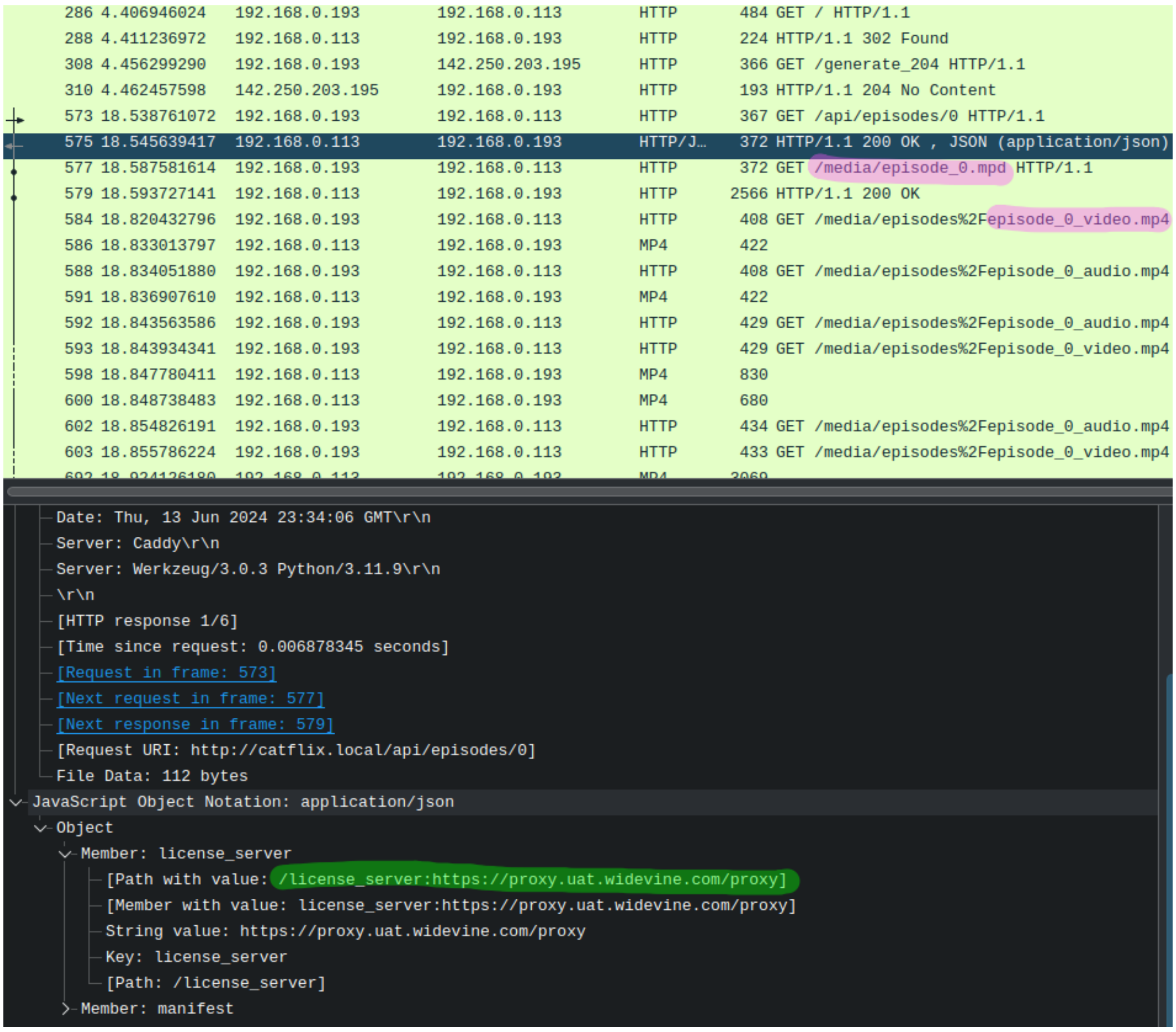
leaving soon (misc, 6 solves)

1. Analyzing the pcap file

We can see multiple mp4 chunks and contact to a license server at `http://proxy.uat.widevine.com`



```
286 4.406946024   192.168.0.193    192.168.0.113    HTTP     484 GET / HTTP/1.1
288 4.411236972   192.168.0.113    192.168.0.193    HTTP     224 HTTP/1.1 302 Found
308 4.456299290   192.168.0.193    142.250.203.195  HTTP     366 GET /generate_204 HTTP/1.1
310 4.462457598   142.250.203.195  192.168.0.193    HTTP     193 HTTP/1.1 204 No Content
573 18.538761072  192.168.0.193    192.168.0.113    HTTP     367 GET /api/episodes/0 HTTP/1.1
575 18.545639417  192.168.0.113    192.168.0.193    HTTP/J…  372 HTTP/1.1 200 OK , JSON (application/json)
577 18.587581614  192.168.0.193    192.168.0.113    HTTP     372 GET /media/episode_0.mpd HTTP/1.1
579 18.593727141  192.168.0.113    192.168.0.193    HTTP    2566 HTTP/1.1 200 OK
584 18.820432796  192.168.0.193    192.168.0.113    HTTP     408 GET /media/episodes%2Fepisode_0_video.mp4
586 18.833013797  192.168.0.113    192.168.0.193    MP4      422
588 18.834051880  192.168.0.193    192.168.0.113    HTTP     408 GET /media/episodes%2Fepisode_0_audio.mp4
591 18.836907610  192.168.0.113    192.168.0.193    MP4      422
592 18.843563586  192.168.0.193    192.168.0.113    HTTP     429 GET /media/episodes%2Fepisode_0_audio.mp4
593 18.843934341  192.168.0.193    192.168.0.113    HTTP     429 GET /media/episodes%2Fepisode_0_video.mp4
598 18.847780411  192.168.0.113    192.168.0.193    MP4      830
600 18.848738483  192.168.0.113    192.168.0.193    MP4      680
602 18.854826191  192.168.0.193    192.168.0.113    HTTP     434 GET /media/episodes%2Fepisode_0_audio.mp4
603 18.855786224  192.168.0.193    192.168.0.113    HTTP     433 GET /media/episodes%2Fepisode_0_video.mp4
```

```
  Date: Thu, 13 Jun 2024 23:34:06 GMT\r\n
  Server: Caddy\r\n
  Server: Werkzeug/3.0.3 Python/3.11.9\r\n
  \r\n
  [HTTP response 1/6]
  [Time since request: 0.006878345 seconds]
  [Request in frame: 573]
  [Next request in frame: 577]
  [Next response in frame: 579]
  [Request URI: http://catflix.local/api/episodes/0]
  File Data: 112 bytes
JavaScript Object Notation: application/json
  Object
    Member: license_server
      [Path with value: /license_server:https://proxy.uat.widevine.com/proxy]
      [Member with value: license_server:https://proxy.uat.widevine.com/proxy]
      String value: https://proxy.uat.widevine.com/proxy
      Key: license_server
      [Path: /license_server]
    Member: manifest
```

So this seems to be a Widevine DRM protected video. Firstly, let's extract all the mp4 chunks from the pcap file using tshark:

```
shark -r catflix.pcapng --export-objects "http,out_path"
```

2. Decrypting the video

After some googling, we found Shaka Player which is a JavaScript library for playing encrypted mpd content. After some issues getting this running on localhost, it’s time play the 34 episodes, which we assume are 34 flag characters.

3. Chunks!

As each episode is split into multiple chunks, we need a webserver which can serve these chunks while still making Shaka Player accept them. Just concatenating all the chunks into one file did not work.

Thus, we simply served the next chunk in the list after the previous one was requested. This was archieved using a simple Flask server.

4. Playing the video

This suprisingly worked! We were able to construct the flag by playing the episodes and writing down the characters.

5. Flag

justCTF{Y0u_w0uldnt_d0wnl04d_a_C4T}