CSC 600 Spring 2025 Final Project

Dr. Si Chen

This project involves group collaboration, with each group consisting of up to 2 students. Alternatively, you may work individually if you prefer. If you decide to work as a team, **please send an email to me before 04/28, 2025, including your group members' names and WCU IDs.** After 04/28, 2025, **you will NOT be able to claim team membership**. However, you can still work on this project individually.

Topic 1: Video Privacy for Public IP Camera

Background

Network or internet cameras have gained popularity for monitoring property and ensuring the safety of loved ones, providing real-time video and audio feeds that can be accessed remotely through an internet browser. In this project, we focus on analyzing the network throughput of public IP cameras, which is a critical yet often overlooked aspect of video surveillance. By investigating the fluctuations in bandwidth for a specific IP camera, we can glean valuable insights. It is also essential to comprehend network throughput in order to fully understand the mechanisms behind IP cameras. We demonstrate how real-time throughput can vary significantly across different scenarios and settings.

For this project, we utilize public IP cameras to study the methods for measuring and analyzing video bandwidth. To begin, select several public IP cameras from the provided list (http://www.insecam.org/en/). Alternatively, you may set up your own IP camera if feasible. Ensure that the chosen camera URL is functional and supports the H.264 CODEC.

Objectives and Targets

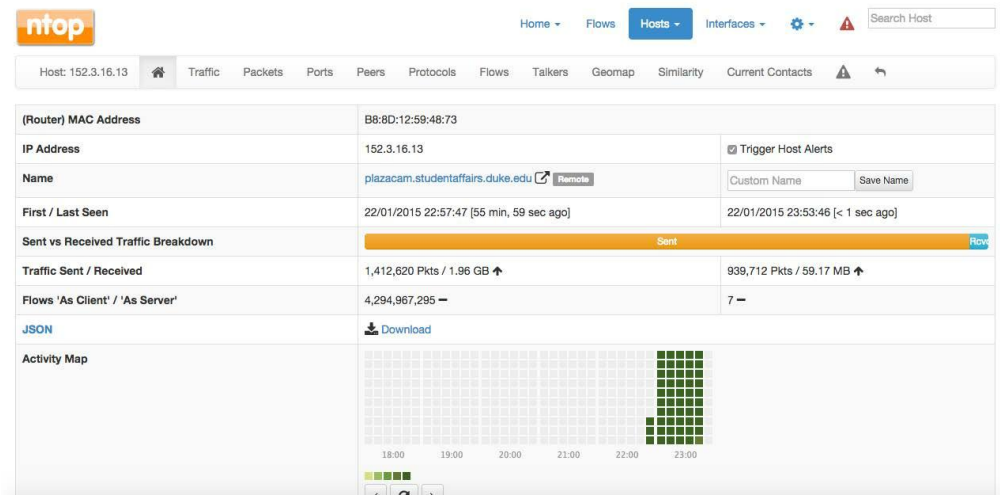
Please address the following questions and include the responses in your report:

Target 0

1. Identify the make and model of this IP camera.
2. Determine the location and IP address of this IP camera.

3. Ascertain the supported encoding CODECs for this camera. (Research the brand online)

Subsequently, a method for monitoring and storing the traffic data pertaining to the selected IP camera must be established. One practical solution is to use ntopng (http://www.ntop.org/products/ntop/), an open-source network traffic monitoring software. With compatibility for Unix, Linux, BSD, Mac OS X, and Windows, ntopng can be acquired from the official website. For further details, refer to the online user guide (https://www.ntop.org/support/documentation/documentation/).



After accessing the ntopng system through a browser, set up the monitoring interface and choose the host IP address you want to monitor. Upon completing the configuration, you can view the sent and received traffic via the web-based graphical user interface (as depicted in the figure). Keep in mind that other methods or software can be used to monitor video traffic as well.

Target 1

Develop a program that continuously monitors and stores the traffic data of the IP camera. Process the traffic data to calculate the IP camera's throughput consumption. Illustrate the results over time with a plotted figure.

Target 2

Create a program to calculate the average, maximum, and minimum throughput usage.

Next, design custom experiments to address the following questions. Provide a detailed write-up describing your methodology, including data capture timing and methods, recording of ground truth, etc.

Target 3

Does movement within a scene affect throughput? If so, what is the relationship? Please provide an explanation.

Target 4

Compare different lighting conditions (e.g., day mode and night mode) and determine which condition is more sensitive to movement within a scene. Explain the reasons behind your findings.

Target 5

Select an IP camera monitoring an indoor environment (e.g., a room). Develop a program that uses throughput data to detect the presence of someone in the scene. Describe your methods and compare your results with the ground truth. Explain the rationale behind your approach.

Topic 2: Malware Analysis: Stuxnet

This project topics entails the analysis of a notorious Trojan malware – Stuxnet. You are required to utilize the provided memory dump file (stuxnet.vmem) and analyze it using Volatility, a process similar to Lecture 12. Subsequently, justify your analysis by referring to the leaked source code available at https://github.com/micrictor/stuxnet.

Target 1

According to the Symantec report (https://nsarchive2.gwu.edu/NSAEBB/NSAEBB424/docs/Cyber-088.pdf), Stuxnet hooks *Ntdll.dll* to monitor requests for loading specially crafted file names. These uniquely crafted filenames are then mapped to a different location specified by Stuxnet.

The functions hooked for this purpose in *Ntdll.dll* include:

- ZwMapViewOfSection
- ZwCreateSection
- ZwOpenFile
- ZwCloseFile
- ZwQueryAttributesFile
- ZwQuerySection

Please analysis the memory dump file and reveal the real memory address for these functions.

Target 2

Stuxnet incorporates two modules: *maxnet.sys* and *mrxcls.sys*. The former module installs a file system registration change callback, which enables it to receive notifications when new file systems become available (allowing immediate spreading or hiding of files). The latter module installs an image load callback, which is utilized to inject code into processes when they attempt to load other DLLs.

Utilize Volatility and consult the command reference available at https://github.com/volatilityfoundation/volatility/wiki/Command-Reference. Determine an approach to identify malicious kernel drivers, kernel callbacks, and pinpoint the malicious devices within memory.

Target 3

Utilize Volatility to analyze the malware memory dump and report how Volatility can be employed to discover relevant artifacts of activity within memory. And refer to the leaked malware source code to justify such activity (e.g., pinpoint which function or line of code performs the Hooks). (Note: This section is crucial; your group must identify the functions or lines of code for each malware activity discovered in the memory dump).

Once done, your malware analysis report should encompass the following information:

1. **Introduction** – Offer a brief overview of the malware.
2. **Findings for each target**
3. **Summary** – Provide a summary of your work and recommend best practices for preventing infections and recovering from them.

Submission

- Each team must submit a detailed report (in PDF format) on D2L. The team leader should submit the report using their student account to D2L. On the first page of your group report, please indicate how your team allocated tasks among its members.
- Your submission should include:
    1. A comprehensive project report detailing your work, complete with screenshots and code snippets. Include explanations for any interesting or surprising observations. You are encouraged to explore beyond the project description's requirements. Bonus points may be awarded for extra efforts.

2. Your program's source files (compressed into a .zip file) in the programming language of your choice. Note: DO NOT submit binary results or raw data files.
3. Academic integrity is expected. If your work is based on others', provide clear attribution. Failure to do so may result in failing the course.