

## ECE 1776 – Project Proposal

### Group Members:

Rita Chiu – 980290250

Jacky Mok – 990872301

Vicky Tsang – 981000580

### Introduction and Motivation

“Conventional security wisdom suggests that code vulnerabilities occur on less commonly executed paths. We will try to verify that hypothesis in this project. We will instrument programs with documented vulnerabilities to gather path-frequency information. They will then try to determine a statistical correlation between the frequency of paths executed and location of the vulnerabilities.”

### Outline of Proposed Solution

- identify open source programs that have documented vulnerabilities
  - o There should be a mix of vulnerabilities (stack overflow, formatted string attack, resource leaks, etc...)
- Instrument program using EXE, inline C/C++ code or gprof
- Gather statistics on the frequency for path executions
- Compare the location of the vulnerabilities and hot paths
- Verify / argue against the thesis

### Related Work

- Provide a list of related work and an explanation of how the advantage they believe their proposal will have over existing solutions

#### Automatic Vulnerability Detection Using Static Source Code Analysis

<http://gcc.vulncheck.org/sotirov05automatic.pdf#search=%22code%20vulnerabilities%20occur%20on%20less%20commonly%20executed%20paths%22>

- classifies three common characteristics present in software vulnerabilities and develop a static source analysis that is able to identify execution paths with these three characteristics and report them as potential vulnerabilities
- our current proposal differs in that it does not restrict the type of vulnerability

#### GPROF instructions manual

<http://www.cs.utah.edu/dept/old/texinfo/as/gprof.html>

GPROF is a tool developed by the GNU community on gathering path statistics in a program. It allows user to learn where the program spent its time and which functions called with which other functions while it was executing. The original intention for the development of this tool is to allow programming to perform code optimization in the hot traces.