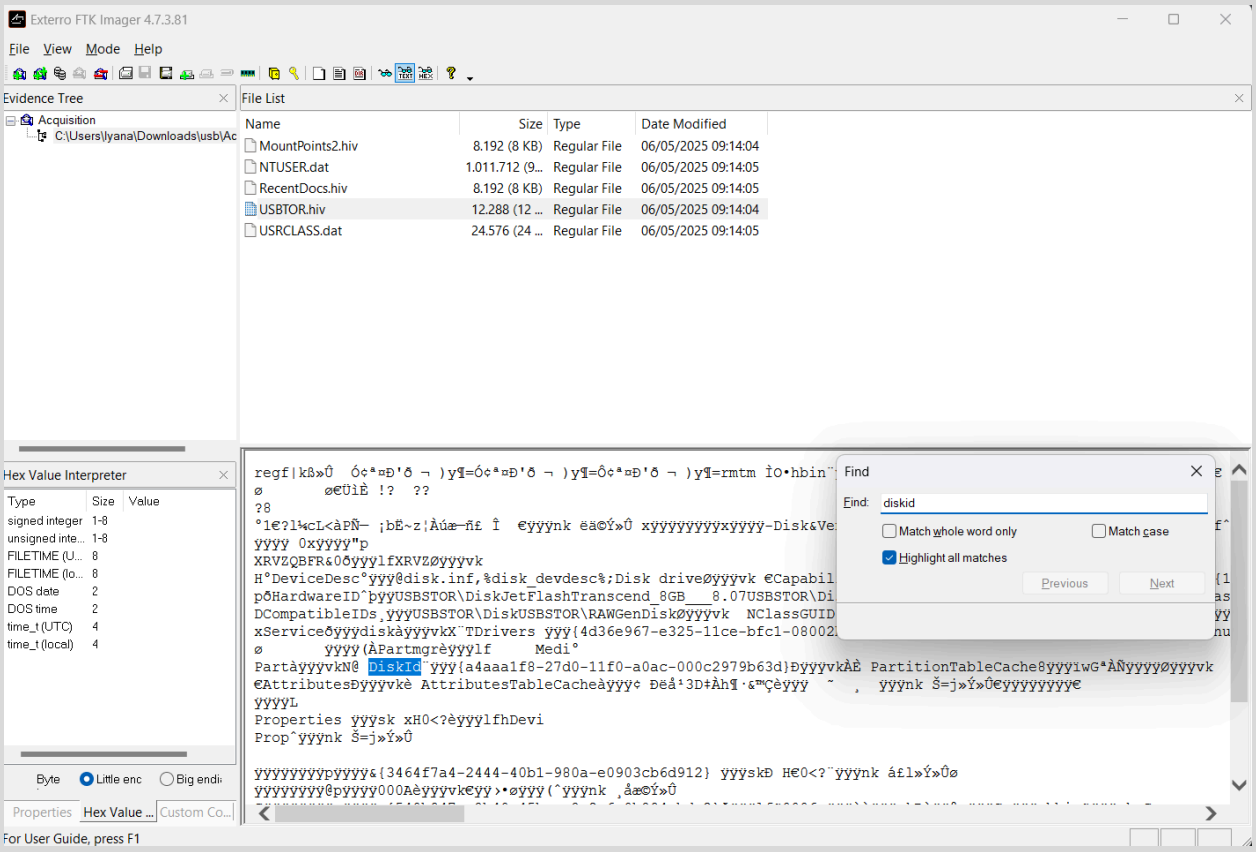


Flag : IDN_FLAG{11775948-7a76-52b3-9bc7-19cb3d487774}

USB Forensics 4

ada hacker, physical acces ke laptop.. bantuin dong !
Apa Disk ID yang dipakai hacker ?
format flag : IDN_FLAG{Jawaban yang disoal}
Author: Aditya Firman Nugroho

Okay now we can try another tool here, which is FTK Imager since there seems to be no value of “DiskID” found. So we mount the folder and searched for it in USBTOR.hiv

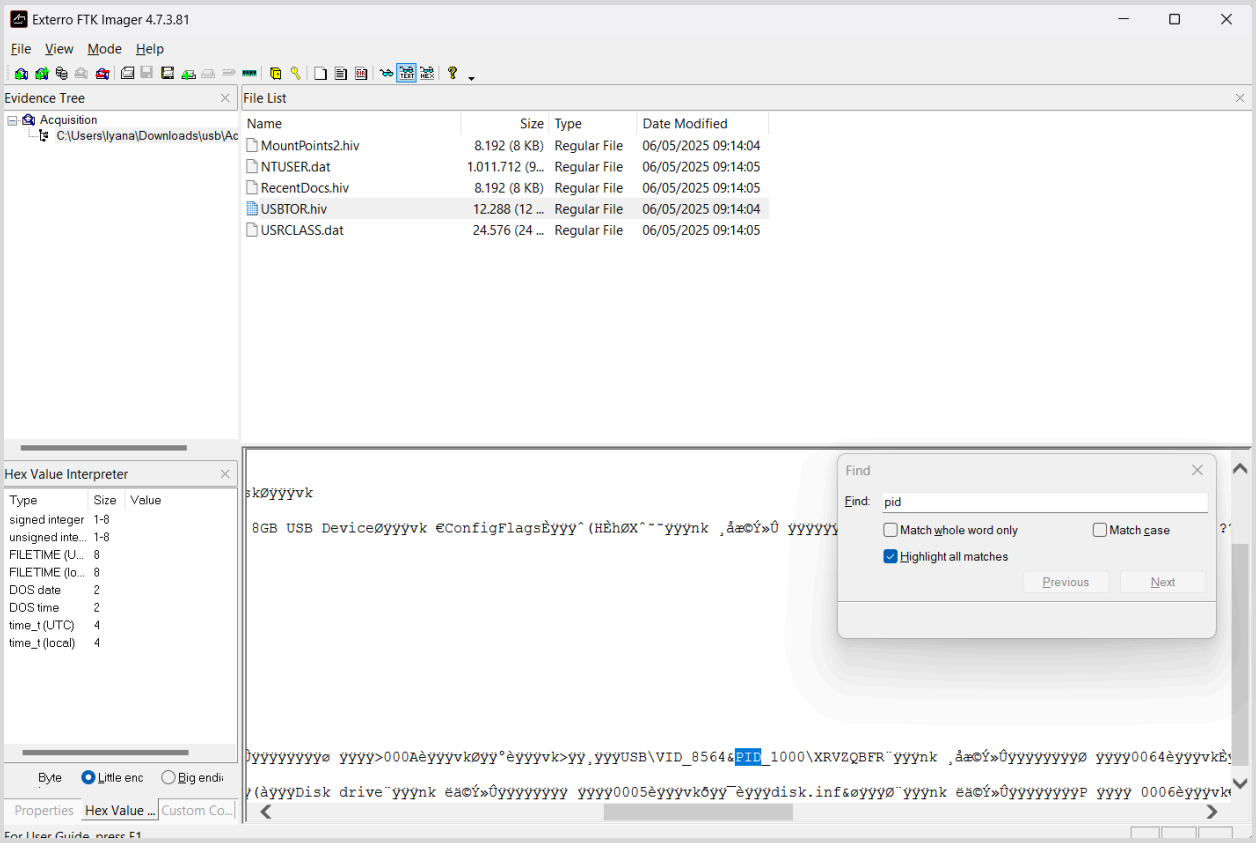


Flag : IDN_FLAG{a4aaa1f8-27d0-11f0-a0ac-000c2979b63d}

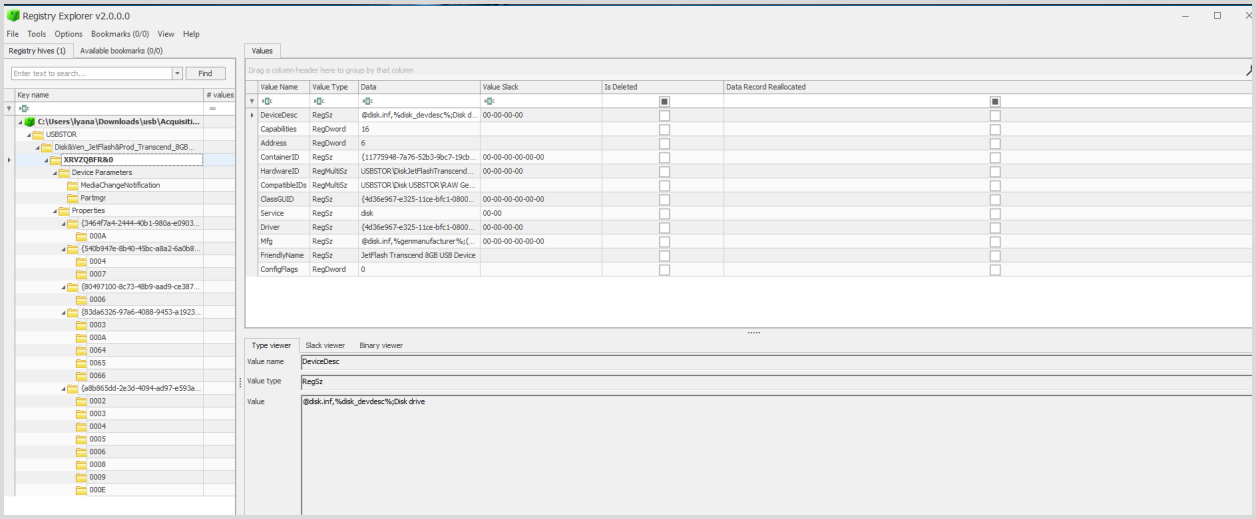
USB Forensics 5

ada hacker, physical acces ke laptop.. bantuin dong !
Apa Serial ID USB yang dipakai hacker ?
format flag : IDN_FLAG{Jawaban yang disoal}
Author: Aditya Firman Nugroho

Okay so i tried using FTK Imager to look for the serial id of the usb by trying to search the PID since the format is usually <VID>\<PID>\<SerialID>



I tried inputting the Serial ID value but its not the correct answer, so i go back to using Registry Explorer:



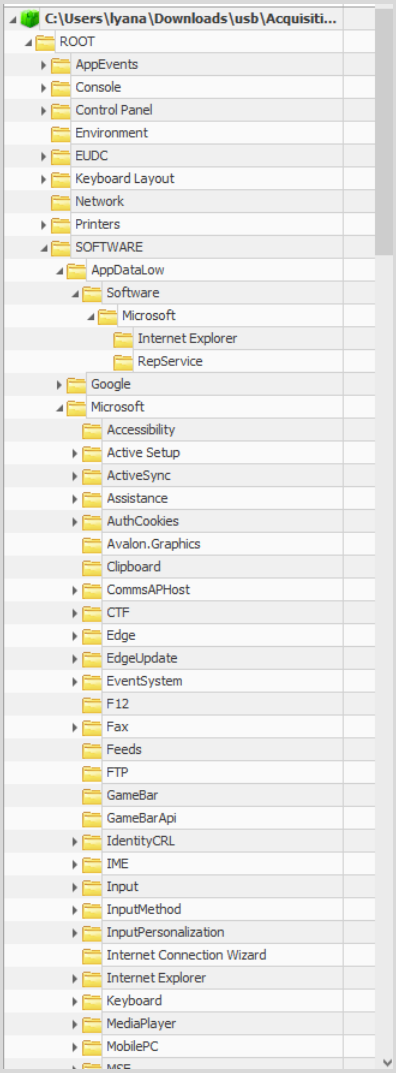
I noticed that the Key im currently in for challenge 1 to 3 is similar to the one on the Serial ID with a slight difference, so i tried inputting it and it is correct

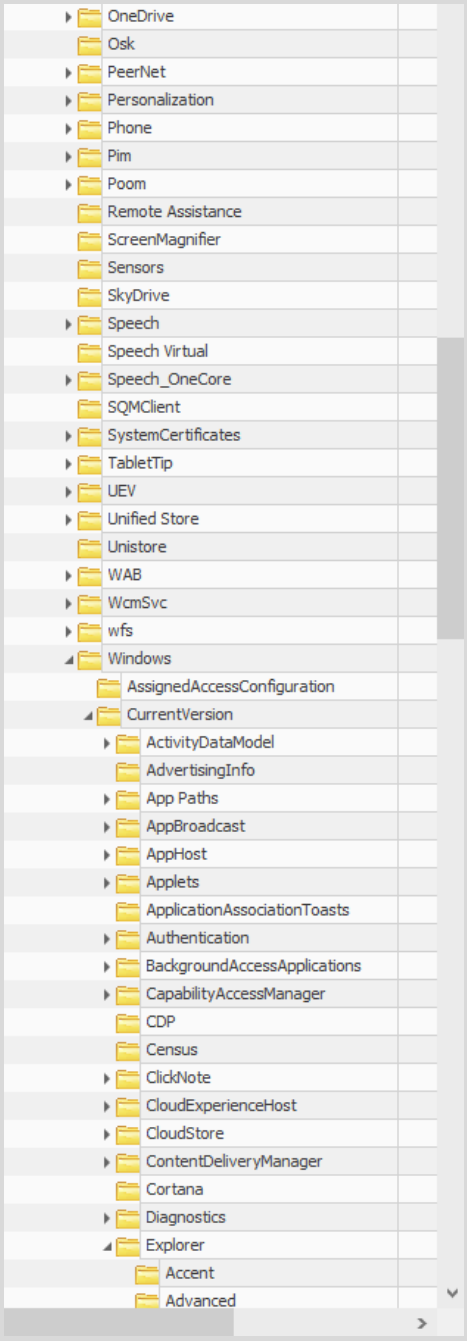
Flag : IDN_FLAG{XRVZQBFR&0}

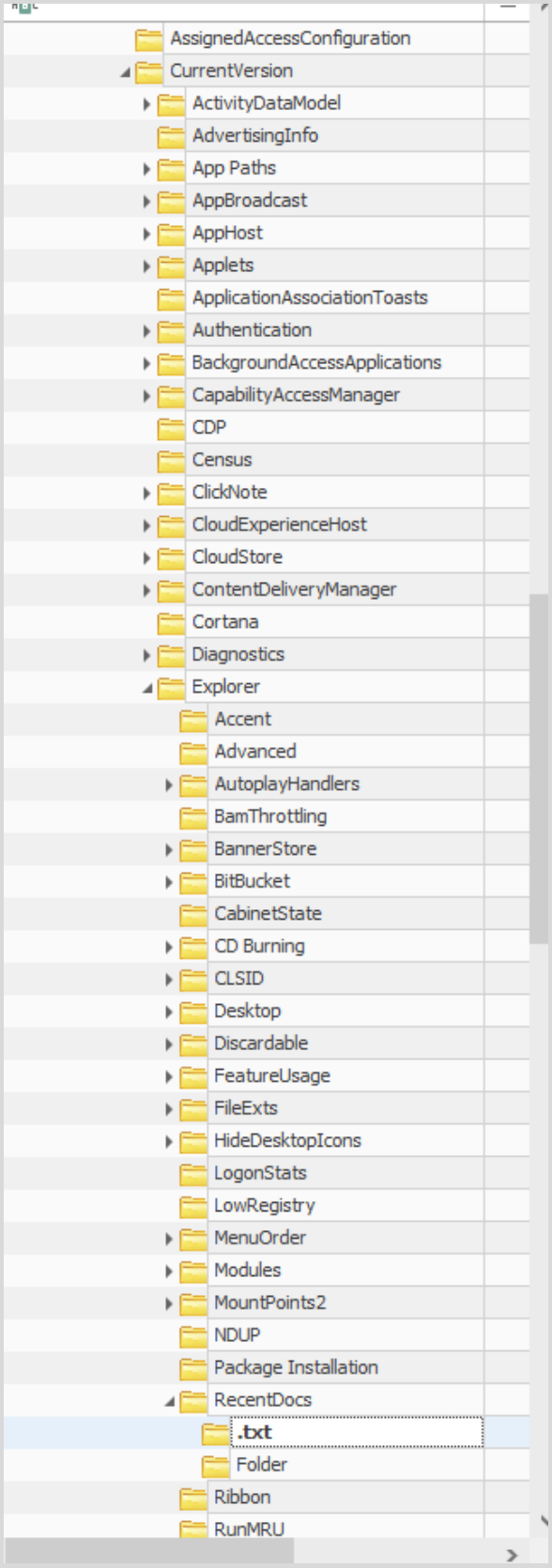
USB Forensics 6

ada hacker, physical acces ke laptop.. bantuin dong !
Nama File Yang ada di USB ?
format flag : IDN_FLAG{Jawaban yang disoal}
Author: Aditya Firman Nugroho

Since we want to find the files that are present in the USB, we're gonna open a different file here which is NTUSER.dat using Registry Explorer. And since the question specifically asks us for what files are present in the USB, we're gonna navigate all the way through recentdocs:




OneDrive
Osk
PeerNet
Personalization
Phone
Pim
Poom
Remote Assistance
ScreenMagnifier
Sensors
SkyDrive
Speech
Speech Virtual
Speech_OneCore
SQMClient
SystemCertificates
TabletTip
UEV
Unified Store
Unistore
WAB
WcmSvc
wfs
Windows
AssignedAccessConfiguration
CurrentVersion
ActivityDataModel
AdvertisingInfo
App Paths
AppBroadcast
AppHost
Applets
ApplicationAssociationToasts
Authentication
BackgroundAccessApplications
CapabilityAccessManager
CDP
Census
ClickNote
CloudExperienceHost
CloudStore
ContentDeliveryManager
Cortana
Diagnostics
Explorer
Accent
Advanced



As you can see we found a .txt file here:

Flag: IDN_FLAG{4fu284428u5984-8308848.txt}

USB Forensics 8

ada hacker, physical acces ke laptop.. bantuin dong !
File dibuka pada jam ?
format flag : IDN_FLAG{Jawaban yang disoal} example : xxxx-xx-xx xx:xx:xx
Author: Aditya Firman Nugroho

Looking at the .txt files value, you can see when the file is opened

Flag : IDN_FLAG{2025-05-03 03:48:32}

Windows Forensics

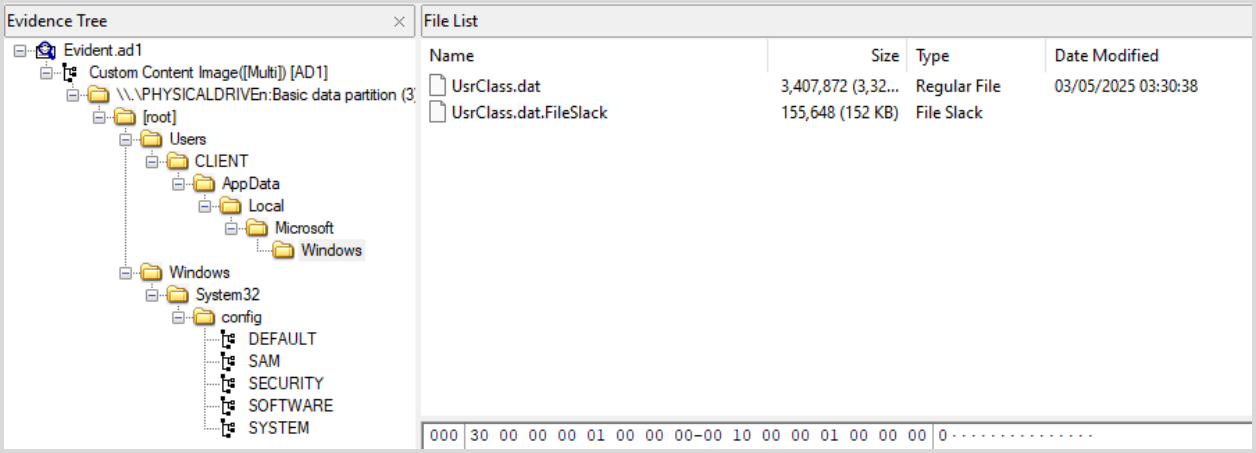
Windows Forensics 1

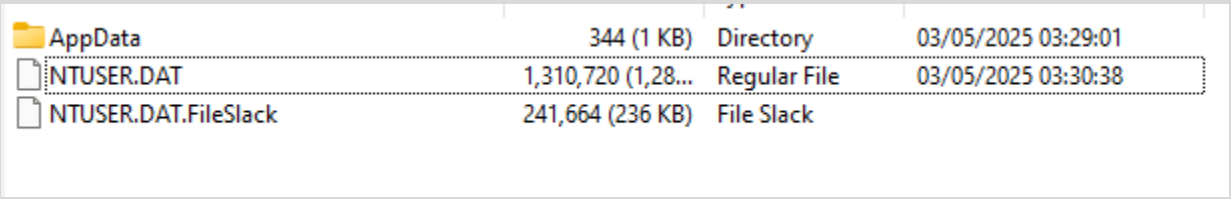
Ceritanya, udh ambil hivenya.. mount, trus cari beberapa informasi tambahan lainya !!
(Filenya ada di pertanyaan pertama)
Nama file yang menyimpan credential ?
format flag : IDN_FLAG{Jawaban yang disoal} example : xxxxxxxx_xxx_xxx
Author: Aditya Firman Nugroho

Given a large AD1 File with a txt file saying that the file was acquitted by FTK imager.

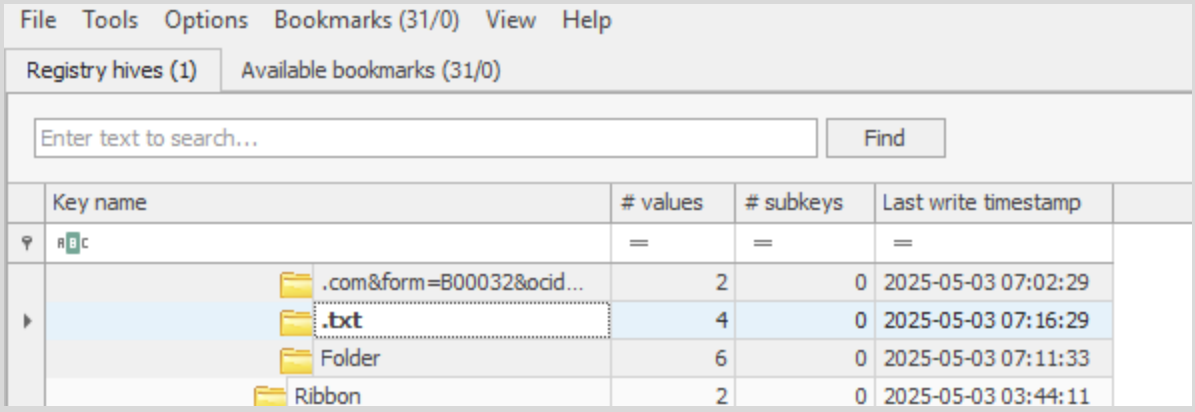
```
Created By Exterro® FTK® Imager 4.7.3.81

Case Information:
Acquired using: ADI4.7.3.81
Case Number:
Evidence Number:
Unique Description:
Examiner:
Notes:
```

Opening it in FTK imager shows us several registry files including a UsrClass.dat file





Exporting it and showing it in registry explorer shows us a lot of stuff, but since the Windows Recent History registry is stored in Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs, we will check it first and foremost, and since the hint shows a txt file, then we'll check the last txt file



Target Name	Lnk Name
password docs.txt	password docs.lnk
flag.txt	flag (2).lnk
4fu284428u5984-8308848.txt	4fu284428u5984-8308848.lnk

Flag : IDN_FLAG{password docs.txt}

Windows Forensics 2

Ceritanya, udh ambil hivenya.. mount, trus cari beberapa informasi tambahan lainya !!
(Filenya ada di pertanyaan pertama)
file yang menyimpan credential, dibuka pada tanggal berapa ?
format flag : IDN_FLAG{Jawaban yang disoal} example : xxxx-xx-xx xx:xx:xx
Author: Aditya Firman Nugroho

Checking the last activity of password docs.txt shows:

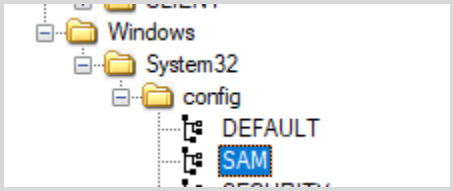
password docs.txt	password docs.lnk	0	2025-05-03 07:16:29

Flag : IDN_FLAG{2025-05-03 07:16:29}

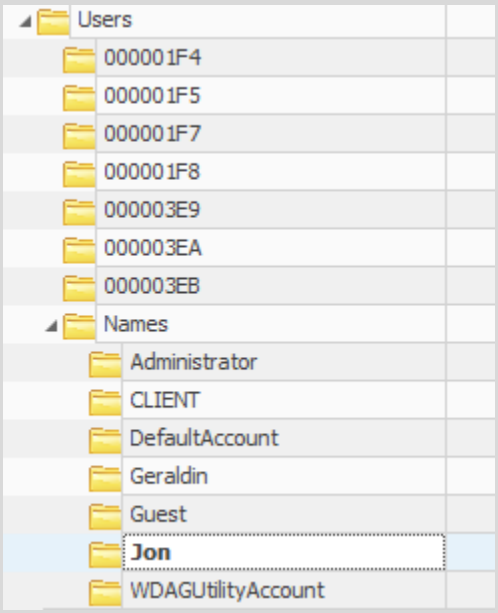
Windows Forensics 3

Ceritanya, udh ambil hivenya.. mount, trus cari beberapa informasi tambahan lainya !!
(Filenya ada di pertanyaan pertama)
User yang dibuat pada tanggal 2025-05-03 07:04:43, Username nya ?
format flag : IDN_FLAG{Jawaban yang disoal}
Auhtor: Aditya Firman Nugroho

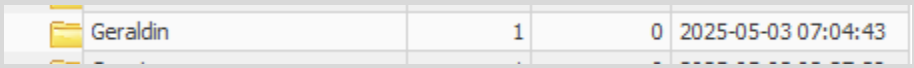
Using the same data, we will now look at SAM hive, the registry file containing account management stuffs.



Exporting it from FTK Imager and importing it to Regexplorer, and looking for the registry where Users are stored (SAM\Domains\Account\Users) shows:



This is huge, since each Name has a value tied to it (the hex id shown directly under Users/), to check the creation date, we could simply cross-reference the id with the names, or look at the last updated metadata in Regexplorer

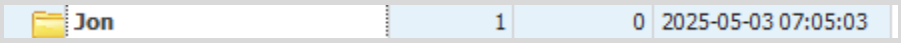


Flag : IDN_FLAG{Geraldin}

Windows Forensics 4

Ceritanya, udh ambil hivenya.. mount, trus cari beberapa informasi tambahan lainya !!
(Filenya ada di pertanyaan pertama)
User yang dibuat pada tanggal 2025-05-03 07:05:03, Username nya ?
format flag : IDN_FLAG{Jawaban yang disoal}
Auhtor: Aditya Firman Nugroho

Same as Windows Forensic 4, just look at the metadata

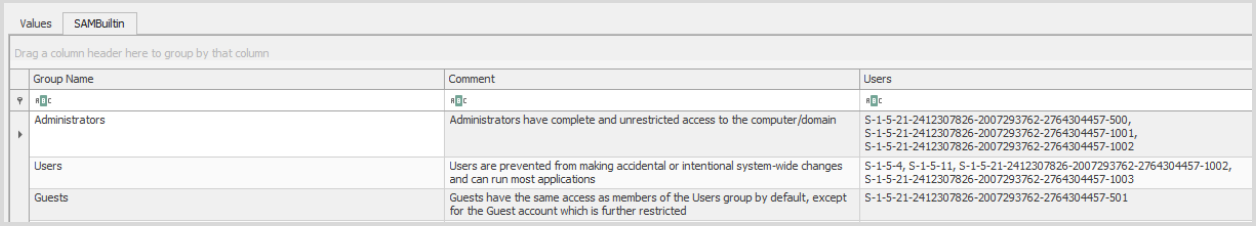


Flag : IDN_FLAG{Jon}

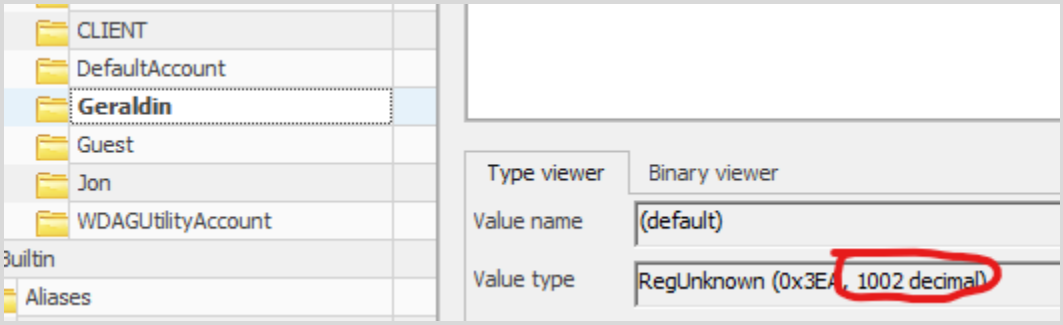
Windows Forensics 5

Ceritanya, udh ambil hivenya.. mount, trus cari beberapa informasi tambahan lainya !!
(Filenya ada di pertanyaan pertama)
User yang localgroupnya ada 2, yaitu ?
format flag : IDN_FLAG{Jawaban yang disoal}
Auhtor: Aditya Firman Nugroho

Going to SAM/Domains/Account/Builtin/Aliases, we can see the SID of the members of each group,



The SID which is not a built-in Windows SID (<500) is SID with 1002 at the end, checking the aforementioned account list shows that the person with SID 1002 is Geraldin.

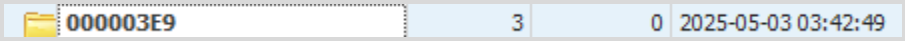


Flag : IDN_FLAG{Geraldin}

Windows Forensics 6

Ceritanya, udh ambil hivenya.. mount, trus cari beberapa informasi tambahan lainya !!
(Filenya ada di pertanyaan pertama)
Last Login Time dari User Cli.. ?
format flag : IDN_FLAG{Jawaban yang disoal} example : xxxx-xx-xx xx:xx:xx
Auhtor: Aditya Firman Nugroho

There's only 1 user starting with Cli.. and that's CLIENT (SID 1001, RID 0x3E9), checking the standard last login time registry, which is located in the RID entry of the user (SAM\Domains\Account\Users\000003E9), shows us the answer

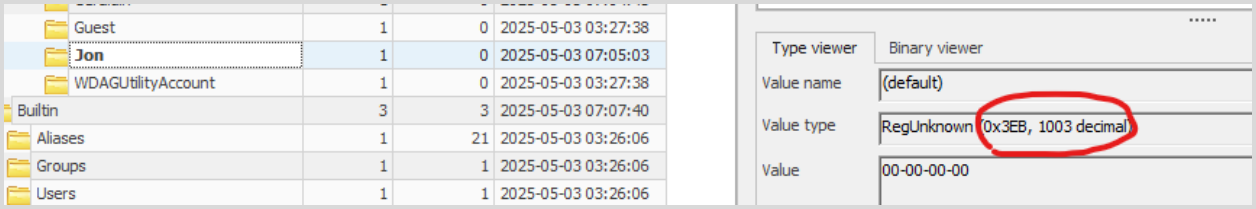


Flag : IDN_FLAG{2025-05-03 03:42:49}

Windows Forensics 7

Ceritanya, udh ambil hivenya.. mount, trus cari beberapa informasi tambahan lainya !!
(Filenya ada di pertanyaan pertama)
User ID dari user dengan 3 huruf ?
format flag : IDN_FLAG{Jawaban yang disoal}
Auhtor: Aditya Firman Nugroho

The user referred to is Jon, and he has an SID of 1003 and RID of 0x3EB

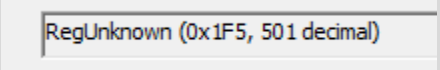


Flag : IDN_FLAG{1003}

Windows Forensics 8

Ceritanya, udh ambil hivenya.. mount, trus cari beberapa informasi tambahan lainya !!
(Filenya ada di pertanyaan pertama)
SID dari user Guest
format flag : IDN_FLAG{Jawaban yang disoal}
Auhtor: Aditya Firman Nugroho

We just navigate to the Guest, the same location as in the Windows Forensic 7 challenge



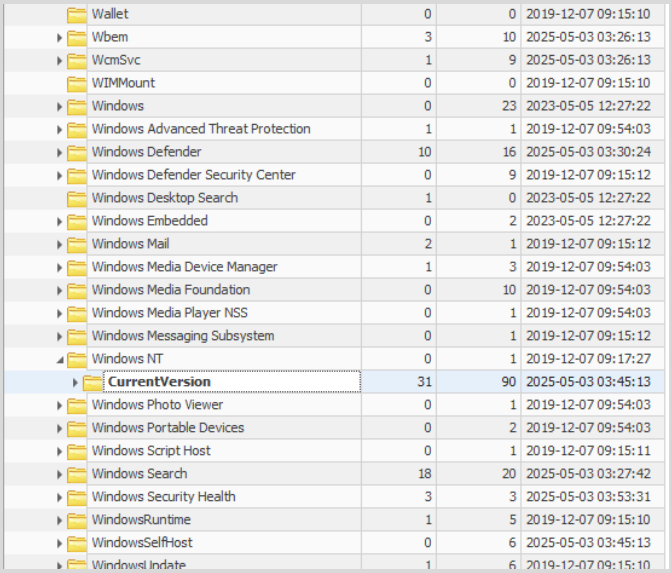
Group Name	Comment	Users
Administrators	Administrators have complete and unrestricted access to the computer/domain	S-1-5-21-2412307826-2007293762-2764304457-500, S-1-5-21-2412307826-2007293762-2764304457-1001, S-1-5-21-2412307826-2007293762-2764304457-1002
Users	Users are prevented from making accidental or intentional system-wide changes and can run most applications	S-1-5-4, S-1-5-11, S-1-5-21-2412307826-2007293762-2764304457-1002, S-1-5-21-2412307826-2007293762-2764304457-1003
Guests	Guests have the same access as members of the Users group by default, except for the Guest account which is further restricted	S-1-5-21-2412307826-2007293762-2764304457-501
Power Users	Power Users are included for backwards compatibility and possess limited	

Flag : IDN_FLAG{S-1-5-21-2412307826-2007293762-2764304457-501}

Windows Forensics 9

Ceritanya, udh ambil hivenya.. mount, trus cari beberapa informasi tambahan lainya !!
(Filenya ada di pertanyaan pertama)
CurrentBuild pada Windows ?
format flag : IDN_FLAG{Jawaban yang disoal}
Auhtor: Aditya Firman Nugroho

CurrentBuild is located in SOFTWARE\Microsoft\Windows NT\CurrentVersion, not SAM, so we have to export it again from FTK Imager and import it to Regexplorer



Wallet	0	0	2019-12-07 09:15:10
Wbem	3	10	2025-05-03 03:26:13
WcmSvc	1	9	2025-05-03 03:26:13
WIMMount	0	0	2019-12-07 09:15:10
Windows	0	23	2023-05-05 12:27:22
Windows Advanced Threat Protection	1	1	2019-12-07 09:54:03
Windows Defender	10	16	2025-05-03 03:30:24
Windows Defender Security Center	0	9	2019-12-07 09:15:12
Windows Desktop Search	1	0	2023-05-05 12:27:22
Windows Embedded	0	2	2023-05-05 12:27:22
Windows Mail	2	1	2019-12-07 09:15:12
Windows Media Device Manager	1	3	2019-12-07 09:54:03
Windows Media Foundation	0	10	2019-12-07 09:54:03
Windows Media Player NSS	0	1	2019-12-07 09:54:03
Windows Messaging Subsystem	0	1	2019-12-07 09:15:12
Windows NT	0	1	2019-12-07 09:17:27
CurrentVersion	31	90	2025-05-03 03:45:13
Windows Photo Viewer	0	1	2019-12-07 09:54:03
Windows Portable Devices	0	2	2019-12-07 09:54:03
Windows Script Host	0	1	2019-12-07 09:15:11
Windows Search	18	20	2025-05-03 03:27:42
Windows Security Health	3	3	2025-05-03 03:53:31
WindowsRuntime	1	5	2019-12-07 09:15:10
WindowsSelfHost	0	6	2025-05-03 03:45:13
WindowsUpdate	1	6	2019-12-07 09:15:10

CurrentBuild	RegSz	19045

Flag : IDN_FLAG{19045}

Windows Forensics 10

Ceritanya, udh ambil hivenya.. mount, trus cari beberapa informasi tambahan lainya !!
(Filenya ada di pertanyaan pertama)
DisplayVersion pada Windows ?
format flag : IDN_FLAG{Jawaban yang disoal}
Auhtor: Aditya Firman Nugroho

Same Location.

DisplayVersion	RegSz	22H2	1E-04

Flag : IDN_FLAG{22H2}

Windows Forensics 11

Ceritanya, udh ambil hivenya.. mount, trus cari beberapa informasi tambahan lainya !!
(Filenya ada di pertanyaan pertama)
Buildlab pada Windows ?
format flag : IDN_FLAG{Jawaban yang disoal}
Auhtor: Aditya Firman Nugroho

Again, same location.

BuildLab	RegSz	19041.vb_release.191206-1406

Flag : IDN_FLAG{19041.vb_release.191206-1406}

Windows Forensics 12

Ceritanya, udh ambil hivenya.. mount, trus cari beberapa informasi tambahan lainya !!
(Filenya ada di pertanyaan pertama)
File exe yang ada di Desktop windows, yang berkaitan dengan attack hacker ?
format flag : IDN_FLAG{Jawaban yang disoal}
Auhtor: Aditya Firman Nugroho

HOWWWWWWWWW

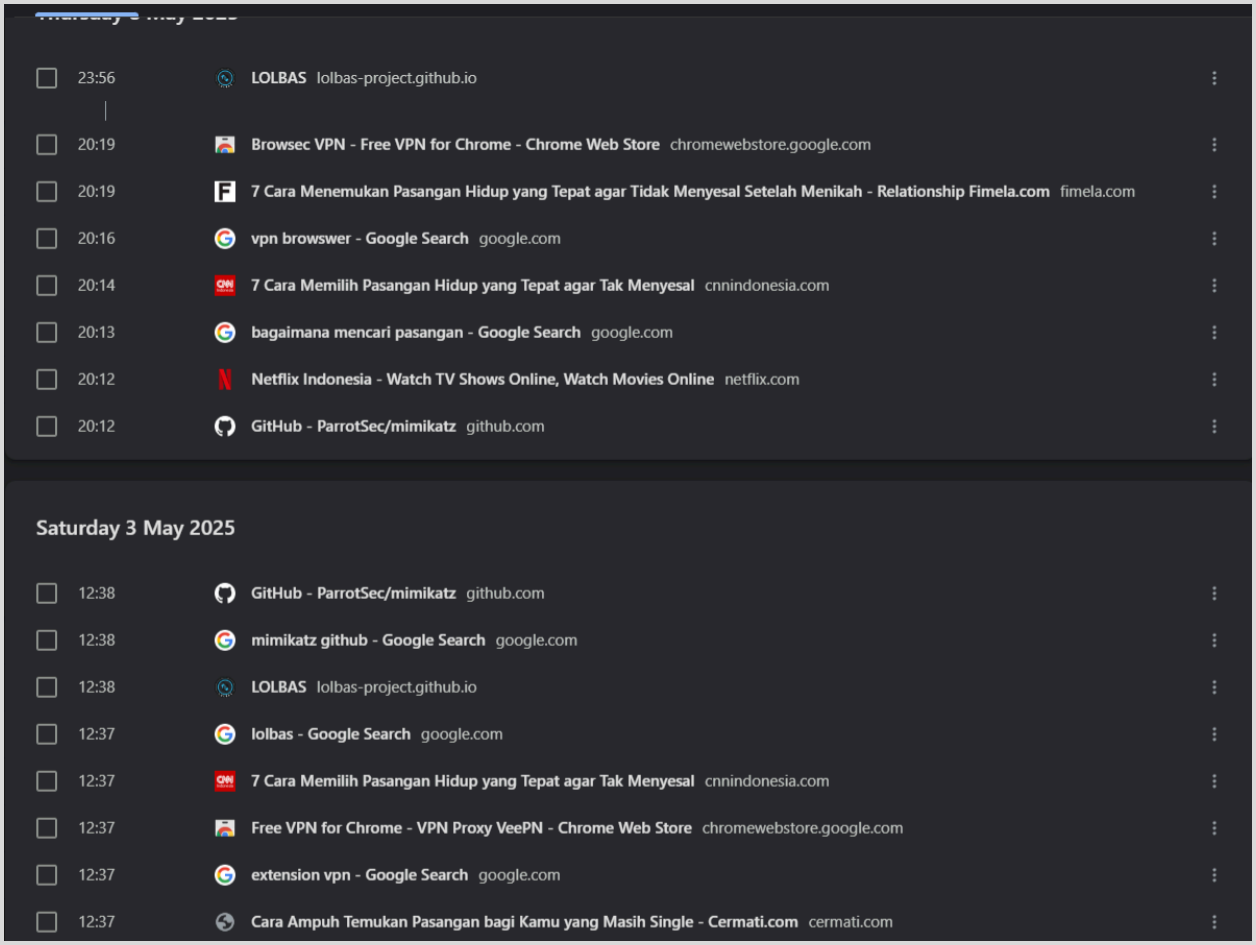
Browser Forensics

Browser Forensic 1

Ada Volation yang dilakukan oleh user di satu laptop, coba bantu forensic browsernya dong !!
(Filenya ada di pertanyaan pertama)
Tools apa yang di cari oleh user ?
format flag : IDN_FLAG{Jawaban yang disoal}

Author: Aditya Firman Nugroho

Given the zip folder, we can extract it and then copy the folder on Chrome Portable using the link here: https://portableapps.com/apps/internet/google_chrome_portable. The location to copy it is at ./GoogleChromePortable/Data/profile (create a new directory named profile here)
After that we can see the browsing history that is in the folder



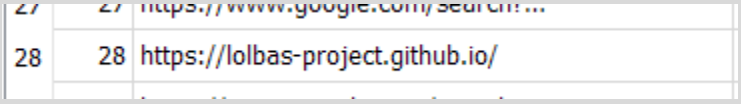
Looking at the history, i deduct that the tools that he wanted to search here is probably related to tools in github, so i just inputted mimikatz and it is correct
Flag : IDN_FLAG{mimikatz}

Browser Forensic 2

Ada Volation yang dilakukan oleh user di satu laptop, coba bantu forensic browsernya dong !!
(Filenya ada di pertanyaan pertama)
Website apa yang dicari oleh user berkaitan dengan Teknik Persistence, Privilage Escalation, DLL Injection etc ?
format flag : IDN_FLAG{Jawaban yang disoal}

Author: Aditya Firman Nugroho

Given the zip folder, we can extract it and then open this path (browser\Acuatiotion\User Data\Default), the we can see a History file containing SQLite database, add the .db extension then open it with SQLite browser in urls table to find this (based on the hint):



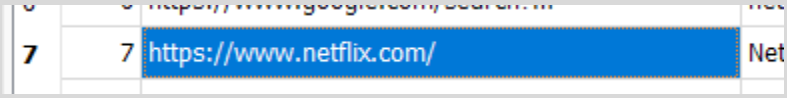
Flag : IDN_FLAG{https://lolbas-project.github.io/}

Browser Forensic 3

Ada Volation yang dilakukan oleh user di satu laptop, coba bantu forensic browsernya dong !!
Streaming Website yang ditonton oleh user ?
format flag : IDN_FLAG{Jawaban yang disoal}

Auhtor: Aditya Firman Nugroho

Given the zip folder, we can extract it and then open this path (browser\Acuatiotion\User Data\Default), the we can see a History file containing SQLite database, add the .db extension then open it with SQLite browser in urls table to find this (based on the hint):



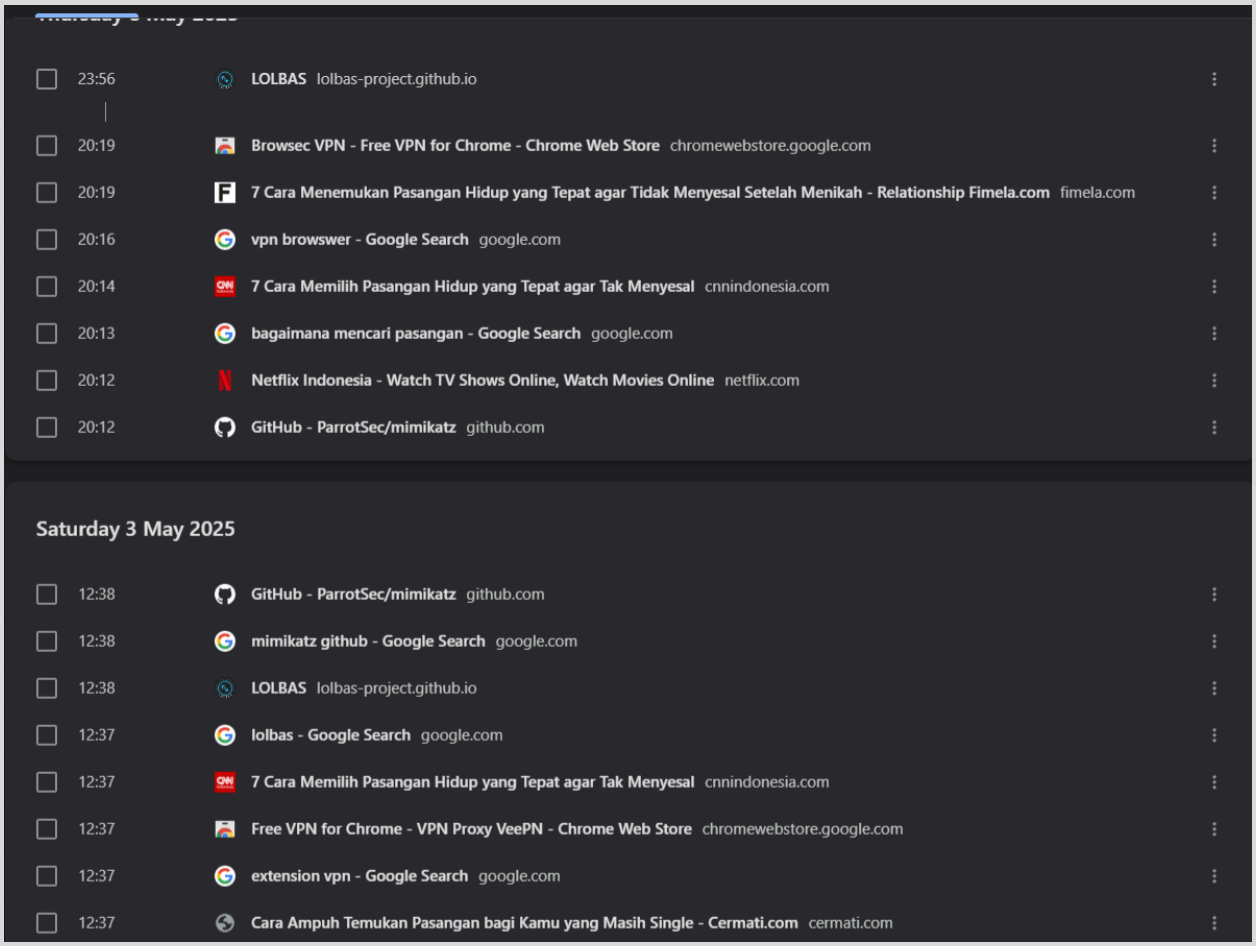
Flag : IDN_FLAG{https://www.netflix.com/}

Browser Forensic 4

Ada Volation yang dilakukan oleh user di satu laptop, coba bantu forensic browsernya dong !!
Vpn apa saja yang diinstal oleh user ?
format flag : IDN_FLAG{VPN_1-VPN_2} example : IDN_FLAG{IPSEC_SECURITY-L2TP_SECURITY}

Auhtor: Aditya Firman Nugroho

Looking back at our Browser History,



We can see that there's two VPNs here, which is VPN Proxy VeePN and Browsec VPN.
Flag : *IDN_FLAG{VPN_PROXY_VEEPN-BROWSEC_VPN}*

Browser Forensic 5

Ada Volation yang dilakukan oleh user di satu laptop, coba bantu forensic browsernya dong !!
Visit Duration di Website yang berkaitan dengan Persistence, Privilage Escalation, DLL Injection ?
format flag : IDN_FLAG{Jawaban yang disoal} example : XX:XX:XX.XXX

Auhtor: Aditya Firman Nugroho

Given the zip folder, we can extract it and then open this path (browser\Acuatiotion\User Data\Default), the we can see a History file containing SQLite database, add the .db extension then open it with SQLite browser in urls table to find this (based on the hint and the url id in browser forensic 2):

28	28	13390724280712417	27		805306368	0	32509459

The number is in ms and needs to be converted to HH:MM:SS.mmm format so we can get this flag:
Flag : IDN_FLAG{00:00:32.509}

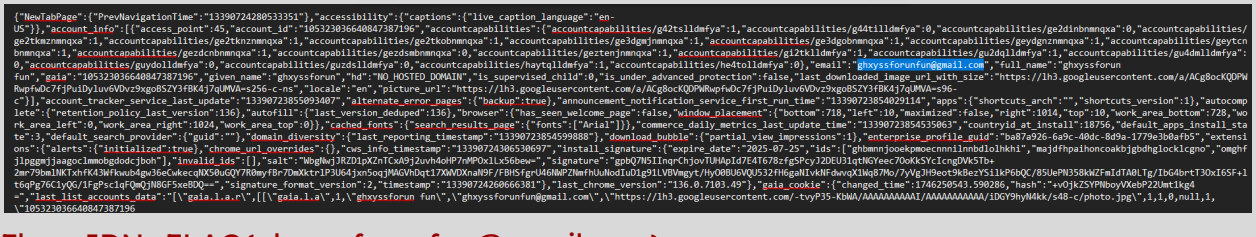
Browser Forensic 5

Ada Volation yang dilakukan oleh user di satu laptop, coba bantu forensic browsernya dong !!
Email yang digunakan pada browser ?

format flag : IDN_FLAG{Jawaban yang disoal}

Auhtor: Aditya Firman Nugroho

Given the zip folder, we can extract it and then open this path (browser\Acuatiotion\User Data\Default), the we can see a Preferences file, then just search for any email contained in the file:



Flag : IDN_FLAG{ghxyssforunfun@gmail.com}

Browser Forensic 7

AAda Volation yang dilakukan oleh user di satu laptop, coba bantu forensic browsernya dong !!
date_created pada email menggunakan tools DB Browser SQLite ?

format flag : IDN_FLAG{Jawaban yang disoal}

Auhtor: Aditya Firman Nugroho

Given the zip folder, we can extract it and then open this path (browser\Acuatiotion\User Data\Default), the we can see a Web data file, thenopen the file using SQLite browser, we can see date_created column in the autofil table:

	name	value	value_lower	date_created	date_last_used	count
	Filter	Filter	Filter	Filter	Filter	Filter
1	identifier	ghxyssforunfun@gmail.com	ghxyssforunfun@gmail.com	1746250363	1746250363	1

Flag : IDN_FLAG{1746250363}

Browser Forensic 8

Ada Volation yang dilakukan oleh user di satu laptop, coba bantu forensic browsernya dong !
url favicon, di website yang dicari oleh user ? (tidak berkaitan dengan hacker !!!)

format flag : IDN_FLAG{Jawaban yang disoal}

Auhtor: Aditya Firman Nugroho

Given the zip folder, we can extract it and then open this path (browser\Acuatiotion\User Data\Default), the we can see a History file containing SQLite database, add the .db extension then open it with SQLite browser in urls table to find this link, and the www.muslima.com frequently shows so we checked the favicon url

(based on the hint):

12	12	https://www.muslima.com/en/lp/paid-search/?...
13	13	https://sso-terra.clickocean.io/?redirect_uri=https%3A%2F%2Fwww.muslima.com%2Fen%2Flp%2Fpaid-...
14	14	https://www.muslima.com/en/lp/paid-search/?...
15	15	https://www.muslima.com/en/lp/paid-search/?...
16	16	https://www.googleadservices.com/pagead/aclk?sa=L&ai=DChcSEwjx-5aI0oaNAxV-q2YCHUXXCYEYABAAGgJzbQ&co=
17	17	https://www.muslima.com/en/lp/paid-search/?...
18	18	https://sso-terra.clickocean.io/?redirect_uri=https%3A%2F%2Fwww.muslima.com%2Fen%2Flp%2Fpaid-...
19	19	https://www.muslima.com/en/lp/paid-search/?...
20	20	https://www.muslima.com/en/lp/paid-search/?...

Flag :
IDN_FLAG{https://www.muslima.com/lp/paid-search/terra-assets/images/favicon-8b7d9ccfa1-3.ico}

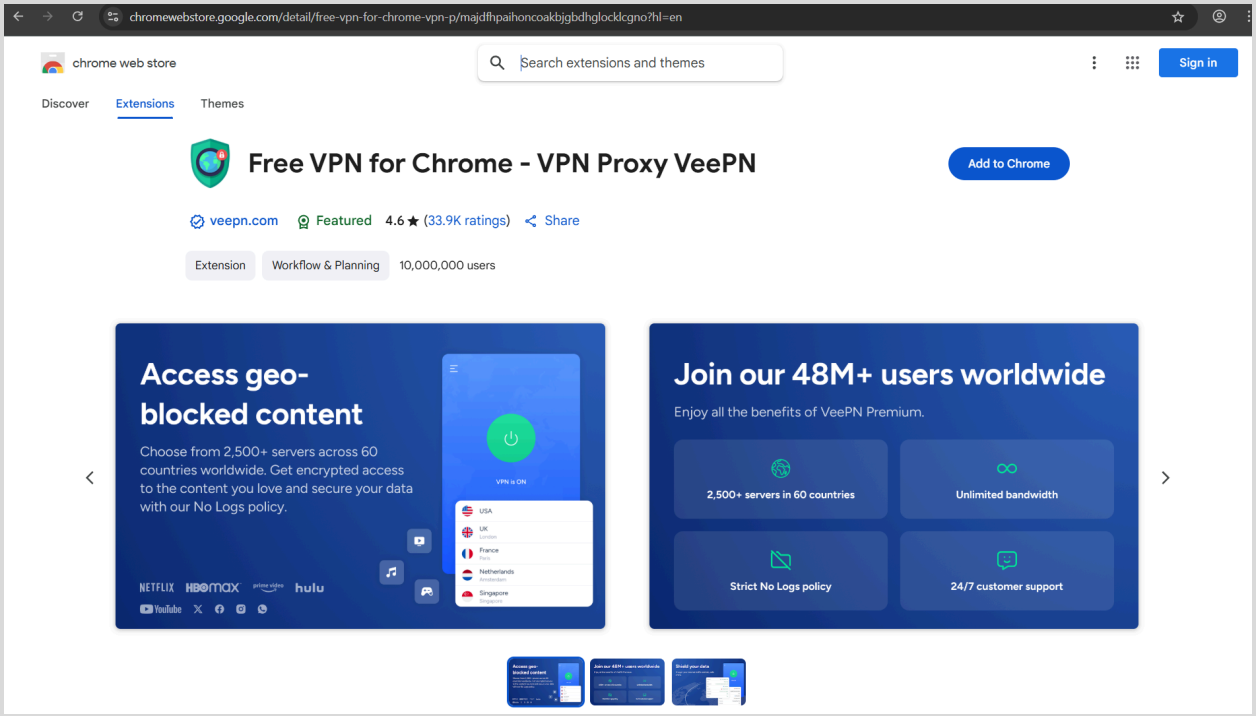
Browser Forensic 9

Ada Volation yang dilakukan oleh user di satu laptop, coba bantu forensic browsernya dong !
extension id dengan icon salah satu vpn yang diinstal V.. !

format flag : IDN_FLAG{Jawaban yang disoal}

Auhtor: Aditya Firman Nugroho

Since the vpn starts with the letter “V”, we can assume its VPN Proxy VeePN. We then navigate into the extensions tab to find its extension id, which is in the URL parameter.



Flag : IDN_FLAG{majdfhpaihoncoakbjgbdhglocklcgno}

Browser Forensic 10

Ada Volation yang dilakukan oleh user di satu laptop, coba bantu forensic browsernya dong !! (Filenya ada di pertanyaan pertama)

Version vpn V.. yang diinstal oleh user ?

format flag : IDN_FLAG{Jawaban yang disoal}

Given the zip folder, we can extract it and then open this path browser\Acuatiotion\User Data\Default\Extensions\majdfhpaihoncoakbjgbdhglocklcgno\3.4.3_0), we can already see that the VPN extension version is 3.4.3_0:

Flag : IDN_FLAG{3.4.3_0}

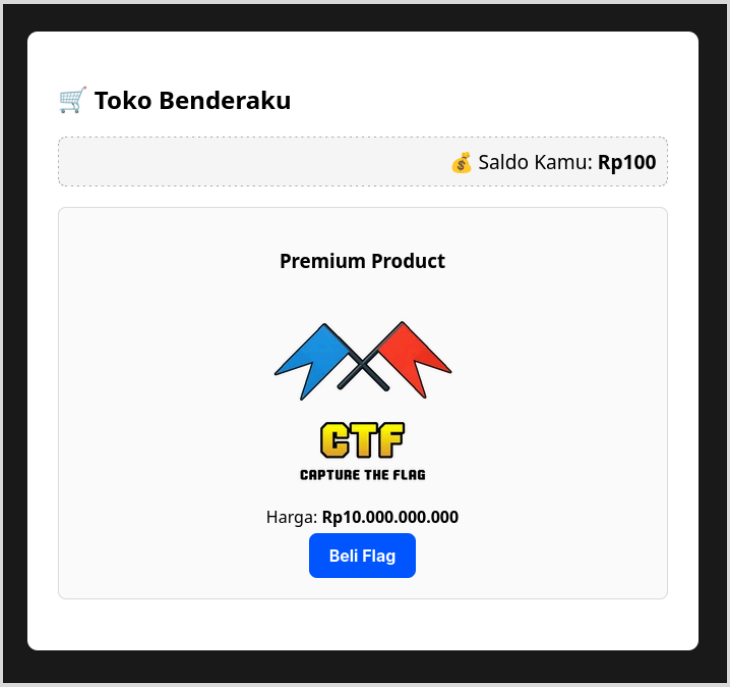
Web Exploitation

Hidden Buy Flag

Tim ID-Network baru saja membuat website, tetapi tim internal saja yang dapat masuk ke dalam website tersebut dengan pointing ke website (idn.id), kami menyuruh kalian para (Pentester) untuk mencoba menemukan celah disana dan masuk ke website tersebut. Didalam website tersebut kalian harus membeli sebuah Flag dengan harga 100000000.

Author : Faiz Ahmad Habibi

Given a website, with a shop as follows:



There must be some place where the balance is stored, turns out it's just a hidden value sent using a form.

```
<p>
  Harga:
  <strong>Rp10.000.000.000</strong>
</p>
<form action="" method="POST">
  <input type="hidden" name="saldo" value="100">
  <button type="submit">Beli Flag</button>
</form>
</div>
</div>
```

So we just need to include it in a POST payload.

```
Shisones@Arch-Thinkpad [~]
➜ http POST https://ctf.solusiber.com/buy_the_flag/ saldo=10000000000 --form | grep IDN
      <p style='color:red; padding-top:3px; font-weight:bold;'>IDN_FLAG{h3ader_wh1telist_4nd_p4r4m3ter_t4mp3r1ng_v3ryy_3zzz}</p>
```

Flag : IDN_FLAG{h3ader_wh1telist_4nd_p4r4m3ter_t4mp3r1ng_v3ryy_3zzz}

Konoha Breach

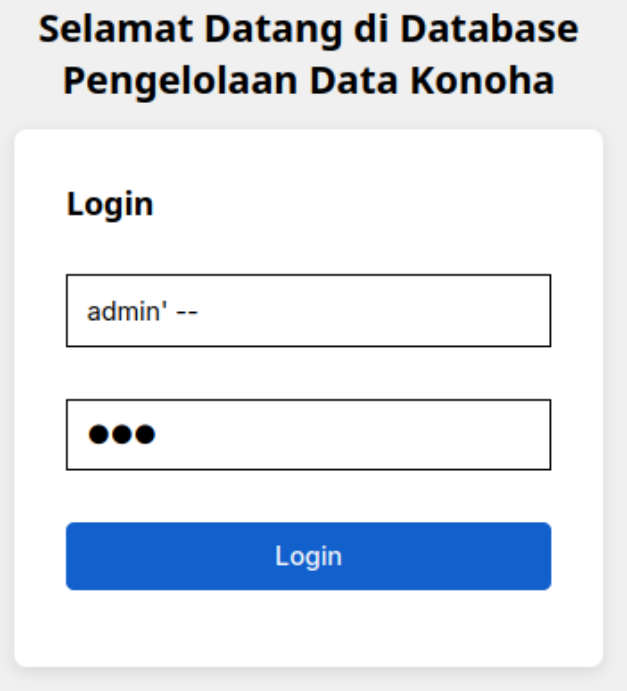
Desa Konoha baru saja meluncurkan sistem data tabel internal untuk para ninja tingkat tinggi. Sistem ini hanya bisa diakses setelah login dengan kredensial resmi admin. Namun, rumor menyebutkan bahwa sistem ini dibangun tergesa-gesa oleh seorang Chuunin yang baru belajar PHP. Konon, ada celah klasik yang memungkinkan siapa pun melewati sistem login dan mengakses dashboard rahasia tanpa kredensial! Bocoran pertama yang muncul berisi daftar shinobi aktif dan lokasi markas Anbu. Keamanan Konoha kini dalam bahaya…
Bisakah kamu menyusup ke sistem tanpa login dan menemukan yang tersembunyi?
Author: Rafly Permana

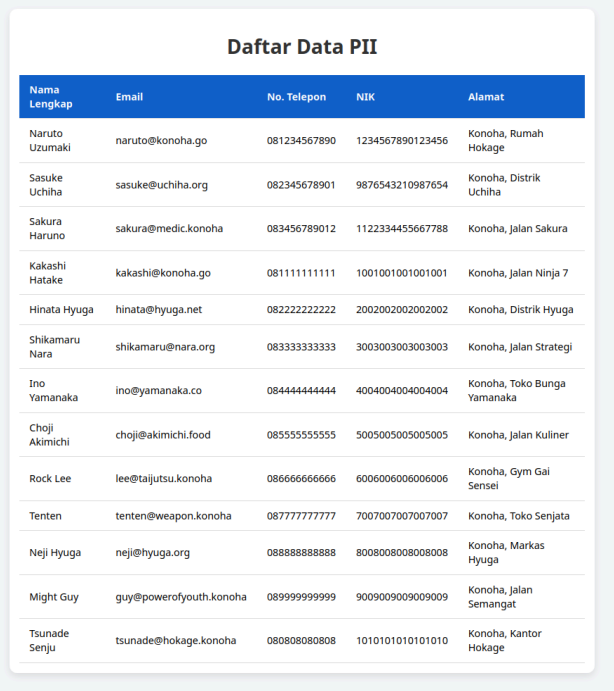
Given a link to the website. when accessed, it's a simple login form built with php. since the description shows that it has a very common exploit, let's try SQLi.

Warning: SQLite3::query(): Unable to prepare statement: 1, near "1": syntax error in **/var/www/html/flag.php** on line **14**

Login gagal.

Looks like it has SQLite, and since the POST request is transferred to /flag.php. let's inject it with some SQLi payload

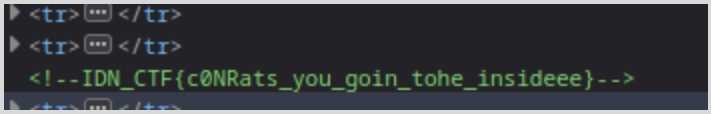




Daftar Data PII

Nama Lengkap	Email	No. Telepon	NIK	Alamat
Naruto Uzumaki	naruto@konoha.go	081234567890	1234567890123456	Konoha, Rumah Hokage
Sasuke Uchiha	sasuke@uchiha.org	082345678901	9876543210987654	Konoha, Distrik Uchiha
Sakura Haruno	sakura@medic.konoha	083456789012	1122334455667788	Konoha, Jalan Sakura
Kakashi Hatake	kakashi@konoha.go	081111111111	1001001001001001	Konoha, Jalan Ninja 7
Hinata Hyuga	hinata@hyuga.net	082222222222	2002002002002002	Konoha, Distrik Hyuga
Shikamaru Nara	shikamaru@nara.org	083333333333	3003003003003003	Konoha, Jalan Strategi
Ino Yamanaka	ino@yamanaka.co	084444444444	4004004004004004	Konoha, Toko Bunga Yamanaka
Choji Akimichi	choji@akimichi.food	085555555555	5005005005005005	Konoha, Jalan Kuliner
Rock Lee	lee@taijutsu.konoha	086666666666	6006006006006006	Konoha, Gym Gai Sensei
Tenten	tenten@weapon.konoha	087777777777	7007007007007007	Konoha, Toko Senjata
Neji Hyuga	neji@hyuga.org	088888888888	8008008008008008	Konoha, Markas Hyuga
Might Guy	guy@powerofyouth.konoha	089999999999	9009009009009009	Konoha, Jalan Semangat
Tsunade Senju	tsunade@hokage.konoha	080808080808	1010101010101010	Konoha, Kantor Hokage

But where's the flag?



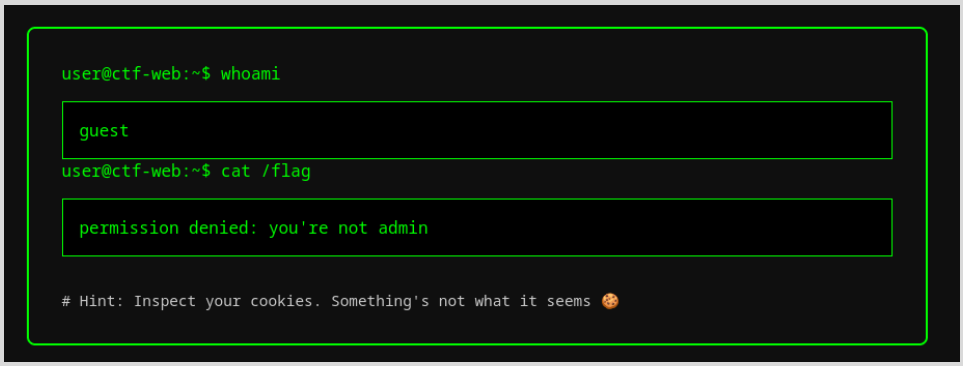
it's in the source code

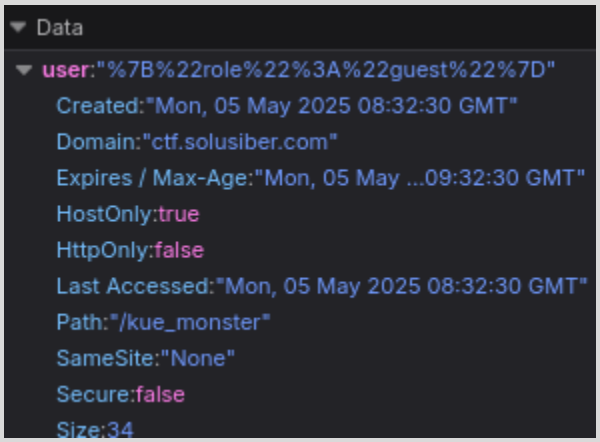
Flag : IDN_CTF{c0NRats_you_goin_tohe_insideee}

Kue Monster

Kamu cuma dikasih kue biasa? Bosen. Upgrade kue-mu jadi kue sultan dan lihat apa yang bisa kamu lakukan! (Jangan makan beneran ya)
Author : Rafly Permana

Looks like a cookie webex challenge. accessing the website returns this webpage





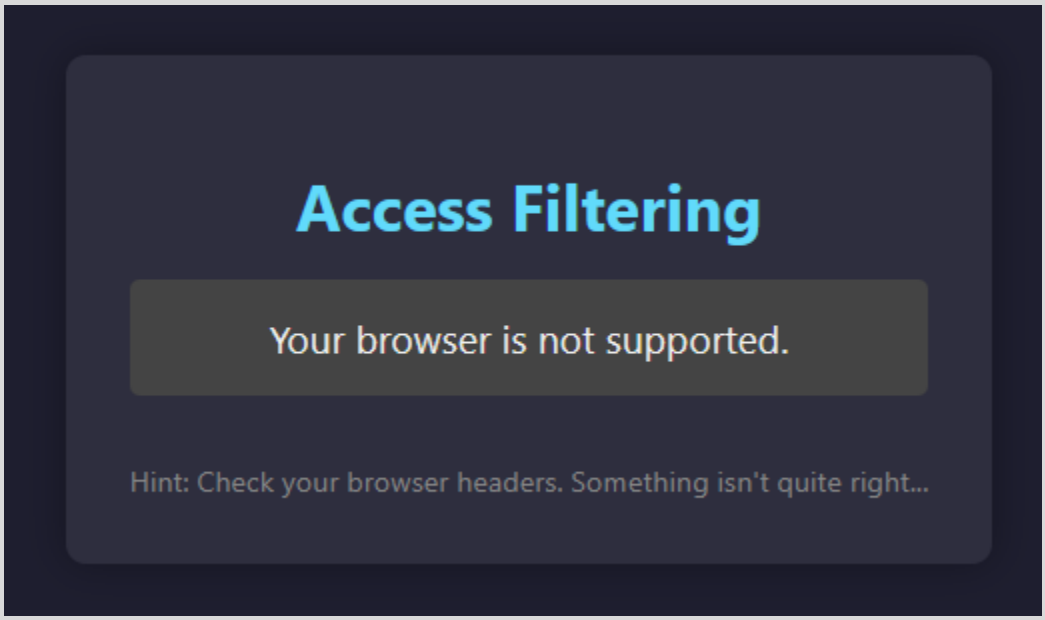
just change the guest to root, and we got the flag.

Flag : IDN_CTF{Y0u_@rE_TH@_C00K|e_M@st$r}

Support Force

Ini klub eksklusif buat agen rahasia. Browser biasa? Maaf, Anda tidak terdaftar. Tapi kalau kamu bisa pura-pura jadi "Agent hackme", pintu rahasia mungkin bakal terbuka buatmu.

Author : Rafly Permana



Since it says it doesn't accept regular browser, the solution is simple: just change the User-Agent header to hackme.

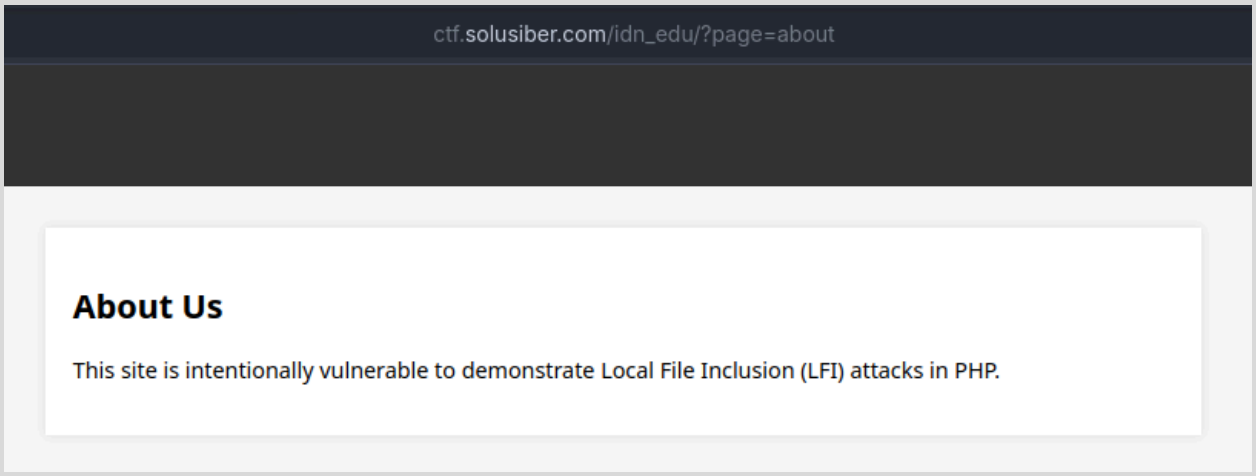
```
Shisones@Arch-Thinkpad  [~]
➜ http GET https://ctf.solusiber.com/support_force/ User-Agent:"hackme" | grep IDN
    IDN_CTF{r7x9_uaSwitch_delta44}
```

Flag : IDN_CTF{r7x9_uaSwitch_delta44}

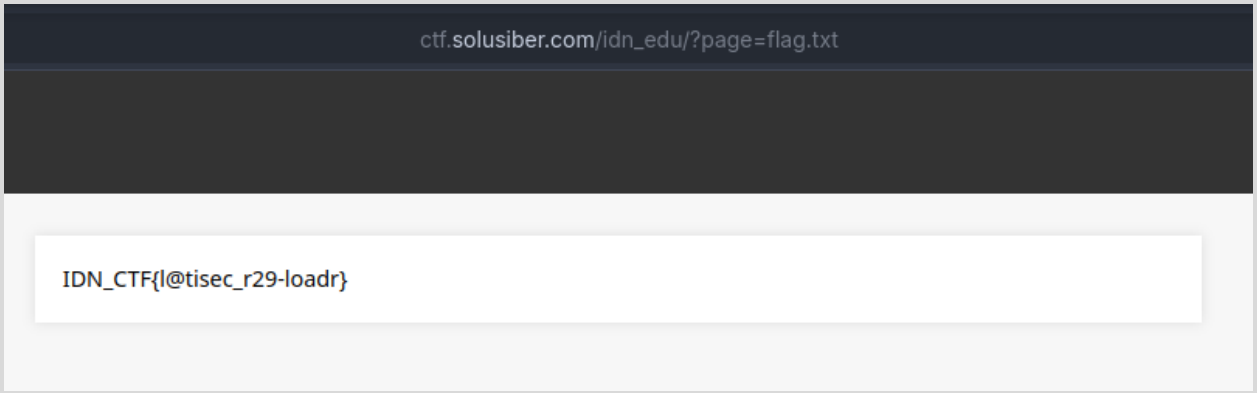
IDN Education

Siapa sangka file-file tersembunyi di balik input sederhana? Coba kamu buka celahnya, biar file yang terpendam itu bisa keluar. Siapa tahu ada kejutan!
Author : Rafly Permana

Right away, it shows us that it's an LFI vuln on the /about page



So, using a common LFI method, we redirected the page to flag.txt



Flag : IDN_CTF{l@tisec_r29-loadr}

Beyond Way

Mungkin kamu nggak pernah diajari buat berjalan keluar dari jalan yang benar... tapi kalau kamu bisa, kamu bakal dapetin sesuatu yang terlarang. Ayo jalanin manipulasi path-nya! 🏃‍♂️💨

Author : Rafly Permana

Sounds like a path traversal vulnerabilirty, let's check the website.

Going to the about page gives us a very peculiar url format, it fetches a file and then render it. So we tried something here by typing ../../ in the url parameter:

Looks like it's an LFI vuln as we predicted. assuming the file is flag.txt, we tried to send a get request with the payload

```
Shisones@Arch-Thinkpad  [~]
➜ http GET "https://ctf.solusiber.com/search_free/?file=../../../../var/www/html/flag.txt" | grep IDN
      IDN_CTF{tvec-resolver_41}
```

Flag : IDN_CTF{tvec-resolver_41}

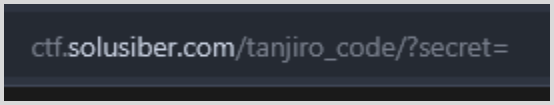
Code Analysis

Tanjiro terus berlatih tanpa henti untuk menguasai Hinokami Kagura demi mengalahkan iblis Bulan Atas. Bantu dia membuka kekuatan sejatinya dengan menganalisis kode yang diberikan. Kunci untuk tingkat kekuatan berikutnya terletak pada pemahaman alur kerjanya kode.
Author : Mohamad Fattyr

Given a website with its source code:

```php
<?php
$input = $_GET["secret"] ?? "";

$clean_input = strtolower(str_replace(" ", "", $input));
$result = preg_replace("/".preg_quote($keyword, '/')."/", "", $clean_input, 1);

if ($result === "tanjiro") {
    echo $flag;
}

?>
```

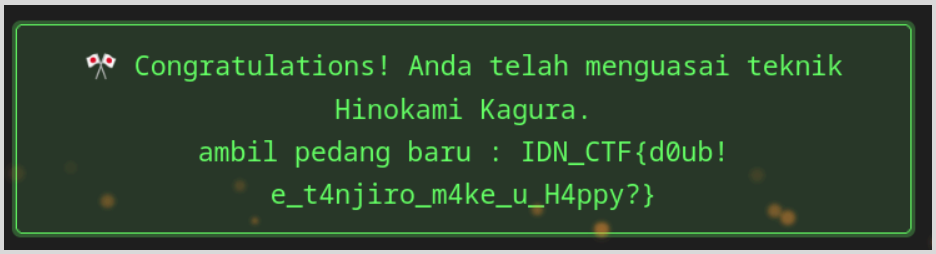
This is an interesting one, basically it will get the secret param from the url, parse it, clean the spaces, make it lowercase, and remove a "keyword" from the string, sending just "tanjiro" to the url param will clear it



but if you look closer, it will only delete one instance of the keyword.

```
', "", $clean_input, 1);
```

so just send 2 tanjiros:



Flag : IDN_CTF{d0ub!e_t4njiro_m4ke_u_H4ppy?}

ID Networkers

Sebuah situs publik baru saja diluncurkan ID-Networkers. Tampilannya sederhana dan tidak mencurigakan—hanya halaman beranda dengan ucapan "Selamat Datang di ID-Networkers" dan beberapa tamabahan lainya.
Namun, informasi mengatakan bahwa developer situs ini terlalu percaya pada "aturan" yang ditulis untuk mesin pencari. Mereka menyembunyikan direktori rahasia dengan harapan crawler tidak akan melihatnya...
Tapi kamu bukan crawler, kamu seorang penyusup yang teliti.
Author: Rafly Permana

Since there are mentions of 'crawlers', we assume that it has to do with robots.txt, a file that hides certain websites from search engine and web crawlers.

```
User-agent: *
Disallow: /asdsa024nsfd01372021.html
```

Sure enough, it's there.

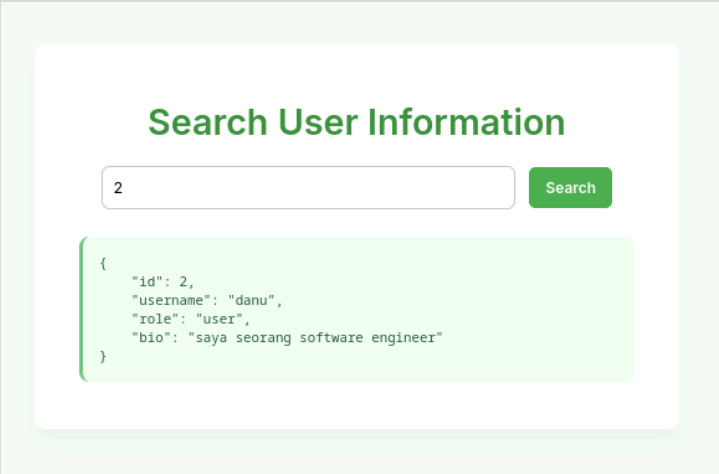
IDN_CTF{@W*_FOuN&_th@_#|**$N_F|@&}

Flag : IDN_CTF{@W*_FOuN&_th@_#|**$N_F|@&}

I'm Not Me, You're Me

Bukan cuma kamu yang punya profil! Coba-coba ganti ID di URL dan lihat apakah kamu bisa jadi orang lain. Mungkin kamu bisa mengakses sesuatu yang seharusnya nggak buatmu!
Author: Rafly Permana

Opening the website shows us a simple json data viewer.



since there might be lots of ids, i simply made a script that scans each id for the flag

```
Shisones@Arch-Thinkpad [~]
➜ for x in {0..100}; do
  http GET "https://ctf.solusiber.com/user_information/?id=$x" | grep "IDN"
done
    &amp;quot;flag&amp;quot;: &amp;quot;IDN_CTF{Y0u_FF0D_the_heN_admin}&amp;quot;
```

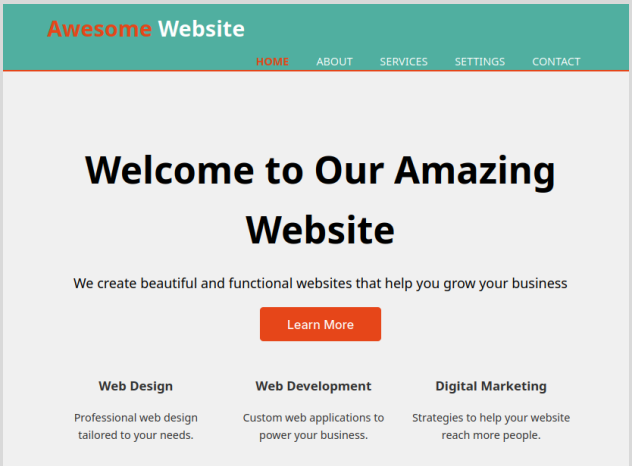
Flag : IDN_CTF{Y0u_FF0D_the_heN_admin}

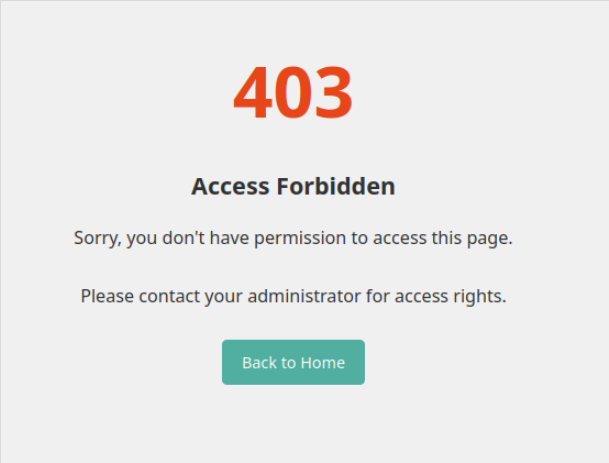
Awesome Website

CARI!!

Author : Mohamad Fattyr

Given a website with a pretty restrictive access as follows:





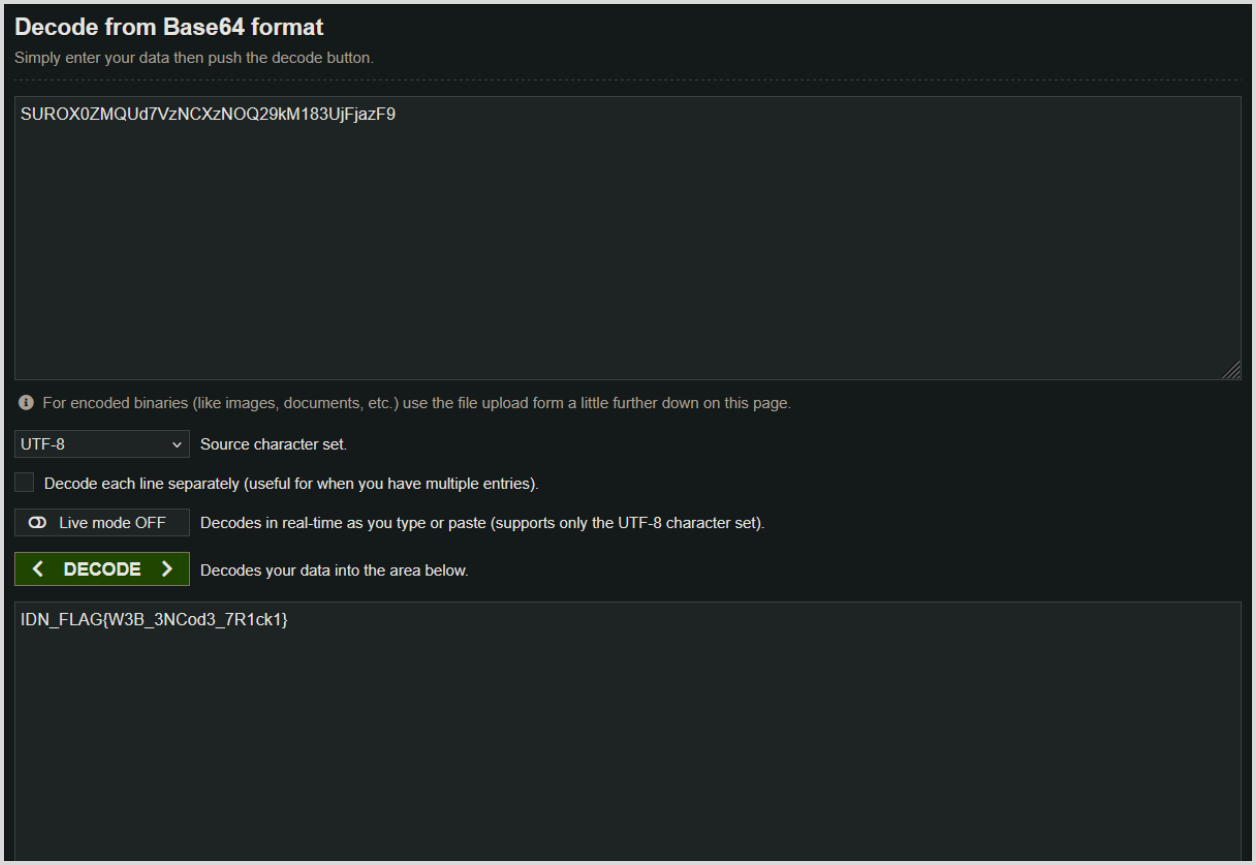
Looking at the source code, it shows a very peculiar js snippet

```
// Site configuration object const config = { // Main site configuration site: { name: "Awesome Website", version: "1.5.2", domain: "example.com", environment: "production", debug: false, maintenanceMode: false, cookiesEnabled: true }, // Feature toggles features: { darkMode: false, comments: true, userAccounts: true, notifications: true }, // API configuration api: { baseUrl: "https://api.example.com/v2", timeout: 5000, retryAttempts: 3, cacheTTL: 3600, accessToken: "SUROX0ZMQUd7VzNCXzNOQ29kM183UjFjazF9" // Access token for API authentication }, // Analytics configuration analytics: { provider: "GoogleAnalytics", trackingId: "UA-XXXXX-Y", anonymizeIp: true, sessionTimeout: 30 }, // Performance settings performance: { lazyLoading: true, prefetch: true, minification: true, compression: true } }; // Modal functionality const modal = document.getElementById("info-modal"); const learnMoreBtn = document.getElementById("learn-more"); const closeBtn = document.querySelector(".close-btn"); learnMoreBtn.addEventListener("click", function() { modal.style.display = "block"; }); closeBtn.addEventListener("click", function() { modal.style.display = "none"; }); window.addEventListener("click", function(e) { if (e.target === modal) { modal.style.display = "none"; } }); // Forbidden page functionality const forbiddenPage = document.getElementById("forbidden-page"); const backBtn = document.getElementById("back-btn"); // Make all navigation links show forbidden page const navLinks = [ document.getElementById("about-link"), document.getElementById("services-link"), document.getElementById("settings-link"), document.getElementById("contact-link") ]; navLinks.forEach(link => { link.addEventListener("click", function(e) { e.preventDefault(); forbiddenPage.style.display = "block"; }); }); // Home link should not show forbidden document.getElementById("home-link").addEventListener("click", function(e) { e.preventDefault(); forbiddenPage.style.display = "none"; }); // Back button backBtn.addEventListener("click", function(e) { e.preventDefault(); forbiddenPage.style.display = "none"; }); // Initialize site function initSite() { if (config.site.debug) { console.log("Site initialized with config:", config); } // Check for maintenance mode if (config.site.maintenanceMode) { alert("This website is currently under maintenance. Some features may not be available."); } // Set up API connection setupAPI(); // Set up analytics if (config.analytics.trackingId) { initAnalytics(); } } // Setup API connection function setupAPI() { if (config.site.debug) { console.log("API initialized with base URL:", config.api.baseUrl); console.log("Using access token:", "********"); // Don't log actual token in production } // In a real application, this would set up the API connection } // Initialize analytics function initAnalytics() { if (config.site.debug) { console.log("Analytics initialized with provider:", config.analytics.provider); } // In a real application, this would set up analytics } // Initialize the site when DOM is loaded document.addEventListener("DOMContentLoaded", initSite);
```

We got the API token

```
// API configuration
api: {
    baseUrl: "https://api.example.com/v2",
    timeout: 5000,
    retryAttempts: 3,
    cacheTTL: 3600,
    accessToken: "SUROX0ZMQUd7VzNCXzNOQ29kM183UjFjazF9" // Access token for API authentication
},
```

The token looks suspicious, so we tried decoding it using a base64decoder and we got the flag:



Flag : IDN_FLAG{W3B_3NCod3_7R1ck1)

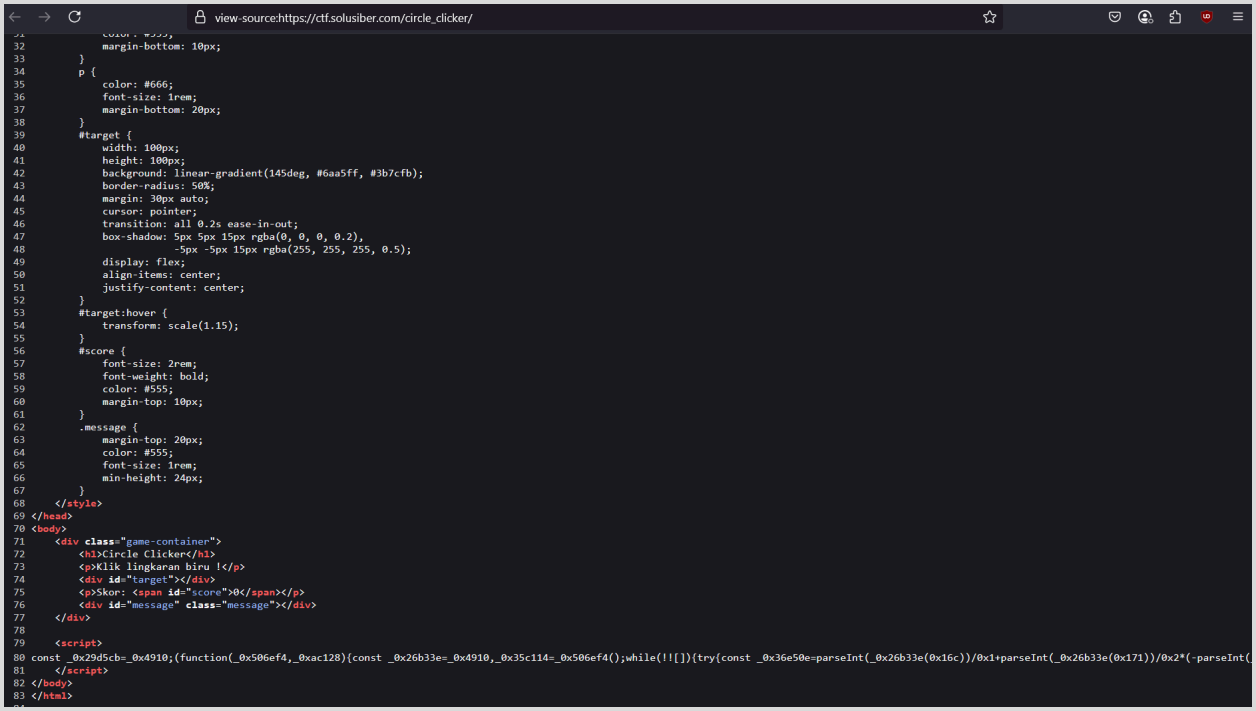
Circle Clicker

Click Sampai 1000 kali!

Flagnya Di Encode Dengan Encoder yang sama dengan bitcoin dan solana

Author : Mohamad Fattyr

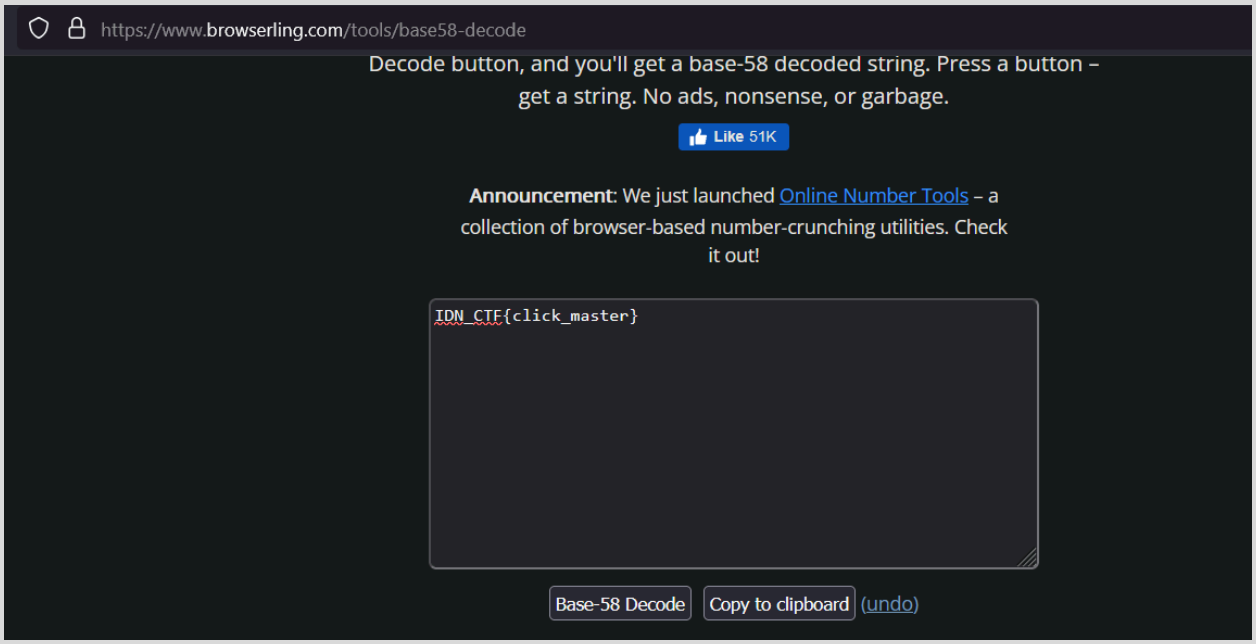
Given the website link, the first thing i do is to inspect the source which yielded this:



As you can see, there is an obfuscated javascript here that's doing something, so i deobfuscated it and it yielded this:

```
      setTimeout(() => messageEl.textContent = '', 0x7d0);
    }
});
function revealSecret() {
    secretUnlocked = true;
    console.log("%cSelamat! Kamu menemukan fungsi rahasia!", "color: green; font-size: 16px;");
    console.log("%cBagian pertama flag: 5WJoJxz5CCVWDSE", "color: blue; font-size: 14px;");
    console.log("%cUntuk bagian kedua, coba berfikir sambil bermain click!", "color: purple;");
}
setInterval(function () {
    if (score >= 0x3e8 && secretUnlocked) {
        console.log("%cLuar biasa! Kamu mendapatkan bagian kedua flag: master}", "color: green; font-size: 14px;");
        console.log("%cFlag lengkap: 5WJoJxz5CCVWDSEpH4E1n77BT5Fec", "color: red; font-size: 16px; font-weight: bold;");
        secretUnlocked = false;
        target.style.background = "linear-gradient(145deg, gold, orange)";
        messageEl.textContent = "Kamu menemukan flag!";
        messageEl.style.color = "green";
        messageEl.style.fontWeight = "bold";
    }
}, 0x3e8);
console.log("Selamat datang di Circle Clicker! Game sederhana atau...?");
```

As you can see, we got the complete flag instantly, so after that we can decode it using the same solana and bitcoin encoder, which is base58 to get the flag:

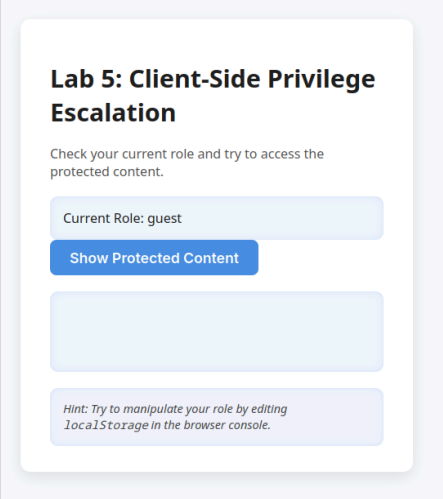


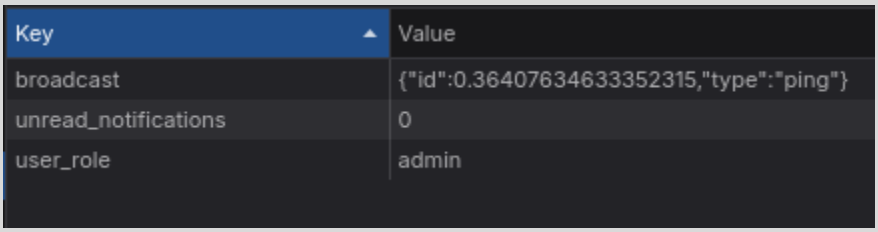
Flag : IDN_CTF{click_master}

Web 303

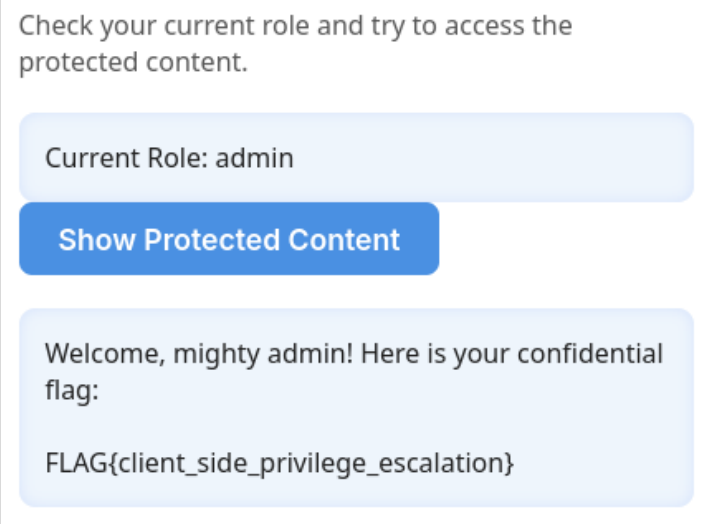
Client Side Privilege Escalation

Given a website and the hint to the challenge:





Simply change the role to admin, and then show the protected content.

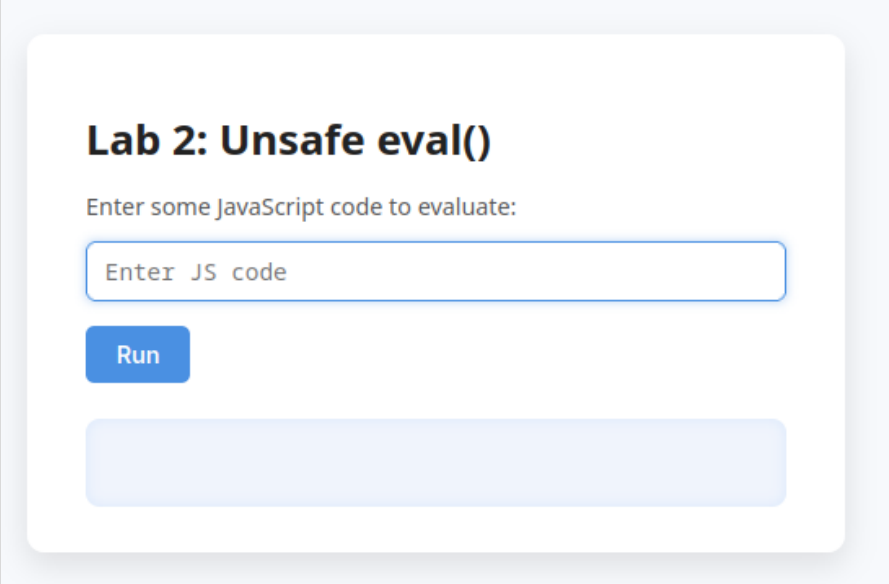


Flag : FLAG{client_side_privilege_escalation}

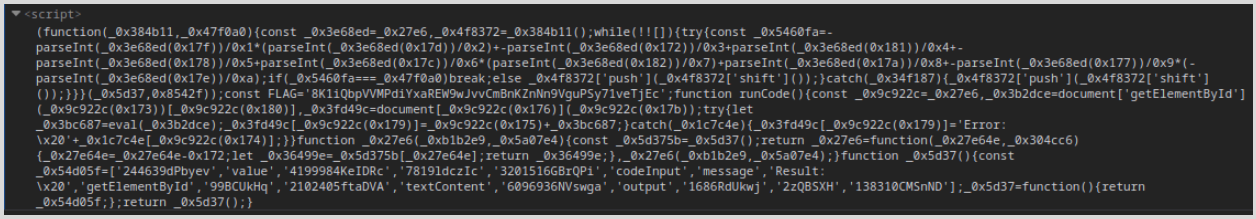
Unsafe Eval

Flagnya Di Encode Dengan Encoder yang sama dengan bitcoin dan solana
Author: Rafly Permana

Given a website with an alleged unsafe eval() function:

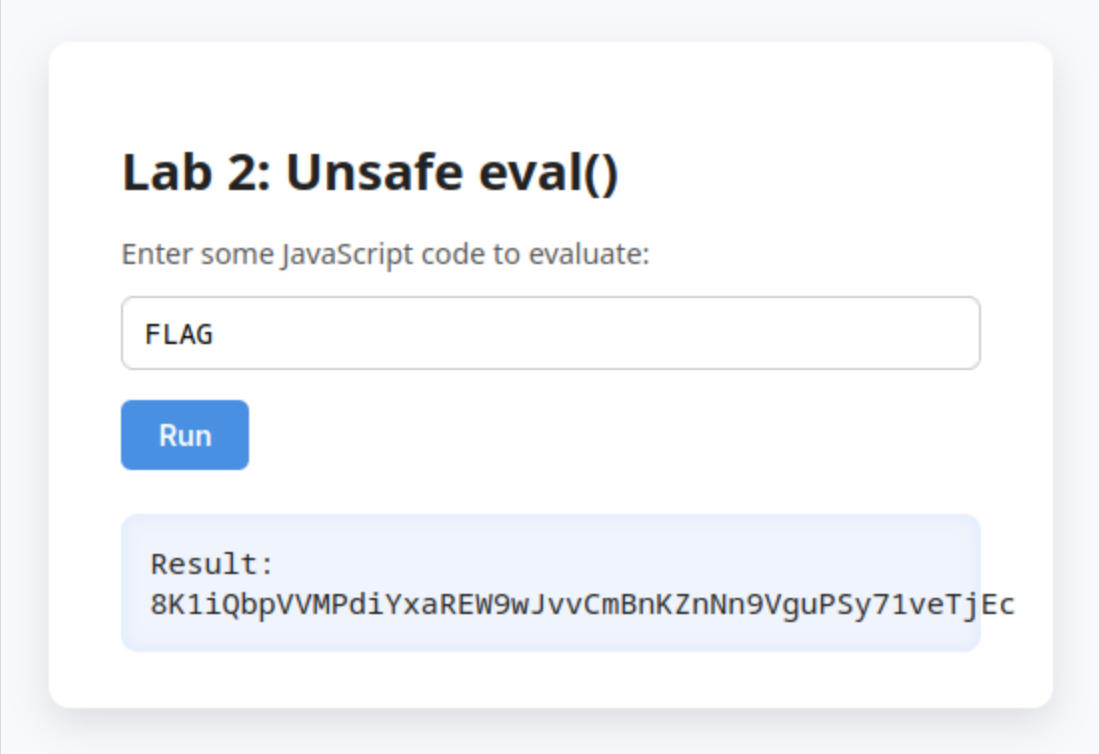


looking at the source code, it's written in the client-side. However it's an obfuscated javascript:

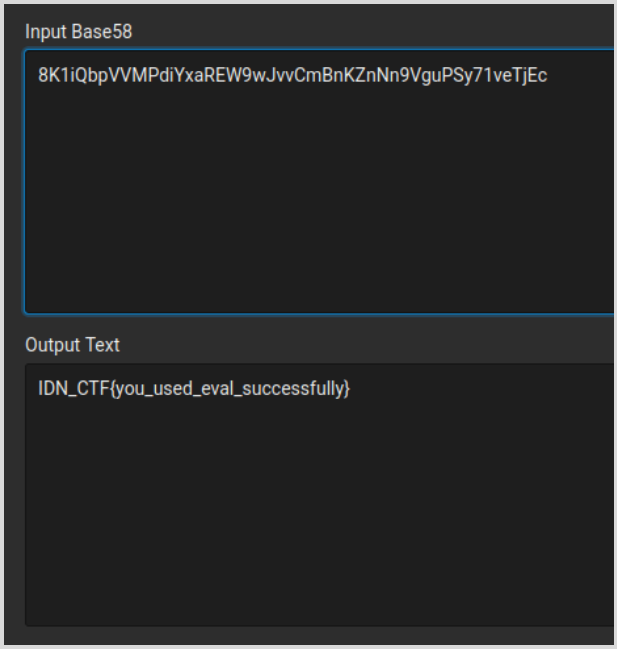



```
;const FLAG='8K1iQbpVVMPdiYxaREW9wJvvCmBnKZnNn9VguPSy71veTjEc
```

It turns out that there is indeed a vulnerable eval() function, However, the flag is stored in a plaintext format, making it easier to check by simply doing eval("FLAG")



The description says it was encoded in the same encoding as bitcoin and solana, which uses Base58



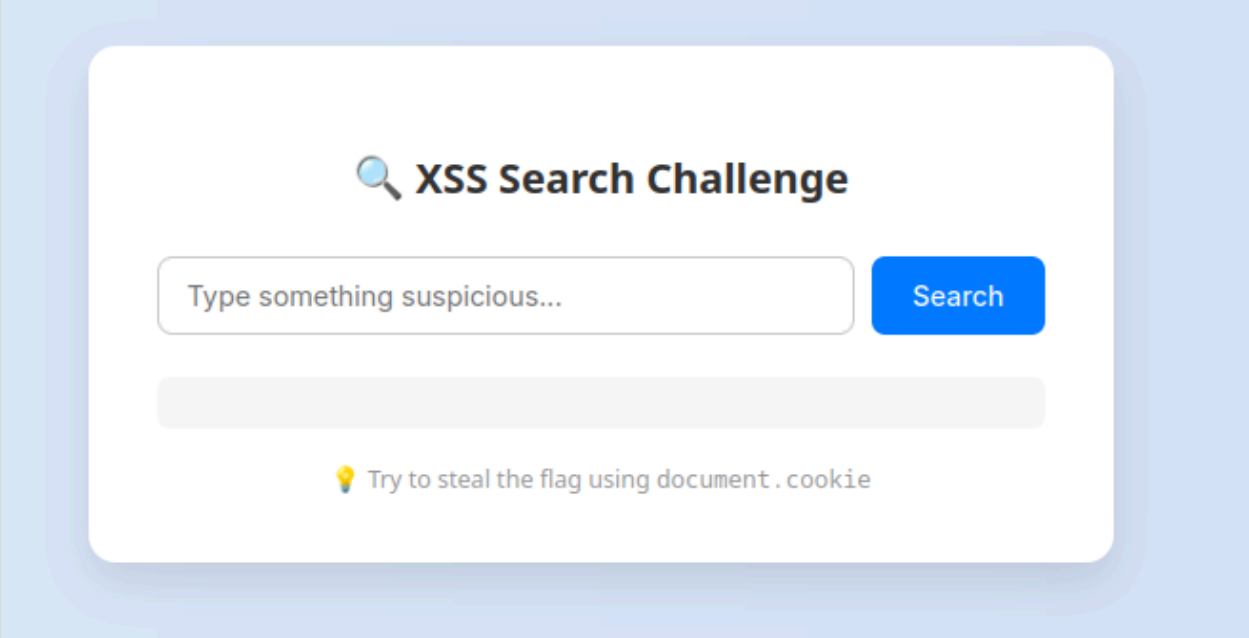
Flag : IDN_CTF{you_used_eval_successfully}

XSS

CURI!

Author : Rafly Permana

Given a website that's seemingly vulnerable to XSS.



The hint is there, so we just execute it and got the flag

flag	IDN_FLAG{XSS_C00K13_ST34L3R}

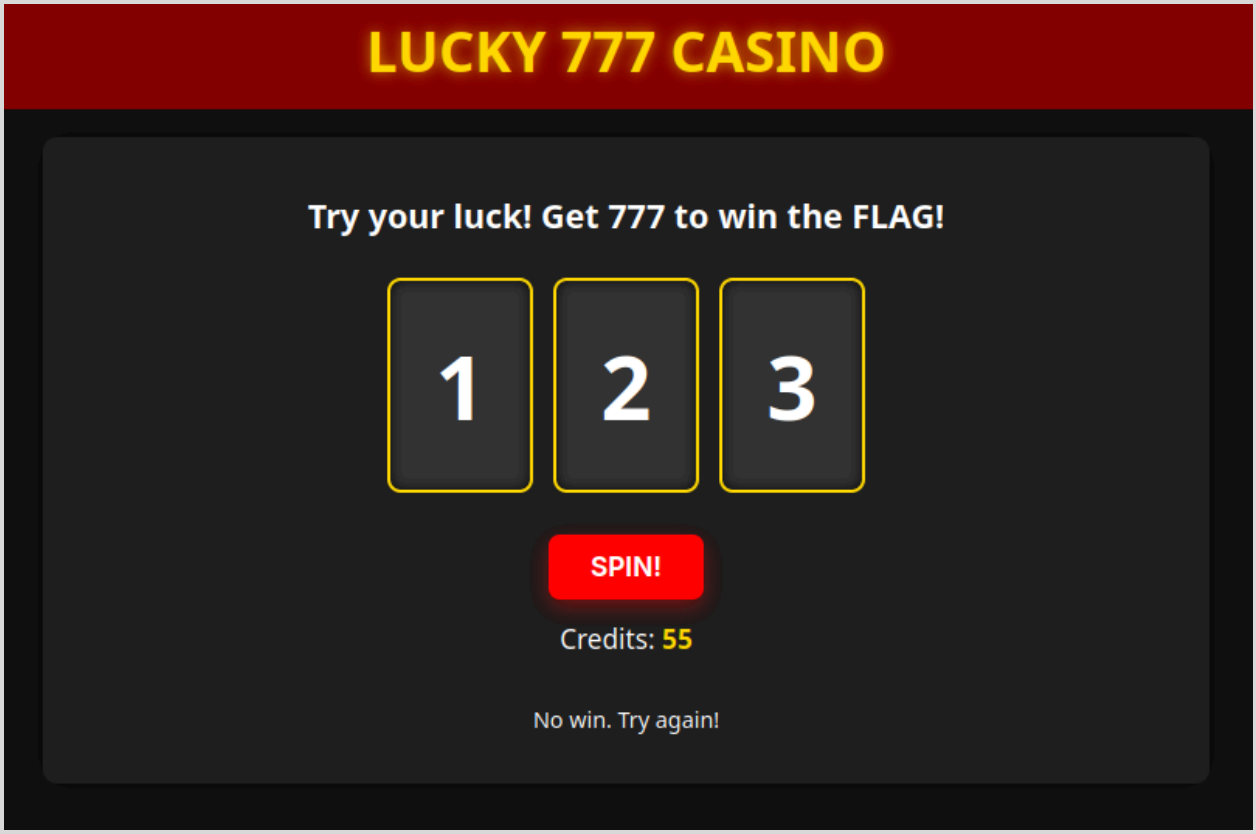
Flag : IDN_FLAG{XSS_C00K13_ST34L3R}

Casino 777

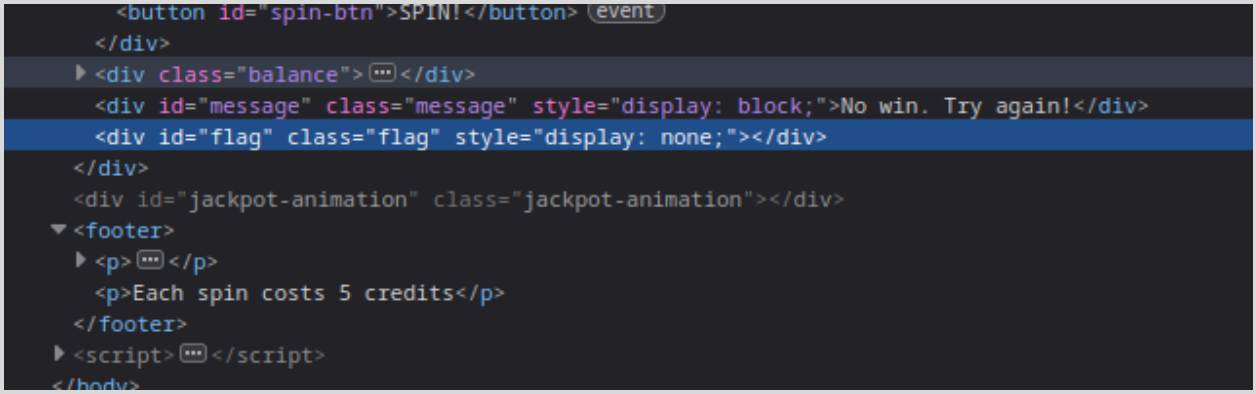
Ternyata aplikasi ini menerima input melalui query parameter. Cobalah eksplorasi URL dan manipulasi nilai slot-nya.

mungkin ada sesuatu yang jika sudah lengkap baru merespon

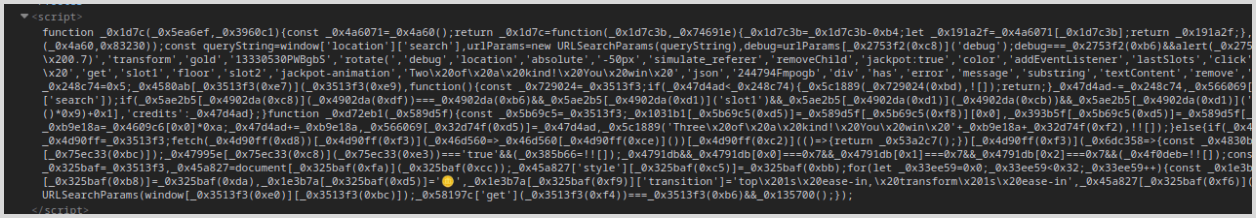
Author : Mohamad Fattyr

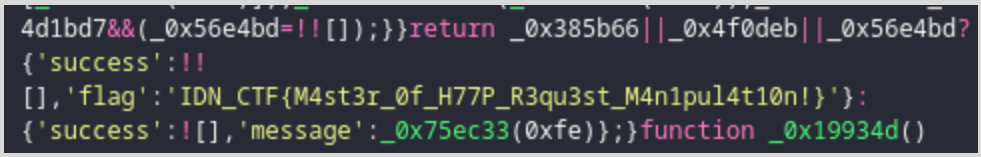


I'm not a gambler, so i would just look at the client source code.



There's something interesting over here, and more obfuscated js

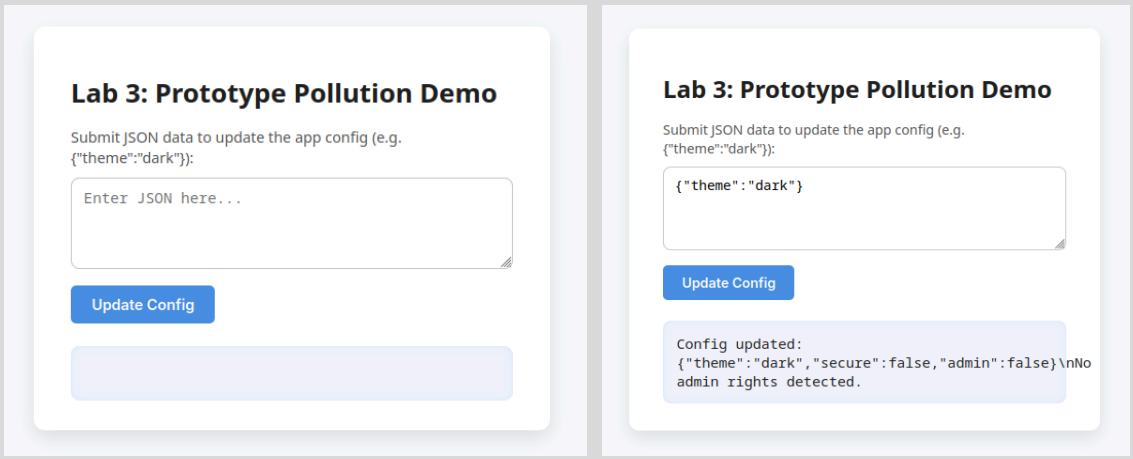




Flag : IDN_CTF{M4st3r_0f_H77P_R3qu3st_M4n1pul4t10n!}

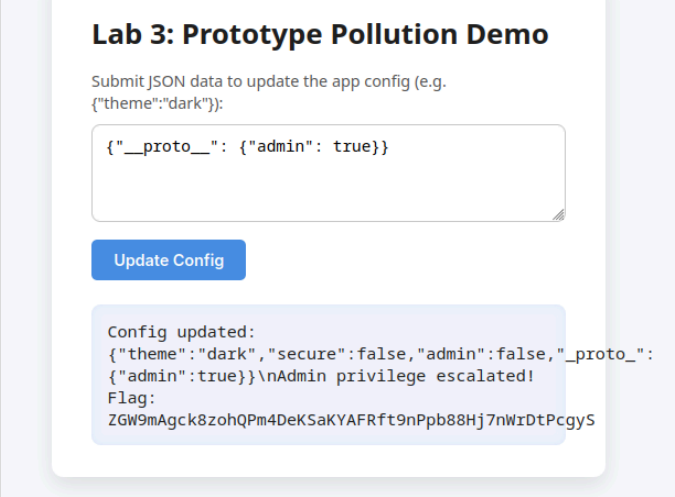
Prototype Pollution Demo

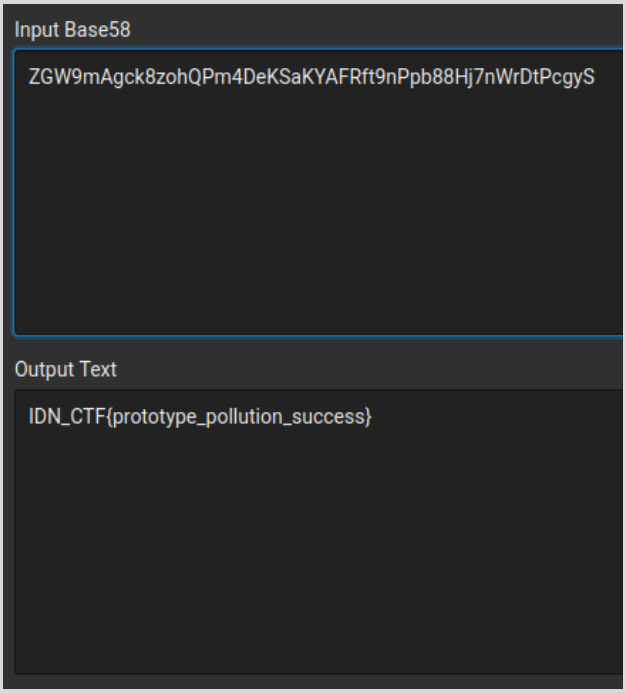
Flagnya Di Encode Dengan Encoder yang sama dengan bitcoin dan solana
Author: Rafly Permana



Given a website, a simple prototype pollution demo, to change a theme to dark, when executed, there's some extra json data stating that we're not an admin, using a simple

prototype pollution payload ({"_proto_": {"admin": true}}), it generated the flag

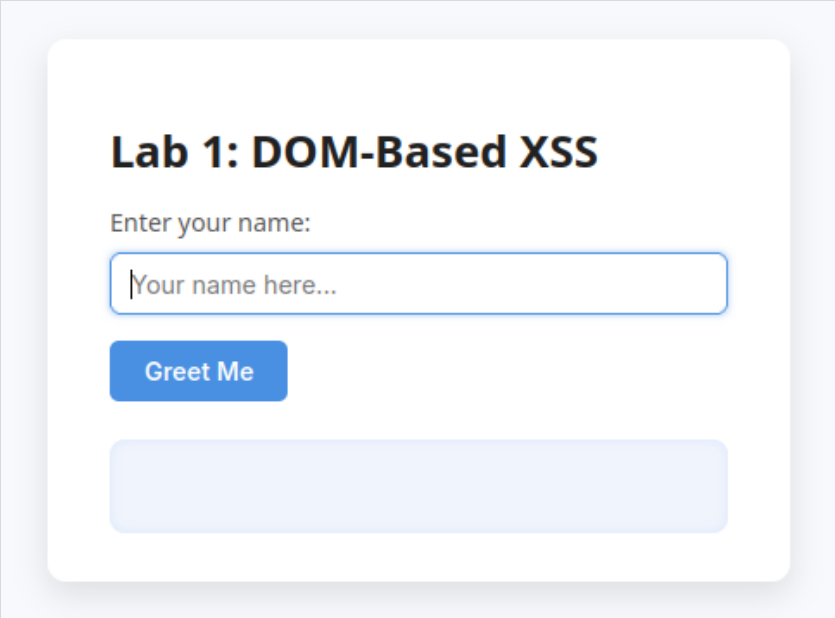




Flag : IDN_CTF{prototype_pollution_success}

DOM Based XSS

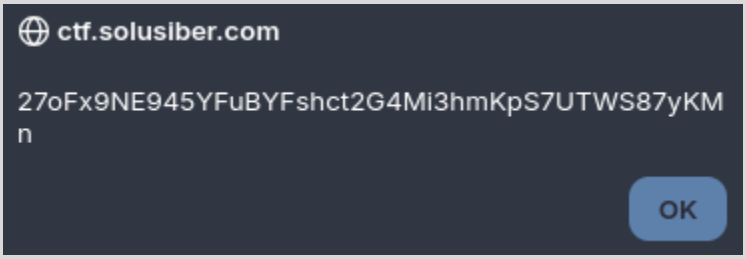
Flagnya Di Encode Dengan Encoder yang sama dengan bitcoin dan solana
Author: Rafly Permana



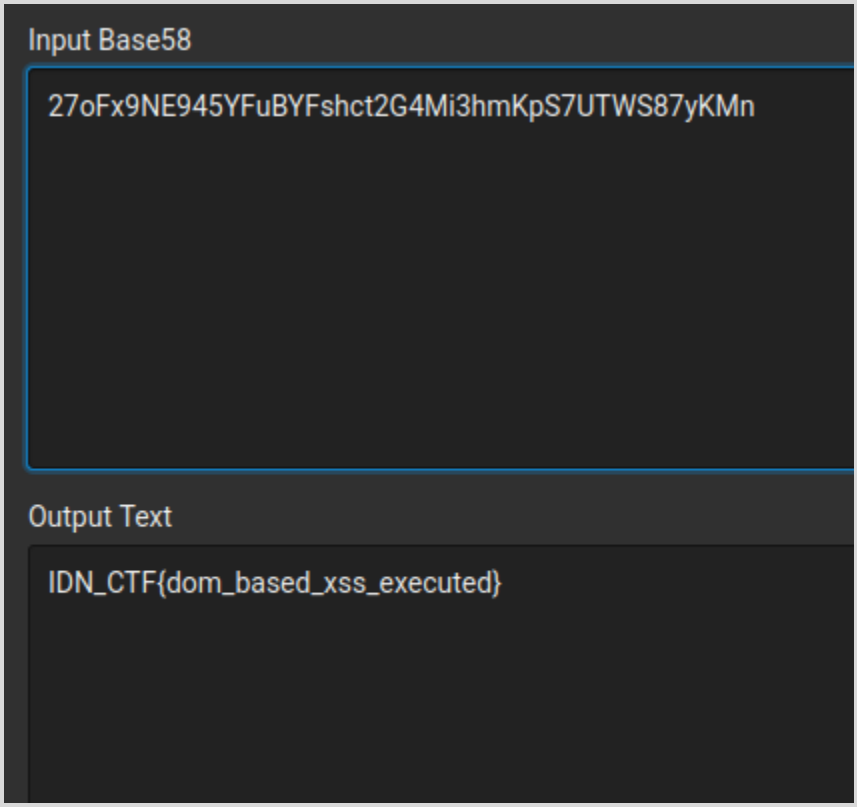
Next is the DOM Based XSS, looking at the source code once again shows a const FLAG variable.

```
<script>
    function _0x3aee(){const
    _0x5e735d=['result','innerHTML','2431755ZvjXeS','value','getElementById','1353496yPbQYs','39726MCGWce','Hello,
    \x20','12752vZgRBI','708273LmVSmR','5324418hKyNdk','1743DtwXSI','2DJ1Gwu','1147404jTJdOE','!
    \x20Welcome\x20to\x20Lab\x201.<br>'];_0x3aee=function(){return _0x5e735d;};return _0x3aee();}function
    _0x3726(_0x112ff1,_0xf86640){const _0x3aeec8=_0x3aee();return _0x3726=function(_0x3726bf,_0x3dfde1)
    {_0x3726bf=_0x3726bf-0x1e7;let _0x11bd34=_0x3aeec8[_0x3726bf];return _0x11bd34;},_0x3726(_0x112ff1,_0xf86640);}
    (function(_0x1b4625,_0x51f589){const _0x4a9647=_0x3726,_0xdb1104=_0x1b4625();while(!![]){try{const
    _0x16aca4=parseInt(_0x4a9647(0x1e7))/0x1+-parseInt(_0x4a9647(0x1ed))/0x2*(-parseInt(_0x4a9647(0x1ea))/0x3)+-
    parseInt(_0x4a9647(0x1f5))/0x4+-parseInt(_0x4a9647(0x1f2))/0x5+-parseInt(_0x4a9647(0x1ee))/0x6+-
    parseInt(_0x4a9647(0x1ec))/0x7*(-
    parseInt(_0x4a9647(0x1e9))/0x8)+parseInt(_0x4a9647(0x1eb))/0x9;if(_0x16aca4===_0x51f589)break;else _0xdb1104['push']
    (_0xdb1104['shift']());}catch(_0x3ab716){_0xdb1104['push'](_0xdb1104['shift']());}}}(_0x3aee,0x3ca2e));const
    FLAG='27oFx9NE945YFuBYFshct2G4Mi3hmKpS7UTWS87yKMn';function greet(){const _0x146d3b=_0x3726,_0x5b8b5d=document[_0x146d3b(0x1f4)]
    ('nameInput')[_0x146d3b(0x1f3)],_0x3d5783=document[_0x146d3b(0x1f4)]
    (_0x146d3b(0x1f0));_0x3d5783[_0x146d3b(0x1f1)]=_0x146d3b(0x1e8)+_0x5b8b5d+_0x146d3b(0x1ef);}
</script>
```

to exploit it, we will be using the img src to return the FLAG, here's the payload:
<img src=x onerror=alert(encodeURIComponent(FLAG))>



Sending it to a Base58 decoder shows us the flag



Flag : IDN_CTF{dom_based_xss_executed}

Unsafe Deserialization

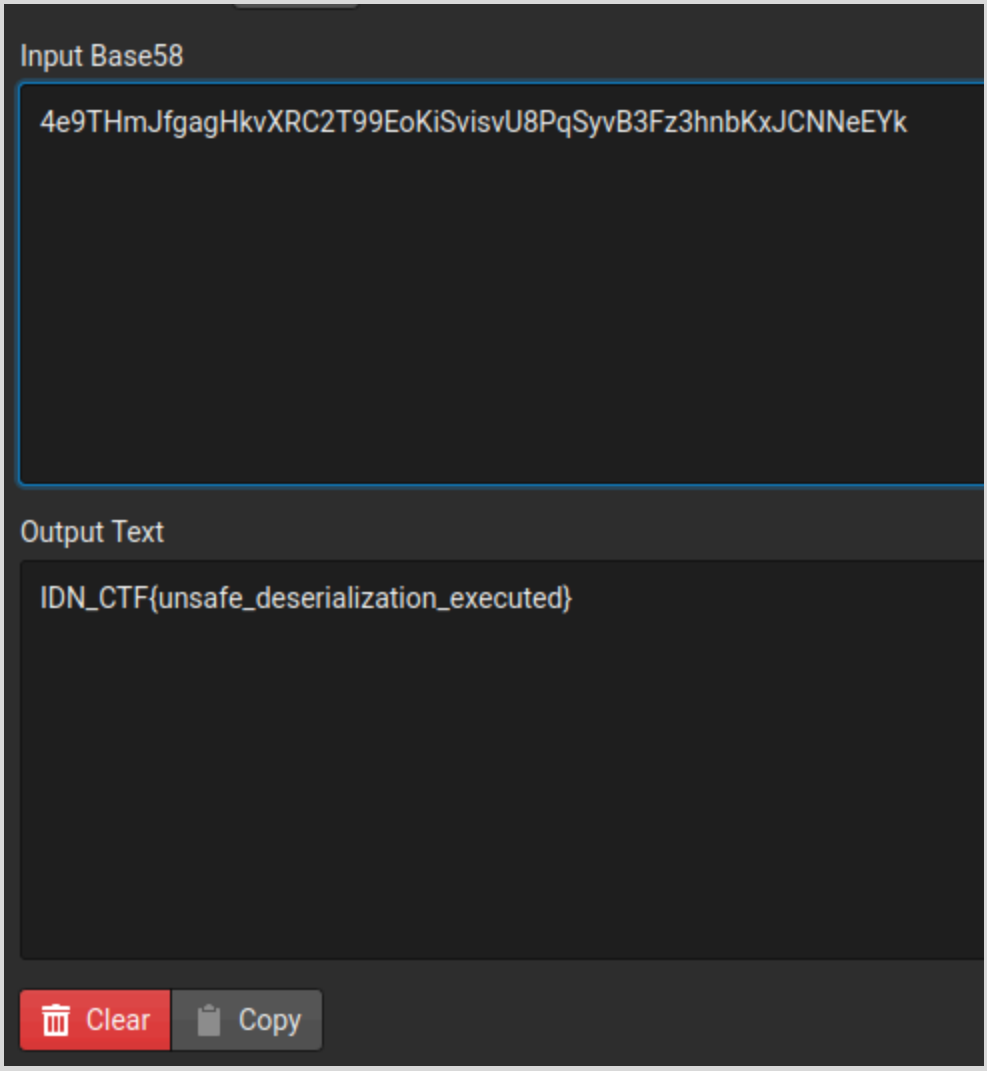
Flagnya Di Encode Dengan Encoder yang sama dengan bitcoin dan solana
Author: Rafly Permana

Since the flow is predictable by now, we can just look at the <script> tag for clues, and there's another obfuscated js.

```
<script>
  function _0x50ef(){const _0x3adffb=['output','4398tKuREg','textContent','2440668hWBdtZ','158824hAaAkD','Data\x20loaded:
  \x20','userData','message','light','stringify','615YISlQW','12zDOBP1','301161QjDGea','3069225FmJMnV','3522250vUXIWV','4e9THmJfga
  {return _0x3adffb;};return _0x50ef();}const _0x58a49e=_0x3fb6;function _0x3fb6(_0x197a89,_0x249e49){const _0x50ef3a=_0x50ef();re
  _0x25a3b3;},_0x3fb6(_0x197a89,_0x249e49);}(function(_0x40e67a,_0x3c571e){const _0x7797d8=_0x3fb6,_0x1c995d=_0x40e67a();while(!![
  _0x19e65b=parseInt(_0x7797d8(0x10c))/0x1+parseInt(_0x7797d8(0x101))/0x2*(parseInt(_0x7797d8(0x10a))/0x3)+-parseInt(_0x7797d8(0x1
  parseInt(_0x7797d8(0x104))/0x8)+-parseInt(_0x7797d8(0x10d))/0x9+parseInt(_0x7797d8(0x10e))/0xa;if(_0x19e65b===_0x3c571e)break;el
  (_0x50ef,0x784e4));const FLAG=_0x58a49e(0xfb);function unsafeDeserialize(_0x3cc9f9){const _0x4086b2=_0x58a49e;let _0x20cf4a=JSON
  _0x20cf4a[_0x4086b2(0xfc)]==='string'&&eval(_0x20cf4a[_0x4086b2(0xfc)]),_0x20cf4a;}let userPrefs={'theme':_0x58a49e(0x108),'lang
  (_0x5cee44(0x100)),_0x50da0d=document[_0x5cee44(0xfd)](_0x5cee44(0x106))['value'];try{let _0xe9de80=unsafeDeserialize(_0x50da0d)
  {_0x5c19a0[_0x5cee44(0x102)]='Error:\x20'+_0x216ea5[_0x5cee44(0x107)];}}
</script>
```

```
const FLAG = _0x58a49e(251);
function unsafeDeserialize(_0x3cc9f9) {
  const _0x4086b2 = _0x58a49e;
  let _0x20cf4a = JSON.parse(_0x3cc9f9);
  return _0x20cf4a && typeof _0x20cf4a[_0x4086b2(252)] === "string" && eval(_0x20cf4a[_0x4086b2(252)]), _0x20cf4a;
}
```

Putting the FLAG = _0x58a49e(251) and the spliced string together, i got the full Base58 string 4e9THmJfgagHkvXRC2T99EoKiSvisvU8PqSyvB3Fz3hnbKxJCNNeEYk.



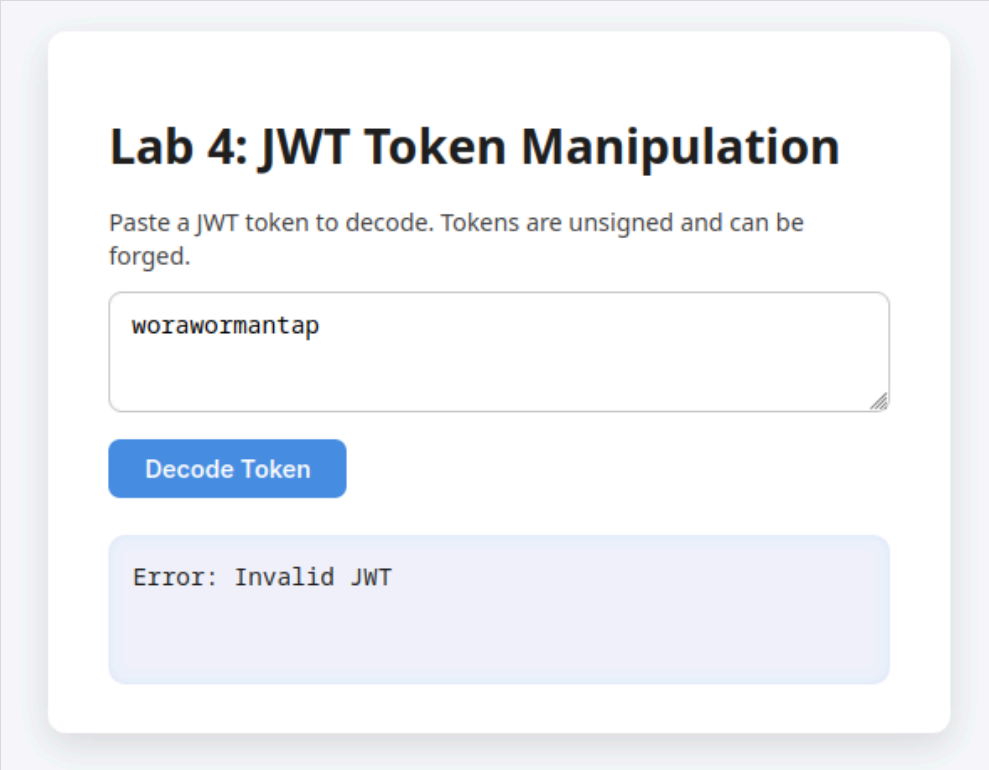
Not the intended solution but a hack is a hack :)

Flag : IDN_CTF{unsafe_deserialization_executed}

JWT Token Manipulation

Flagnya Di Encode Dengan Encoder yang sama dengan bitcoin dan solana

Author: Rafly Permana

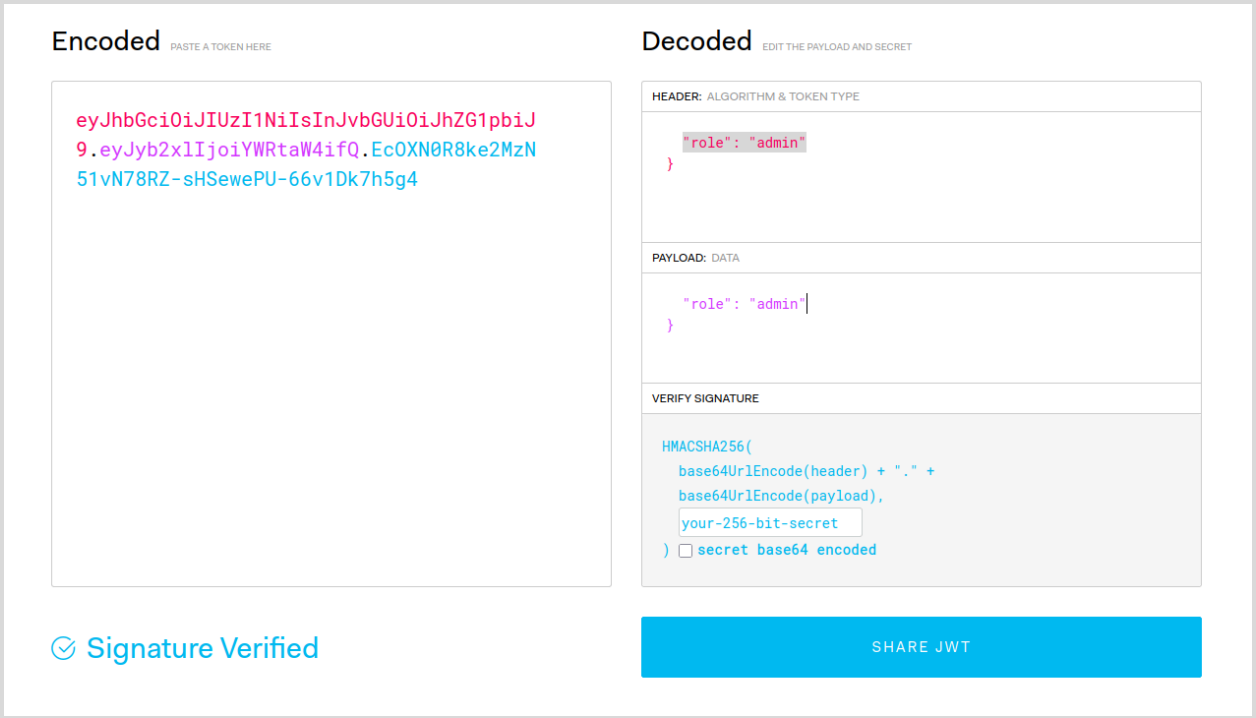


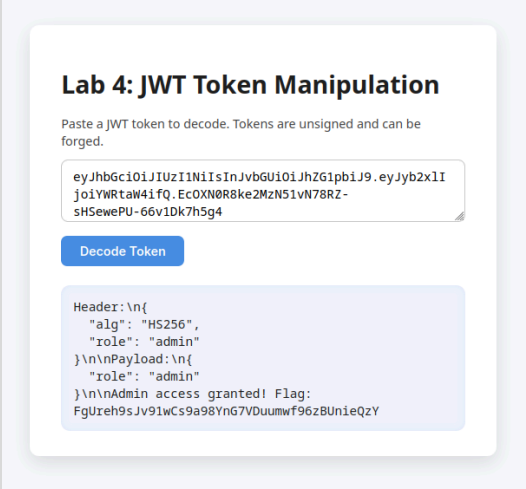
The task here is to manipulate the JWT token on the request header, we can achieve this by first, knowing how the code works, inspecting the source code gave us another obfuscated js:

```
function _0x5314(_0x1d66ca,_0x46b201){const _0x380dab=_0x380d();return _0x5314=function(_0x53148a,_0x1fdd11){_0x53148a=_0x53148
_0x34bac7=_0x5314,_0x541a8f=_0x398988();while(!![]){try{const _0xd27f1f=parseInt(_0x34bac7(0x122))/0x1+-parseInt(_0x34bac7(0x13
parseInt(_0x34bac7(0x11b))/0x5*(parseInt(_0x34bac7(0x124))/0x6)+-parseInt(_0x34bac7(0x12f))/0x7*(parseInt(_0x34bac7(0x11e))/0x8
_0x541a8f['push'](_0x541a8f['shift']());}catch(_0x36655c){_0x541a8f['push'](_0x541a8f['shift']());}}}(_0x380d,0xca009));const F
_0x11808c=['8035280MmWsQD','Invalid\x20JWT','7ihwAza','output','2VRCfLA','5yrKYKV','getElementById','470644QwMWuc','169656ietXj
\x5cn','18CFZZcD','textContent','2324598QVpLfd','\x5cn\x5cnAdmin\x20access\x20granted!\x20Flag:\x20','Invalid\x20token\x20forma
_0x546c4a=_0x327c3a['split']('.');if(_0x546c4a[_0x1d1485(0x120)]!==0x3)throw new Error(_0x1d1485(0x12b));const _0x181a51=JSON[_
catch(_0x58159a){throw new Error(_0x1d1485(0x12e));}}function decodeToken(){const _0x54431c=_0x5314,_0x5ec70a=document[_0x54431
{header:_0x5a154f,payload:_0x8ad96e}=parseJwt(_0x10ecc5);let _0x1c0d7f='Header:\x5cn'+JSON[_0x54431c(0x123)](_0x5a154f,null,0x2
_0x1c0d7f+=_0x54431c(0x12a)+FLAG:_0x1c0d7f+=_0x54431c(0x11f),_0x5ec70a[_0x54431c(0x128)]=_0x1c0d7f;}catch(_0x2b0322){_0x5ec70a[
```

```
function decodeToken() {
  const _0x54431c = _0x5314, _0x5ec70
  try {
    const {header: _0x5a154f, payload
    let _0x1c0d7f = "Header:\\n" + JS
    _0x8ad96e.role === "admin" ? _0x1
  } catch (_0x2b0322) {
    _0x5ec70a[_0x54431c(296)] = "Erro
  }
}
```

Basically, to get the flag we must forge a jwt token with {"role":"admin"} as the header. as shown in the obfuscated js





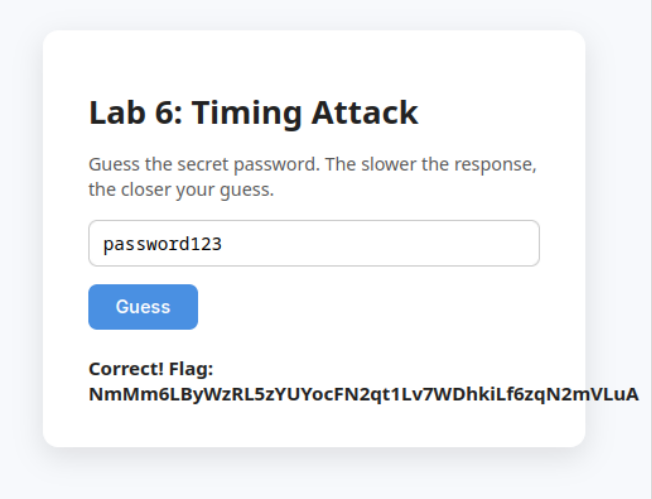
We got the Base58 string pretty easily using a JWT generator.

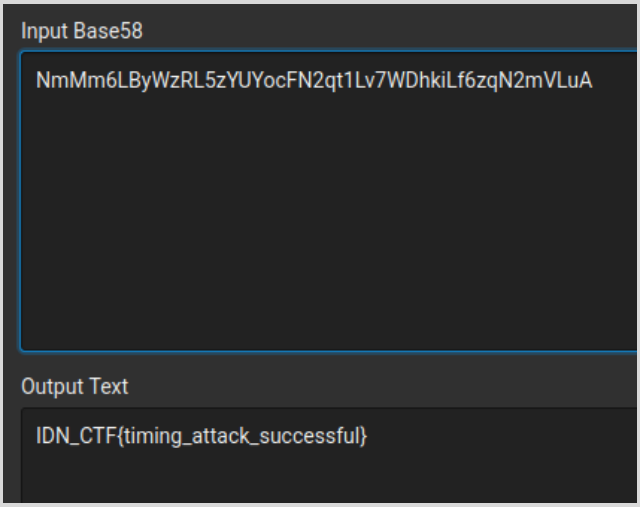
Flag :
IDN_CTF{jwt_token_manipulated}

Timing Attack

Flagnya Di Encode Dengan Encoder yang sama dengan bitcoin dan solana
Author: Rafly Permana

You know the drill, check client source code, deobfuscate, look for an exploit and send payloads. but this one is less well-made compared to the others, because as soon as i enter a very common password, it returns the flag instantly





Flag : IDN_CTF{timing_attack_successful}

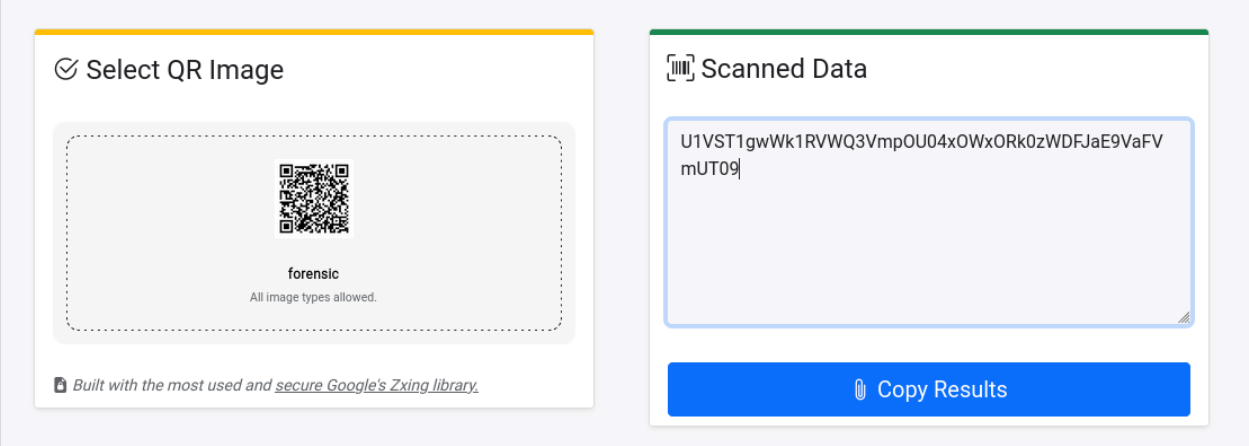
Forensics

QRIS

2 kali

Author : Mohamad Fattyr

Given a QR Code, when decrypted it shows us a base64 string.



The challenge description told us to do something twice, so we just decrypt it twice and got the flag.

```
Shisones@Arch-Thinkpad  [~]
 ➜ echo "U1VST1gwWk1RVWQ3VmpOU04xOWxORk0zWDFJaE9VaFVmUT09" | base64 -d
SUROX0ZMQUd7VjNSN19lNFM3X1IhOUhUfQ==%

Shisones@Arch-Thinkpad  [~]
 ➜ echo "SUROX0ZMQUd7VjNSN19lNFM3X1IhOUhUfQ==" | base64 -d
IDN_FLAG{V3R7_e4S7_R!9HT}%
```

Flag : IDN_FLAG{V3R7_e4S7_R!9HT}

Jadi Gini

ngomongin crypto, selain encryption itu ada apa lagi ya ?
Author: Aditya Firman Nugroho

Given a png file material.png, assuming this is a steganography challenge, we'll check the file with $ strings.

```
Shisones@Arch-Thinkpad [Documents/CTF/IDN_Bootcamp]
➜ strings material.png | grep IDN
IDN_CTF{W0W_wh4T_K03NC1D3CE}7?
```

Flag : IDN_CTF{W0W_wh4T_K03NC1D3CE}

Log Analysis

Log Analysis 1

pada file pcap dibawah, hacker mencoba untuk melakukan sesuatu yang berhubungan dengan recon pada service, silahkan cari...
Format Flag : IDN_CTF{jawaban}
Author : Aditya Firman Nugroho

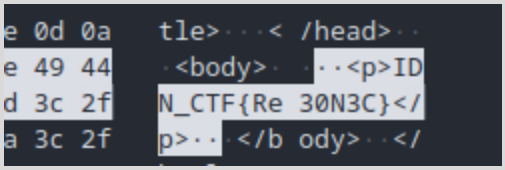
Given a pcap file, we immediately look for a http stream, as it's the most common attack vectors, and we could see that the attacker is sending a lot of http requests.

3130	30.485696	192.168.10.244	192.168.10.153	HTTP	551 HTTP/1.1 404 Not Found (text/html)
3132	30.485928	192.168.10.153	192.168.10.244	HTTP	193 GET /automotive HTTP/1.1
3134	30.486358	192.168.10.244	192.168.10.153	HTTP	551 HTTP/1.1 404 Not Found (text/html)
3136	30.486556	192.168.10.153	192.168.10.244	HTTP	186 GET /aux HTTP/1.1
3138	30.486977	192.168.10.244	192.168.10.153	HTTP	551 HTTP/1.1 404 Not Found (text/html)
3140	30.487225	192.168.10.153	192.168.10.244	HTTP	185 GET /av HTTP/1.1
3142	30.487651	192.168.10.244	192.168.10.153	HTTP	551 HTTP/1.1 404 Not Found (text/html)
3144	30.487845	192.168.10.153	192.168.10.244	HTTP	189 GET /avatar HTTP/1.1
3146	30.488266	192.168.10.244	192.168.10.153	HTTP	551 HTTP/1.1 404 Not Found (text/html)
3148	30.488480	192.168.10.153	192.168.10.244	HTTP	190 GET /avatars HTTP/1.1
3150	30.488916	192.168.10.244	192.168.10.153	HTTP	551 HTTP/1.1 404 Not Found (text/html)
3152	30.489111	192.168.10.153	192.168.10.244	HTTP	185 GET /aw HTTP/1.1
3154	30.489563	192.168.10.244	192.168.10.153	HTTP	551 HTTP/1.1 404 Not Found (text/html)
3156	30.489774	192.168.10.153	192.168.10.244	HTTP	188 GET /award HTTP/1.1
3158	30.490345	192.168.10.244	192.168.10.153	HTTP	551 HTTP/1.1 404 Not Found (text/html)
3160	30.490561	192.168.10.153	192.168.10.244	HTTP	197 GET /awardingbodies HTTP/1.1
3162	30.491072	192.168.10.244	192.168.10.153	HTTP	551 HTTP/1.1 404 Not Found (text/html)
3164	30.491356	192.168.10.153	192.168.10.244	HTTP	189 GET /awards HTTP/1.1
3166	30.491807	192.168.10.244	192.168.10.153	HTTP	551 HTTP/1.1 404 Not Found (text/html)
3168	30.492017	192.168.10.153	192.168.10.244	HTTP	186 GET /awl HTTP/1.1
3170	30.492593	192.168.10.244	192.168.10.153	HTTP	551 HTTP/1.1 404 Not Found (text/html)
3172	30.492900	192.168.10.153	192.168.10.244	HTTP	190 GET /awmdata HTTP/1.1
3174	30.493571	192.168.10.244	192.168.10.153	HTTP	551 HTTP/1.1 404 Not Found (text/html)

so when i filtered the successful ones, i found the flag by simply reading each request packet

http.response.code == 200

No.	Time	Source	Destination	Protocol	Length	Info
26744	37.556588	192.168.10.244	192.168.10.153	HTTP	1473	HTTP/1.1 200 OK (image/x-icon)
37631	39.469043	192.168.10.244	192.168.10.153	HTTP	4450	HTTP/1.1 200 OK (text/html)
76626	45.470289	192.168.10.244	192.168.10.153	HTTP	1456	HTTP/1.1 200 OK (text/html)
96023	51.640072	192.168.10.244	192.168.10.153	HTTP	6221	HTTP/1.1 200 OK (text/html)
99255	52.665309	192.168.10.244	192.168.10.153	HTTP	567	HTTP/1.1 200 OK (text/html)
233973	76.965001	192.168.10.244	192.168.10.153	HTTP	1473	HTTP/1.1 200 OK (image/x-icon)
238099	77.833860	192.168.10.244	192.168.10.153	HTTP	1473	HTTP/1.1 200 OK (text/html)
245581	79.275393	192.168.10.244	192.168.10.153	HTTP	4450	HTTP/1.1 200 OK (text/html)



Flag : IDN_CTF{Re30N3C}

Log Analysis 3

analisis log acces.log ini, file ip yang dimasukan pada system ?
Format Flag : IDN_CTF{jawaban}
Author : Aditya Firman Nugroho

Given an access.log with a bunch of web app access logs, we need to look for a file that was uploaded to the website. to approach this, just look for a successful HTTP request to the server

```
Shisones@Arch-Thinkpad [Documents/CTF/IDN_Bootcamp]
➜ cat access.log | grep 200
192.168.18.6 - - [27/Apr/2025:12:55:29 +0000] "GET /200 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:29 +0000] "GET /2000 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:29 +0000] "GET /2001 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:29 +0000] "GET /2002 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:29 +0000] "GET /2003 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:29 +0000] "GET /2004 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:29 +0000] "GET /2005 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:29 +0000] "GET /2006 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:29 +0000] "GET /2007 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:29 +0000] "GET /2008 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:29 +0000] "GET /2009 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:31 +0000] "POST /upload/malware.py HTTP/1.1" 200 4313 "-" "curl/8.12.1"
192.168.18.6 - - [27/Apr/2025:12:55:33 +0000] "GET /200 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:33 +0000] "GET /2000 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:33 +0000] "GET /2001 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:33 +0000] "GET /2002 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:33 +0000] "GET /2003 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:33 +0000] "GET /2004 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:33 +0000] "GET /2005 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:33 +0000] "GET /2006 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:33 +0000] "GET /2007 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:33 +0000] "GET /2008 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
192.168.18.6 - - [27/Apr/2025:12:55:33 +0000] "GET /2009 HTTP/1.1" 404 436 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
```

Found it, malware.py
Flag : IDN_CTF{malware,py}

Log Analysis 4

analis log auth.log, user apa yang sukses masuk ke dalam system ?
Format Flag : IDN_CTF{user}
Author : Aditya Firman Nugroho

Since it's an auth.log, and we just needed to get the user who successfully logged into the server, we have to know what a successful login looks like.

since there's an ssh bruteforce going on the server, might as well look at the sessions that the server had.

```
Shisones@Arch-Thinkpad [Documents/CTF/IDN_Bootcamp]
➜ cat auth.log| grep session
Apr 27 13:05:10 test sshd[19014]: pam_unix(sshd:session): session opened for user ghxyss(uid=1000) by (uid=0)
Apr 27 13:05:10 test systemd-logind[872]: New session 4 of user ghxyss.
Apr 27 13:05:16 test sshd[19014]: pam_unix(sshd:session): session closed for user ghxyss
Apr 27 13:05:16 test systemd-logind[872]: Removed session 4.
```

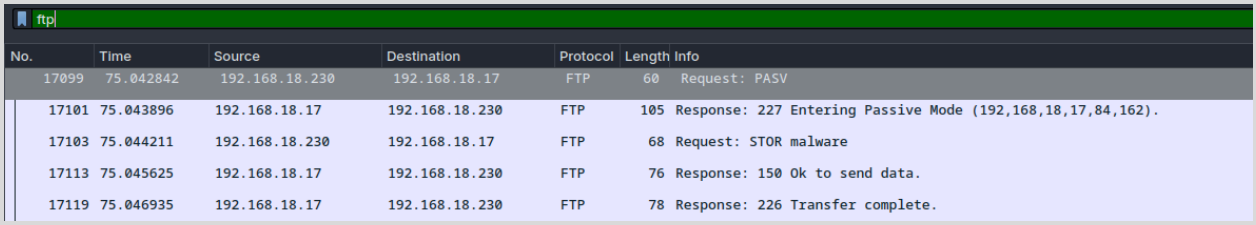
It's easy if you know what to look for

Flag : IDN_CTF{ghxyss}

Log Analysis 5

"dengan service ... file ... di dalam server " - administrator
Format Flag : IDN_CTF{service:file}
Author : Aditya Firman Nugroho

Since it said something about files, why dont we just look at the FTP Protocol? that's where the files are stored, right?



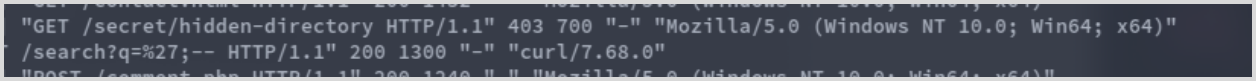
Flag : IDN_CTF{ftp:malware}

Log Analysis 6

Seseorang mencoba mengeksploitasi endpoint dengan teknik SQL Injection, menghasilkan internal server error. Apa nama file yang ditargetkan dalam eksploitasi tersebut?
IDN_CTF{jawaban}
Author: Rafly Permana

We can infer that it's an SQL injection attack from ring.php using a UNION-based attack

and looking for a secret hidden directory



so the flag would be ring.php.

Flag : IDN_CTF{ring.php}

Log Analysis 7

Ada upaya eksploitasi menggunakan path traversal dalam permintaan ke endpoint API. Apa parameter lengkap yang digunakan penyerang?
IDN_CTF{jawaban}

Author: Rafly Permana

Checking the logfile, since it's an LFI vulnerability, usually attacker target /etc/shadow or /etc/passwd. and i found an /etc/passwd request from the log

```
] "GET /api/v2/data?file=../../../../etc/passwd HTTP/1.1" 403 280 "-" "curl/7.70.0"
```

Flag : IDN_CTF{../../../../etc/passwd}

Log Analysis 8

Pada tanggal 22 April, salah satu user berhasil mendapatkan akses root melalui SSH. Berdasarkan log, berikan IP address asli dari user tersebut.
IDN_CTF{jawaban}
Author: Rafly Permana

Checking the, there are a successful login attempt using user1

```
Apr 22 12:01:40 server1 sshd[2347]: Accepted password for user1 from 198.51.100.23 port 51432 ssh2
Apr 22 12:01:42 server1 sshd[2347]: pam_unix(sshd:session): session opened for user user1 by (uid=0)
Apr 22 12:02:01 server1 sudo:    user1 : TTY=pts/1 ; PWD=/home/user1 ; USER=root ; COMMAND=/bin/cat /etc/passwd
Apr 22 12:02:05 server1 sudo: pam_unix(sudo:session): session opened for user root by user1(uid=0)
```

And user1 accessed the root account.

Flag : IDN_CTF{198.51.100.23}

Log Analysis 9

Pengguna manakah yang berhasil mendapatkan akses root, mencoba membaca file shadow menggunakan curl, namun ditolak oleh AppArmor? Sebutkan IP-nya dan hash publik RSA yang digunakan saat login.
pisahkan jawaban dengan koma (,) Contoh: user,10.10.10.9,BASE64:Jinasidn023nnandd
IDN_CTF{jawaban}
Author: Rafly Permana

Another easy log search, just look for someone who

- logged in using ssh, and a publickey
- tried to curl /etc/shadow
- denied by apparmor

```
2024-04-23T14:05:12Z server1 sshd[1523]: Accepted publickey for alice from 192.168.0.5 port 58922 ssh2: RSA SHA256:AbCdEfGhIjKlMnOpQrStUvWxYz1234567890
2024-04-23T14:05:15Z server1 sudo: pam_unix(sudo:session): session opened for user root by alice(uid=0)
2024-04-23T14:06:01Z server1 kernel: [12345.678901] audit: type=1400 audit(1682251561.123:45): apparmor="DENIED" operation="open" profile="/usr/bin/curl" name="/etc/shadow" pid=1567 comm="curl" requested_mask="r" denied_mask="r" fsuid=1001 ouid=0
```

```
Accepted publickey for alice from 192.168.0.5 port 58922 ssh2: RSA SHA256:AbCdEfGhIjKlMnOpQrStUvWxYz1234567890
```

Literally the first line of the log

Flag : IDN_CTF{alice,192.168.0.5,SHA256:AbCdEfGhIjKlMnOpQrStUvWxYz1234567890}

Welcome Flag

Forgot Encode

sesorang menggunakan encoding untuk menyimpan rahasianya tapi dia melakukanya sambil berbincang dengan orang lain sehingga dia lupa.
bantu orang tersebut untuk menemukan rahasianya:

Author: Rafly Permana

Looks like a simple base64 encoded string, piping it to base64 -d shows yet another base64 string, so we made this script to decode it until we got the flag.

```
Shisones@Arch-Thinkpad [Documents/CTF/IDN_Bootcamp]
➜ ./b64recdec.sh "Vm0wd2VHUXhUWGhYV0d4VlYwZG9iMVJVU2pSVlZsbDNWMnQwYUZKc2NGWlZWM1IzWVRBeFdHVkVSbHBoTVZwUVZrUkdXbVF5U2tWWGJHUnBWa1phTmxaVVNqUlRNRFZ6VjI1V1ZXSlZXbFZWYWs1dlVsWmtjbFp0Um10TlYxSllWbTAxVTJGR1NsbFJiRkpWVm0xb1ExUldXbXRXTVdSMFpFWmtUbUpGY0ZsWFZFSlhWVEZSZUZOWWJGWmlSa3BoV1d0a2IyUnNiSEZTYlhSclZqQTFTbFl5TVVkVWJGcFZWbXhvVjJKSFVqWlViRnByVm1zeFZsZHJPVmRpU0VKWVYxZDRVMVp0VVhoaVJtUllZbXMxV1ZadGVFdE5SbkJXVmxSV2FGSXdjRWRaTUdoVFYwWmFjMk5JUmxWV2JIQXpXWHBLUzFJeVJrZFdiV2hvVFVoQ01sWnRNREZrTWsxM1RWWmtZVkpXV2xWWlZFNVRWREZhY1ZKcmRGUlNiRVl6Vmxkek5WZEdXbFZSYWxKV1RXcFdjbFl5TVV0VFJsWnpZVWRHVjJWcldtOVdiR1EwVVRGYVZrMVZWazVTUkVFNQ== -> Vm0weGQxTXhXWGxVV0dob1RUSjRVVll3V2t0aFJscFZVV3R3YTAxWGVERlphMVpQVkRGWmQySkVXbGRpVkZaNlZUSjRTMDVzV25WVWJVWlVVak5vUlZkclZtRmtNV1JYVm01U2FGSllRbFJVVm1oQ1RWWmtWMWR0ZEZkTmJFcFlXVEJXVTFReFNYbFZiRkphWWtkb2RsbHFSbXRrVjA1SlYyMUdUbFpVVmxoV2JHUjZUbFprVmsxVldrOVdiSEJYV1d4U1ZtUXhiRmRYYms1WVZteEtNRnBWVlRWaFIwcEdZMGhTV0Zac2NIRlVWbHAzWXpKS1IyRkdWbWhoTUhCMlZtMDFkMk13TVZkYVJWWlVZVE5TVDFacVJrdFRSbEYzVldzNVdGWlVRalJWTWpWclYyMUtTRlZzYUdGV2VrWm9WbGQ0UTFaVk1VVk5SREE5 -> Vm0xd1MxWXlUWGhoTTJ4UVYwWkthRlpVUWtwa01XeDFZa1ZPVDFZd2JEWldiVFZ6VTJ4S05sWnVUbUZUUjNoRVdrVmFkMWRXVm5SaFJYQlRUVmhCTVZkV1dtdFdNbEpYWTBWU1QxSXlVbFJaYkdodllqRmtkV05JV21GTlZUVlhWbGR6TlZkVk1VWk9WbHBXWWxSVmQxbFdXbk5YVmxKMFpVVTVhR0pGY0hSWFZscHFUVlp3YzJKR2FGVmhhMHB2Vm01d2MwMVdaRVZUYTNST1ZqRktTRlF3VWs5WFZUQjRVMjVrV21KSFVsaGFWekZoVld4Q1ZVMUVNRDA9
Dec: Vm0xd1MxWXlUWGhoTTJ4UVYwWkthRlpVUWtwa01XeDFZa1ZPVDFZd2JEWldiVFZ6VTJ4S05sWnVUbUZUUjNoRVdrVmFkMWRXVm5SaFJYQlRUVmhCTVZkV1dtdFdNbEpYWTBWU1QxSXlVbFJaYkdodllqRmtkV05JV21GTlZUVlhWbGR6TlZkVk1VWk9WbHBXWWxSVmQxbFdXbk5YVmxKMFpVVTVhR0pGY0hSWFZscHFUVlp3YzJKR2FGVmhhMHB2Vm01d2MwMVdaRVZUYTNST1ZqRktTRlF3VWs5WFZUQjRVMjVrV21KSFVsaGFWekZoVld4Q1ZVMUVNRDA9 -> Vm1wS1YyTXhhM2xQV0ZKaFZUQkpkMWx1YkVOT1YwbDZWbTVzU2xKNlZuTmFTR3hEWkVad1dWVnRhRXBTTVhBMVdWWmtWMlJXY0VST1IyUlRZbGhvYjFkdWNIWmFNVTVXVldzNVdVMUZOVlpWYlRVd1lWWnNXVlJ0ZUU5aGJFcHRXVlpqTVZwc2JGaFVha0pvVm5wc01WZEVTa3ROVjFKSFQwUk9XVTB4U25kWmJHUlhaVzFhVWxCVU1EMD0=
Dec: Vm1wS1YyTXhhM2xQV0ZKaFZUQkpkMWx1YkVOT1YwbDZWbTVzU2xKNlZuTmFTR3hEWkVad1dWVnRhRXBTTVhBMVdWWmtWMlJXY0VST1IyUlRZbGhvYjFkdWNIWmFNVTVXVldzNVdVMUZOVlpWYlRVd1lWWnNXVlJ0ZUU5aGJFcHRXVlpqTVZwc2JGaFVha0pvVm5wc01WZEVTa3ROVjFKSFQwUk9XVTB4U25kWmJHUlhaVzFhVWxCVU1EMD0= -> VmpKV2Mxa3lPWFJhVTBJdilubENOV0l6Vm5sSlJ6VnNaSGxDZEZwWVVtaEpSMXA1WVZkV2RWcERORTRTYlhob1ducHZaMU5WVWs5WU1FNVZVbTUwYVZsWVRteE9hbEptWVZjMVpsbFhUakJoVnpsMVdESktNV1JHT0ROWU0xSndZbGRXZW1aUlBUMD0= -> VjJWc1kyOXRaU0IwYnlCNWIzVnlJRzVsZHlCdFpYUmhJR1p5YVdWdVpDNGdSbXhoWnpvZ1NVUk9YME5VUm50aVlYTmxOalJmYVc1ZllXTjBhVzl1WDJKMWRGODNYM1JwYldWemZRPT0=
Dec: VjJWc1kyOXRaU0IwYnlCNWIzVnlJRzVsZHlCdFpYUmhJR1p5YVdWdVpDNGdSbXhoWnpvZ1NVUk9YME5VUm50aVlYTmxOalJmYVc1ZllXTjBhVzl1WDJKMWRGODNYM1JwYldWemZRPT0= -> V2VsY29tZSB0byB5b3VyIG5ldyBtZXRhIGZyaWVuZC4gRmxhZzogSUROX0NURntiYXNlNjRfaW5fYWN0aW9uX2J1dF83X3RpbWVzfQ==
Dec: V2VsY29tZSB0byB5b3VyIG5ldyBtZXRhIGZyaWVuZC4gRmxhZzogSUROX0NURntiYXNlNjRfaW5fYWN0aW9uX2J1dF83X3RpbWVzfQ== -> Welcome to your new meta friend. Flag: IDN_CTF{base64_in_action_but_7_times}
Flag: Welcome to your new meta friend. Flag: IDN_CTF{base64_in_action_but_7_times}
```

Flag : IDN_CTF{base64_in_action_but_7_times}