

# PCAP1

HOLOGY7{4n\_E4sy\_1CmP\_Ch4ll3ngE}

```
import pyshark

# Define the pcap file and the IP address to filter by
pcap_file = 'chall/chall.pcapng'
source_ip = '192.168.216.129'

cap = pyshark.FileCapture(pcap_file, display_filter='icmp')

# Loop through each packet and filter by source IP and ICMP type
(echo request)

for packet in cap:
    try:
        if packet.ip.src == source_ip and packet.icmp.type == '8':
            data_hex = packet.icmp.data.replace(':', ' ')[:32]
            print(data_hex) ## decode hex 2x
    except AttributeError:
        # Skip packets that do not have the expected fields
        continue

# Close the capture
cap.close()
```

Recipe

From Hex

Delimiter  
Auto

From Hex

Delimiter  
Auto

STEP

BAKE!

Auto Bake

Input

3663666373236353230363132303737  
3666373236636343230373736383635  
373236353230363437320a3631363736  
66366537323230373236663631373236  
35363432303734363837323666373536  
37363832303633373237393733373436  
313663323037330a3662363936353733  
32303631366536343230366436313637  
3639363332303632366366636663664  
3635363432303633639366236353230  
373736390a366363643636366366637  
37363537323733326530610a00000000  
000000454e444f646556494c50494e47  
|

71181 2158

Raw Bytes LF (detected)

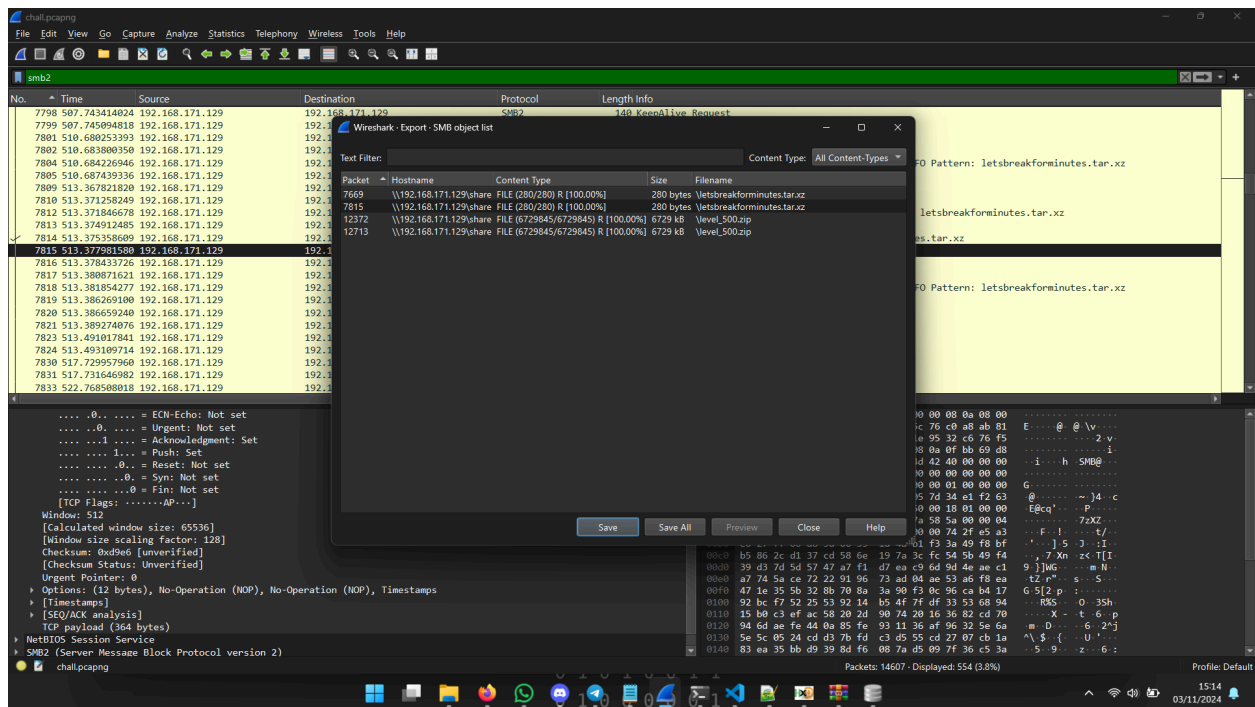
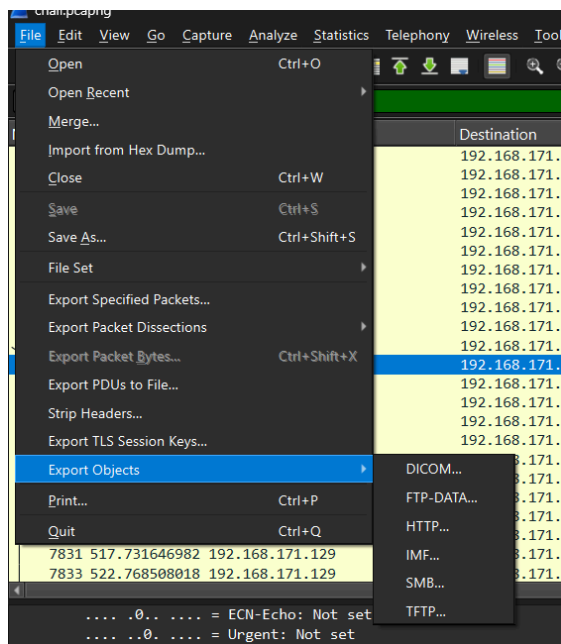
Output

chest. â••I wonâ••t give up,â•• he said, louder this time. â••Every puzzle has a heart, and Iâ••ll find this oneâ••s secret.â•• The forest watched, waiting to see if the boy who had stumbled into the realm of Astranor could turn his first challenge into the beginning of a true legend.  
â••Finn took a steady breath, trying to calm the racing thoughts in his mind. He considered the inscription again: HOLOGY7{4n\_E4sy\_1CmP\_Ch4ll3ngE}. The code looked like a jumble of letters, numbers, and symbols, but he knew there had to be a hidden meaning behind it. He remembered something his grandmother had told him when he was younger, back when they used to do puzzles together: â••Every challenge has a pattern, and every puzzle has a heart.â•• Finn wondered if this puzzle, too, had a heart that he could find and unravel.  
â••Once upon a time, in the bustling city of Astranor, there was a peculiar little bookstore tucked away in a narrow alley. It was called The Whispering Pages, and its door was marked with a brass handle shaped like a dragonâ••s tail. The store was run by an old woman named Madame Lyria, who had hair the color of freshly fallen snow and eyes that glimmered like moonlit rivers. She was rumored to be a collector of secrets, and her bookstore held more than just dusty tomes and ancient scrolls.

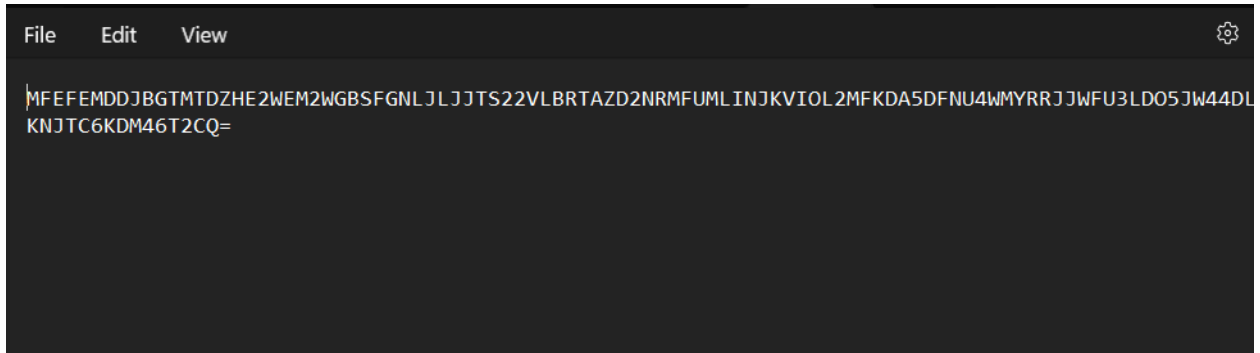
16296 67

Raw Bytes LF (detected)

## PCAP 2



Use Extract File -> SMB to dump



(rickroll)

Extract level\_500.zip sampai level\_1.zip ditemukan folder berisi profile mozilla firefox

```
import os
import zipfile
import shutil

def extract_until_target(zip_file_path, target_file, output_folder):
    # Create the output folder if it doesn't exist
    os.makedirs(output_folder, exist_ok=True)

    # Initialize the list of files to extract
    files_to_extract = [zip_file_path]

    while files_to_extract:
        current_zip = files_to_extract.pop()

        # Extract the current ZIP file
        with zipfile.ZipFile(current_zip, 'r') as zip_ref:
            zip_ref.extractall(output_folder)
            extracted_files = zip_ref.namelist()

        # Check if the target file is extracted
        if target_file in extracted_files:
            print(f"Found {target_file} in {current_zip}.")
            # Copy the target file to the output folder
            shutil.copy(os.path.join(output_folder, target_file),
output_folder)
            print(f"Copied {target_file} to {output_folder}.")
            break

        for file in extracted_files:
            if file.endswith('.zip'):
```

```

files_to_extract.append(os.path.join(output_folder, file))

initial_zip_file = 'level_500.zip' # Path to your initial zip file
target_zip_file = 'level_1.zip'    # The target zip file you want to
find

reachable_folder = 'aaa' # The folder to copy the target zip file into
extract_until_target(initial_zip_file, target_zip_file, reachable_folder)

```

Name	Date modified	Type	Size
Crash Reports	22/10/2024 17:18	File folder	
firefox-mpris	24/10/2024 10:45	File folder	
ju1ay0ip.default	22/10/2024 17:18	File folder	
Pending Pings	22/10/2024 17:18	File folder	
zuh147e6.default-release	03/11/2024 15:12	File folder	
installs.ini	22/10/2024 17:18	Configuration setti...	1 KB
profiles.ini	22/10/2024 17:18	Configuration setti...	1 KB

Terdapat file menarik yaitu logins.json dengan data login instagram terenkripsi

```

S.dat  tools.dat  extracted_file.dat  p2p.bin  new 2  capture.pcap  udp_filtered_data.txt  new 3  data.safe.bin  cookies.sqlite-wal  recovery.bak124  logins.json
1 [{"id":1,"hostname":"https://www.instagram.com","httpRealm":"null","formSubmitURL":"","usernameField":"","passwordField":"","encryptedUsername":"MDIEEPgAAAAAAAAAAAAAAAAAAwFAYIKoZThvcNAwcECJ"}]

```

Gunakan ini untuk dekripsi

[https://github.com/unode/firefox\\_decrypt](https://github.com/unode/firefox_decrypt)

```

Select the Mozilla profile you wish to decrypt
1 -> ju1ay0ip.default
2 -> zuh147e6.default-release
2

Website:  https://www.instagram.com
Username:  'Sam'
Password:  'HOLOGY7{y0U_ar3_4_gR34t_D3tecT1v3!!!}'

```