Austin Peterson (926006358)
Ethan Benkel (625001477)
CSCE451

CSCE 451 Binary Filters Proposal

Binary filters are used when extracting segments of code from a binary executable file. The use of these filters is often vital to the reconstruction of a given program and the process of understanding how a program operates. In addition, binary filters allow us to reuse code found in various executables, which can be especially helpful in the field of encryption, often found in malware and other malicious programs. Binary filters can be used to analyze and recreate the algorithms used to encrypt these packages and allow us to use the same algorithms to decrypt similar programs in the future and understand the methods used to protect such programs. These filters are also useful when it comes to malware detection, which can be beyond useful when protecting a high risk system or database.

Beyond malware detection and encryption, binary filters allow us to understand how the malware itself functions and how it's able to attack a given system. Malicious programs use a variety of methods to operate, including call-stack tampering, stolen-byte techniques, exception-based control transfers, and anti-relocation techniques. Understanding and recreating these techniques using these filters allow us to properly create defensive measures against such malicious methods and react more quickly when they’re used against us. To work with and analyze these malicious programs, as well as non-malicious programs, cybersecurity experts have a variety of tools at their disposal. Kali Linux, a distribution designed for penetration testing, reverse engineering, digital forensics and network security assessments, come prepackaged with useful programs and has become the most efficient linux image for these purposes.

There are tools available to aid in the process of binary extraction. Binwalk is a tool commonly used to extract embedded files from within images, allowing hidden executables to be detected and quarantined before any unwanted programs are run. BinPro is also useful for matching binary patterns to higher-level source code regardless of compiler convention, expediting the reversing process as a whole. Another program useful for analyzing malware functionality is Cuckoo, a system used for analyzing the actions of programs when executed inside an isolated environment.

Our project goal is to ultimately determine how we can use existing binary filtering and extractions tools to manipulate binaries in specific ways. This manipulation will most likely involve the isolation and changing of branch instructions. Such instructions are the cause of any deviation from a program’s default behavior, which is the primary reason why they are often targeted by malware with the purpose of manipulating a program, or system. We intend to study these binary filtering tools such as Binwalk, BinPro and the Dyninst Binary Instrumentation and Analysis Framework,

etc. so that we may first understand the complexities and goals of binary filtering. We then hope to either use these tools themselves, or implement a personally constructed simplified tool to exemplify the importance and overall purpose of binary filters within the industries of cybersecurity and reverse engineering.

The schedule for our project is highly tentative, but will consist of three phrases, each encompassing either 1 to 2 weeks. The first phase will be our compilation of research on binary filters. All of the articles below, as well as a few others will be studied thoroughly to develop a solid understanding of binary filtering and in general, binary instrumentation, or manipulation **(estimated 1 week completion time)**. The second phase will be to utilize our newly found knowledge of binary filtering to either efficiently operate an existing binary filtering tool, or implement a simplified binary filtering program of our own design. In doing so, we hope to effectively display the applications of binary filtering in malware defense and execution, as well as general program altering and manipulation. Regardless of whether we decide to use an existing tool, or develop our own, we will create and collect a plethora of examples to provide a well-rounded representation of the various uses of binary filtering **(estimated 2 week completion time)**. Our last phase will be the consolidation of all of the results of our binary filtering tools, as well as a thorough analysis of all of our findings to develop our final report. **(estimated 1 week completion time)**. Every phase will be a collaborative effort between both teammates, with each waiting to move to the next phase until the previous phase has been unanimously deemed completed.

Reading Articles:

- https://people.eecs.berkeley.edu/~dawnsong/papers/2010%20binary%20code%20extraction.pdf
- https://www.hindawi.com/journals/sp/2017/3273891/
- ftp://ftp.cs.wisc.edu/paradyn/papers/Roundy12Packers.pdf
- ftp://ftp.cs.wisc.edu/paradyn/papers/Bernat11AWAT.pdf
- https://arxiv.org/pdf/1711.00830.pdf