Info:

One of our DFIR-Guys is analyzing an infected host but is kinda stuck. We know this host is communicating with

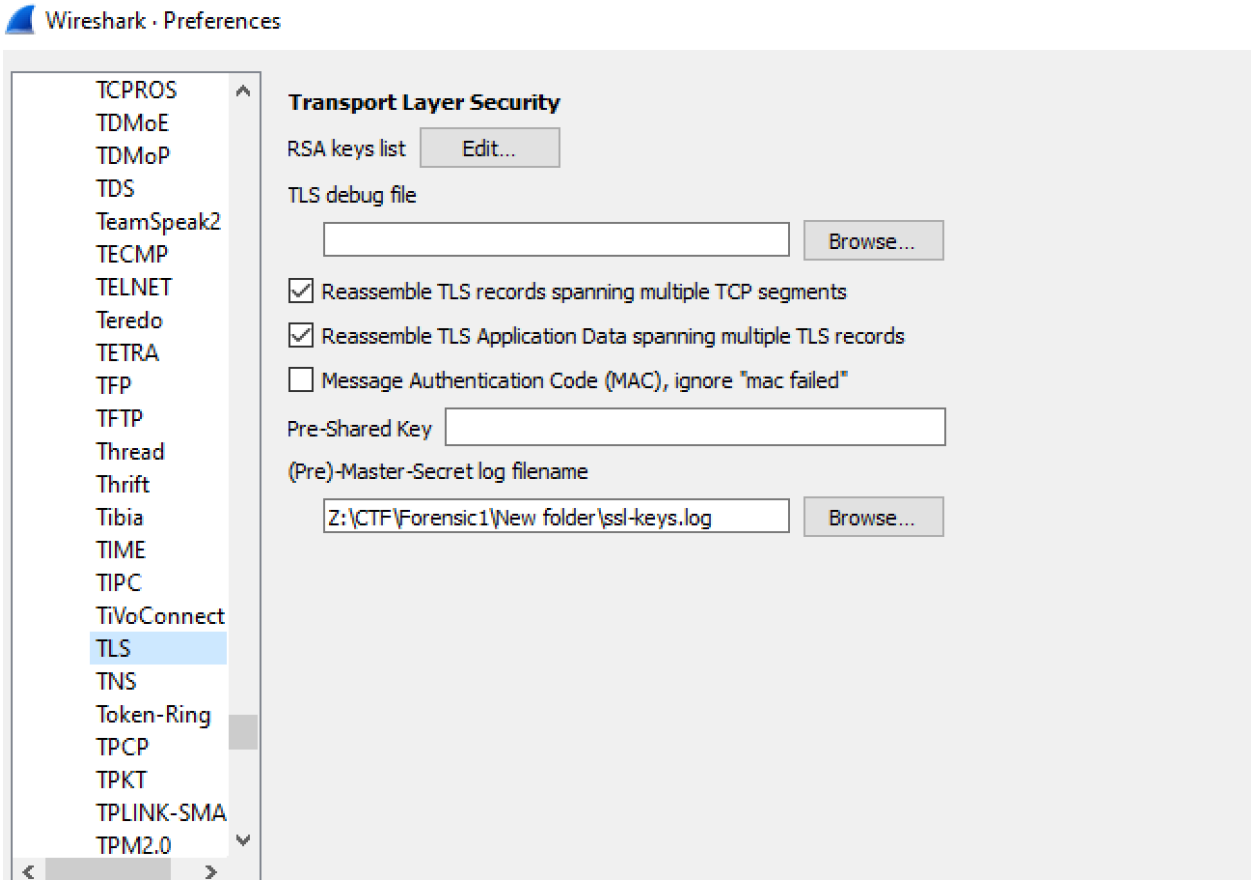
some form of C2 but we do not know how it works. We need to crack this to get to the encryption key. Can you help us?

Task:

Find the Key (Flag)

---

1. Provided are 2 files. Forensic1.pcapng and ssl-keys.log. Both can be analyzed with Wireshark. PCAPNG is the captured traffic and ssl-keys.log can be used to decrypt encrypted traffic.
2. Open up Wireshark and load the pcap-file.
3. Edit > Preferences > Protocols > TLS > (Pre)-Master-Secret log filename > Select the ssl-keys.log



4. Go through the traffic and check if you find anything suspicious. There are several ways to filter out the traffic. In this case we find a lot of irrelevant

data belonging to twitch.tv and usual browsing. Going through the traffic you will also find IMAP-Traffic with seemingly encrypted content.

Protocol	Length	Info
TLSv1.3	134	Change Cipher Spec, Finished
TCP	60	993 → 50818 [ACK] Seq=4296 Ack=598 Win=65535 Len=0
IMAP	687	Response: * OK Gimap ready for requests from 77.185.141.220 q6mb34591238wrc
IMAP	94	Request: BKIA0 CAPABILITY
TCP	60	993 → 50818 [ACK] Seq=4929 Ack=638 Win=65535 Len=0
IMAP	296	Response: BKIA0 OK Thats all she wrote! q6mb34591238wrc
TCP	54	50817 → 993 [FIN, ACK] Seq=806 Ack=5924 Win=63247 Len=0
TCP	60	993 → 50817 [ACK] Seq=5924 Ack=807 Win=65535 Len=0
IMAP	137	Request: BKIA1 LOGIN gustavhanswurst123@gmail.com "edopqvpodjboakuq"
TCP	60	993 → 50818 [ACK] Seq=5171 Ack=721 Win=65535 Len=0
TCP	60	993 → 50817 [FIN, ACK] Seq=5924 Ack=807 Win=65535 Len=0
TCP	54	50817 → 993 [ACK] Seq=807 Ack=5925 Win=63247 Len=0
IMAP	357	Response: BKIA1 OK gustavhanswurst123@gmail.com authenticated (Success)
IMAP	96	Request: BKIA2 SELECT inbox
TCP	60	993 → 50818 [ACK] Seq=5474 Ack=763 Win=65535 Len=0
IMAP	433	Response: BKIA2 OK [READ-WRITE] inbox selected. (Success)

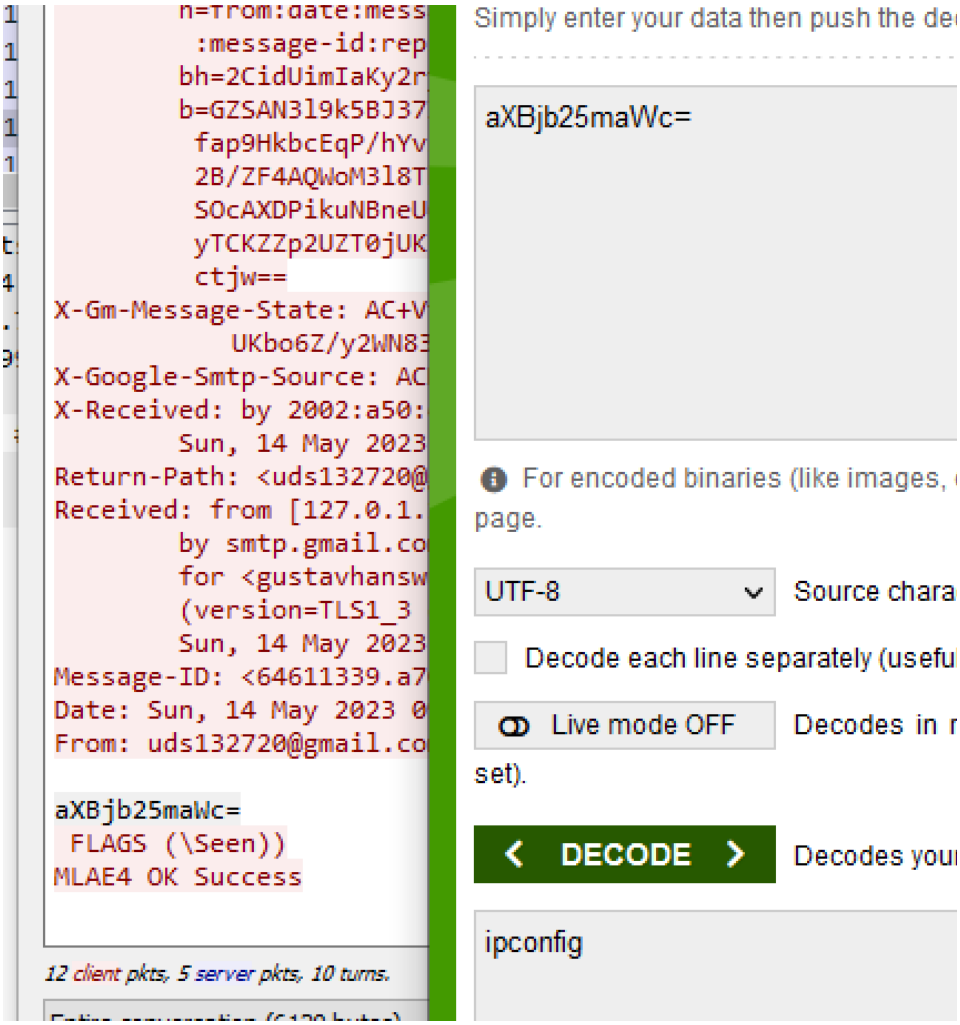
5. Filtering out IMAP-Traffic will leave you with a lot of encrypted content

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

imap

No.	Time	Source	Destination	Protocol	Length	Info
1703…	349.763019	173.194.76.108	10.0.2.15	IMAP	688	Response: * OK Gimap ready for requests from 77.185.141.220 r15mb61618862wmo
1703…	349.763416	10.0.2.15	173.194.76.108	IMAP	94	Request: FBFP0 CAPABILITY
1703…	349.787876	173.194.76.108	10.0.2.15	IMAP	297	Response: FBFP0 OK Thats all she wrote! r15mb61618862wmo
1703…	349.789140	10.0.2.15	173.194.76.108	IMAP	137	Request: FBFP1 LOGIN gustavhanswurst123@gmail.com "edopqvpodjboakuq"
1706…	350.873239	173.194.76.108	10.0.2.15	IMAP	357	Response: FBFP1 OK gustavhanswurst123@gmail.com authenticated (Success)
1706…	350.873570	10.0.2.15	173.194.76.108	IMAP	96	Request: FBFP2 SELECT inbox
1707…	351.043885	173.194.76.108	10.0.2.15	IMAP	433	Response: FBFP2 OK [READ-WRITE] inbox selected. (Success)
1707…	351.044236	10.0.2.15	173.194.76.108	IMAP	97	Request: FBFP3 SEARCH UNSEEN
1707…	351.212628	173.194.76.108	10.0.2.15	IMAP	123	Response: FBFP3 OK SEARCH completed (Success)
1713…	353.290013	173.194.76.108	10.0.2.15	IMAP	687	Response: * OK Gimap ready for requests from 77.185.141.220 h8mb35204470wmo
1713…	353.290345	10.0.2.15	173.194.76.108	IMAP	94	Request: MLAE0 CAPABILITY
1713…	353.314360	173.194.76.108	10.0.2.15	IMAP	296	Response: MLAE0 OK Thats all she wrote! h8mb35204470wmo
1713…	353.315072	10.0.2.15	173.194.76.108	IMAP	137	Request: MLAE1 LOGIN gustavhanswurst123@gmail.com "edopqvpodjboakuq"
1714…	354.366611	173.194.76.108	10.0.2.15	IMAP	357	Response: MLAE1 OK gustavhanswurst123@gmail.com authenticated (Success)
1714…	354.366998	10.0.2.15	173.194.76.108	IMAP	96	Request: MLAE2 SELECT inbox
1716…	354.535724	173.194.76.108	10.0.2.15	IMAP	433	Response: MLAE2 OK [READ-WRITE] inbox selected. (Success)
1716…	354.536212	10.0.2.15	173.194.76.108	IMAP	97	Request: MLAE3 SEARCH UNSEEN

6. At first some of the message seem to include base64-encoded data, in this case the command ipconfig. It seems the C2 is communicating over Email.



7. Going through the IMAP-Traffic we can not find anything looking like a flag so we need to check if the client send anything via mail to the C2 and inspect SMTP.
8. Filtering out SMTP will give you several base64-encoded data. One of them is the flag DS-CTF{d4mnwh0w0u1dh4v37h0u9h7h3yu53dm411}

Simply enter your data then push the decode button.

SERTLUNURntkNG1ud2gwdzB1MWRoNHYzN2gwdTloN2gzeXU1M2RtNDExfQ==

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

DECODE Decodes your data into the area below.

HDS-CTF{d4mnwh0w0u1dh4v37h0u9h7h3yu53dm411}

```
250 SMTPUTF8
AUTH PLAIN AGd1c3RhdmhhbnN3dXJzdDEyM0BnbWFpbC5jb20AZWRvcHF2cG9kam
235 2.7.0 Accepted
mail FROM:<gustavhanswurst123@gmail.com> size=60
250 2.1.0 OK w12-20020aa7da4c000000b0050bc5727507sm6159863eds.73
rcpt TO:<uds132720@gmail.com>
250 2.1.5 OK w12-20020aa7da4c000000b0050bc5727507sm6159863eds.73
data
354  Go ahead w12-20020aa7da4c000000b0050bc5727507sm6159863eds.73
SERTLUNURntkNG1ud2gwdzB1MWRoNHYzN2gwdTloN2gzeXU1M2RtNDExfQ==
.
250 2.0.0 OK  1684083549 w12-20020aa7da4c000000b0050bc5727507sm61
quit
221 2.0.0 closing connection w12-20020aa7da4c000000b0050bc5727507
```