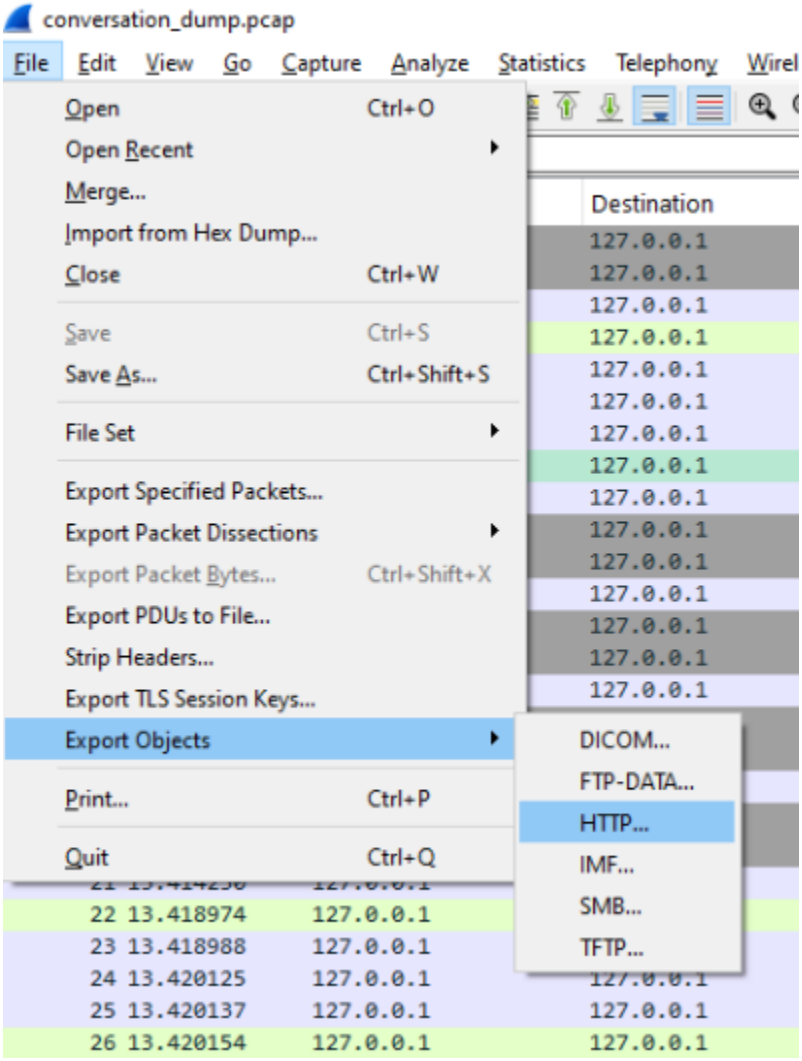
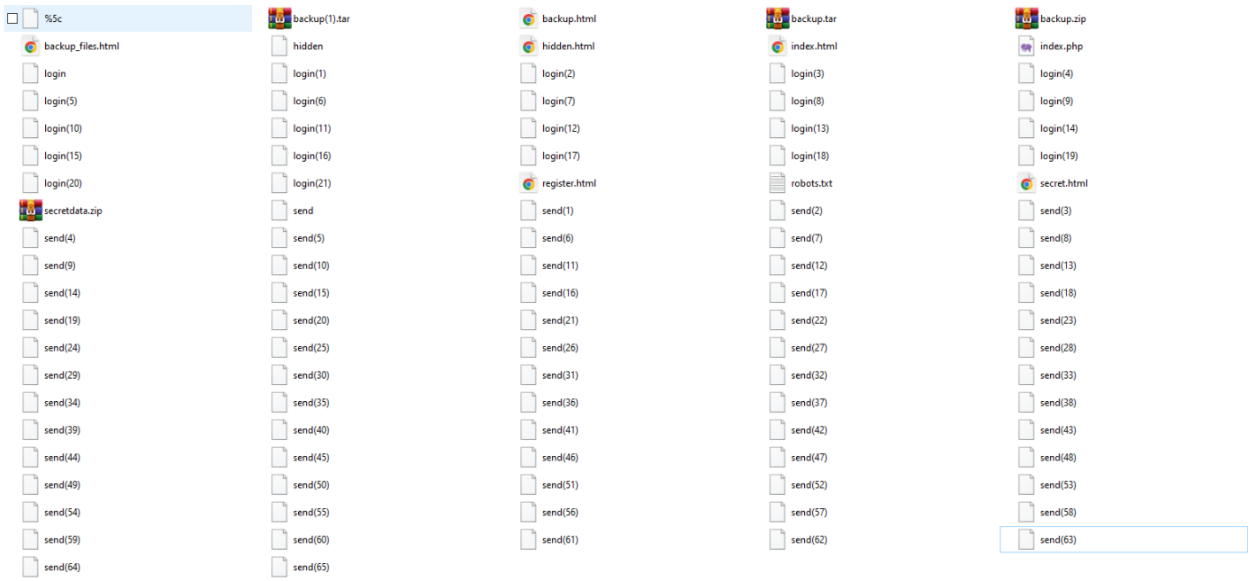
Dark Web Packer

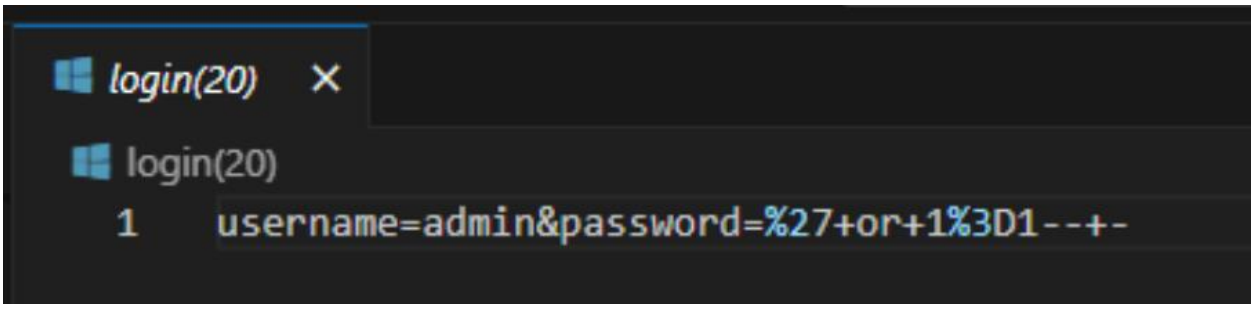
Después de abrir el archivo entregado en Wireshark y verificar algunos paquetes HTTP, vemos que tenemos algunos archivos HTML y otros.



Exportamos todos estos a una nueva carpeta y podemos ver los siguientes archivos. El que resalta es el secretdata.zip, el cual necesita una contraseña.



El login, login (2), login (4), etc. (con números pares) son intentos de un hacker de iniciar sesión, siendo el último exitoso después de realizar una inyección.

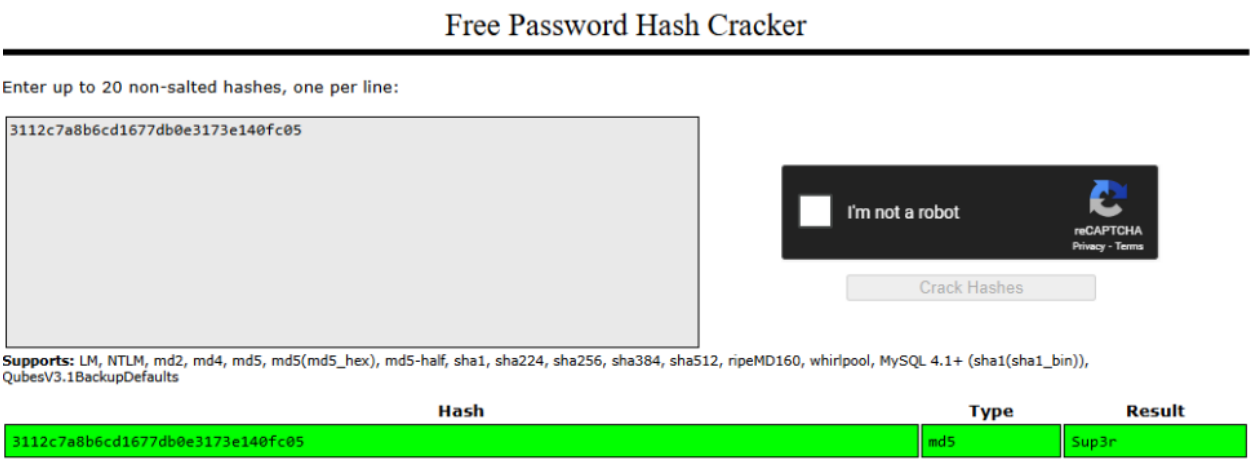


Posteriormente, al inspeccionar los 66 archivos de envío, podemos ver una conversación entre R1kS y Charon VI:

```
R1kS: Yo, are you here? Got something for you.
Charon VI: Hey, what's up?
R1kS: Was crawling on their website, 68b36d42880c527fb70086b1b97f4f34e49bc0d538f52607ec4009c552c2a63b.onion, or BlackVault as they call it. Managed to retrieve the
Charon VI: Great job. Thought since the begining that they encrypt their files in case of a breakout.
Charon VI: Luckily, more or less, they were holding some of their passowrds in a database located on another server. Glad I was able to find it.
R1kS: That's very good. So, will you give me the password for this secret archive?
Charon VI: I will, but something is not right. There are multiple strings, and I don't understand what to do with them. Maybe you can find out.
R1kS: Sure. Drop 'em here. Those guys will finally learn Dark Web is not for everyone. They will not even know what struck them.
Charon VI: I will drop 'em one by one.
Charon VI: 5dbc98dcc983a70728bd082d1a47546e
Charon VI: f72c915d8f575a5c0999b5f37b6d99b7
Charon VI: a20bba554bfa1580a9d4aa2b6879ed46
Charon VI: 02beeea47ee3cfe212e6bd843b9ce7d3
Charon VI: 3112c7a8b6cd1677db0e3173e140fc05
Charon VI: 50f4646135205fd4a5417e460cf71d3c
Charon VI: eb22cfa0890a2df3177966854a7176bc
Charon VI: 845f49aa19c955b849d57593bf09d224
Charon VI: 87f63931da79aa969ac4a776ce6cfb03
Charon VI: 9793d9d6041c80f46ad7c1f530c8bbf8
Charon VI: 2f88d89a8f50426a6285449be3286708
Charon VI: 61bd22f017588208a0cacdf9a1a7ca1e
Charon VI: a7623c8b76316e10538782371b709415
Charon VI: c6cca42180caba17e9e6882dc66cc6ee
Charon VI: 7c854900e46ebc5ee5680032b3e334de
Charon VI: ac81882b848b7673d73777ca22908c0d
Charon VI: 4ce97d67963edca55cdd21d46a68f5bb
Charon VI: 4abb62a00bccb775321f2720f2c7750b
Charon VI: 67e00e8ef738fe75afdb42b22e50371e
Charon VI: b561052e5697ee5f1491b5e350fb78e1
Charon VI: That's all.
R1kS: Wow, that's a lot of them! I'll see what I can do.
Charon VI: You are on your own. I feel like someone is listening to us right now. We've never met, bye!
R1kS: Yeah, I feel the same. Even this channel ain't secure no more. Bye!
```

Solo 33 de ellos son "utilizables", ya que el resto son una confirmación (OK).

Podemos usar un descifrador MD5 en línea para encontrar la contraseña del archivo zip (es el único archivo cifrado).



Después de esto, podemos construir un script mínimo en Python para obtener los demás.

```python
#!/usr/bin/env python3
import hashlib
import string

# 1) Hashes are pasted here
raw_hashes = """
5dbc98dcc983a70728bd082d1a47546e
f72c915d8f575a5c0999b5f37b6d99b7
a20bba554bfa1580a9d4aa2b6879ed46
02beeea47ee3cfe212e6bd843b9ce7d3
```

```
3112c7a8b6cd1677db0e3173e140fc05
50f4646135205fd4a5417e460cf71d3c
eb22cfa0890a2df3177966854a7176bc
845f49aa19c955b849d57593bf09d224
87f63931da79aa969ac4a776ce6cfb03
9793d9d6041c80f46ad7c1f530c8bbf8
2f88d89a8f50426a6285449be3286708
61bd22f017588208a0cacdf9a1a7ca1e
a7623c8b76316e10538782371b709415
c6cca42180caba17e9e6882dc66cc6ee
7c854900e46ebc5ee5680032b3e334de
ac81882b848b7673d73777ca22908c0d
4ce97d67963edca55cdd21d46a68f5bb
4abb62a00bccb775321f2720f2c7750b
67e00e8ef738fe75afdb42b22e50371e
b561052e5697ee5f1491b5e350fb78e1
""".strip().splitlines()

# 2) Clean up each hash
hashes = [h.strip().lower() for h in raw_hashes]

# 3) Candidate character set
charset = string.ascii_letters + string.digits + string.punctuation

def md5hex(s: str) -> str:
return hashlib.md5(s.encode("utf-8")).hexdigest()

password = ""

for index, target_hash in enumerate(hashes, start=1):
found_char = None

for c in charset:
if md5hex(password + c) == target_hash:
found_char = c
password += c
print(
f"{index:2d}: matched hash → char '{c}', "
f"password so far: '{password}'"
)
break

if not found_char:
```

```
print(f"Couldn't match hash #{index}: {target_hash}")
print(
"• Check for typos in your list, stray whitespace, "
"or if your charset needs expanding."
)
raise SystemExit(1)

print("\n Full password recovered:", password)
```

ESto nos revela la contraseña del archivo secretdata.zip, el cual contiene una imagen, que al revisar su metadata con exiftool encontramos la flag!

```
Custom Rendered                 : Normal
Exposure Mode                   : Auto
White Balance                   : Auto
Focal Length In 35mm Format     : 26 mm
Scene Capture Type              : Standard
Contrast                        : Normal
Sharpness                       : Normal
PrintIM Version                 : 0300
Compression                     : JPEG (old-style)
Thumbnail Offset                : 7660
Thumbnail Length                : 10557
Comment                         : FIS{d4rk_w3b_m3t4d4t4}
Image Width                     : 1920
Image Height                    : 1080
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:2 (2 1)
```