A Multiplayer Peer-to-Peer Cyber Attack and Defense Infrastructure

Mr. Wesley Allen Hotalen Jr., ECU Department of Technology Systems

Mr. Wesley Hotalen is an Undergraduate student studying Computer Science at East Carolina University. His research interests include programming languages and web application graphical user interfaces (GUI's). Mr. Hotalen is a student web developer for the Department of Engineering and Technology at East Carolina University.

Dr. Te-Shun Chou, East Carolina University

Dr. Te-Shun Chou is an Associate Professor in the Department of Technology Systems at ECU. He received his Bachelor degree in Electronics Engineering at Feng Chia University and both Master's degree and Doctoral degree in Electrical Engineering at Florida International University. He serves as the program coordinator of the Master program in Network Technology for the Department of Technology Systems and the lead faculty of Digital Communication Systems concentration for the Consortium Universities of the Ph.D. in Technology Management. He is also the point of contact of ECU National Centers of Academic Excellence in Cyber Defense Education (CAE-CDE). Dr. Chou teaches IT related courses, which include network security, network intrusion detection and prevention, wireless communications, and network management. His research interests include machine learning, wireless communications, technology education, and information security, especially in the field of intrusion detection and incident response.

A Multiplayer Peer-to-Peer Cyber Attack and Defense Infrastructure

Abstract

This paper will discuss an infrastructure used for cyber security education. The infrastructure includes a set of identical student network environments and each student will have access to one environment to conduct cyber security experiments. Students will be required to identify other students' system vulnerabilities and model the actions of the attacks. Students will also need to recognize the attacks from others and apply certain techniques to mitigate other students' corresponding attacks. The infrastructure will ensure that all of the crafted malicious activities are confined inside the environments. It will provide an isolated environment so no sensitive information can be released to those outside of the infrastructure.

To help students learn the knowledge of network security assessments, computer systems' vulnerabilities exploitation, and intrusion detection and prevention, a series of labs will be designed for students to practice what they have learned and gain hands-on experiences in the field of cyber security. In order to make the labs more manageable and to maximize the effectiveness of navigational instructions, a graphic user interface (GUI) application will be designed. The application will provide a link to all required tools, introductions, and instructions to help students perform the lab activities. The proposed infrastructure will be accessible through a secure Internet connection. With the help of virtualization technology and designed GUI application, students will be able to access the infrastructure anytime and anywhere in the world. Upon completion of these activities, students will be well-prepared and ready to contribute their knowledge and hands-on experiences of cyber security to a high demand workforce.

Keywords: Cybersecurity, virtualization, graphic user interface

1. Introduction

For the last five hundred years or more, the way students learn has not evolved very much. Most often, students sit in massive lecture halls passively listening to lectures. With this traditional approach, the lecturer drives the learning process rather than students actively taking part in the learning process themselves [1]. A more solid and modern approach on learning, called Problem-Based Learning (PBL), was introduced through John Dewey's philosophies. PBL comes from idea that learning is grounded in discovery and is better achieved through mentoring by teaching professionals, rather than the traditional "transmission of knowledge" approach. The PBL approach leads to higher student engagement and involvement in the learning process, which allows for increased levels of deep thought and discovery since students are actively using their critical thinking skills, as opposed to listening to lectures and reading textbooks. This ultimately results in more creative problem solving and retention of the knowledge gained when solving problems hands-on.

The rise of the digital age has introduced the proliferation of computers into the everyday lives of almost every human on the planet. While computers are an extraordinary tool that has the potential to improve the lives of all inhabitants of the earth, they open people, companies, and governments up to increased levels of secretive data intrusions at the hands of hackers. In many situations, such as the December 2013 data breach on Target retail stores, unknowing people had their data compromised because of criminal intrusions into vulnerable systems [2]. In the Target

data breach, the criminal hackers accessed over 40 million credit and debit card accounts. This single event gave these criminals access to all these account holders information including names, mailing addresses, email addresses, phone number, and card numbers.

With the ever-increasing skill of criminal hackers, our educational institutions have an obligation to train even more skilled cyber security professionals to defend critical infrastructure, including retailers like Target, government databases, banking institutions, air traffic control, and more. The best way to accomplish this goal is by engaging students and creating more advanced training simulations to improve the quality of overall understanding in the next generation of cyber security professionals [3-6]. Using the PBL philosophy, the proposed approach will provide real-world simulations in the form of labs that can be performed online through the web application. This web application will use virtualization technology that will be accessed through a user-friendly GUI online. The labs will simulate real world cyber security situations and give students experience using critical thinking to solve cyber security issues that are present in the world today. This approach will train more advanced cyber security professionals with hands-on skills that have been learned through completing cyber security tasks in the design labs.

2. GUI Design

In order to maximize responsiveness of GUI, the application will be built using Angular open source framework [7] that includes HTML5, JavaScript, Typescript, and CSS. The framework will only reload pieces of a page needed for each different component. There will be a login component where users will log into the system. There will be a registration component where users will register for an account. The registration will include the users' first and last name, username, password, and email address. There will be a navigation bar at the top of every page that will contain any necessary links to navigate around the application. The navigation bar will contain links such as labs, account, logout, and a dropdown menu for the information section of the application. Figure 1 illustrates the navigation bar.

Figure 1. Navigation Bar

The labs page will contain all the labs and will have buttons linking to their specific lab page, as illustrated in Figure 2. These links will lead users to the specific lab page where a user will select whether to complete the lab in attack or defend mode. Figure 3 shows the lab page when clicking the DHCP Starvation button. Cybersecurity covers a broad spectrum of topics and only the most important cybersecurity issues will be selected for the lab activities. The tentative labs include password cracking, denial of service attack, web defacement, honeypot, session hijacking, SQL injection, vulnerability exploitation, and digital forensics. Once the user has selected which mode (attack or defend) to complete the lab in, they will be directed to a page containing a detailed introduction to the lab that explains all aspects of the scenario. This is the learning part of the lab. The student will read the introduction page and make sure they understand the concepts before moving on with the next section of the lab. After the user reads the introduction they will be sent to a quiz via a next button. This quiz will test the users understanding of the lab to make sure

they have gained an understanding of the concepts related to the respective lab. Once the user has received a minimum score of 80% on the quiz, they will be sent to a page that explains the lab exercise. The student will have a step by step process to follow to help them complete the lab attack or defend exercise at hand. The student will launch their VM from a *Launch VM* button at the beginning of the lab exercise. The student will be given a series of steps, each including a very generic explaination of the current step that must be completed in their VM. If the students gets stuck and just cannot figure out what they must do to move forward, there are *help* buttons that provide a pop-up help window. The help window will provide more detailed information on the current step and possible required commands. Once the student completes all steps of the lab; the introduction, quiz, and lab exercise, they will have completed that respective lab. Once the user finishes the lab, they will be given a score that is based on their performance in the lab. This score will be stored in the scoreboard, which allows classmates to compete to see who can complete more labs. There will also be a message board that is coupled with the scoreboard, where users can review what labs their classmates have recently completed and their classmates current score.

Figure 2. Lab buttons link to Attack and Defend button options

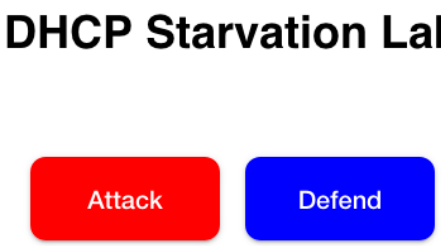

Figure 3. Lab page example

Two virtual machines (VMs) will be included in each lab for students to conduct the attack and defense lab activities. The VM will be launched by the student during the respective lab exercise portion of the lab. These VMs will be configured by the student completeing the lab. When a student configures a VM to defend against a certain attack, the rest of the students class with be tasked with attacking the VM that the student has set up. Each student will attack other student's VMs and will attempt to prevent theirs from being attacked. This is what provides the peer-to-peer multiplayer functionality of the GUI. The Angular 5 framework uses node package manager (npm) and Node.js to communicate with the MongoDB backend. The GUI application coupled with virtualization technology will allow students to easily navigate the application and complete labs without the complexity of setting up these scenarios on their own machines.

3. Specification

The Angular 5 framework allows for a seamless experience when navigating between pages. Traditional websites build a new HTML page for every different page that needs to be displayed. For instance, the labs page needs a HTML page and all associated session logic. When a user clicks on a link directing the user to the *Account* section of the site, an entirely separate page will be fetched from the server and loaded to the users view. This can often lead to delayed responsiveness and a less user friendly experience. With Angular 5, the application is built using Angular's modularity system called NgModules. An Angular 5 application is created by composing HTML templates with imbedded Angular markup, then writing component classes that manage the templates [7]. There is then added application logic in an Angular service. Figure 4 shows the modules that include HTML templates, components, and services.

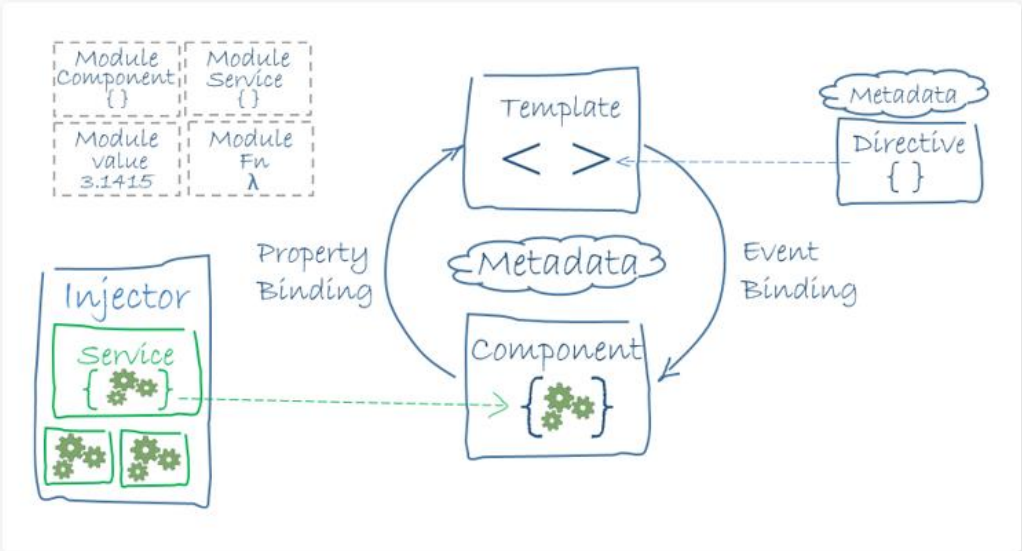

Figure 4. Angular 5 General Architecture overview

The GUI application is built using Angular 5 and will provide a smooth and user friendly experience due to the use of the Angular components ability to create custom HTML5 tags or *selectors* to embed in HTML5. This means there can be a simple main web page that consists only of HTML5 tags. Each tag has a corresponding component that it represents. The navigation component, for instance, can be created and embedded into the *main.html* page; it consists of a

nav.component.html template, a *nav.component.css* style sheet, and a *nav.component.ts* where the component and associated logic are defined. As shown in Figure 5, the navigation component will display no matter what page the user navigates to because it is in *main.html*.

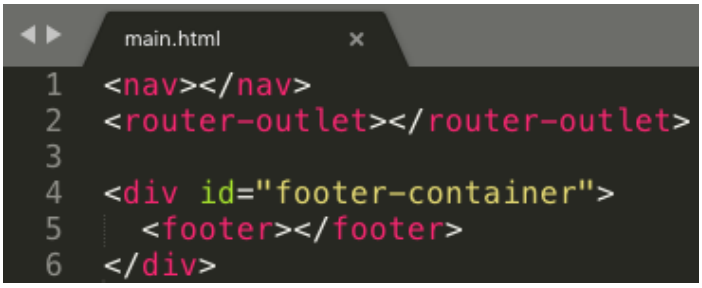

Figure 5. Example of *main.html* file for the web application

There is a router-outlet tag added to the *main.html* which allows the creation of many different content components; these can then be routed using the router outlet. These different components include: login, labs, information, and account pages. This approach enables the application to only reload the pieces of the web page that it needs and not have to repeatedly fetch entire HTML pages with associated logic. As an example, the *login.component.ts* file defines the behavior of the component and is where the logic associated with this component is stored. The *login.component.html* is where the actual html content for the particular component is stored. Lastly the *login.component.css* is where css styling for the component is stored. These files work in conjunction to render the portion of the page needed at a given time. The login page is only rendered once the router outlet directs a user to the login page. Once a user successfully logs in, they are routed to the labs page and only the labs component is loaded from the server. On both the login and labs page, the navigation and footer of the application remain the same and are not reloaded.

In Angular 5 these main content components are called parent components because they are in *main.html*. However, when we add a selector for a component into the template of another component, this is called a child component. The child component idea will be used to display the score/messege board (a table) containing all users name, score, and related messages in the labs component. Not only does this provide for enhanced responsiveness, but it helps keep the application code organized and in small pieces. For instance, if the above mentioned table needs to have its HTML or CSS edited, there is just one place to go: the table component. Likewise, if the navigation component needs to be modified or debugged, there is just one place to apply these changes and the changes reflect everywhere the navigation component is displayed.

Another feature of Angular 5 are *ngIf directives, which are used to dynamically place extra links on the navigation bar, such as account and logout, once the user has been authenticated. NGIf is an Angular directive used to conditionally show the inline HTML5 template as seen in Figure 6. This directive will only display these list items in the DOM (Document Object Model) if the condition *isLoggedIn()* is true. The account link will direct users to their account page which will display their full name, username and email. The account component will also have a child component password form which allows users to change their password. The Labs link routes users to the labs component of the application, which will display all labs and the table component. Each lab will be represented by a button that links users to the selected lab activity.

```
<li *ngIf="isLoggedIn()">
    <a routerLink="/account">Account</a>
</li>
<li *ngIf="isLoggedIn()">
    <a (click)="logout()">Logout</a>
</li>
```

Figure 6. Example of an *NGIf* directive for the web application

Like the NGIf directive, another useful directive provided by the Angular 5 framework is the NGFor directive. This allows us to write code once for a (list item tag) and produce multiple tags using the NGFor directive. For each question in an array of questions, Angular will produce another list item tag with the question embedded in the list item. This is illustrated in Figure 7 below.

```
<li *ngFor="let question of questions; let i = index">
    <h3>{{question.name}}</h3>
    <div *ngFor="let answer of question.answers" class="radio" >
        <label *ngIf="answer">
            <input [(ngModel)]="question.selectedAnswer" name="answer{{i}}" type="radio" [
            value]="answer"> {{answer}}
        </label>
    </div>
    <br>
</li>
```

Figure 7. Example of an *NGFor* directive for the web application

4. Conclusions

In this paper, implementation of a GUI application is discussed. An educational learning infrastructure will be constructed for students to practice the operations of cyber security activities. The proposed GUI application will be designed using Angular 5 framework. Having completed the design, we expect that the GUI will provide an effective solution that facilitates students to conduct both attack and defense lab activities. In the future, we will enhance the functions of the proposed GUI application. We will research the best approach of the communications between the buttons shown on the GUI and the VMs on the backend.

Achnowledge

This research is made possible by the National Science Foundation under grant 1723650. The authors are grateful to the support of Department of Technology Systems in the College of Engineering and Technology at ECU.

References

1. Michael O'Grady, "Practical Problem-Based Learning in Computing Education," *ACM Transactions on Computing Education*, vol. 12, no. 3, pp. 1–16, Jan. 2012.

2. Data breach FAQ, Target. https://corporate.target.com/about/shopping-experience/payment-card-issue-faq (Last browsed in February 2018)
3. Richard Weiss, Jens Mache, Michael Locasto, and Frankly Turbak, “Hands-on Cybersecurity Exercises That are Easy to Access and Assess,” Proceedings of the 2017 ACM SIGCSE Technical Symposium on computer science education, Seattle, Washington, March 2017.
4. Jessica Chisholm, “Analysis on the Perceived Usefulness of Hands-on Virtual Labs in Cybersecurity Classes,” Ph.D. dissertation, Colorado Technical University, Colorado Springs, CO, 2015.
5. Dongqing Yuan, “Design and Develop Hands on Cyber-security Curriculum and Laboratory,” Computing Conference, London, UK, July 2017.
6. Melissa Carlton, ”Development of a Cybersecurity Skills Index: A Scenarios-Based, Hands-On Measure of Non-IT Professionals' Cybersecurity Skills,” Ph.D. dissertation, College of Engineering and Computing, Nova Southeastern University, Fort Lauderdale, FL, 2016.
7. Angular. https://angular.io/guide/architecture (Last browsed in February 2018)