# 18-731 Project Proposal: Chromecast Traffic Analysis

Alison Kao
Carnegie Mellon University
ackao@andrew.cmu.edu

Philip Lee
Carnegie Mellon University
plee2@andrew.cmu.edu

Wellington Lee
Carnegie Mellon University
wklee@andrew.cmu.edu

## I. Project Description

The Chromecast is a streaming device developed by Google to allow users to view media on the connected TV ”cast” from devices such as a computer or mobile phone or tablet. This allows any user to cheaply convert a TV into a smart internet connected device. The second generation of the Chromecast was released on September 29, 2015. There are differences in protocols used by the two generations for communication, so our project will be targeted on this newer device. This will allow our research to be relevant to the current available product sold by Google.

The goal of this project is to discover possible vulnerabilities with the Chromecast in a setting where the attacker is not connected to the same network as the Chromecast. Our initial step will be to determine any basic privacy issues from sniffed data. The useful leaked data would contain information of device being used, media being streamed, account information, and other personal details. Further research, if time permits, will be into the effects of multiple Chromecasts in the same physical area with the same general goal of determining the data being cast from a particular Chromecast.

## II. Related Work

Work directly related to the Chromecast is very scarce. From our preliminary research, the only work done on Chromecasts examined the traffic caused by utilizing the Chromecast in various ways, such as idling, streaming, and mirroring content. The work showed that control packets to the Chromecast are in clear text, allow possible replay attacks or session hijacking attacks. Also, private information is leaked through these packets, such as YouTube account, video, and device information. Tekeoglu et al.[1] also mention methods to help determine the presence of a Chromecast behind a NAT device based on behavioral patterns. This will be useful for us to help determining the existence of the Chromecast while not connected to the network. It is important to note that these experiments and tests were done on the first generation Chromecast on the same network.

Chromecast is vulnerable to a deauth attack common to wireless devices. The attack works in by sending a deauth packet which will cause the Chromecast to revert to configuration mode. In this mode, the Chromecast broadcasts its own wireless network for setup, allowing malicious entities to hijack the device. Dan Petro of Bishop Fox (a global security consulting firm) demonstrated this vulnerability by creating the RickMote, which attempts to deauth any Chromecast device within range and stream the Rick Astley video ”Never Gonna Give You Up” [2]. Kaspersky Lab reconfirmed this issue recently in November of 2015.

The new Chromecast uses multicast Domain Name System (mDNS) for discovery purposes. There are a few published vulnerabilities with mDNS, but initial research shows very few that are relevant towards our goal. The previous generation Chromecast utilized DIAL, which allowed commands to be sent to return device information, such as the app or video currently active or local information such as available wifi and detailed device information. No such related capabilities have been published for the newest Chromecast.

## III. Outline of Research Contribution

We intend to expand on Tekeoglu et al. and discover new vulnerabilities in the Chromecast to promote a stronger sense of security during development of such streaming devices. Tekeoglu et al. assumed an attacker on the same network as the Chromecast. We want to take this further by changing the threat model. The presence of a vulnerability in a streaming device in the general vicinity of the attacker would prove useful in improving the security of next-generation devices.

We hope that our research will open the doors to a new class of vulnerabilities for streaming devices, not just specific to the Chromecast. With the presence of these vulnerabilities, we want to see future protocols developed and used for streaming devices finding a balance between ease-of-use and security.

## IV. Timeline of Implementation Milestones

1) Feb 15 - Project Proposal
2) Feb 22 - Obtain materials, test known exploits [1] [2]
3) Mar 7 - Design and test channel sniffing using SDR
4) Mar 14 - Project Milestone, meet with course staff
5) Mar 28 - Start project paper, continue vulnerability assessment
6) Apr 25 - Project paper, working demo of vulnerabilities (if found)

NDSS '16, 21-24 February 2016, San Diego, CA, USA
Copyright 2016 Internet Society, ISBN 1-891562-41-X
http://dx.doi.org/10.14722/ndss.2016.23xxx

## V. Evaluation Metrics

We will evaluate the success of our project using standard machine learning evaluation metrics because we plan to attempt to differentiate different types of web traffic using a Naive Bayes classifier. As such, we will train the model on a training dataset and measure both precision and recall of a test set. For stronger analysis, we will perform crossvalidation by running our classifier multiple times on different training and test sets.

We will perform separate evaluation for multiple granularities. For example, we may evaluate whether we can distinguish broad categories of content like streaming video via native apps vs. streaming video through casting the whole screen vs. casting the whole screen with non-video content. We may also evaluate whether we can make more fine-grained distinctions such as determining if video content is from Youtube or Netflix.

We will also perform separate analysis for each type of attack tried at each granularity. If time permits, we will attempt to identify traffic from one particular Chromecast when multiple devices are on the same network. This will be evaluated separately from the single Chromecast on a network scenario.

## References

[1] Ali Tekeoglu, Ali Tosun, ?A Closer Look into Privacy and Security of Chromecast Multimedia Cloud Communications?, IEEE INFOCOM?15, International Workshop on Multimedia Cloud Communication, Hong Kong, April 2015

[2] http://www.bishopfox.com/blog/2014/07/rickmote-controller-hacking-one-chromecast-time/