CNS 2020 Project Proposal - Survey on the Download Prevention Schemes of Video Streaming Services

CHI-FENG TSAI, B07902123
KAI-EN LIN, B07902075
MING-HSUAN TSAI, B07902064
YU-HENG CHEN, B07902026

1 PROBLEM DESCRIPTION

Recently, university courses are forced to become online due to corona-virus. Most courses in National Taiwan University are moved to NTU COOL. Previously, some student tried to download the course videos on NTU COOL, but they are banned since the operator of NTU COOL claimed that these videos should only be watched on NTU COOL. Downloading the videos will violate the lecturer's digital rights.

Currently, to download the videos on NTU COOL, all you need is press F12 and you will see the URL to the video. Downloading the video is only within very few clicks. However, if the lecturer uploaded their videos to other video streaming platforms such as YouTube or U-Webinar, we will not be able to find the exact URL to the video. We wonder how these websites protects their videos from being easily downloaded.

In this project, we want to analyze how popular video-streaming and live-streaming platforms makes it harder for their users to download the videos. We mainly aim at YouTube, U-Meeting, U-Webinar, and Google Meets since these are the most popular platforms among National Taiwan University. We hope to compare their prevention method, find out their respective advantage and disadvantage, and propose a more secure mechanism to prevent unauthorized downloading.

We assume the attacker is familiar with HTML, JavaScript, and is able to track the network traffic between the client and the server. We do not consider the case where the user simply records a screencast, and the case where the user is able to observe how the video is stored in the memory on his own machine.

2 RELATED WORK

The Dynamic Adaptive Streaming over HTTP(DASH) [5] technique, which is commonly used by servers, can act as a basic protection that prevents a user from downloading the videos directly. Similar to other streaming protocols, a video is partitioned into segments and each of them is delivered independently. This way, the client does not receive the entire video as a file that can be saved directly. However, one can still investigate the implementation of the protocol, and develop a tool that captures the segments and assembles them to retrieve the original file.

To address the problem, some have attempted to combine media streaming protocols with existing DRM solutions. Hartung et al. proposed changes and extensions to DASH [3] in order to incorporate DRM. Many huge companies also have developed their own solution for their own products, such as Google's Widevine [2], Microsoft's PlayReady [4], and Apple's FairPlay [1]. However, these solutions often rely on a *Trusted Execution Environment* on the client side, which is established with hardware support, so it may not be cost-efficient to apply it to lower-value contents.

To fully protect a video from being downloaded, a hardware solution may be inevitable. Regardless of the encryption method used during the video transfer, the plain content must be loaded to memory in order to render the video on the screen. Once the video is loaded into memory, it is completely at the client's disposal [6]. Thus, our goal is to investigate how video content could be protected, or to what level the contents can be protected, without hardware support on several platforms, and how different scraping tools break through the protection. We hope that our analysis can server as a reference for future development.

3 PLAN

In general, we plan to analyze how online video-sharing platforms prevent users from downloading videos. Firstly, we would investigate the techniques used in some mainstream media services, e.g. Youtube, Umeeting, Google Meet.

The next step is to analyze the mechanism of existing download tools . A tool named youtube-dl [7] particularly attracts our interest as it claims to allow users to download videos from Youtube and other video platforms. We would like to look into the machanism behind this tool.

Finally, we summarise and compare the advantages and disadvantages of different download-preventing methods. We hope our result can help the development of media platforms in the future.

4 TIMELINE

5/13-5/26

- Study communication protocols and technologies that are used by streaming platforms for access control and duplicate prevention.

5/27-6/09

- Study the implementation of mainstream media services and find the difference between the methods they use.
- Analyze how youtube-dl [7] and other scraping tools download videos from media platforms.

6/10-6/23

- Conclude the previous results and compare different solutions.

6/24-7/07

- Adjust the report according to questions and recommendations from other groups.

5 DELIVERABLES

- Clearly illustrate the structure of the download protection approaches applied by Youtube, Umeeting and Google Meet.
- Clearly illustrate how youtube-dl [7] and other scraping tools circumvent the download protection approaches applied by the mainstream platforms.
- A measurement of the effectiveness of the download protection approaches in terms of the amount of effort required to hack them.

REFERENCES

[1] Apple. [n.d.]. *FairPlay.* https://developer.apple.com/streaming/fps/

[2] Google. [n.d.]. *Widevine.* https://www.widevine.com/

[3] Frank Hartung. 2011. *DRM protected dynamic adaptive HTTP streaming.* ACM SIGMM Conference on Multimedia Systems, MMSys.

[4] Microsoft. [n.d.]. *PlayReady.* https://www.microsoft.com/playready/

[5] Thomas Stockhammer. 2011. *Dynamic adaptive streaming over HTTP: Standards and design principles.* IEEE International Conference on Multimedia Systems.

[6] Ruoyu Wang, Yan Shoshitaishvili, Christopher Kruegel, and Giovanni Vigna. 2013. *Steal this movie - automatically bypassing DRM protection in streaming media services.* the 22nd USENIX conference on Security.

[7] ytdl org. [n.d.]. *youtube-dl.* ytdl-org. https://github.com/ytdl-org/youtube-dl