Anomaly Detection System for Camera Cyberattacks
Sai Sarath Karry, Pratik Satam and Salim Hariri

Motivation

- Today is the era of smart cars self driving cars. These cars are entirely controlled by the Central computer with the data from the various sensors such as cameras, proximity sensors, speedometers etc. One of the most Critical devices on a such systems is the camera. But the camera is susceptible to a wide range of attacks. Hence it is absolutely necessary we secure such devices.
- In this project we aim to develop an intrusion detection system to identify threats to the camera devices.

Objectives

- Develop models to predict the behavior of different applications using the camera device
- Develop operational scenarios to conduct what-if-else analysis
- Develop an Intrusion Detection System to identify Malicious Camera Applications.

Threat Model

- The main vulnerability that affects the system is the lack of control for applications accessing the camera device.
- In the system, any application can access the camera as long as it has the required Interface modules.
- This leads to a situation where the OS cannot regulate the use of camera device.
- This depicts a security vulnerability which leads to the misuse of the device

Consequences of a camera attack

LEAKING PRIVATE INFORMATION
- The camera in the car can be used to locate the position of it’s user by analysing the photo data it takes during travelling such a road signs. This leads to a loss of privacy and can have significant financial impact on the person involved.

LOSS OF LIFE
- Let us assume if an attacker seizes control of the camera and feeds false information to the central processor, It may result in a fatal consequence.
- This may also be exploited by the Anti-social elements to create mass hysteria.

In this section, we describe how the Event database of the camera application behaviour is generated.
- This unit consists of two scripts – the first one collects the data while the second script organises the events into the database.
- The Android Debugging Bridge (ADB) logcat script collects the log related to the camera application and passes them to a perl script which quantifies them and adds them to the database.
- The perl script first divides each log into different columns such as Event number, application, PID, Details.
- Then each different event is assigned as numerical id such as event1, event2 etc.
- The database records all the events related to the camera application from the logs.
- We use the available database to create a normal model of the camera application.
- This is done as a set of events that are normally executed by the camera application which is extracted from the database of events using machine learning algorithms.