DAY-6 : CYBERSECURITY

CyberChef :

- CyberChef is a web-based tool developed by GCHQ (Government Communications Headquarters) to help analyze and process data in a user-friendly interface. It offers a wide range of functionalities like encryption, encoding, data parsing, conversion, and much more. What makes CyberChef powerful is its drag-and-drop recipe feature, where you can combine multiple operations in a sequence to manipulate and analyze data efficiently.
- CyberChef is particularly useful in cybersecurity and forensics analysis, where investigators need to quickly decode or manipulate data to find useful information like passwords, encrypted data, or hidden messages.

Use Case: Decoding Email and Password from SMTP Requests in Wireshark using CyberChef

Captured SMTP traffic using Wireshark, the email and password might be base64 encoded. To analyze and decode this information using CyberChef, follow these steps:

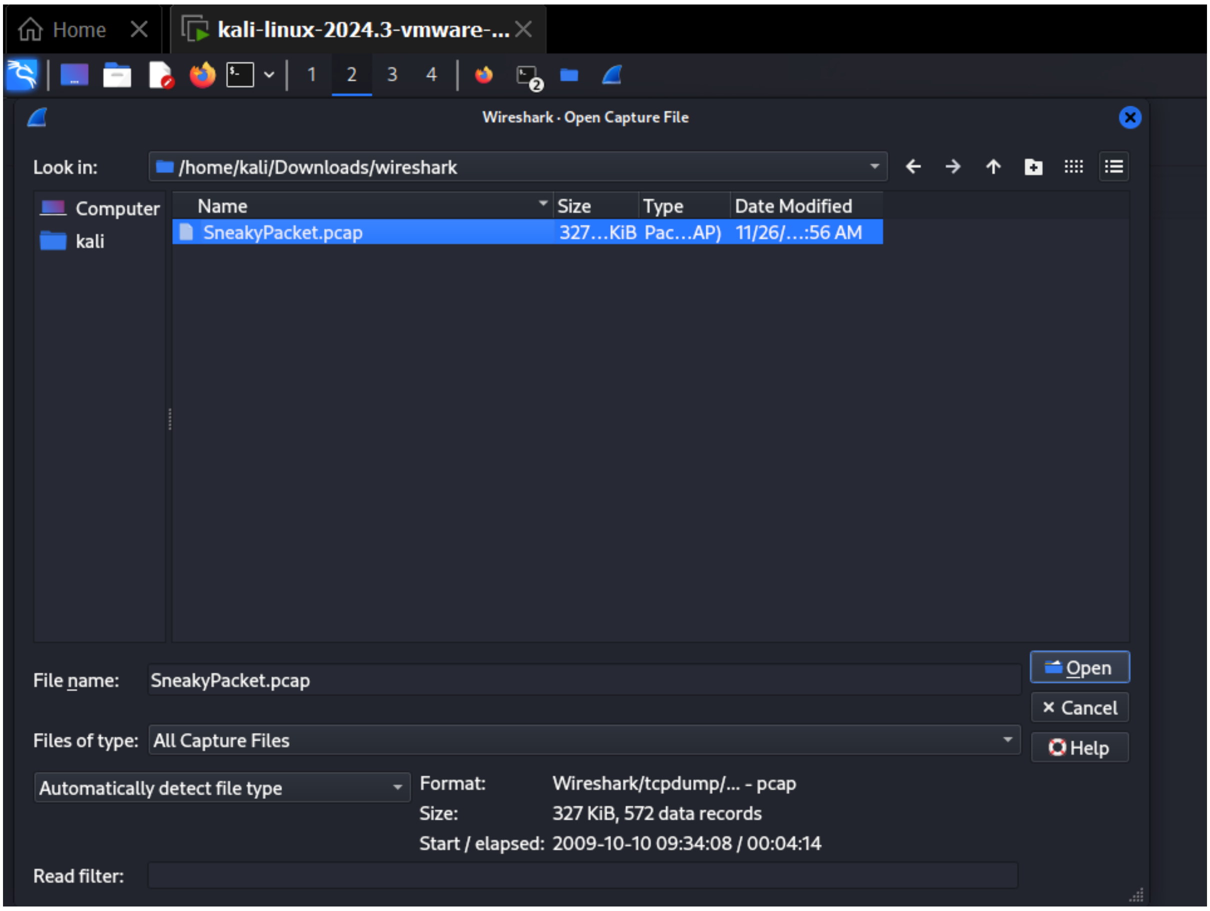
# 1. Capture SMTP Traffic in Wireshark:

- Open Wireshark and start a capture session.

- Apply a filter for SMTP traffic:

smtp

- Look for `**AUTH LOGIN**`, which typically contains the base64-encoded username (email) and password in SMTP authentication requests.



Look in: /home/kali/Downloads/wireshark

Name	Size	Type	Date Modified
SneakyPacket.pcap	327…KiB	Pac…AP)	11/26/…:56 AM

File name: SneakyPacket.pcap

Files of type: All Capture Files

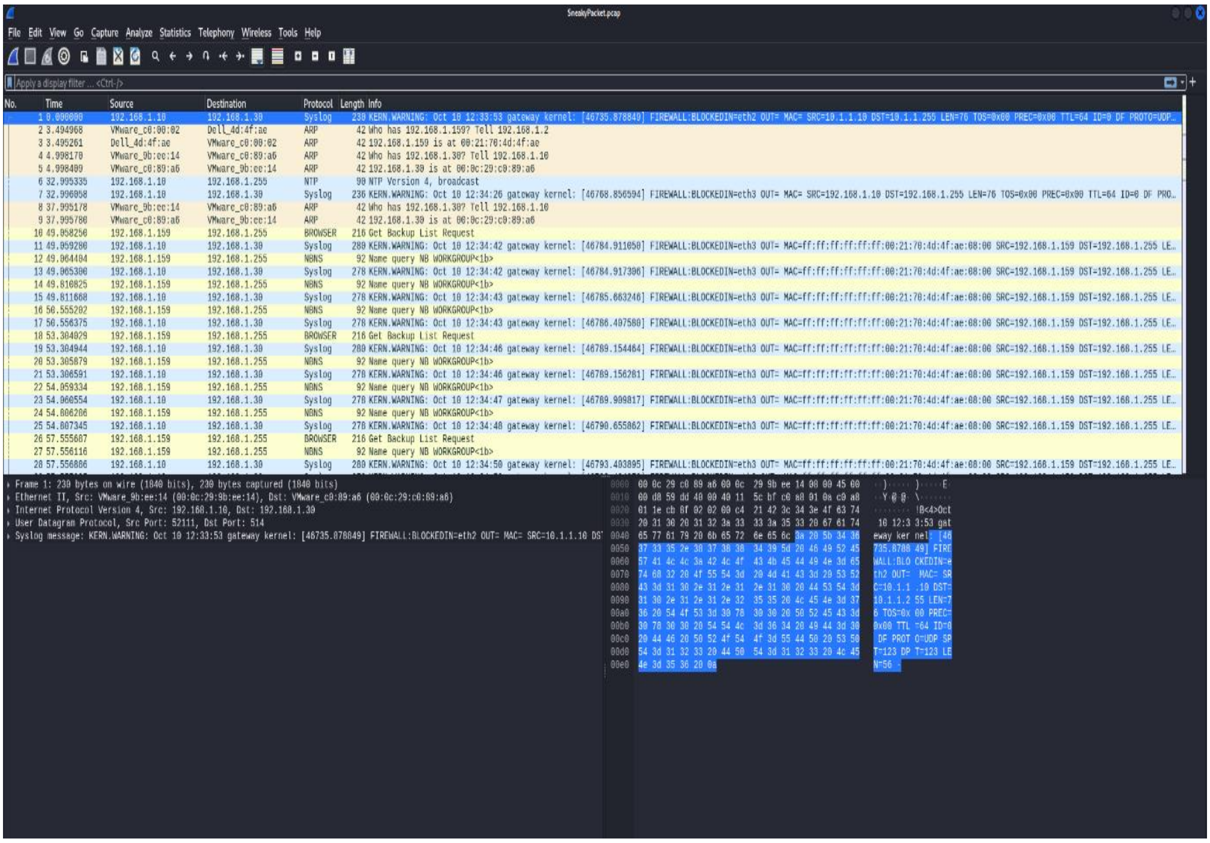
Automatically detect file type

Format: Wireshark/tcpdump/... - pcap

Size: 327 KiB, 572 data records

Start / elapsed: 2009-10-10 09:34:08 / 00:04:14

Read filter:



```
No.  Time       Source             Destination        Protocol Length Info
  1 0.000000   192.168.1.10       192.168.1.30       Syslog   230 KERN.WARNING: Oct 10 12:33:53 gateway kernel: [46735.878849] FIREWALL:BLOCKEDIN=eth2 OUT= MAC= SRC=10.1.1.10 DST=10.1.1.255 LEN=76 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF PROTO=UDP…
  2 3.494968   VMware_c0:00:02    Dell_4d:4f:ae      ARP       42 Who has 192.168.1.159? Tell 192.168.1.2
  3 3.495261   Dell_4d:4f:ae      VMware_c0:00:02    ARP       42 192.168.1.159 is at 00:21:70:4d:4f:ae
  4 4.998170   VMware_9b:ee:14    VMware_c0:89:a6    ARP       42 Who has 192.168.1.30? Tell 192.168.1.10
  5 4.998409   VMware_c0:89:a6    VMware_9b:ee:14    ARP       42 192.168.1.30 is at 00:0c:29:c0:89:a6
  6 32.995335  192.168.1.10       192.168.1.255      NTP       90 NTP Version 4, broadcast
  7 32.996058  192.168.1.10       192.168.1.30       Syslog   236 KERN.WARNING: Oct 10 12:34:26 gateway kernel: [46768.856594] FIREWALL:BLOCKEDIN=eth3 OUT= MAC= SRC=192.168.1.10 DST=192.168.1.255 LEN=76 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF PRO…
  8 37.995178  VMware_9b:ee:14    VMware_c0:89:a6    ARP       42 Who has 192.168.1.30? Tell 192.168.1.10
  9 37.995786  VMware_c0:89:a6    VMware_9b:ee:14    ARP       42 192.168.1.30 is at 00:0c:29:c0:89:a6
 10 49.058256  192.168.1.159      192.168.1.255      BROWSER  216 Get Backup List Request
 11 49.059286  192.168.1.10       192.168.1.30       Syslog   280 KERN.WARNING: Oct 10 12:34:42 gateway kernel: [46784.911050] FIREWALL:BLOCKEDIN=eth3 OUT= MAC=ff:ff:ff:ff:ff:ff:00:21:70:4d:4f:ae:08:00 SRC=192.168.1.159 DST=192.168.1.255 LE…
 12 49.064404  192.168.1.159      192.168.1.255      NBNS      92 Name query NB WORKGROUP<1b>
 13 49.065306  192.168.1.10       192.168.1.30       Syslog   278 KERN.WARNING: Oct 10 12:34:42 gateway kernel: [46784.917306] FIREWALL:BLOCKEDIN=eth3 OUT= MAC=ff:ff:ff:ff:ff:ff:00:21:70:4d:4f:ae:08:00 SRC=192.168.1.159 DST=192.168.1.255 LE…
 14 49.810825  192.168.1.159      192.168.1.255      NBNS      92 Name query NB WORKGROUP<1b>
 15 49.811668  192.168.1.10       192.168.1.30       Syslog   278 KERN.WARNING: Oct 10 12:34:43 gateway kernel: [46785.663246] FIREWALL:BLOCKEDIN=eth3 OUT= MAC=ff:ff:ff:ff:ff:ff:00:21:70:4d:4f:ae:08:00 SRC=192.168.1.159 DST=192.168.1.255 LE…
 16 50.555202  192.168.1.159      192.168.1.255      NBNS      92 Name query NB WORKGROUP<1b>
 17 50.556375  192.168.1.10       192.168.1.30       Syslog   278 KERN.WARNING: Oct 10 12:34:43 gateway kernel: [46786.407580] FIREWALL:BLOCKEDIN=eth3 OUT= MAC=ff:ff:ff:ff:ff:ff:00:21:70:4d:4f:ae:08:00 SRC=192.168.1.159 DST=192.168.1.255 LE…
 18 53.304029  192.168.1.159      192.168.1.255      BROWSER  216 Get Backup List Request
 19 53.304944  192.168.1.10       192.168.1.30       Syslog   280 KERN.WARNING: Oct 10 12:34:46 gateway kernel: [46789.154464] FIREWALL:BLOCKEDIN=eth3 OUT= MAC=ff:ff:ff:ff:ff:ff:00:21:70:4d:4f:ae:08:00 SRC=192.168.1.159 DST=192.168.1.255 LE…
 20 53.305879  192.168.1.159      192.168.1.255      NBNS      92 Name query NB WORKGROUP<1b>
 21 53.306591  192.168.1.10       192.168.1.30       Syslog   278 KERN.WARNING: Oct 10 12:34:46 gateway kernel: [46789.156281] FIREWALL:BLOCKEDIN=eth3 OUT= MAC=ff:ff:ff:ff:ff:ff:00:21:70:4d:4f:ae:08:00 SRC=192.168.1.159 DST=192.168.1.255 LE…
 22 54.059334  192.168.1.159      192.168.1.255      NBNS      92 Name query NB WORKGROUP<1b>
 23 54.060554  192.168.1.10       192.168.1.30       Syslog   278 KERN.WARNING: Oct 10 12:34:47 gateway kernel: [46789.909817] FIREWALL:BLOCKEDIN=eth3 OUT= MAC=ff:ff:ff:ff:ff:ff:00:21:70:4d:4f:ae:08:00 SRC=192.168.1.159 DST=192.168.1.255 LE…
 24 54.806206  192.168.1.159      192.168.1.255      NBNS      92 Name query NB WORKGROUP<1b>
 25 54.807345  192.168.1.10       192.168.1.30       Syslog   278 KERN.WARNING: Oct 10 12:34:48 gateway kernel: [46790.655862] FIREWALL:BLOCKEDIN=eth3 OUT= MAC=ff:ff:ff:ff:ff:ff:00:21:70:4d:4f:ae:08:00 SRC=192.168.1.159 DST=192.168.1.255 LE…
 26 57.555607  192.168.1.159      192.168.1.255      BROWSER  216 Get Backup List Request
 27 57.556116  192.168.1.159      192.168.1.255      NBNS      92 Name query NB WORKGROUP<1b>
 28 57.556886  192.168.1.10       192.168.1.30       Syslog   280 KERN.WARNING: Oct 10 12:34:50 gateway kernel: [46793.403895] FIREWALL:BLOCKEDIN=eth3 OUT= MAC=ff:ff:ff:ff:ff:ff:00:21:70:4d:4f:ae:08:00 SRC=192.168.1.159 DST=192.168.1.255 LE…
```

```
› Frame 1: 230 bytes on wire (1840 bits), 230 bytes captured (1840 bits)
› Ethernet II, Src: VMware_9b:ee:14 (00:0c:29:9b:ee:14), Dst: VMware_c0:89:a6 (00:0c:29:c0:89:a6)
› Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.1.30
› User Datagram Protocol, Src Port: 52111, Dst Port: 514
› Syslog message: KERN.WARNING: Oct 10 12:33:53 gateway kernel: [46735.878849] FIREWALL:BLOCKEDIN=eth2 OUT= MAC= SRC=10.1.1.10 DST
```

```
0000  00 0c 29 c0 89 a6 00 0c  29 9b ee 14 08 00 45 00   ··)·····)·····E·
0010  00 d8 59 dd 40 00 40 11  5c bf c0 a8 01 0a c0 a8   ··Y·@·@·\·······
0020  01 1e cb 8f 02 02 00 c4  21 42 3c 34 3e 4f 63 74   ········!B<4>Oct
0030  20 31 30 20 31 32 3a 33  33 3a 35 33 20 67 61 74    10 12:33:53 gat
0040  65 77 61 79 20 6b 65 72  6e 65 6c 3a 20 5b 34 36   eway kernel: [46
0050  37 33 35 2e 38 37 38 38  34 39 5d 20 46 49 52 45   735.878849] FIRE
0060  57 41 4c 4c 3a 42 4c 4f  43 4b 45 44 49 4e 3d 65   WALL:BLOCKEDIN=e
0070  74 68 32 20 4f 55 54 3d  20 4d 41 43 3d 20 53 52   th2 OUT= MAC= SR
0080  43 3d 31 30 2e 31 2e 31  2e 31 30 20 44 53 54 3d   C=10.1.1.10 DST=
0090  31 30 2e 31 2e 31 2e 32  35 35 20 4c 45 4e 3d 37   10.1.1.255 LEN=7
00a0  36 20 54 4f 53 3d 30 78  30 30 20 50 52 45 43 3d   6 TOS=0x00 PREC=
00b0  30 78 30 30 20 54 54 4c  3d 36 34 20 49 44 3d 30   0x00 TTL=64 ID=0
00c0  20 44 46 20 50 52 4f 54  4f 3d 55 44 50 20 53 50    DF PROTO=UDP SP
00d0  54 3d 31 32 33 20 44 50  54 3d 31 32 33 20 4c 45   T=123 DPT=123 LE
00e0  4e 3d 35 36 20 0a                                  N=56 ·
```

# 2. Extract Base64 Encoded Data from SMTP Packets:

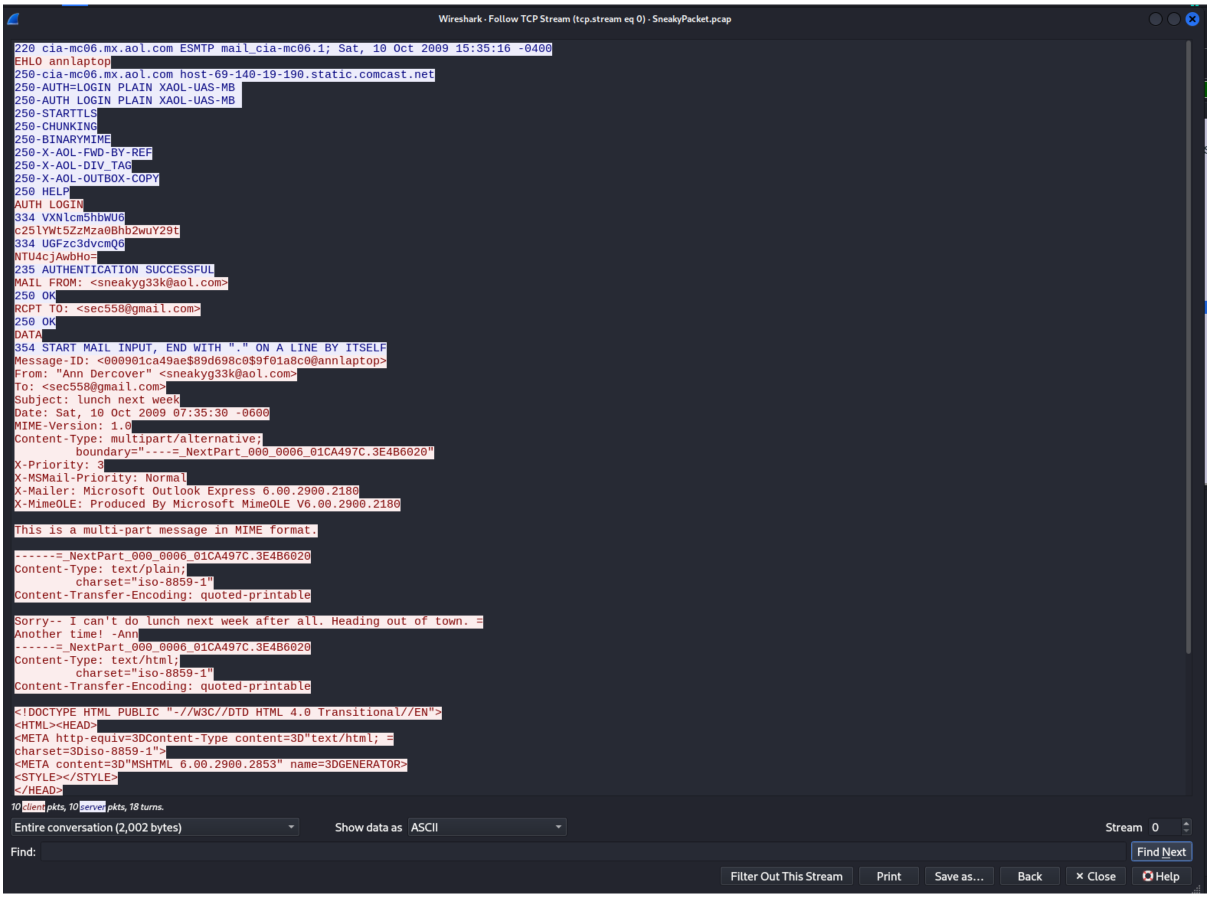
- Find the packets with `AUTH LOGIN`.

- The email (username) and password will be base64 encoded. Right-click on the packet, choose "Follow TCP Stream", and extract the base64 data from the packet content.

- The base64 string will look something like this:

dXNlcm5hbWU6IEV4YW1wbGVAbWFpbC5jb20=

cGFzc3dvcmQ6 IEV4YW1wbGVQYXNz



Wireshark · Follow TCP Stream (tcp.stream eq 0) · SneakyPacket.pcap

```
220 cia-mc06.mx.aol.com ESMTP mail_cia-mc06.1; Sat, 10 Oct 2009 15:35:16 -0400
EHLO annlaptop
250-cia-mc06.mx.aol.com host-69-140-19-190.static.comcast.net
250-AUTH=LOGIN PLAIN XAOL-UAS-MB
250-AUTH LOGIN PLAIN XAOL-UAS-MB
250-STARTTLS
250-CHUNKING
250-BINARYMIME
250-X-AOL-FWD-BY-REF
250-X-AOL-DIV_TAG
250-X-AOL-OUTBOX-COPY
250 HELP
AUTH LOGIN
334 VXNlcm5hbWU6
c25lYWt5ZzMza0Bhb2wuY29t
334 UGFzc3dvcmQ6
NTU4cjAwbHo=
235 AUTHENTICATION SUCCESSFUL
MAIL FROM: <sneakyg33k@aol.com>
250 OK
RCPT TO: <sec558@gmail.com>
250 OK
DATA
354 START MAIL INPUT, END WITH "." ON A LINE BY ITSELF
Message-ID: <000901ca49ae$89d698c0$9f01a8c0@annlaptop>
From: "Ann Dercover" <sneakyg33k@aol.com>
To: <sec558@gmail.com>
Subject: lunch next week
Date: Sat, 10 Oct 2009 07:35:30 -0600
MIME-Version: 1.0
Content-Type: multipart/alternative;
         boundary="----=_NextPart_000_0006_01CA497C.3E4B6020"
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 6.00.2900.2180
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.2180

This is a multi-part message in MIME format.

------=_NextPart_000_0006_01CA497C.3E4B6020
Content-Type: text/plain;
         charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

Sorry-- I can't do lunch next week after all. Heading out of town. =
Another time! -Ann
------=_NextPart_000_0006_01CA497C.3E4B6020
Content-Type: text/html;
         charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD>
<META http-equiv=3DContent-Type content=3D"text/html; =
charset=3Diso-8859-1">
<META content=3D"MSHTML 6.00.2900.2853" name=3DGENERATOR>
<STYLE></STYLE>
</HEAD>
```

10 client pkts, 10 server pkts, 18 turns.

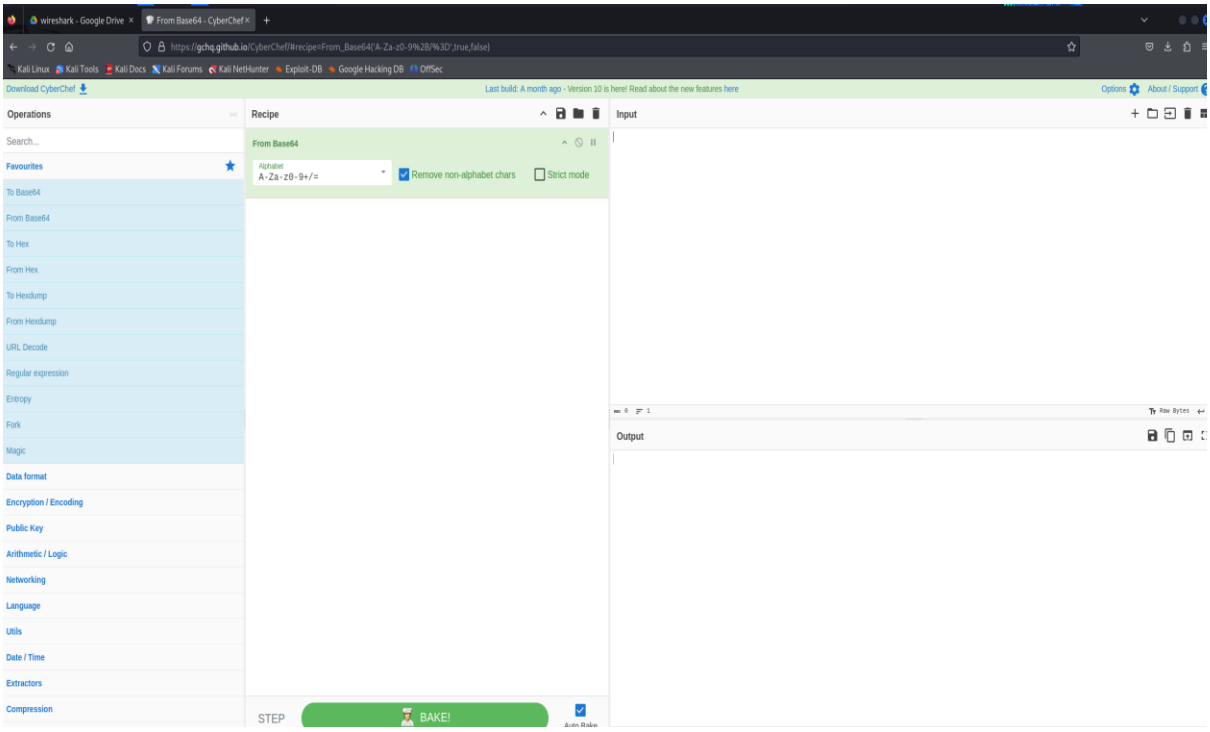
Entire conversation (2,002 bytes) Show data as ASCII Stream 0

Find:

# 3. Analyze the Data in CyberChef:

- Open CyberChef in your browser.

- Paste the base64-encoded username (email) and password into the input section.



https://gchq.github.io/CyberChef/#recipe=From_Base64('A-Za-z0-9%2B/%3D',true,false)

From Base64

Alphabet A-Za-z0-9+/=

☑ Remove non-alphabet chars

□ Strict mode

# 4. Use the "From Base64" Operation in CyberChef:

- In CyberChef's left-hand panel, search for the "From Base64" operation and drag it to the "Recipe" section.

- Apply the operation, and CyberChef will decode the base64 strings, revealing the plaintext email and password.

Example:

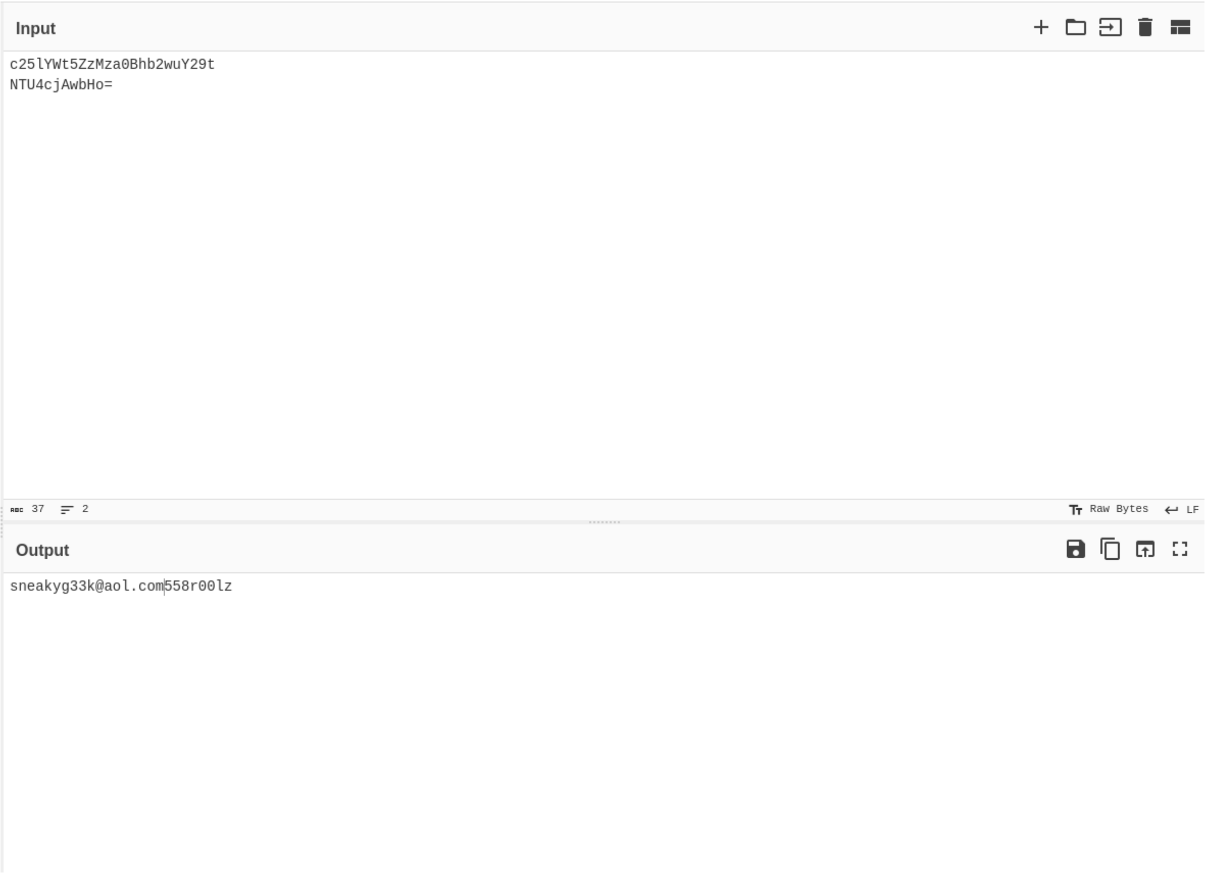
- Base64 string: `dXNlcm5hbWU6IEV4YW1wbGVAbWFpbC5jb20=`

- Decoded output: `username: Example@mail.com`

Similarly, for the password string:

- Base64 string: `cGFzc3dvcmQ6IEV4YW1wbGVQYXNz`

- Decoded output: `password: ExamplePass`

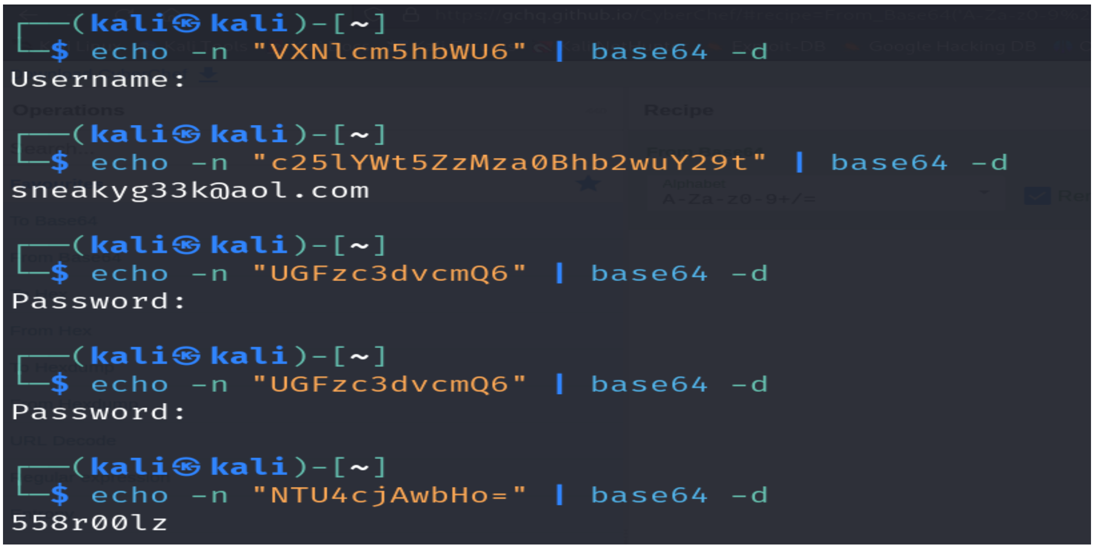


```
c25lYWt5ZzMza0Bhb2wuY29t
NTU4cjAwbHo=
```

```
sneakyg33k@aol.com558r00lz
```

# 5. Analyze the Results:

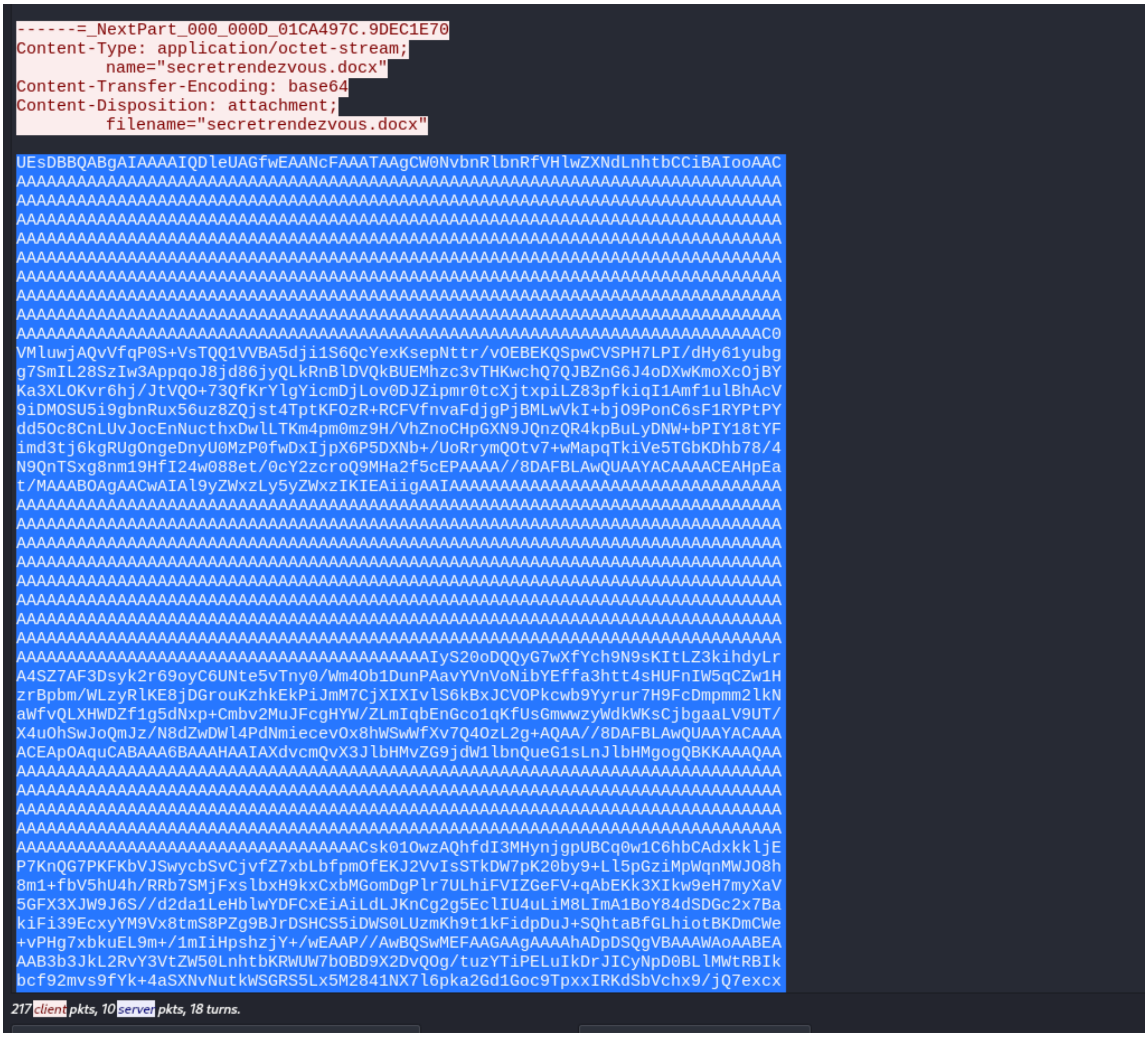
- Once the email and password are decoded, you can analyze them further for forensics purposes. This could involve checking if the credentials have been compromised, inspecting the source or destination of the emails, or performing further analysis on the associated traffic.



```
┌──(kali㉿kali)-[~]
└─$ echo -n "VXNlcm5hbWU6" | base64 -d
Username:
┌──(kali㉿kali)-[~]
└─$ echo -n "c25lYWt5ZzMza0Bhb2wuY29t" | base64 -d
sneakyg33k@aol.com
┌──(kali㉿kali)-[~]
└─$ echo -n "UGFzc3dvcmQ6" | base64 -d
Password:
┌──(kali㉿kali)-[~]
└─$ echo -n "UGFzc3dvcmQ6" | base64 -d
Password:
┌──(kali㉿kali)-[~]
└─$ echo -n "NTU4cjAwbHo=" | base64 -d
558r00lz
```

Figure: *Decoding using terminal with a base64 -d flag*

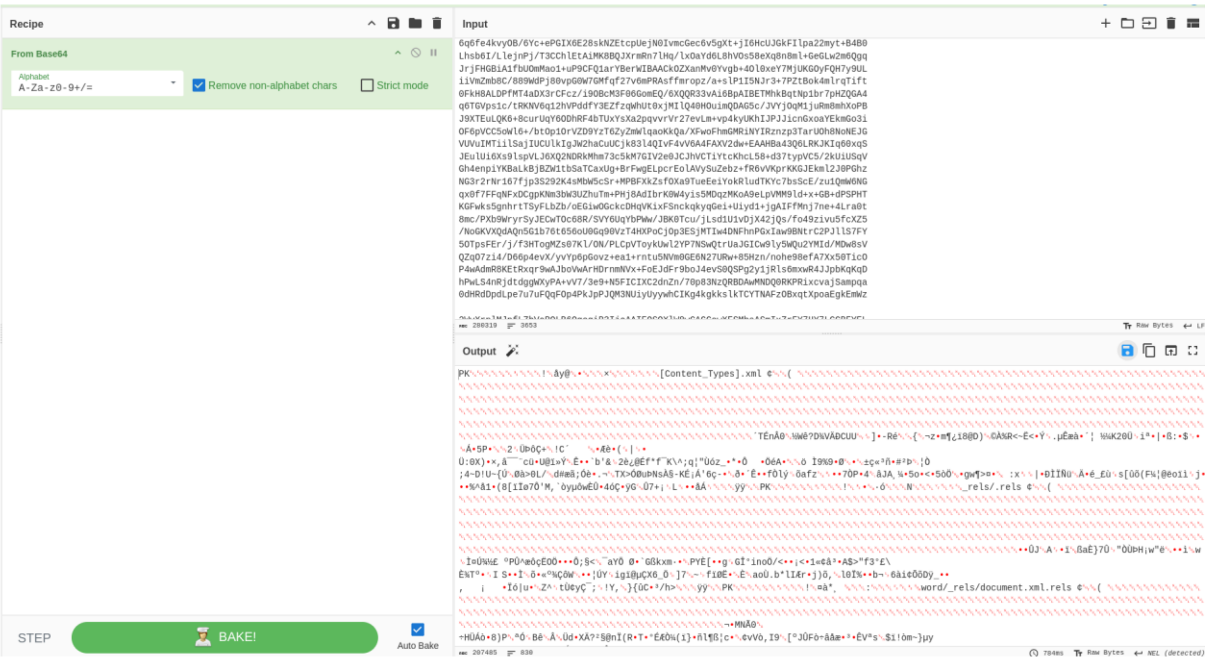
Using Further Attached data to find out further information:



```
------=_NextPart_000_000D_01CA497C.9DEC1E70
Content-Type: application/octet-stream;
        name="secretrendezvous.docx"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
        filename="secretrendezvous.docx"
```

```
UEsDBBQABgAIAAAAIQDleUAGfwEAANcFAAATAAgCW0NvbnRlbnRfVHlwZXNdLnhtbCCiBAIooAAC
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAC0
VMluwjAQvVfqP0S+VsTQQ1VVBA5dji1S6QcYexKsepNttr/vOEBEKQSpwCVSPH7LPI/dHy61yubg
g7SmIL28SzIw3AppqoJ8jd86jyQLkRnBlDVQkBUEMhzc3vTHKwchQ7QJBZnG6J4oDXwKmoXcOjBY
Ka3XLOKvr6hj/JtVQO+73QfKrYlgYicmDjLov0DJZipmr0tcXjtxpiLZ83pfkiqI1Amf1ulBhAcV
9iDMOSU5i9gbnRux56uz8ZQjst4TptKFOzR+RCFVfnvaFdjgPjBMLwVkI+bjO9PonC6sF1RYPtPY
dd5Oc8CnLUvJocEnNucthxDwlLTKm4pm0mz9H/VhZnoCHpGXN9JQnzQR4kpBuLyDNW+bPIY18tYF
imd3tj6kgRUgOngeDnyU0MzP0fwDxIjpX6P5DXNb+/UoRrymQOtv7+wMapqTkiVe5TGbKDhb78/4
N9QnTSxg8nm19HfI24w088et/0cY2zcroQ9MHa2f5cEPAAAA//8DAFBLAwQUAAYACAAAACEAHpEa
t/MAAABOAgAACwAIAl9yZWxzLy5yZWxzIKIEAiigAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAIyS20oDQQyG7wXfYch9N9sKItLZ3kihdyLr
A4SZ7AF3Dsyk2r69oyC6UNte5vTny0/Wm4Ob1DunPAavYVnVoNibYEffa3htt4sHUFnIW5qCZw1H
zrBpbm/WLzyRlKE8jDGrouKzhkEkPiJmM7CjXIXIvlS6kBxJCVOPkcwb9Yyrur7H9FcDmpmm2lkN
aWfvQLXHWDZf1g5dNxp+Cmbv2MuJFcgHYW/ZLmIqbEnGco1qKfUsGmwwzyWdkWKsCjbgaaLV9UT/
X4uOhSwJoQmJz/N8dZwDWl4PdNmiecevOx8hWSwWfXv7Q4OzL2g+AQAA//8DAFBLAwQUAAYACAAA
ACEApOAquCABAAA6BAAAHAAIAXdvcmQvX3JlbHMvZG9jdW1lbnQueG1sLnJlbHMgogQBKKAAAQAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACsk01OwzAQhfdI3MHynjgpUBCq0w1C6hbCAdxkkljE
P7KnQG7PKFKbVJSwycbSvCjvfZ7xbLbfpmOfEKJ2VvIsSTkDW7pK20by9+Ll5pGziMpWqnMWJO8h
8m1+fbV5hU4h/RRb7SMjFxslbxH9kxCxbMGomDgPlr7ULhiFVIZGeFV+qAbEKk3XIkw9eH7myXaV
5GFX3XJW9J6S//d2da1LeHblwYDFCxEiAiLdLJKnCg2g5EclIU4uLiM8LImA1BoY84dSDGc2x7Ba
kiFi39EcxyYM9Vx8tmS8PZg9BJrDSHCS5iDWS0LUzmKh9t1kFidpDuJ+SQhtaBfGLhiotBKDmCWe
+vPHg7xbkuEL9m+/1mIiHpshzjY+/wEAAP//AwBQSwMEFAAGAAgAAAAhADpDSQgVBAAAWAoAABEA
AAB3b3JkL2RvY3VtZW50LnhtbKRWUW7bOBD9X2DvQ0g/tuzYTiPELuIkDrJICyNpD0BLlMWtRBIk
bcf92mvs9fYk+4aSXNvNutkWSRS5Lx5M2841NX7l6pka2Gd1Goc9TpxxIRKdSbVchx9/jQ7excx
```

217 client pkts, 10 server pkts, 18 turns.

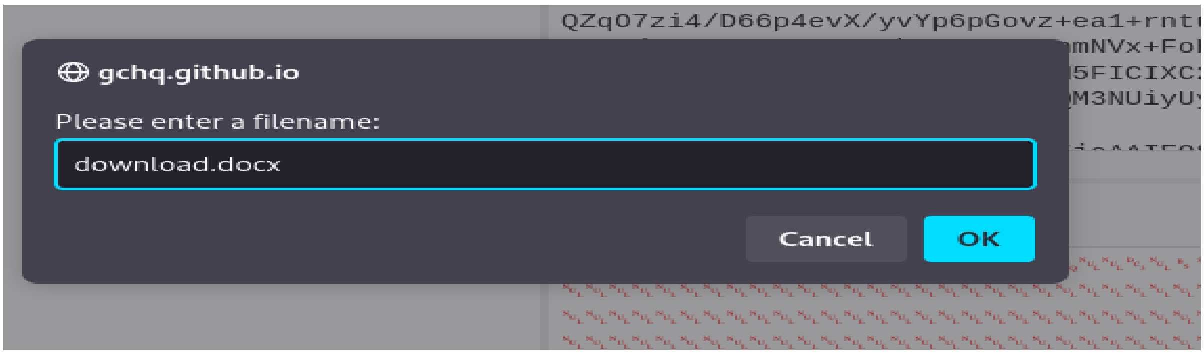


```
6q6fe4kvyOB/6Yc+ePGIX6E28skNZEtcpUejN0IvmcGec6v5gXt+jI6HcUJGkFIlpa22myt+B4B0
Lhsb6I/LlejnPj/T3CChlEtAiMK8BQJXrmRn7lHq/lxOaYd6L8hVOs58eXq8n8ml+GeGLw2m6Qgq
JrjFHGBiA1fbUOmMao1+uP9CFQ1arYBerWIBAACkOZXanMv0Yvgb+4Ol0xeY7MjUKGOyFQH7y9UL
iiVmZmb8C/889WdPj80vpG0W7GMfqf27v6mPRAsffmropz/a+slP1I5NJr3+7PZtBok4mlrqTift
0FkH8ALDPfMT4aDX3rCFcz/i9OBcM3F06GomEQ/6XQQR33vAi6BpAIBETMhkBqtNp1br7pHZQGA4
q6TGVps1c/tRKNV6q12hVPddfY3EZfzqWhUt0xjMIlQ40HOuimQDAG5c/JVYjOqM1juRm8mhXoPB
J9XTEuLQK6+8curUqY6ODhRF4bTUxYsXa2pqvvrVr27evLm+vp4kyUKhIJPJJicnGxoaYEkmGo3i
OF6pVCC5oWl6+/btOp1OrVZD9YzT6ZyZmZmWlqaoKkQa/XFwoFhmGMRiNYIRznzp3TarUOh8NoNEJG
VUVuIMTiilSajIUCUlkIgJW2haCuUCjk83l4QIvF4vV6A4FAXV2dw+EAAHBa43Q6LRKJKIq60xqS
JEulUi6Xs9lspVLJ6XQ2NDRkMhm73c5kM7GIV2e0JCJhVCTiYtcKhcL58+d37typVC5/2kUiUSqV
Gh4enpiYKBaLkBjBZW1tbSaTCaxUg+BrFwgELpcrEolAVySuZebz+fR6vVKprKKGJEkml2J0PGhz
NG3r2rNr167fjp3S292K4sMbW5cSr+MPBFXkZsfOXa9TueEeiYokRludTKYc7bsScE/zu1QmW6NG
qx0f7FFqNFxDCgpKNm3bW3UZhuTm+PHj8AdIbrK0W4yis5MDqzMKoA9eLpVMM9ld+x+GB+dPSPHT
KGFwks5gnhrtTSyFLbZb/oEGiwOGckDHqVKixFSnckqkyqGei+Uiyd1+jgAIFfMnj7ne+4Lra0t
8mc/PXb9WryrSyJECwTOc68R/SVY6UqYbPWw/JBK0Tcu/jLsd1U1vDjX42jQs/fo49zivu5fcXZ5
/NoGKVXQdAQn5G1b76t656oU0Gq90VzT4HXPoCjOp3ESjMTIw4DNFhnPGxIaw9BNtrC2PJllS7FY
5OTpsFEr/j/f3HTogMZs07Kl/ON/PLCpVToykUwl2YP7NSwQtrUaJGICw9ly5WQu2YMId/MDw8sV
QZqO7zi4/D66p4evX/yvYp6pGovz+ea1+rntu5NVm0GE6N27URw+85Hzn/nohe98efA7Xx50TicO
P4wAdmR8KEtRxqr9wAJboVwArHDrnmNVx+FoEJdFr9boJ4evS0QSPg2y1jRls6mxwR4JJpbKqKqD
hPwLS4nRjdtdggWXyPA+vV7/3e9+N5FICIXC2dnZn/70p83NzQRBDAwMNDQ0RKPRixcvajSampa
0dHRdDpdLpe7u7uFQqFOp4PkJpPJQM3NUiyUyywhCIKg4kgkslkTCYTNAFzOBxqtXpoaEgkEmWz
```

```
280319   3653                                         Raw Bytes   LF
```

```
PK ! åy@ × [Content_Types].xml ¢ (
 ´TÉnÂ0 ½Wê?D%VÄÐCUU ]-Ré { ¬z m¶¿ì8@D) ©À%R<~É<Ý .µÊæà ´¦ ½¼K20Ü iª |ß: $
Á 5P 2 ÜÞôÇ+ !C´ Æè ( |
Ü:0X) ×,â¯¯cü U@ï»Ý Ê `b'& 2è¿@Éf*f¯K\^;q¦"Üóz_ * Ô ÖéA ö Ì9%9 Ø ±ç«³ñ #²Þ ¦Ò
;4~D!U~{Ú Øà>0L/ d#æã;Óè .¬ TX>ÓØuÞNsÀ§-KÉ¡Á'6ç- ð ´Ê fÒlý õafz 7ÒP 4 âJA¸¼ 5o < 5òÖ gw¶>¤ :x |ÐÌÏÑü Ä é_£ù s[ûõ(F¼¦@èoîì j
%^å1 (8[ïÌø7Ô'M,`òyµôwÉÛ 4óÇ ÿG Û7+¡ L åÁ ÿÿ PK ! ó N _rels/.rels ¢ (
ÛJ A ï ßaÊ}7Û "ÒÙÞH¡w"ë ì w
Ì¤Ú%½£ ºPÛ^æôçÉOÖ Ô;§< ¯aYÕ Ø `Gßkxm· PYÈ[ g GÎ°inoÕ/< ¡< 1«¢å³ A$>"f3°£\
È%Tº I S Ì õ «º%ÇõW ¦ÚY igï@µÇX6_Õ ]7 ~ fïØË È aoÙ.b*lIÆr j)õ, l0Ï% b¬ 6àí¢ÕõDÿ_
, ¡ Ïó|u Z^ tÙ¢yÇ¯; !Y, }{ûC ³/h> ÿÿ PK ! ¤à*¸ : word/_rels/document.xml.rels ¢ (
¬ MNÃ0
÷HÜÁò 8)P ªÓ Bê Â Üd XÄ?²§@nÏ(R T °ÉÆÒ¼(ï} ñl¶ß¦c ¢vVò,I9 [ºJÛFò÷ââæ ³ ÊVªs $ï!òm~}µy
```

```
207485   830                          784ms   Raw Bytes   NEL (detected)
```



QZqO7zi4/D66p4evX/yvYp6pGovz+ea1+rnt

gchq.github.io

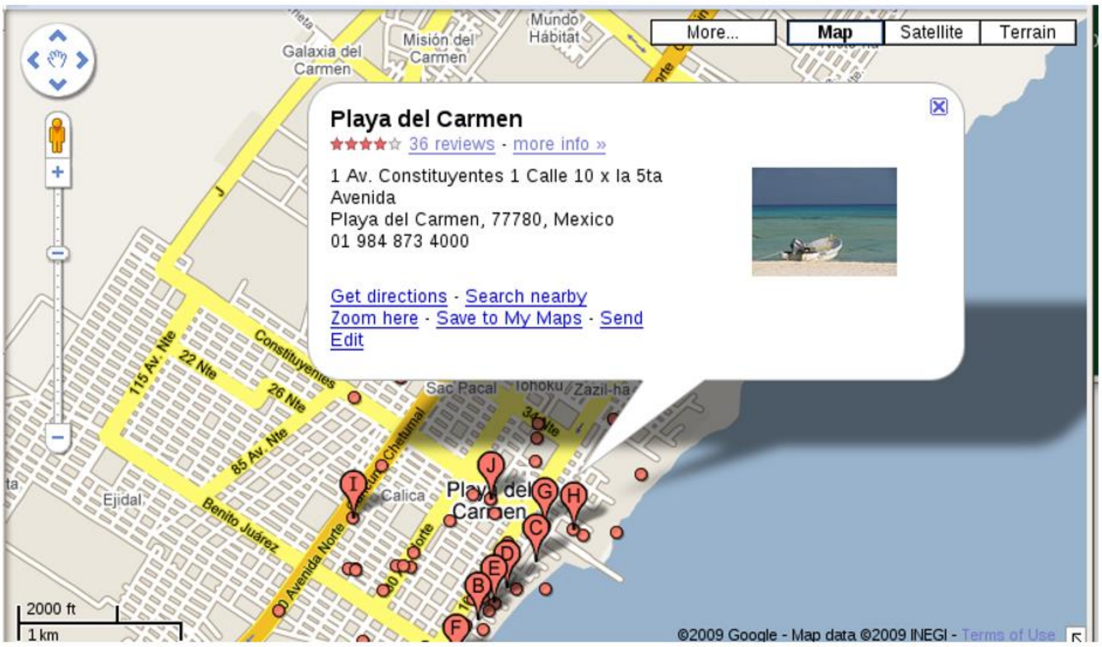
Please enter a filename:

download.docx

Figure: *Download the encoded zip file as .docx*

Analyzing the .docx which reveals an address:

Meet me at the fountain near the rendezvous point. Address below. I'm bringing all the cash.



Playa del Carmen

★★★★☆ 36 reviews - more info »

1 Av. Constituyentes 1 Calle 10 x la 5ta
Avenida
Playa del Carmen, 77780, Mexico
01 984 873 4000

Get directions - Search nearby
Zoom here - Save to My Maps - Send
Edit

Figure: *Decoding the .docx file*

Autopsy:

To decode information from a file, such as a forensic disk image like Treasure.E01, using Autopsy, follow these steps:

Step-by-Step Guide to Analyze Treasure.E01 with Autopsy

# 1. Download and Install Autopsy:

- Ensure you have Autopsy installed on your machine. You can download it from the official website: [Autopsy Download](https://www.sleuthkit.org/autopsy/download.php).

# 2. Create a New Case:

- Launch Autopsy.

- Click on "Create New Case" and enter a case name (e.g., Treasure_Analysis), a base directory, and any other required details. Click Next.

- Fill in the investigator's name and any other optional details.

- Once ready, click Finish to create the case.

# 3. Add the Treasure.E01 Image to the Case:

- After creating the case, Autopsy will prompt you to add a data source.

- Select "Add Data Source" and choose "Disk Image or VM File" as the data source type.

- Browse for the Treasure.E01 file and select it. Click Next.

# 4. Configure the Ingest Modules:

- Autopsy offers several modules to analyze the file. You can select the ones relevant to your investigation.

Some common modules to include are:

- File Type Identification: Identifies the file types present in the image.

- Keyword Search: Allows you to search for specific strings or patterns within the image.

- Extract EXIF Metadata: Useful if you're analyzing images with metadata.

- Web Artifacts: Helpful for extracting internet activity data (e.g., history, cookies).

- Hash Lookup: Matches files against known hashes in databases (like NSRL).

- File Analysis: Scans for anomalies or suspicious file types.

- Email Parser: Extracts and parses email-related data if any are found in the image.

After selecting the appropriate modules, click Next.

# 5. Start the Analysis:

- Autopsy will now start processing the Treasure.E01 file based on the selected modules. Depending on the size of the file, this can take some time.

- Once the analysis is complete, you can navigate the following sections:

  - File Browser: Allows you to browse through the file system of the disk image.

  - Results: Lists the files and artifacts discovered during analysis, including any anomalies, images, documents, or encrypted files.

# 6. Searching for Artifacts (Decrypted or Encoded Data):

- To find specific information, like encoded or encrypted data, do the following:

  - Use the Keyword Search to look for specific terms (e.g., base64 encoded strings, password hashes, etc.).

  - Check the File Types or File Signatures module to identify unusual files, such as encrypted or compressed files that might hold hidden data.

  - Look under the Results section for identified artifacts, such as web activity, user documents, or hidden files.

# 7. Analyze Suspicious Files:

- If you find files that are encoded (like base64 or other formats), you can export those files from Autopsy.

- Once exported, you can use tools like CyberChef or other decoding tools to decode the content.

# 8. Examine File Metadata and Carve Files:

- Autopsy will automatically extract metadata from images, documents, and other file types.

- Use the Carved Files feature to recover deleted or fragmented files.

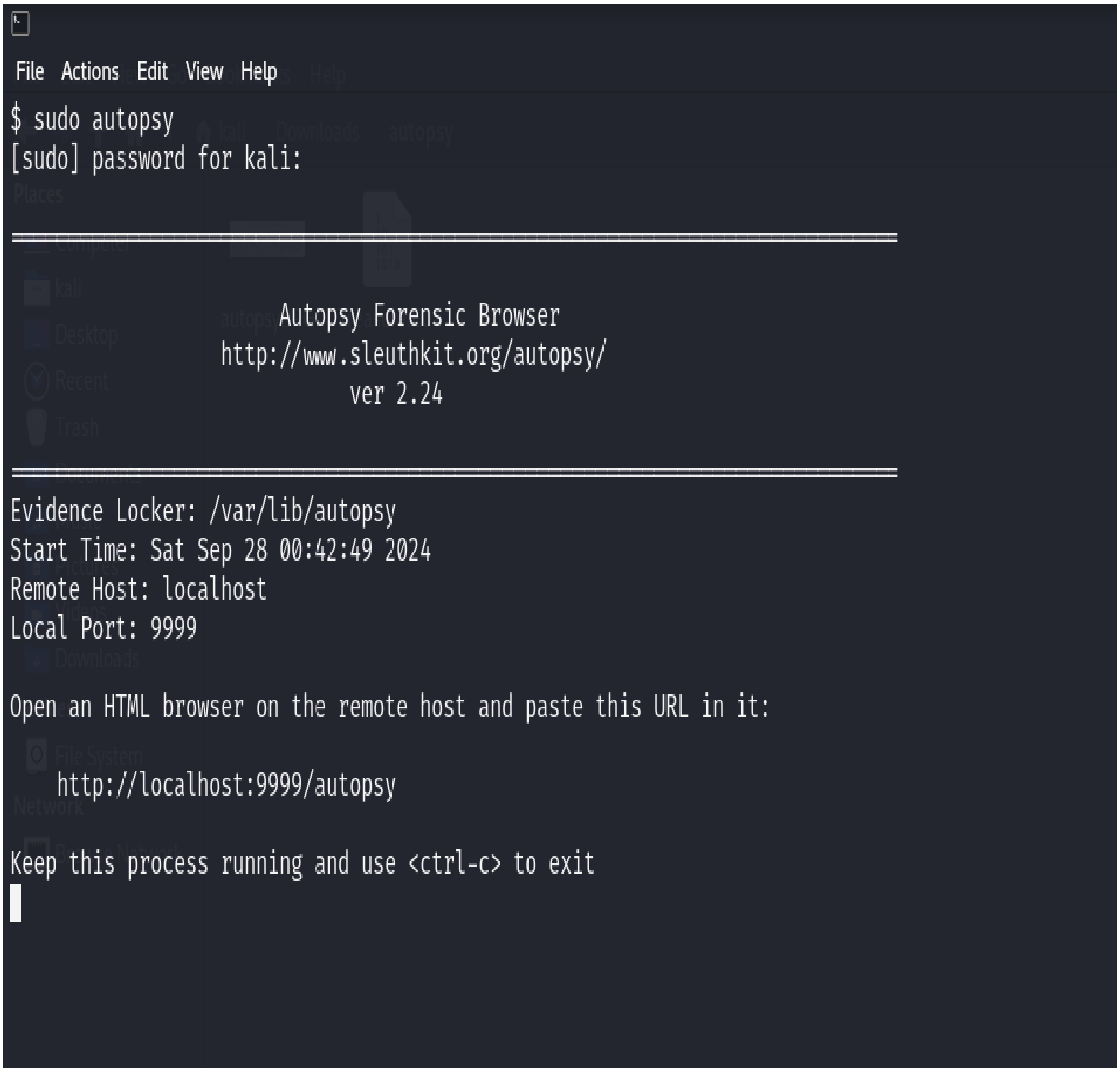
# 9. Export and Report the Findings:

- After analyzing the disk image, you can export the findings and generate a report.

- Go to Generate Report and choose the format (e.g., HTML, CSV, or Excel).

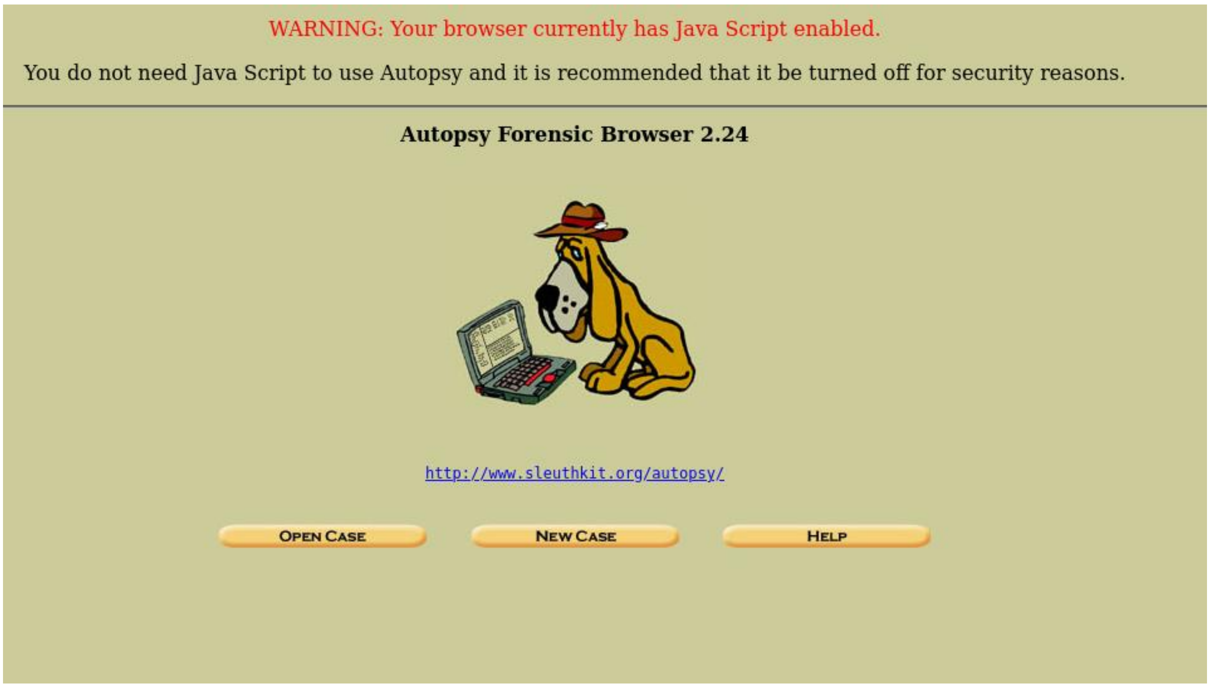
- Review the report, which will include the artifacts, decoded data, and other important forensic findings.

Step by Step Examples:



```
$ sudo autopsy
[sudo] password for kali:

============================================================================

                       Autopsy Forensic Browser
                  http://www.sleuthkit.org/autopsy/
                              ver 2.24

============================================================================
Evidence Locker: /var/lib/autopsy
Start Time: Sat Sep 28 00:42:49 2024
Remote Host: localhost
Local Port: 9999

Open an HTML browser on the remote host and paste this URL in it:

    http://localhost:9999/autopsy

Keep this process running and use <ctrl-c> to exit
```

Figure: *Loading autopsy from terminal*



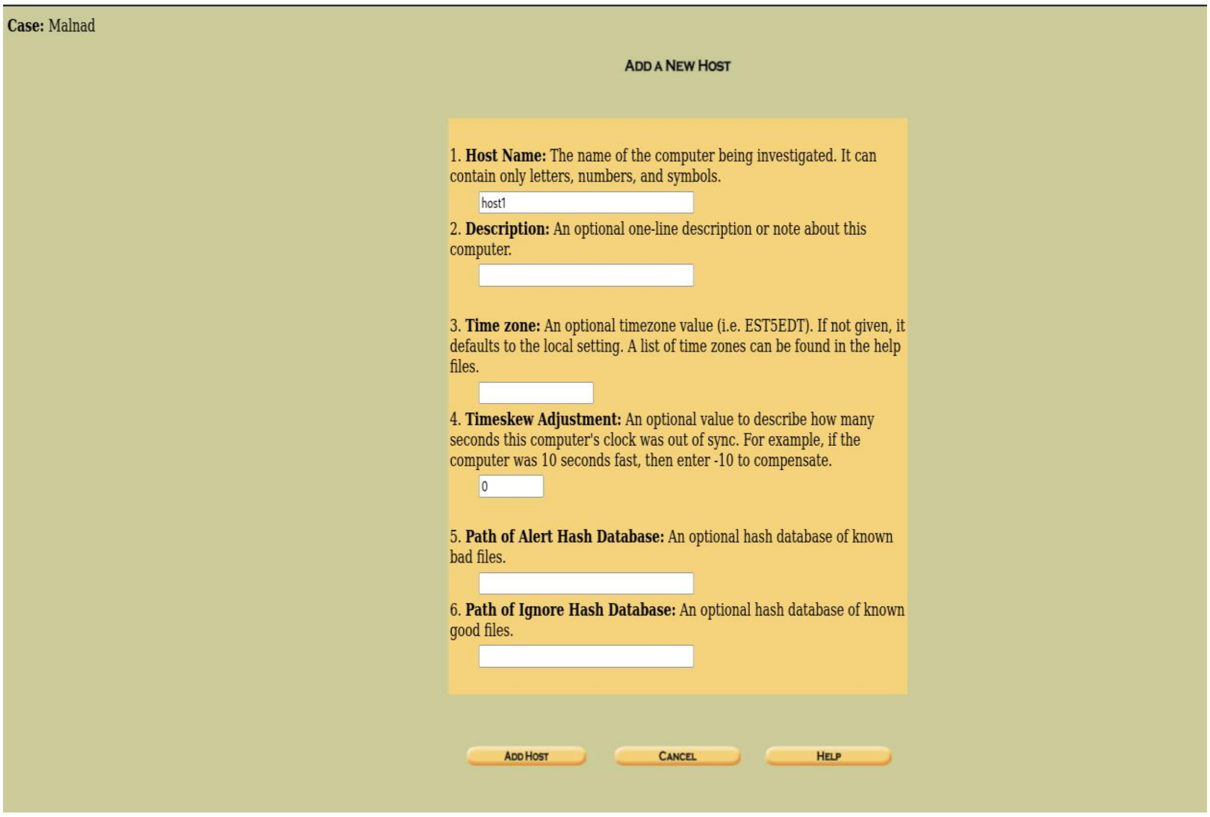
WARNING: Your browser currently has Java Script enabled.

You do not need Java Script to use Autopsy and it is recommended that it be turned off for security reasons.

Autopsy Forensic Browser 2.24

http://www.sleuthkit.org/autopsy/

Figure: *Autopsy interface on localhost*



Case: Malnad

ADD A NEW HOST

1. **Host Name:** The name of the computer being investigated. It can contain only letters, numbers, and symbols.

host1

2. **Description:** An optional one-line description or note about this computer.

3. **Time zone:** An optional timezone value (i.e. EST5EDT). If not given, it defaults to the local setting. A list of time zones can be found in the help files.

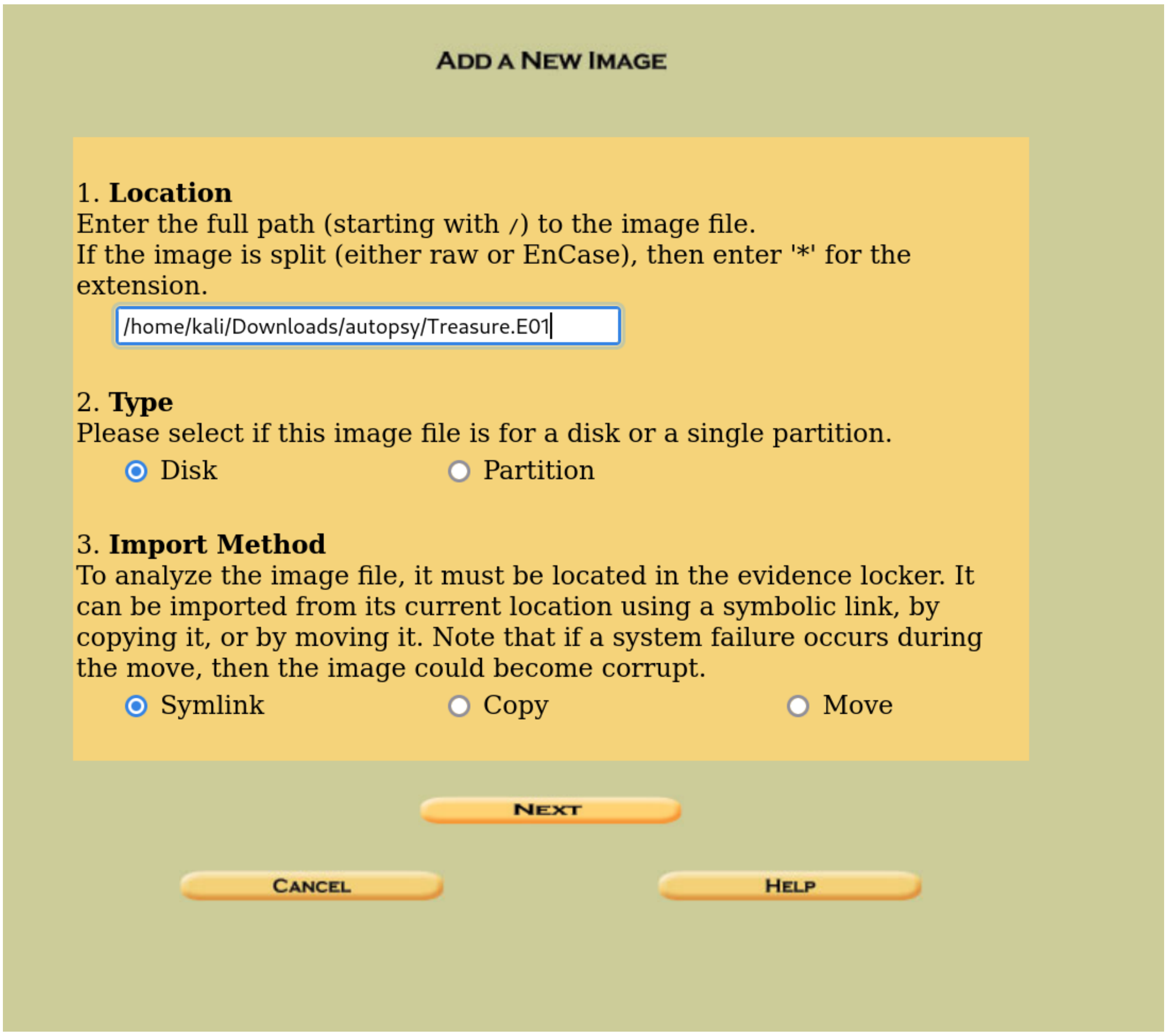
4. **Timeskew Adjustment:** An optional value to describe how many seconds this computer's clock was out of sync. For example, if the computer was 10 seconds fast, then enter -10 to compensate.

0

5. **Path of Alert Hash Database:** An optional hash database of known bad files.

6. **Path of Ignore Hash Database:** An optional hash database of known good files.

Figure: *New Case*



ADD A NEW IMAGE

1. **Location**
Enter the full path (starting with /) to the image file.
If the image is split (either raw or EnCase), then enter '*' for the extension.

/home/kali/Downloads/autopsy/Treasure.E01

2. **Type**
Please select if this image file is for a disk or a single partition.

● Disk ○ Partition

3. **Import Method**
To analyze the image file, it must be located in the evidence locker. It can be imported from its current location using a symbolic link, by copying it, or by moving it. Note that if a system failure occurs during the move, then the image could become corrupt.

● Symlink ○ Copy ○ Move

Figure: *Adding a new image by providing the absolute path of Treasure.E01*

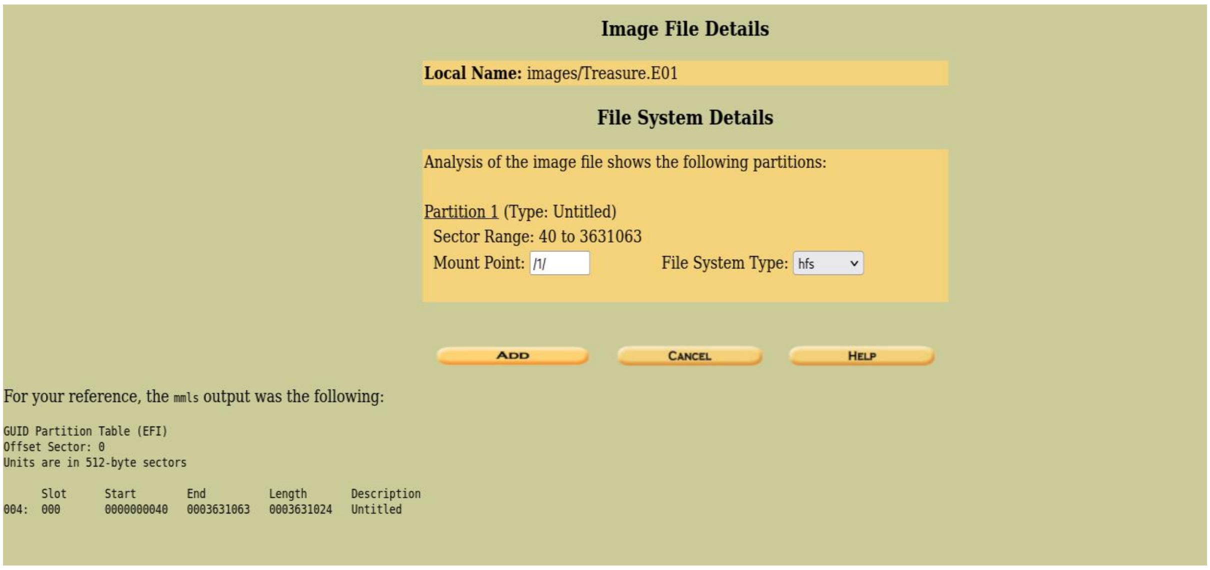


Image File Details

Local Name: images/Treasure.E01

File System Details

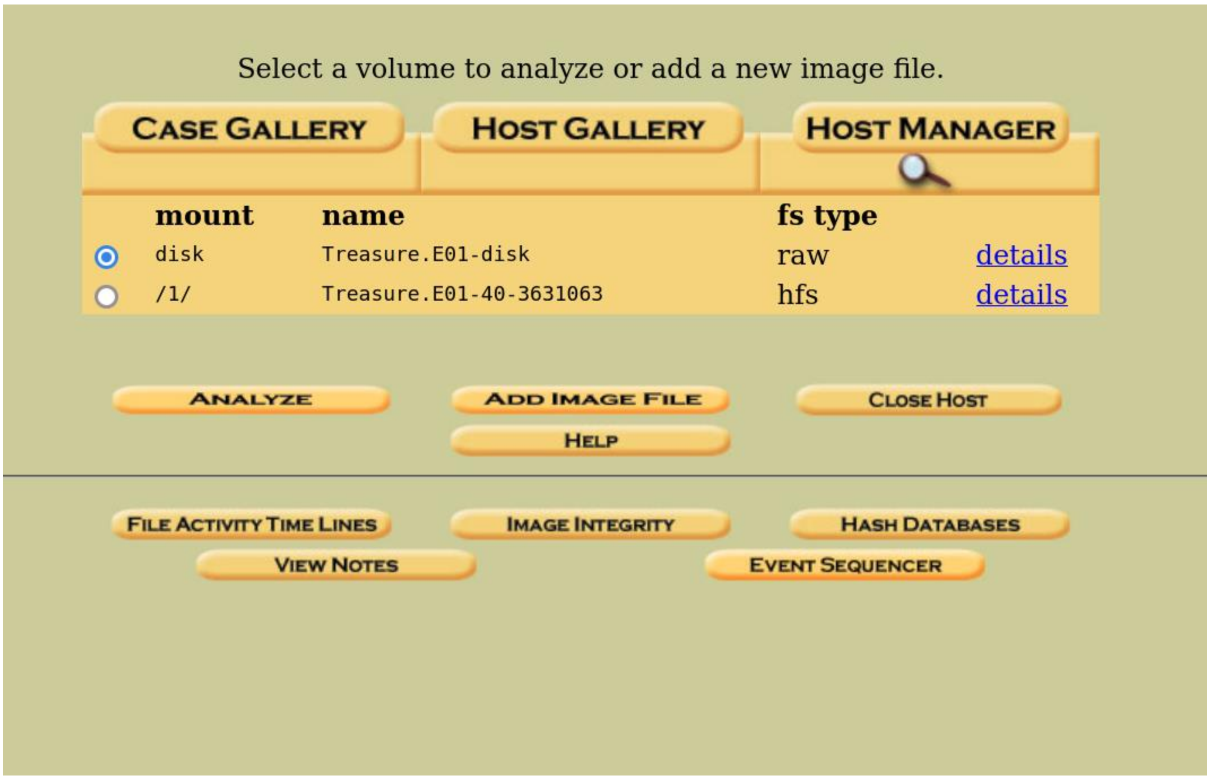
Analysis of the image file shows the following partitions:

Partition 1 (Type: Untitled)
Sector Range: 40 to 3631063
Mount Point: /1/ File System Type: hfs

For your reference, the mmls output was the following:

```
GUID Partition Table (EFI)
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
004:  000       0000000040   0003631063   0003631024   Untitled
```

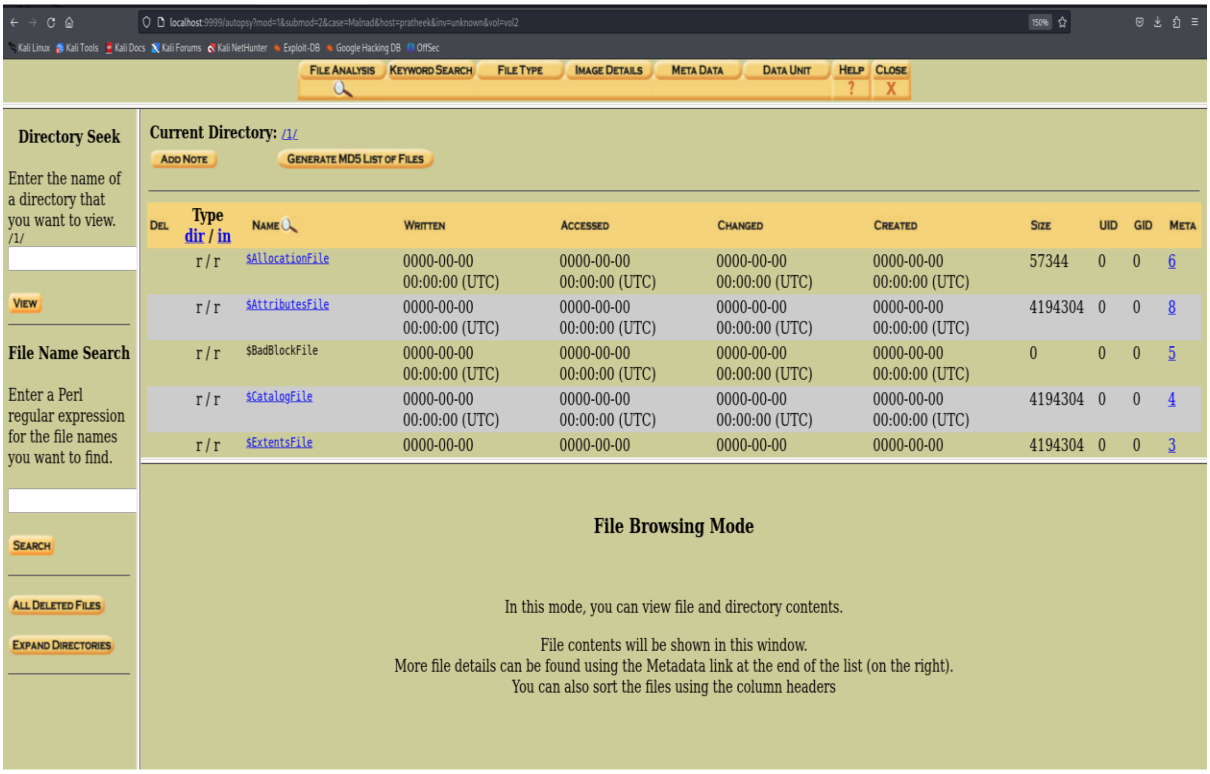
Figure: *Change file system type to hfs*



Select a volume to analyze or add a new image file.

	mount	name	fs type	
●	disk	Treasure.E01-disk	raw	details
○	/1/	Treasure.E01-40-3631063	hfs	details

Figure: *Select /l/ and click analyze*



Directory Seek

Enter the name of a directory that you want to view.
/1/

File Name Search

Enter a Perl regular expression for the file names you want to find.

Current Directory: /1/

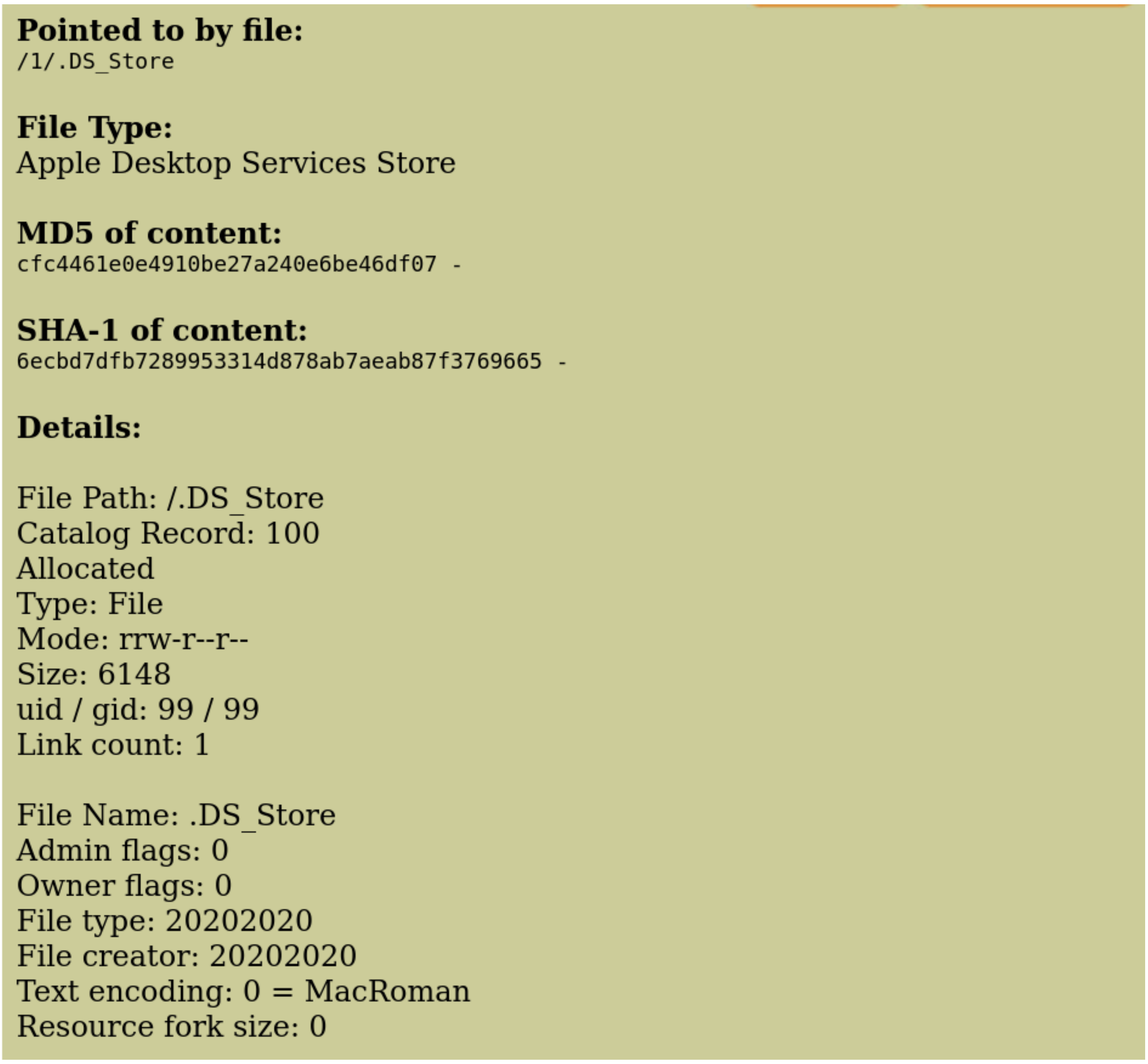
Del	Type dir / in	Name	Written	Accessed	Changed	Created	Size	UID	GID	Meta
	r / r	$AllocationFile	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	57344	0	0	6
	r / r	$AttributesFile	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	4194304	0	0	8
	r / r	$BadBlockFile	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0	0	0	5
	r / r	$CatalogFile	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	4194304	0	0	4
	r / r	$ExtentsFile	0000-00-00	0000-00-00	0000-00-00	0000-00-00	4194304	0	0	3

File Browsing Mode

In this mode, you can view file and directory contents.

File contents will be shown in this window.
More file details can be found using the Metadata link at the end of the list (on the right).
You can also sort the files using the column headers

Figure: *List of all directories in the image*



Pointed to by file:
`/1/.DS_Store`

File Type:
Apple Desktop Services Store

MD5 of content:
`cfc4461e0e4910be27a240e6be46df07 -`

SHA-1 of content:
`6ecbd7dfb7289953314d878ab7aeab87f3769665 -`

Details:

```
File Path: /.DS_Store
Catalog Record: 100
Allocated
Type: File
Mode: rrw-r--r--
Size: 6148
uid / gid: 99 / 99
Link count: 1

File Name: .DS_Store
Admin flags: 0
Owner flags: 0
File type: 20202020
File creator: 20202020
Text encoding: 0 = MacRoman
Resource fork size: 0
```

Figure: *Searching Meta Data to find the type of text encoding*