Initiale Analyse

Gegeben ist ein Netzwerkmitschnitt in Form einer `.pcap` Datei. Wie die Challenge Beschreibung schon verrät, können wir die Datei mit Wireshark analysieren. Die Challenge Beschreibung deutet darauf hin, dass wir einen Flyer aus der Datei rekonstruieren und anschließend reparieren müssen.

Öffnen wir den Mitschnitt mit Wireshark sehen wir folgendes:

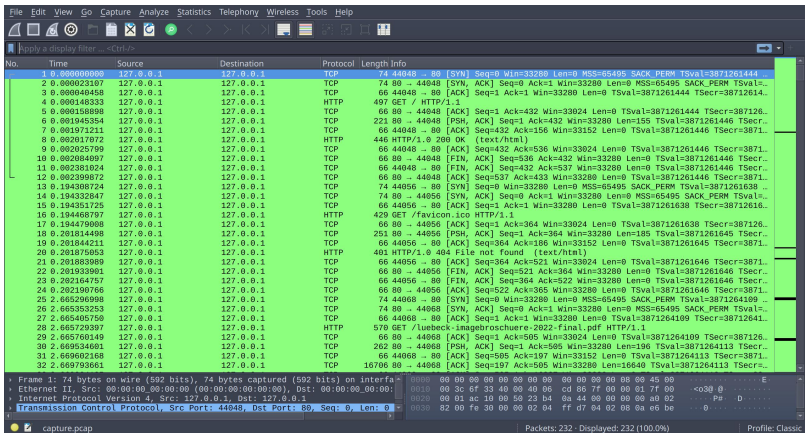

Figure 1: capture.pcap geöffnet mit Wireshark

Auf den ersten Blick können wir erkennen, dass viele Pakete über TCP oder HTTP versendet wurden. TCP und HTTP sind hierbei Netzwerkprotokolle.

Wenn wir uns den Infotext der einzelnen Pakete anschauen, sehen wir bei Paket Nr. 28, dass anscheinend das Herunterladen einer Broschüre als PDF mitgeschnitten wurde. Das scheint für unsere Aufgabe, den Flyer zu rekonstruieren, interessant zu sein.

Extrahieren der PDF

Mit einer Recherche im Internet können wir einen Weg finden, um die PDF zu extrahieren.

Die PDF können wir extrahieren, indem wir das Paket Nr. 28 mit Rechtsklick auswählen und `Follow -> TCP Stream` selektieren. Nun haben wir die Anfrage “Ich möchte PDF XYZ herunterladen” des Clients an den Server und die Antwort “Hier ist die PDF Datei” des Servers herausgefiltert.

Figure 2: Selektierter TCP Stream

Die roten, in hexadezimal kodierten Zeichen stellen hierbei die Anfrage des Clients dar und das Blaue repräsentiert die Antwort des Servers. Da uns nur die Antwort des Servers interessiert, wählen wir unten links die Antwort des Servers `127.0.0.1:80 -> 127.0.0.1:44068` aus, anstatt `Entire conversation`. Bevor wir die PDF mittels `Save as...` extrahieren, müssen wir sicherstellen, dass wir `Show data as Raw` auswählen. Man beachte allerdings, dass wir jetzt noch nicht ausschließlich die PDF extrahiert haben. Da der Server die PDF mittels HTTP übertragen hat, enthalten unsere extrahierten Daten noch den Header des HTTP Pakets. Wir können einfach mit einem Texteditor alles bis zum Header der PDF Datei `"%PDF"` löschen.

Rekonstruieren der PDF

Leider wird uns nur die erste Seite der PDF angezeigt. Warum das so ist, würde den Rahmen dieses Writeups sprengen.

Für Interessierte: Um zu verstehen, in wiefern die PDF Datei beschädigt ist, empfehlen wir, sich in das PDF Dateiformat einzulesen. Als kleiner Tipp: Das Tool `exiftool` gibt die Warnung: `Error reading xref table` aus.

Im Wettbewerb haben wir aber keine Zeit und versuchen unser Glück deshalb mit freizugänglichen Online Tools, die behaupten, dass sie PDFs reparieren können. Die Seite ilovepdf.com ist zum Beispiel in der Lage, die vollständige PDF zu rekonstruieren.

Finden der Flagge

Wir haben nun die vollständige Broschüre, aber wenn wir mit `STRG + F` nach dem Flaggenformat `SSH{` suchen, finden wir leider nichts. Scrollen wir einmal durch die PDF, sticht die letzte Seite besonders heraus, da sie außschließlich einen großen QR-Code enthält.

Scannen wir diesen QR-Code ein, erhalten wir eine kryptische Zeichenkette:

```
U1NIe24wd191X2tuMHdfdGgzX2I0c2ljc30=
```

Die Zeichenkette scheint verschlüsselt oder in einer anderen, für uns nicht lesbaren Kodierung zu sein. Das geschulte Auge kann erkennen, dass es sich um eine base64 kodierte Zeichenkette handelt, weil die Zeichenkette auf `=` endet, was für base64 kodierte Strings verwendet wird, um sie auf eine gewisse Länge aufzufüllen. Alternativ hätte auch eine Frage an ChatGPT mit hoher Wahrscheinlichkeit ausgereicht, um zu erkennen, dass es sich um einen base64 kodierten String handelt.

Der letzte Hint aus der Challenge weist uns auf die Webseite CyberChef hin, welche unter anderem in der Lage ist, Texte zwischen verschiedenen Kodierungen umzuwandeln. Auf dieser Webseite kann man den base64 kodierten String wieder dekodieren und erhält damit die Flagge. Die Webseite verfügt außerdem über eine **Magic** Funktion, welche den gegebenen String auf viele Kodierungen überprüft, um die richtige Kodierung zu finden.

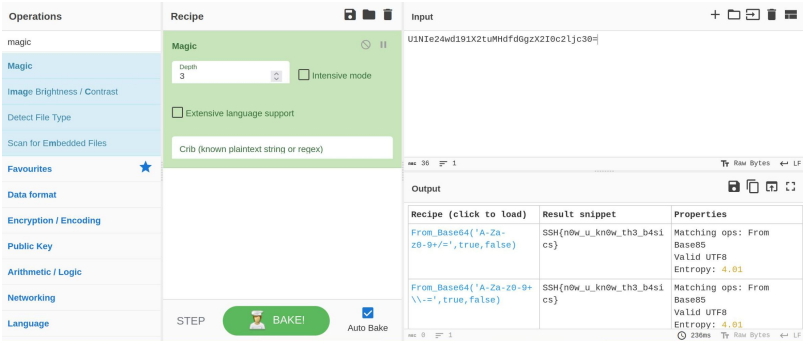

Figure 3: Verwendung der Magic Option von CyberChef