Capstone – Introducing Students to Research through Application Development in Teams

Katia Mayfield, Matthew Perry, Christopher Pounders, Lucas Pruitt, and Ory Wigington
Department of Mathematical, Computing, and Natural Sciences
Athens State University, Athens, AL, USA

Abstract—Computer science, as many other fields, is currently requiring students to be better prepared for their entrance in the workforce or for pursuing advanced degrees by getting involved in advanced research and projects. Service learning, where the students apply their knowledge to solve a real life problem instead of a fictitious scenario, has become a usual method of giving the student the required experience. Beyond the application of their individual knowledge is also the requirement of getting acquainted with teamwork. This study describes the experience of a team of students of different backgrounds working to develop a tool applicable to ongoing research in the cybersecurity area. This project involved the expected software development field stages such as analysis, design, coding and testing, with an extra research component, requiring the student to plan and schedule the necessary readings to be able to understand the problem. This study describes the research background as well as the students planning and execution of the project. A brief description of the resulting tool, which is now part of the full research project, is also shown.

Keywords: service learning, cybersecurity, petri nets, CAPEC, capstone project

1. Introduction

In today’s society it can be difficult to find a computer science undergraduate program that teaches the same way they did twenty years ago. Before the boom of technology and every piece of information that is now available at one’s finger tips, courses would be taught with assignments and projects given from a textbook, and twenty years ago professors would expect that students ended up doing their own work.

Now, with websites like chegg.com and different forums available where students can post any question (including those associated to homework assignments) and have strangers provide them with solutions and vague explanations that do not help the student learn, professors have had to become more creative than what was required once upon a time. Even when coming up with their own homework problems and exams, the cycle of making changes and producing new material has to happen almost on a semester basis as the information will get circled around between students. Most of the time, such material will get posted online, forever, because a student chose to ask a question to a complete stranger instead of asking their professor for help.

Instead of just re-creating problems for students to solve, many professors have started to implement service learning into their classroom. This is what the Computer Science department at Athens State University has done in their Senior Capstone class. Instead of having a made up scenario where students are required to go through a standardized system analysis, design, and implementation, students are put into teams of three to five and assigned to work with a community client. The professors will partner up with community representatives, usually non-profit organizations, which have not always been the case, and discuss a service that can be provided to them by the students. The majority of the projects in the last five years in the authors’ institution have been of a web application system, with a few others completed in the areas of game development, security analysis, and research.

Throughout their time at Athens State University, students take several classes with four different concentrations available: computer science, computer networking, information technology, or information security. When students reach their capstone class they may be grouped with students from other concentrations and they must work together to achieve the task that they have been given. This is a good opportunity for students, especially because many of them will not have held an internship during the time of their studies, but they will be able to state that they have work experience with a client in the development of some software. In this study, the focus will be on the research component of the Capstone class, combining development into the mix. In the next section, the research portion of this study will be introduced, followed by the requirements of the capstone project that was assigned. Once the details have been mentioned, the work completed by the team will be discussed, ending with a summary and future works.

2. Background

Cybersecurity, cyberattacks, cyberwarfare, cybercriminals, cyber-something, these are the areas of focus from many different organizations. Software and hardware development is no longer about only developing or creating what has been provided from the specifications, but it is also making sure that all of it is secure. However, it is hard to come up with all the
possible vulnerabilities because not all test cases can be developed and tested, especially if one may not know what exactly to test for. Due to this, a fairly new knowledge area has been the focus of many studies, focused on the process that an attacker takes to be successful. If developers are able to understand how an attacker behaves then they should be able to safeguard against it. MITRE Corporation thought of this and wrote a report called Common Attack Pattern Enumeration and Classification (CAPEC) [8]. This document lists all known possible attacks and provides as much detailed information as is known, from the knowledge that the attacker must have to begin an attack, to the goals that the attacker will attain when successful. The report has even suggestions of areas of mitigation. The CAPEC report allows developers and others to start the process of becoming familiar with the procedures taken by an attacker.

In the area of cyberattacks, one of the best ways to study an attack is through the development of a model. In this study the basis for creating a cyberattack model was through the use of Petri Nets with Players, Strategies and Cost (PNPSC) [6, 7]. A PNPSC is an extension to the well-known Petri nets, consisting of a bipartite graph made up of nodes referred to as places and transitions. A place within the PNPSC is a true/false state while a transition represents an action which the attacker can take in the process of their attack. A single PNPSC represents one form of attack. There is a token that is found in the beginning places of the PNPSC and as the attacker takes action the token travels through the PNPSC until the attacker has either accomplished his attack goal, failed to accomplish his attack goal, or has been detected and blocked by a defender. Figure 1 displays the basic concept of a PNPSC.

![Figure 1 A simple Petri net, basis for a PNPSC - Basic components][9]

The formalism of PNPSC, described in “Petri Nets with Players, Strategies, and Cost: A Formalism for Modeling Cyberattacks,” along with the CAPEC report became the basis for the Capstone project that allowed students to develop a web application that would introduce them to newly conducted research from a group of authors collaborating from Athens State University, the University of Alabama in Huntsville, and Lipscomb University [6, 8].

3. Research Background

The capstone project assigned to the students was the development of an application. However the application is also associated to an ongoing research project. This research project encompasses the realm of modeling cyberattacks, which is split into four phases, all of which are ongoing and have had several publications associated to them [9]. Before any of the research phases could evolve, the research group first came up with a formalism for PNPSC [6, 7]. Petri Nets with Players and Strategies has been briefly introduced by other researchers not including the concept of cost that could be associated to resources [13]. With the formalism of the model that would be used by the research group specified, the four phases of the project began to unroll:

Phase I – Uses CAPEC reports as a basis to design fault trees that represent the attack. Once the fault trees are created, determines the process for creating a PNPSC that not only corresponds to the fault tree but also corresponds to the CAPEC report. A second part to this phase is to compare existing tools to determine the most ideal way of displaying the PNPSC and being able to not only model it but also simulate the attack represented by it [10, 11].

Phase II - This phase builds off of the generated PNPSCs resulting from Phase I of this research. This phase has three different parts to it. The first is to have a cyberattack model repository created, which became the focus of the capstone project. The repository is needed to hold the information associated to the CAPEC reports but also the full model design, details, and metadata of the constructed PNPSCs. The second part of this phase is to determine a way for the PNPSCs to be designed as components, resulting in the basis for the third part where the components are them used to create a composite and more complex cyberattack model [5].

Phase III - This phase consists of validating the cyberattack models that are designed. At this moment in time the formalism of PNPSC has been validated through a face validation process where comparisons of the PNPSC were done with the details found in the
CAPEC report. Since the CAPEC report is widely used as a reference for the different cyberattacks, it has been assumed that MITRE Corp. has gone through and verified and validated the information that they have included in the report. There is ongoing work for composing more complex cyberattack models based on the single attack models, therefore the composition of the more complex attacks will still need to be validated. There is also ongoing research into using different methods to conduct the validation of the models and to not only rely on face validation. [2, 3]

Phase IV - The last phase of this research project is to take the cyberattack models that are created, simulate a cyberattack and implement machine learning to recognize attack and defense strategies. Being able to have this machine learning component allows for a vulnerability assessment to be developed [1].

Figure 2 shows a graphical representation of the four phases of the research being conducted. In the next section the details of the capstone project to create the repository associated to Phase II will be discussed.
4. The Capstone Project

The capstone project was originally introduced to students as a two part project, the implementation of a database and the design of a GUI to interact with that database. The team that worked on this project was made up of four students all from different backgrounds within the Computer Science program at Athens State University, therefore each one of them bringing a different level of skill sets to the project. Since this was not an ordinary database that was being created where it would be simple for the students to understand the data that would need to be stored, there was a learning curve that they had to overcome, and it was through this that they were introduced to research.

The research component that the students had to focus on before starting the project involved the following:

- Learning about Petri nets and how a model was used
- Become familiar with three different cyberattacks
  (1) SQL injection,
  (2) cross-site scripting,
  (3) spear phishing.
- Understand the formalism of a PNPSC
- Comprehend the PNPSC models that were designed for the three different cyberattacks.

Once the team familiarized themselves with the research portions of the study, they then had enough knowledge to be able to start working out the specifications of how to design the database and the graphical user interface to work with that database. The scope of their project was to build a database that will store the 508 CAPEC report entries as well as a web interface for the user to be able to lookup as well as enter new records. The deliverable would consist of four parts: 1) the database that will hold the CAPEC records, 2) web interface for searching records, 3) web interface for entering new records, and 4) database table that will store information about PNPSC models.

The capstone teams attempt to work as closely as possible, following an agile methodology and one of the first things that were accomplished was to determine the user stories that needed to be completed during the nine week semester.

As a user, one wants to be able to:

- filter the database by specific attack criteria
- enter new attack entries into the database
- search for attack patterns with matching purposes
- search for attack patterns by level of CIA impact
- search for attack patterns by level of typical severity
- search for attack patterns by the likelihood of exploit
- search for attack patterns by method of attack
- search for attack patterns by abstraction
- search for attack patterns by attack motivation
- store PNPSC models
- modify PNPSC models
- enter new PNPSC models
- generate PNPSC model details (to use with graphing software)

The students had nine weeks to be able to learn about the research and accomplish the implementation of all of their user stories which was agreed upon by student and professor to fulfill the requirement of the course.

5. Implementation

With thirteen required user stories to complete on top of becoming familiar with the research, the students had to be sure to plan their time wisely. To do so, a timeline was established to keep them on track. Figure 3 displays the timeline that was followed for the implementation of the project.

<table>
<thead>
<tr>
<th>Task</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Setup/Research</td>
<td>May 20th</td>
<td>June 11th</td>
</tr>
<tr>
<td>Design</td>
<td>June 12th</td>
<td>June 20th</td>
</tr>
<tr>
<td>Database Build</td>
<td>June 21st</td>
<td>June 25th</td>
</tr>
<tr>
<td>Parse IML File into Database</td>
<td>June 25th</td>
<td>July 9th</td>
</tr>
<tr>
<td>Build front-end interface</td>
<td>July 9th</td>
<td>July 23rd</td>
</tr>
<tr>
<td>Finalize technical documentation</td>
<td>July 23rd</td>
<td>July 30th</td>
</tr>
<tr>
<td>TOTAL</td>
<td>9 weeks</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 3 Project Timeline**

The result of the design of the specification is presented below:

A. Introduction to the System

The Repository of Cyberattacks provides storage capability for the CAPEC attack pattern information and formatting for generating graphical models. [8]. The system allows users to filter through the CAPEC attack patterns as well add, edit, and delete attack pattern model components. The system allows users to view a textual representation of the attack pattern models. Familiarity with cyberattacks and the CAPEC reports is assumed.
B. System Overview

The Repository of Cyberattacks has two primary components: a database component and a graphical user interface component. A set of scripts will run in the GUI generating and submitting queries to the database component for the purposes of storage and reference. The database initialization was accomplished in an abbreviated format using third party software and therefore deserves a refactor but provides enough utility in storing attack pattern information for this project. The output generated by the GUI can be validated by third party software.

C. System Architecture

The system is broken down into two components, a backend database and a web graphical user interface. The database is designed to be stored and accessed locally using XAMPP [12]. The reason for this is due to the fact that the repository being created is part of ongoing research which has not all been made public and therefore access to this implementation needed to be kept private. The database is composed of five tables as shown in Figure 4.

Four of the tables contain information associated to the stored models, place, transitions, arc, and inhibitor arc information, and one table stores the information drawn from the CAPEC reports to allow for the searching of attack patterns based on specific filters.

The user does not access the database directly. There is a web based graphical user interface that was designed to run a set of scripts that provide information on the formatting and design of the PNPSC models. It also has a set of scripts that takes user input and queries the database based on filtering options that the user provides. Figure 5 displays the home screen of the graphical user interface where it starts by listing all attacks that are stored from the CAPEC report, and through the drop down option on the top left side allows the user to start filtering the data to narrow down the list of attacks. Figure 6 displays one of the screens associated to modifying, editing, adding information to the database. In Figure 6, a Place is either being added to a model or deleted. Across the top portion of the GUI the user is able to select other options associated to the PNPSC models, or he/she can go back to Filter Attacks further. Lastly, Figure 7 displays the generated information for a PNPSC, in this example that of an SQL Injection. At this point the user would need to copy the text that is displayed on the website and paste it into Graphviz, a third party software, to have the graphical model generated [4].

Figure 4 Database Tables

![Figure 4 Database Tables](image)

Figure 5 GUI Home Screen

![Figure 5 GUI Home Screen](image)
6. Summary

Through the requirements of designing a web application, this team of students have now been introduced to three different areas that are associated to ongoing research. They have been introduced to the CAPEC reports that document cyberattacks, a method of graphically representing a model by first becoming familiar with the basic concept of Petri Nets and then studying the formalism of PNPSCs, and lastly they have been introduced to a full four phase research project and how their work in developing this system will assist in the fulfillment of the overall research project.

The system that has been introduced here is a first prototype system. The students were able to complete all of the required user stories and in the process have identified features that are being completed for version two of this prototype. One of the main items that will be worked on is the table that stores all of the CAPEC information. The students used SQLizer to be able to generate an SQL file from the xml file containing the CAPEC information. This made some areas of the table to be duplicated. One of the first actions is to normalize the database and make sure that there are no
associated anomalies or implications of redundancy. In addition to normalizing the database, additional features will need to be added to keep track of PNPS models or pieces of models that are broken down into components and then used to compose a larger PNPS model that would represent a full cyberattack and not just a single attack. In addition to the new information that would need to be tracked, the addition of a plugin or something similar that will allow for the user to not just generate the information for a model but to have the model displayed graphically on the webpage itself instead of just textually. This will eliminate the step of having to use an additional application to view the graphical representation of the model. Lastly, a script needs to be developed to integrate the automatic generation of PNPS from Phase I of the research project to where all of the model information is uploaded into the database.

References


