Preparation and Delivery of Material for Special Topics in Information Assurance Courses at an Undergraduate and Graduate level

Steven D. Bennett
Master in Computer Science
Jeffrey Duffany, Ph.D.
Electrical & Computer
Engineering and Computer
Science Department
Polytechnic University
of Puerto Rico

Abstract *- I produced coursework for the undergraduate course Penetration Testing and the graduate course Reverse Engineering. The Spring 2017 trimester was the first time that either of the courses had been given. This presented a challenge in the need to dominate the material beforehand, to divide topics in a way that would facilitate the introduction of more complex material, and finally to present the information in an organized manner that will allow the students to digest and comprehend it. As both classes progressed, there was a continual need for change and adaption to the students' progression of understanding. Some of the course material needed extending while others were quickly dominated. What was most noticeable and the best approach, was providing the students with hands-on exercises to be completed in class, and ensuring that they had all the tools necessary to accomplish it. Active learning and engaging exercises were the keys to success. For the Penetration Testing course, the National Cyber League competition was an invaluable resource.*

Introduction

Cybersecurity is a growing field with a high demand for employment. Though there are many with technical backgrounds and degrees in computer science or computer engineering, the major problem with these individuals is their lack of experience related to security [1] and its real world application. My project has been centered on taking part in the design and implementation of two courses in an initiative for advancing cybersecurity education and producing qualified students.

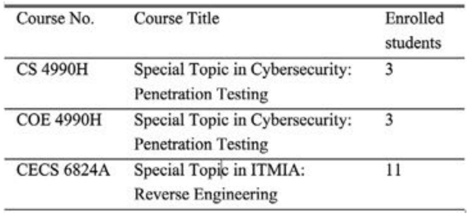
There are two courses that I have worked on for this project: Penetration Testing and Reverse Engineering. Both courses require that the students acquire practical knowledge and understand the theory behind the applied practice. That balance between theory and practice needs to not only be even, but also intertwined. The approach that will be seen throughout the project is too incorporate as much active learning as possible, and to abstain from passive learning, as it doesn't yield a high percentage of remembrance, and lacks the practice that a qualified individual would need in a career requiring technical experience.

A teaching method that has been adapted is the use of capture the flag exercises. All aspects that I have worked on uses these types of exercises. This includes the delivery of theory, discussing concepts, class practice and homework.

Course Background

The courses were given at the Polytechnic University of Puerto Rico during the Spring 2017 trimester. The course names with their respective number of enrolled students can be seen in Table 1 below. Also note that Special Topics require that each department have their own section. That's why the Penetration Testing course was divided between computer engineering (COE) and Computer Science (CS), instead of lumping them together as CECS, but they both ran concurrently.

Next I'm going to provide a brief description on both of the courses and what is the National Cyberleague. There will tend to be a larger emphasis on the Penetration Course, as I designed most of the course program, including the syllabus, and gave half the course under the supervision of my mentor and the courses designated professor, Jeffrey Duffany.

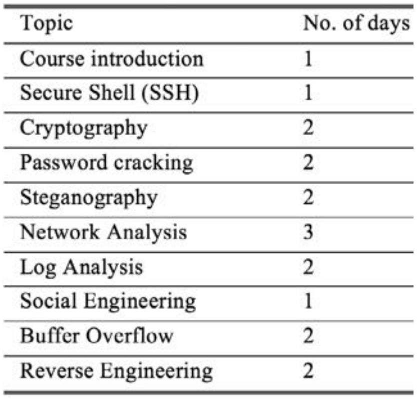


Course No.	Course Title	Enrolled students
CS 4990H	Special Topic in Cybersecurity: Penetration Testing	3
COE 4990H	Special Topic in Cybersecurity: Penetration Testing	3
CECS 6824A	Special Topic in ITMIA: Reverse Engineering	11

Table 1 - Course Information

Penetration Testing

Penetration testing is a process for finding exploitations, and also a subset of ethical hacking. Throughout the course, the students will be introduced to the different steps of penetration testing, and how it ties in with the exercises they are completing. Table 2 is a list of all the topics discussed and the number of days that were given to each. There are 24 days in a trimester, but only 18 were used for teaching new material, the additional 6 days are for exams, reviewing challenges from the competition that were complex, or to make space in case of unforeseen events that may delay or cancel a day of class.



Topic	No. of days
Course introduction	1
Secure Shell (SSH)	1
Cryptography	2
Password cracking	2
Steganography	2
Network Analysis	3
Log Analysis	2
Social Engineering	1
Buffer Overflow	2
Reverse Engineering	2

Table 2 - Penetration Testing Course Topics

The penetration Testing course, or pen testing for short, is a course I proposed to the Polytechnic University of Puerto Rico as a special Topic. The objective is to incorporate the National Cyber League into a classroom. Students are given the opportunity to practice pen testing techniques in class, and are then submitted into competitions that run parallel to the course. These competitions will bring critical thinking and teamwork skills.

Reverse Engineering

Reverse Engineering in terms of software is the analysis of undocumented machine language code, to learn more about how it functions [2]. The techniques used can be for legitimate reasons, such as analyzing malware or adding interoperability. It is also vastly used to circumvent copyright protection mechanisms and for finding vulnerabilities.

The way this course is to be taught is through the exposure of different tools and techniques throughout the trimester. Providing the students with a breadth of knowledge that will increase their overall understand of reversing. I worked on partial of the material, including the following topics:

- Assembly recap
- PE analysis
- Debugging
- Disassembly

National Cyber League

The National Cyber League (NCL) [3] is an organization that offers a biyearly cybersecurity competition aimed at college students of any skill level. Prior to the competition starting there are some virtual environments provided to every competitor to practice cybersecurity exercises aimed at teaching necessary skills for the Security + and EC-Council Certified Ethical Hacker certificates.

Each competition period is known as a season, and contain three competitions. The Fall season is the first one in the school year, and takes place from November to December. The Spring season is second, and takes place in April. Due to how the trimesters at the PUPR are broken up, the Penetration Testing course will only be able to take advantage of the NCL resources during the April season. At the end of every competition, the teachers are sent a report with the results of how each student performed, including what their accuracy was in answering challenges.

Teaching Metholodogy

Creating the course material is half the battle, the other half is presenting it in a manner that the students can most effectively learn it. Whether it's the Penetration Testing course or the Reverse Engineering course, my methods for lecturing and demonstration were identical.

Any lectures I gave, had to be through the use of demonstrations. It is imperative to me that the students are provided resources and instructions so that they may follow along, practice in their spare time and duplicate any results. Both courses are special topics and advanced, and as such there is the understanding that the students have the pre-requisites necessary, minimizing the need for detailing basic theory.

All exercises given were discussed in a conversational manner. Students were free to state their questions and doubts, and help was provided in a collaborative manner between the students and the teacher. Particularly in the Penetration Testing course, where teamwork was encouraged, so were leadership skills between the students to help others and work together to accomplish challenges.

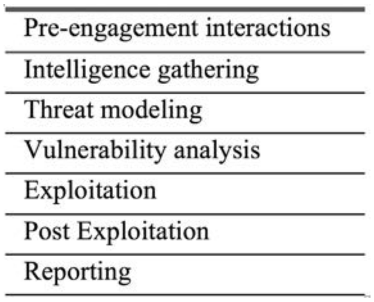
Penetration Testing Course Material

Due to the course being new and the extensive number of topics to cover, I taught only half the course, and then received help from a colleague on the other half. On the half I did not directly work on, I did provide references, constructive criticism, assisted in the supervision of its course material, in some instances I co-taught and I ultimately provided feedback and documented the material. For this section though, I'm only going to focus directly on the material that I personally worked on.

Course Introduction

The first day of class is an introduction to the course. This is the day that most heavily concentrated on what Penetration Testing is. I described and reviewed the different steps that compose it, as seen in Table 3. Also because the PUPR has a course in ethical hacking, the contrast between the two subjects and courses are explained. Lastly the National Cyber League is discussed and how everyone would be evaluated.

At the end of the day the students are advised on three different methods for installing Kali Linux, the preferred operating system to complete this course and the competitions.



Pre-engagement interactions
Intelligence gathering
Threat modeling
Vulnerability analysis
Exploitation
Post Exploitation
Reporting

Table 3 - Penetration Testing Steps

Secure Shell

The history of Telnet is explained along with its different applicable uses. Afterwards its securities flaws are discussed and how it led to the creation of its successor the secure shell. Example uses of Telnet are done in class, and references to accessible websites are passed out.

The secure shell protocol is detailed to the students, and a heavy concentration is given to its security aspects. For the remainder of this class, the students practice using secure shell on their first capture the flag exercises. The first few I explain the rest are left up to the student.

Some students at this point may not have Kali Linux yet, this is an opportunity to work with those students so that they may install it. Alternatives such as Putty and Cygwin for Windows were not left as an option, as Kali has many tools that are not Windows compatible and will be needed in future courses.

Another caveat is that permission to open port 22 for course will be required. Upon not having the port 22 available, the students will not be able to leave the universities network to access the capture the flag exercises. The option available would have been to practice remote connection between students, but it didn't come down to that.

Hashes

The concept of hashing is thoroughly explained, including its relationship with encryption. The process of creating digital signatures by adding hashing to encryptions schemes was discussed, the importance of integrity and the different ways that it can be found employed today with the use of passwords.

Emphasis is given on the different hashing algorithms, the security flaws found in certain algorithms still used today, and the different methods of attacking them. The attacks reviewed include brute force, dictionary, precomputed look up tables and rainbow tables. The first of two classes on hashes ends with a discussion on rainbow tables [4] and how to effectively use salt to thwart these attacks.

For the second class, tools dedicated to password cracking were introduced. These tools include John the Ripper, Hashcat and Ophcrack. Students are provided with a series of folders, each with a list of hashes and instruction on how to create a wordlist in John the Ripper, to crack the hash. Students were free to use any tools and any resources, but encouraged to learn well one specific tool that they can quickly refer to during a competition.

Steganography

For this topic I had an open conversation with the class about how they could hide something within some innocuous medium. Once they understood the concept of steganography, how it provides security through obscurity and its differences with encryption, a colleague then discussed a couple of steganography methods used today in computers. I reviewed the Least Significant Bit method with the students, explaining why it can be difficult to detect and what the most effective ways are. For the remainder of the class, the students learned command line steganography tools, and worked on exercises where they had to find files or images that were hiding within other files or images.

Social Engineering

The social engineering aspect was reviewed to show the students how it can be used during the intelligence gathering phase of penetration testing. Phishing and vishing techniques were discussed in depth. A look into how the pre-engagement interactions step was also discussed, to see what the limitations were of a pen tester, as ultimately some of his actions may be illegal. A penetration tester will need an agreement before performing any services on or off site, so that later they will not be prosecuted, and this includes falsification of documents, recording conversation in a two party consent State, impersonation and many overlooked skills that can go as far as dumpster diving and gaining unauthorized entry to a premises through piggybacking or even lock picking.

After all the above was reviewed, no practical exercises were given out. The Social Engineering Toolkit was introduced as multi-tool for different social engineering phishing and vishing attacks. From the beginning of class there was a set of lock picks and locks going around for the students to casually get the feel of what's like. Lastly students were introduced to the Social Engineering Capture the Flag (SECTF) [5] event held yearly in Defcon, and the results of the previous year's challenge was given out as an example of the different type of information that can be collected through open source intelligence gathering and social engineering attacks.

Reverse Engineering

Capture the Flag events incorporate a multitude of different cybersecurity skills, including those needed for reverse engineering. With a limitation of two days on such a complex topic, the topics chosen were very focused. The first day included a list of most commonly used instructions, the different types of jump instructions, and a look at the registers.

For the second class, a debugger called Immunity Debugger was presented and the interface was thoroughly explained. The students took advantage of most of the class to work their way around an executable and find out how they can bypass and/or make changes to the assembly code. The exercise simulated the requirement of a password, and taught them how to jump over assembly code that one doesn't want to execute, or how to simply change the assembly to execute example what one wants.

Homework and Tests

I wanted the homework to be related to penetration testing, capture the flag and being current on security breaches. With those three requirements in mind, two assignments were created. For the first assignment the students had to periodically write a report on recent security breaches and explain how the exploits took place.

The second homework was to make a write up of capture the flag exercises completed. A write up consisted of showing the steps that one took to complete the challenged. We can quickly see which homework required research of security breaches and which one incorporated capture the flag events. But both assignments targeted documentation, because it's important that a penetration tester be able to communicate his results, and that takes practice. It is said that if a pen tester who cannot communicate how he exploited a system, an organization cannot effectively defend against it.

The exams for the course came in the form of multiple choice questions and very basic capture the flag exercises that could be solved with pen and paper. The objective of the exam was that it would have been easily completed by a student who attended every class and participated in the competitions without cheating.

Reverse Engineering Course Material

This course was given by the professor Jeffrey Duffany, my involvement was to assist in the creation of material for the students. The professor was in control of the course, and my job was to ensure there was material for its objectives to be met. The following are the different topics I covered and their respective exercises.

Assembly Recap

Reverse engineering requires knowledge of low level languages such as assembly. For the opening of the course, I created a workbook that goes over the basics of x86 assembly language. The entire workbook was designed to be answered using Intel® 64 and IA-32 architectures software developer manuals [6]. Because the central processing unit or the operating system don't speak in base 10, the first part covered base conversion, and two's compliment. The second part included the 8 general registers, the instruction pointer register and the FLAGS register, and their different sizes and what type of data they store. The third part covered some of the most common instructions used in assembly. The fourth focused on when flag values changed. The fifth section covered the different types of jump, and the final part was made up of exercises that tested everything learning together.

PE Analysis

Portable executables can tell a lot about themselves without having to use any disassembly or debugging tools. For this class exercise the students are provided with an executable and using a hex dump the student analyzes the headers to find information such as how to tell if it's a PE or an ELF file, and if it's an x86 or x64 executable. Other data that can be extracted includes the names of the dynamic linked libraries, strings and the description of the executable. For this exercise the student can also use PE viewing tools that gathers all the previously mentioned information and sorts it in a more organized manner.

The second part of this exercise is that the students are given the same executable, but this one has been obfuscated. The goal is to compare both of the hex dumps for signs of compression, or missing data that was previously visible.

Debugging

Debugging is a form of dynamic analysis where the executable is ran and its behavior is observed. Because you're running code, there's always the chance of triggering malware, so it is recommended to do any debugging in a controlled environment, like a virtual machine. For this exercise the students are given an executable that instantly tell you that your evaluation period is over, and will then ask for key. To complete it, you would have to bypass the license requirement by controlling whether or not a jump would occur and then patching the executable. The purpose of this exercise is to introduce Immunity debugger [7] to the students and get them used to the interface.

There is a second debugging exercise using the same executable. This time the students would have to learn about the windows application programming interface to understand the code, and without altering any of the assembly code, find out which is the license required.

Disassembler

Disassemblers are used for static analysis. During static analysis no code is executed, instead it's simply translated from machine language to assembly language to be analyzed. This exercise can be completed with the free version or the demo version of IDA Pro. The students will be introduced to the interface of IDA, have the opportunity to explore the environment, and practice. Disassembling x86 executables. Their objective is learn to analyze and trace through an executable, without the help of a debugger.

The exercise is known as a KeygenMe, where there's an executable that takes as input a serial number, but it's not a static serial number and it can change based on the user. The students will need to read the assembly code and use the features of IDA to assist in finding out how the serial numbers are generated. Afterwards they will need to write a key generator in a high-level language. The key generator will ask the user for their name and will output the corresponding serial key that needs to function in the KeygenMe exercise.

Conclusion

All the course material is educational and reusable for future classes. There were several difficulties encountered, the first was that the courses were never given before. Because it was the first time, there was a need for dominating the material and being able to provide backup material at the last moment. The second was gaging the students experience to not underwork or overwork them, while ensuring that the material was sufficient to cover that day's class.

All exercises and references used in class have been documented and provided to my mentor. The penetration testing course now has sufficient material for it to be given in the future. The reverse engineering course now has supplemental material in the event of it also being given again.

References

[1] J. Stark. (2016). *There simply are not enough cyber-security specialists* [Online].
Available: https://www.complianceweek.com/blogs/john-reed-start/ there-simply-are-not-enough-cyber-security-specialists.

[2] E. Eilam, Reverse: *Secrets of Reverse Engineering*, 1st ed. Indianapolis: Wiley Publishing, Inc, 2005.

[3] J. Cote. (2016). *Online Cyber Security Students Hone Skills at Cyber Competitions* [Online].
Available: http://www.snhu.edu/about-us/news-and-events/2016/10/ online-cyber-security-students-hone-skills-at-cyber-competition.

[4] K. Kuliukas. (2006). *How Rainbow Tables Work* [Online]. Available: http//kestas.kuliukas.com/Rainbow/Tables/.

[5] *The DEF CON 24 Social Enginering Capture the Flag Report*, 1st ed. Brooklyn, 2016.

[6] Software.intel.com. (2017). *Intel® 64 and IA-32 Architectures Software Developer Manuals | Intel® Software* [Online].
Available: https://software.intel.com/ en-us/articles/intel-sdm.

[7] R. Nardella and R. Carbone. (2016). *Basic Reverse Engineering with Immunity Debugger* [Online].
Available: https://www.sans.org/reading-room/whitepapers/malicious/ basic-reverse-engineering-immunity-debugger-36982.