**Automated Reverse Engineering Project Proposal**

**Factors Influencing Gadget Discovery with ROPGadget**

**EEL4930**

**Shane Ferrell**

**October 18, 2024**

**Goal:**

The purpose of this project is to examine the factors influencing an attacker’s ability to discover return-oriented programming (henceforth referred to as ROP) gadgets in x86 executable code. This project investigates one of the main claims made in the 2007 paper by Shacham, which states that for any sufficiently large x86 binary, there will exist at least one chain of gadgets that enables arbitrary computation.

**Tools:**

* gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0

GCC (GNU Compiler Collection) is used to create the benchmark dataset of executable binaries in x86 architecture from various C++ programs.

* ROPGadget 7.5

ROPGadget is an open-source tool that automates the discovery of gadgets for ROP exploitation in executable binaries. ROPGadget will be used to identify gadgets in the benchmark dataset.

**Dataset:**

The dataset consists of 18 programs written in C++ that are categorized under 3 purposes: Arithmetic Operations, String Manipulation, and File I/O. For each purpose, the programs are categorized under 2 structures: Dense Loops and Sparse Function Calls. Each program has also been implemented with 3 code sizes: small, medium, and large. The source code for each program in a file called main.cpp.

Each main.cpp program has been compiled with 4 levels of optimization using gcc: -O0, -O1, -O2, and -O3 with a Bash script compile.sh, producing 72 executable binaries.

The 18 programs, 72 executable binaries that make up the dataset, and the script for compiling these executables can be found at the following repository: <https://github.com/shaneferrellwv/gadget-discovery>

**Description:**

**Challenges & Considerations:**