Summary of MCS Portfolio Reports

Kiyoko Mangrobang
CSE 539: Applied Cryptography
CSE 545: Software Security
Arizona State University
kmangrob@asu.edu

ABSTRACT

On the following page, I will briefly provide a description of the two reports for two courses I included in my portfolio. I completed the two courses during my Master of Computer Science degree. These courses include CSE 539 Applied Cryptography and CSE 545 Software Security. I found the required projects for these courses to be very fascinating and helped me further my educational growth.

1. CSE 539 Applied Cryptography

I chose this course to be the first report included in my portfolio. I was required to complete four projects for this course. The titles of these projects are as follows: Steganography and Cryptanalysis project; Hash project; Diffie-Hellman and Encryptions project; and RSA project.

The overall goal for Project 1 was to attain experience using steganography, cryptanalysis, and random number generators. The project included two parts; the first was the steganography portion, which included programmatically modifying a file to hide a message inside the bitmap image, and the second part was applying cryptanalysis, for which I broke the encryption by finding the key through a weakness within the random number generator.

Within Project 2, I had to apply cryptographic libraries to hash strings using the MD5 library. I gained experience in utilizing many of the available modern hash functions in C#'s hashing library. I learned how to convert bytes from a string and then applied salt to that string for further secure hashing through the application of an MD5 hashing function. This project also taught me how to find/identify collisions in a hash function using a birthday attack.

The overall goal for Project 3 was to attain experience utilizing existing libraries to implement encryption using a 265-bit key. I learned how to implement a part of the Diffie-Hellman algorithm to generate a key. I also learned how to work with extremely large numbers that were too large to store in a standard data type. For my program to accommodate these arbitrarily larger integers, I utilized C#'s BigInteger struct to support these numbers.

Lastly, the goal for Project 4 was to acquire first-hand experience in implementing the RSA algorithm. This project also tested my ability to work with extremely large prime numbers, given that all the provided prime numbers were greater than 200 bits. I learned how to implement the Extended Euclidean algorithm and solve for the value of the greatest common divisor. I also learned how to encrypt and decrypt using the RSA algorithm within this project.

All four of these projects taught me about different cryptographic algorithms and how to implement them in a real-world scenario. I learned how to write code in the language of C# syntactically correct within this course.

2. CSE 545 Software Security

I chose this course to be my second report included within my portfolio. This course required me to complete the following five projects: Linux Lifter, Hacking Network Highways, Unwinding Binaries (Reversing), Hijacking Binary Power (Exploiting), and Wrecking the Web World. However, I was only required to include projects 2-5 within my report, leaving out the Linux Lifter project. I was required to use pwn.college to complete all five projects for this course. Each project included various amount of challenges to complete said project.

Within Project 2, I learned how local networks work and what techniques hackers use to attack these networks. I connected to IP addresses on specified ports and listened to traffic sent across them. The challenges in this project taught me how to utilize Scapy for sending packets across the network, sniffing packets and their data using tcpdump, and hijacking traffic by configuring my network interface to something else. For a few of these challenges, I utilized Wireshark to read a tcpdump packet file sent over a network and looked at the contents of these packets.

Project 3 taught me how to analyze the assembly code of a program and use different software such as angr-management, IDA Freeware (x64), and Ghidra to decompile the assembly into readable c code. Within the challenges of this project, I learned how to decipher binary numbers and convert them into usable strings. I used the different applications for decompiling source code so that I could understand what the program was doing so that I could exploit the code's vulnerability.

In Project 4, I learned about the many vulnerabilities a hacker could exploit when running a program. Many of these challenges taught me about the importance of user input scrubbing. I learned how to inject commands into user input when a program doesn't sanitize it correctly. I also learned how segmentation faults are triggered and how to debug your code after receiving one. For these challenges, I continued to utilize the applications for decompiling a challenge's source code to read/understand the flow of the source code and how it was executing the commands within so that I could exploit it.

Lastly, in Project 5, I learned about the vulnerabilities of web applications and how to exploit them. These challenges entailed learning to utilize MySQL and encoded special characters within a URL to hack these web applications. I also learned how to write an HTML script to create a Flask server so that I could leak secret information. I continued my knowledge of the Python language by writing Python scripts. Within this project, I also learned to create HTML scripts from scratch.

The projects for this course included several difficult challenges that required me to use the knowledge taught in the lecture videos and apply it in a hacking setting. I learned about many hacking techniques and methods for safely writing programs that could defend against potential hackers. Some knowledge I gained throughout these projects was how a computer connects and communicates over a local network. Additionally, how to translate the least significant and most significant bits into a binary number and then convert that binary into an ASCII string; how to create command injections and craft a payload to overload the buffer to take control of a program when user input is not scrubbed correctly; how to write an HTML script from scratch to create a Flask server and steal secret data