The USENIX Journal of Education in System Administration
Volume 1, Number 1 • September 2015
www.usenix.org/jesa/0101

The Development of a Computer & Network Security Education Interactive Gaming Architecture for High School Age Students

GUY HEMBROFF, Michigan Technological University
LUCAS HANSON, Michigan Technological University
TIM VANWAGNER, Michigan Technological University
SCOTT WAMBOLD, Michigan Technological University
XINLI WANG, Michigan Technological University

Computer and network security cases continue to rise each year, playing an important role within our society. With a growing job market in this field, there remains little formal education at the high school level to become familiar with this profession. We proposed to develop an interactive computer and network security game which differs from other security-based games previously created, as it does not focus only on computer science security. Our development focuses on a wide range of topics and layers of the OSI Model to offer computer and network security education critical in areas of network and system administration. We have created a storyline, in which each level relates to the story in sequence, creating an engaging story for the player. We also provide details how our gaming architecture is configured. Early results from players who have tested the game from a student and teacher perspective show encouraging results.

1. INTRODUCTION

Computer and network security education can be difficult due to the growing number of threats on computing devices or platforms, such as mobile technologies, routers, or databases. The wide array of applications, protocols, and procedures make computer and network security complex and difficult to fully understand. Universities have begun to offer courses and majors based around field, however most of the development surrounds computer science-oriented courses and often omits a considerable amount of computer and network security. Within most high schools, computer and network security education is often non-existent, and yet this age range is very impressionable and responsible for beginning to choose subjects of interest, leading to an eventual career path. Providing exposure, education, and ethical considerations surrounding the subject of computer and network security can prove beneficial in developing our world's next computer and network security specialists.

The motivation of this project focuses on providing a wide range of computer and network security to high school students through gaming, extending the research we conducted in developing information technology labs for universities with system and network administration programs [Wang et al 2013]. However, instead of offering lengthy documents about how to properly secure a network and computing environment, we provide a hands-on gaming environment to help immerse high school students into learning more about this field. This approach allows the project to encompass avenues of offense and the methods used to defend these types of attacks. The game is designed to provide a gateway for students to obtain knowledge of computer and network security in a fun and interactive environment.

Our motivation for this paper has two distinct purposes. First, we wish to describe our solution's mission to provide computer and network security education through gaming for high school students within a safe environment. Second, we provide detail in explaining the gaming architecture that was developed, providing details in its configuration. We chose to do this due to frequent complications associated with implementing a gaming architecture solution into your own network environment. We also thought the readers of the Journal of Education and System Administration would find the overall configuration details relevant due to its configuration content in system administration.

2. A RISE IN THE NEED OF COMPUTER AND NETWORK SECURITY PROFESSIONALS

A higher volume of cybercrime is conducted each year, while technology continues to evolve, placing more computing into our daily lives, giving humans more opportunities to store and access electronic data. According to the Bureau of Labor Statistics, "Since 2003, employment in the IT industry has grown by 37%." These careers ranged from programmers, systems design, computer facilities management, and other computer related services [Csorny 2013]. Cisco's Annual Security Report states that there will be a global shortage of over a million IT security professionals, stating: "Most people don't have the people or the systems to monitor their networks consistently" [Cisco Systems 2014]. With the expanding job market in computer and network related positions, the shortage for computer and network security personnel to secure these systems, and the complexities in providing this comprehensive security education, provide a compelling argument aimed to expose and educate high school students in this growing field by providing an engaging, safe, and fun-learning environment through gaming.

3. PREVIOUS WORK

Computer security gaming is not a new concept. Over the past ten years, there have been many developments in this field. While many past solutions differ from ours in terms of the audience they target or the approach taken, we have listed below those which have provided the most relevant content to our approach and the greatest influence to our project.

Research by Srikwan, uses cartoon comics to improve security awareness and understanding among typical Internet users, and targets similar age ranges as our research [Srikwan 2007]. The security topics covered include malware, spoofing, phishing, pharming and password safety. The cartoon comics provide simplified examples of common security threats/attacks, with the intent for the reader to understand the importance of internet security by drawing parallels between the example and the reader's life experiences. Although the visual comic provides an effective method for helping the reader understand why security awareness is important, it lacks the ability to go into detail in providing educational awareness for the user.

Development conducted by Jordan, Knapp, and Mitchell provide an interactive gaming concept, similar to ours, in training users in computer security [Jordan et al 2011]. Their creation of an interactive gaming concept to provide basic security training with surveys to evaluate the users' interests and opinions of the game was well-developed and helped to further our research efforts in this area. Our solution differs in providing a more comprehensive educational role within each module providing a continuing storyline to help keep the user engaged, and offer different security content.

Research conducted by ISECOM, a non-profit and open-source research group focuses primarily on security awareness [Hacker High School – ISECOM 2013]. The program uses a classroom lesson approach designed for teenagers as the main audience. There are currently nine subject areas with ten more in development. Topics range from learning basic commands in Windows and Linux, social engineering defenses, to using firewalls and Intrusion Detection Systems (IDS). Although the lessons include security-related exercises, this solution differs from ours in that it provides a much smaller amount of graphical content and does not offer detailed educational and module gaming material.

The group PicoCTF has developed a very interesting and comprehensive computer security gaming solution, which is accessible from a Web browser [Chapman et al 2014]. Although targeted age incorporates a wider range of students than ours, as it includes middle school age children, the security education gaming experience is rich in graphics and content, which is updated on an annual basis and offers open-source capabilities for development. Our security education differs in offering computer and network security content and education for system and network administration and does not focus solely on a computer science approach, as our solution includes modules associated with layers one, two, and three of the Open System Interconnection (OSI) model. The OSI model remains an integral reference and teaching tool into the organization of computer and network education.

4. METHODOLOGY

The project's goal is to create an engaging computer and network security game designed to broaden knowledge of basic to more advanced topics for users ages 14-18 years of age. A scoring server, to keep the scores of each player and allow them to compare and compete with other players is part of the architecture. Secondly, an intriguing story is important to capture the user's interest and keep them playing to find out what happens next. The story, as well as the game itself, is broken into modules, or levels. This helps to insure the player has smaller, manageable size modules to tackle, with each level providing a new computer and network security education topic.

We have aimed to improve upon existing tools already in use, and develop computer and network security education for the targeted audience. We plan to provide our gaming solution in a virtual environment, working with teachers of high schools to provide additional web-enabled tools and education for them to evaluate, improving the education of their students in this field.

5. GAME STORY

Our goal in writing the story and the layout of the game was to develop a linear path that would allow players to see how each scenario that applied in the real world. We wanted to show scenarios from each layer of the OSI model and present discussion on the legal and ethical implementations of each scenario, as well as give an overview of how organizations approach such a case. In the story, players take the role of a competitive programmer/hacker 'Oddball'. Your friend 'Shortstack' won the Pwn2Own contest with a Cross-Site Scripting attack against Amazon. Upon your arrival home, Shortstack is arrested as his program, 'pikpokit' was found to be in use. The FBI agent approaches you to be a consultant on the case. In the first level, players will learn a little bit about social engineering and communication as they talk with Shortstack. The next level has the player looking at Shortstack's laptop, trying to find proof of it being tampered with. The player will find a USB key, which contains a program they will have to interpret. The program is a keylogger, sending keystrokes to an IP that is scanned by the user in the next level using nmap. The networking levels use VMs to create safe yet realistic environments for the players to work in. Scanning the IP turns up a home Web server of a script kiddie. There the player will learn about social engineering and communication. The player will then discover that the script kiddie was just passing along the information to a larger cybercrime organization. The next hop shows Shortstack's data was sent to a Web server for a small store in northern Vermont. Here, the player will show how the server was hacked using SQL injection. They will then write access-control lists (ACLs) and configure to help prevent these kinds of attacks in the future, as well as look at phishing emails used by the crime syndicate to gain information about their target. These phishing emails point to another server where the data is being stored to distribute to 'pikpokit'. Using a man in the middle attack, the players will find this information. The player will then decrypt the data found using python to run the Blowfish and AES encryption algorithms. Seeing that they are compromised, the syndicate runs a DDoS attack against the FBI, which the player helps mitigate. The story ends with a final interrogation of an underboss in the syndicate, and has different endings based on the success of the player.

When choosing the security content of the game, we aimed at providing education around critical computer and network areas, along with considering the various layers of the OSI model. We choose to embed strong conceptual content over specific applications-type of computer security, such as Heartbleed, to ensure students would learn fundamentals, and not merely the latest attacks. Modules were also constructed to provide the users with combinations of communication, observation, and technical skills.

6. LIST OF MODULES

Modules were developed to provide a fun and interesting way to learn about computer and network security concepts through interactive demonstrations. The modules cover a wide variety of security topics coinciding with OSI model layers. Additional modules can be created and added to the game, permitting the game to evolve with computer and network security education. The only caveat depends on following the sequence for the existing storyline.

6.1. Module 1: Physical Security

This module simulates the "friend's" laptop that had been tampered with at the hackathon event. The player will look at various different images to view parts of the laptop, which contain clues to signify a malicious event, such as missing screws from the laptop chassis, or damaged physical ports on the device. The player to pick from certain options to identify information needed to understand how the code was removed from the laptop and used for malicious purposes. The interface is simple enough to see the images of the laptop before and after the tampering occurred. This allows the player to identify certain areas of the laptop that have these indications.

6.2. Module 2: Port Scanning

In this level, the player is asked to look at a machine discovered from the keylogger software on the friend's computer. The player will use Nmap, a piece of software designed to scan and interpret what ports are open on a computer. With this information, the player can determine what the machine is being used for and if there are any open ports used against the machine. This level is designed more around information gathering than it is about attacking or defending.

6.3. Module 3: Keylogger

In this module, the user is presented with a piece of code that was discovered on their friend's laptop. The goal of this level is for the player to determine what the code does and where it is sending the information. With help from the affiliated education section, the player will determine it is a key logger used to send information to a server controlled by the malicious cartel.

6.4. Module 4: Social Engineering

The social engineering module for the project changes focus from the technical aspects of computer security to the interpersonal relationships which take place within the environment of an established network. By definition, social engineering is the idea of a network's security being penetrated through human interaction and the manipulation of individuals. Although it is important to focus on securing a network through its perimeter, the idea of an individual within the company also poses a risk which often goes neglected. A study by Check Point Software Technologies, which included 833 IT professionals from around the world, discovered that among these professional companies, only 26% of participants actively train employees on the threat, 34% did not have any initiatives in place during the time of the study, and that 40% put the responsibility on the employee to read and understand their organization's overall security policy documents to prevent data loss, security attacks, and social engineering-based threats [Check Point Software Technologies LTD 2011]. This module is intended to enlighten the user of the effects of social engineering in its many forms, and often as a non-technical approach in computer and network security.

The primary focus of the module is to give the player an understanding of nonverbal communication to assess a situation. Using Ren'py, a distribution of the Python scripting language specifically designed around ease of storytelling and education, the user switches from being the player who is attempting to set their friend free, to an FBI agent during an interrogation with the initial suspect. The module is intended to be broken into two separate sections. The first occurring in the early stages of the game to familiarize the user to the concept of shared body tells among humans. It functions as a tutorial, providing information presented in flashbacks to the night of the initial crime based on given reactions to the accused during questioning. In the secondary part of the module, which occurs towards the end of the game, the user is able to use what they had gathered from the initial module and perform the interrogation with a suspect where no tutorial is provided.

Scoring on the combined modules is dictated towards an internal point system the scoring engine uses. Points are added together based on reactions and information the player receives from the accused, and from there, an ending is given based on the amount of points received. However, if the user returns with no further information, and the accused refuses to speak, a scripted video will appear as if a new interviewer were to go in with the accused and obtain the information themselves, providing a tutorial in how this technique works.

6.5. Module 5: Encryption

This Encryption is the process of scrambling data so that a user would need a key to properly view the data. This module focuses on the math behind two common symmetrical encryption algorithms, AES and Blowfish. Symmetrical algorithms use a single key to encrypt and decrypt information, and are usually done to protect files at rest. Users will be presented with unfinished scripts for AES and Blowfish implementations, where they must finish the lists of matrix transformations for two algorithms.

6.6. Module 6: DDoS Mitigation with SYN Flood

The DDoS mitigation module will educate users on how to mitigate a syn flood attack against a Web server. The player is given a computer on a public network trying to access a Web page that resides on one of the FBI's private networks. The player will enter this module as the attack has already begun and is tasked with needing to regain access to the Web server. The more time a player takes, the total amount of potential points of the level diminish. The Web page entails information for the player to submit, validating they have got the Web server back up and running. The player will be asked to do all of this by using SSH to access the NetBSD packet filter placed between the public and private networks which the Web server resides. On the NetBSD machine, only certain commands are permitted for the player to issue. These commands are already set up in a sudoer file to keep the player on track. For example, if the player issues a *tcpdump* command on either interface pcn1 or pcn2, it will display a large amount of syn's being sent to the server, with zero of the handshakes completing successfully. This will inform the user it is most likely a syn flood attack. The player must alter the rules of the packet filter and stop the attack. The rules to be used in the pf.conf file will be TCP SYN proxy and anti spoof. With the correct rules put in place, the player will be able to view the Web server's Web page from their computer and answer the question to validate they have completed this module successfully.

6.7. Module 7: SQL Injection

This level brings players to an offensive exercise, where the user is given a Web page for a small business, which has been configured with vulnerabilities, allowing the malicious cartel to access the small company's machine and use it as a data drop. The player is given the address to a Website of the vulnerable Web server and performs various SQL injection techniques to gain access to information stored in a SQL database. This information can then be used to log onto the server being used by the small business. This information will also give them root access to the machine, allowing them to locate the data linked to the malicious cartel and the next clue in tracking this group.

6.8. Module 8: Phishing

This level is designed with a list of emails. All of the emails are presented to the player in their source format. The player goes through these emails and is shown various forms that a phishing email can take. Most of the emails presented to the player are considered legitimate and are not Phishing emails. The player's task is to select the correct phishing emails to successfully complete the level and gain experience in detecting this type of malicious content. Each module we have developed offers a corresponding education section to help players learn about the respective subject areas and become knowledgeable enough to accomplish the level, regardless of prior education in this area.

6.9. Module 9: Man in the Middle

In this level, the player is attacking a server they believe has the *pickpocket* program, leading the user to the cartel. The player simulates a man in the middle attack using Kali Linux, while using specific tools to perform such an attack. The user utilizes Ettercap, which will permit them to locate active traffic on the network where the player will then poison this traffic link between two hosts and obtain the sensitive information as an acting host to each of the two original hosts. Using Wireshark, they can capture more detail about the traffic, such as the protocol being used and the ability to follow the TCP stream, finding the details of the information being sent over the link. With this module, it is important for the user to learn what types of attacks are common, and the methods that are performed, so they can learn how to prevent such an attack.

7. GAMING ARCHITECTURE

7.1. Virtual Machine Web Service Platform

To provide an interactive gaming solution, we created a Web-based virtual machine platform. The solution was developed to be hosted on open-source and provide a user-interface which connected to a backend of virtual machines, coordinating the security, scoring, and virtual machines needed within each module or level. The solution was also developed to be free, or relatively inexpensive, as this project was designed to keep costs at an affordable level. A total of five different services were compared.

The first alternative used VMware WSX to connect to the virtual machines remotely from a browser [VMware 2012]. This platform was very fast and provided a variety of options to turn off, pause, or administrate virtual machines remotely. This tool however, was not open source, and as a result could not be configured to the gaming structure's specifications. VMware VSphere Web client featured a rich set of options, including even more tools to administer the virtual machines [VMware 2013]. This connection, however, could not become Web-based and was not open source. PHPVirtualBox is an open source product and included administrative controls needed to administrate the servers, as well as protect the virtual machines and communication [PHPVirtual Box 2013]. However, this option struggled to integrate the environment and tools we had developed, and therefore was not considered a viable solution. The next option was a commercial solution utilizing RealVNC [RealVNC 2013]. This solution would cost approximately $900.00 for the license. The last option was the most optimal for our design. It also consisted of utilizing VNC, however, this solution incorporated VNC over a WebSocket, or NoVNC [Martin 2013].

NoVNC was an open source solution and also included a powerful pre-built library to incorporate many of the simple features we needed. The copy function also provided a means to pull data from the user's virtual machines within the game and send over WebSockets to the scoring server securely. The connection options also provided the debugging features we required. A wide array of protocols and security features are included with NoVNC to ensure the game is played properly and securely. Cookies are used to remember settings and restore defaults when connecting to the virtual machine's Web service. Secure WebSockets (wss://) are used to transfer information from the clients to the servers. This includes the remote frame buffer (rfb) information while controlling the virtual machines, as well as the data from the user to be scored for each module. The connection starts on the Website where the WebSocket is opened, and a connection using SSL (wss://) is created. The controlled virtual machines host a WebSocket proxy and receive the information from the wss://. This proxy forwards the stripped rfb protocol to the VNC server, which is hosted on the virtual machine controlled machine.

7.2. The Virtual Machines of the Gaming Architecture

The Multiple administrative servers exist on the architecture, outside of the virtual machines designated for each module. Scoring servers, Web servers, and MySQL servers are needed to ensure secure and robust transmission of data between the virtual machines. Figure 7.1 illustrates the computer and network gaming architecture.

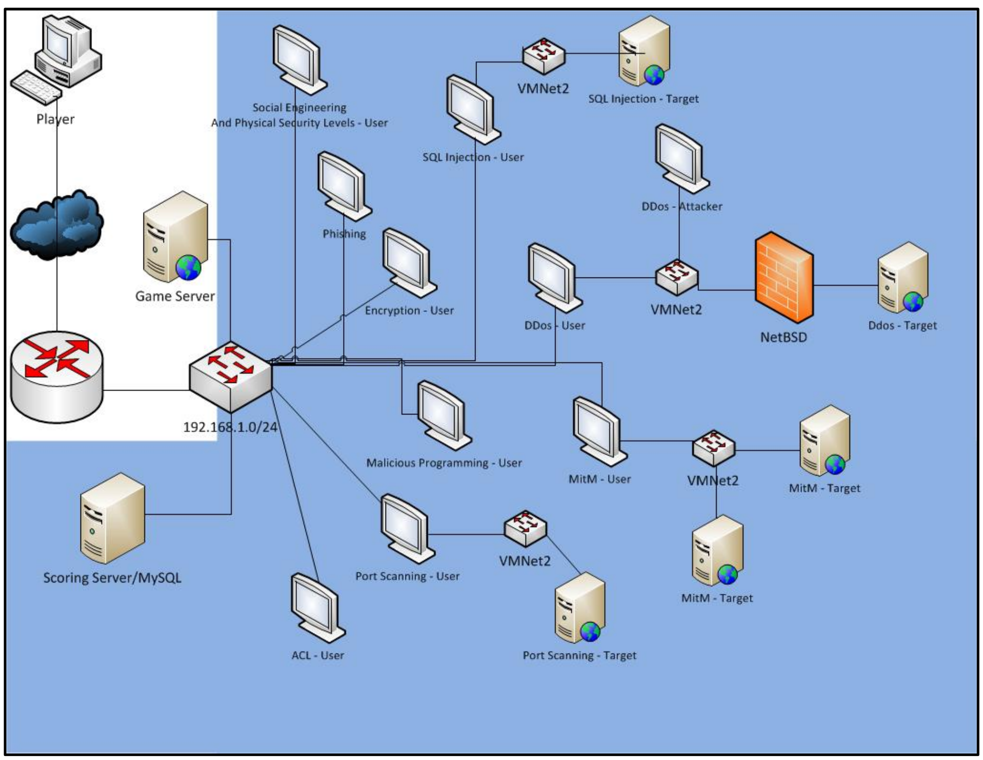

Figure 7.1. Computer & Network Gaming Architecture

Each separate module's virtual machines are Kali, Fedora, Windows, or NetBSD based. Each module's virtual machines require the Websockify proxy, as well as a VNC server running at all times, permitting access from Web clients to the virtual machines.

The Web server uses the solution created by Apache called Xampp on the machine facilitating the Website for clients to use. The Web server incorporates NoVNC as the virtual machine Web service platform, as well as a WebSocket client for communication to the scoring server and virtual machines. Help topics, Camtasia walkthroughs, network diagrams and technical help have also been added and hosted on the Web server and served by an "as requested" system.

7.3. VM Resource Management

The use of one or multiple VM's for each module presents potential resource allocation issues and can slow or halt the game sequence, which leaves users frustrated or disinterested in the game. Early in the development of the game, we experienced some levels having too many virtual machines for our physical computer to run efficiently, causing slow response times. To rectify this issue, we implemented the VMs for each level to start and stop on command. VMware workstation has a useful command called *vmrun*, which allowed the game's VMs to start, stop, and revert to snapshots using the command line. These commands were then added via a batch file, which we call PHP when machines needed to start, stop, or be in another state required for each level.

Each batch file starts off with stopping every VM that is part of the game's respective levels. This ensures that no other machines are turned on by accident, interfering with the game or consuming its resources. The batch file then runs the command *vmrun reverttosnapshot*, which will revert any machine for that level back to the state before a player had started to use it.

7.4. IP Address Schemes

The Connections to machines utilize the private space of 192.168.1.0/24 network. All VM's used by the player have a 192.168.1.x IP address on a network interface which has a bridged connection to the physical machines network. The VM's also have additional virtual private network space they require to perform the offense and defense exercises within each module. These networks range from 172.16.0.0/16 to 10.0.0.0/8.

7.5. Web Frontend

The Web frontend, created in PHP, is used as the Web interface for the players to interact with the game, providing paths to interact with the backend servers. If a user accesses the education portion of the module, they have an option to press the *Play* button on the right side of the screen and view the virtual machine(s) used for the specific level, to becoming accessible. Other buttons on the screen are intentionally disabled when they are not relevant and help to keep players on track. Figure 7.2 provides an example of the gaming Web frontend.

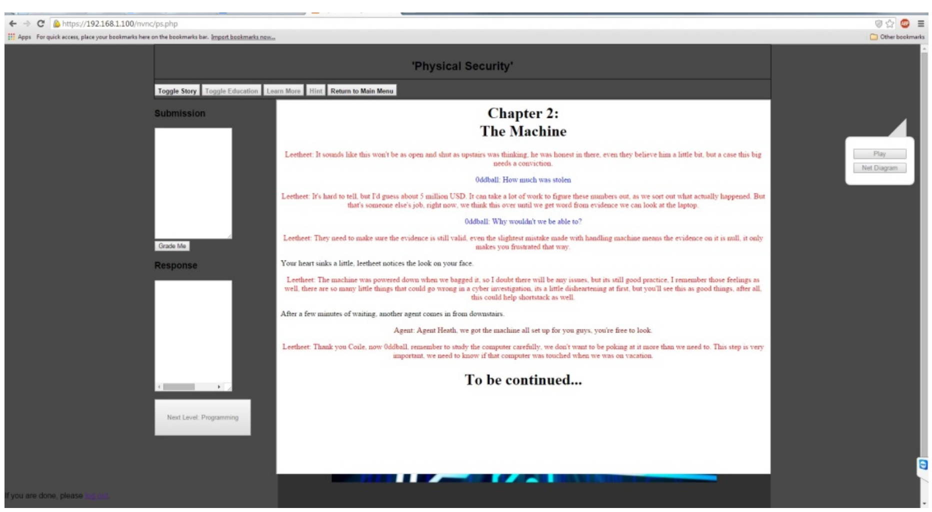

Figure 7.2. Gaming Frontend Web View Example

The main screen provides a window mirroring an instance of the chosen module's virtual machine. The virtual machine is selected from the right sidebar, with each button representing a different machine. When the button is clicked, the designated identifier is pulled from the utils.js JavaScript file and called into the Web interface's PHP file. Full control of the machine is allowed through this portal.

Other buttons, listed on top, such as the *Learn More* and *Hint* buttons, provide additional information specific to each. The *Learn More* button provides additional education about the attack or method to prevent it. The *Hint* button provides useful hints for the player, helping to highlight the area of code which is pertinent to solving the level. The *Return to Main Menu* button returns the user to the game's main screen where levels can be selected.

The left sidebar contains two boxes, *Submission* and *Response*. The *Submission* box is to be used by the player to place the plaintext that he or she deems as the answer to the module. Upon clicking the *Grade Me* button, the plaintext is sent securely using a wss:// socket connection to the Fedora Scoring server, where the data from the user is analyzed and scored. The *Response* box echoes the results and allows the player to see what the player has inputted for submission.

The Main page for the game presents users with a couple of different options. They can select to start from the beginning or level one. They can select the *Continue from Last Point* button where it will bring them to the last level they had played, but not completed. The level selection buttons can be selected to bring them to the exact level they want to play. The buttons will be disabled and grayed if the player has not played that level yet. They can only select a level's button by playing the previous levels to completion and receiving a score for the previous level. Once a level is selected, a player is presented with a loading screen while the VM's are started for that particular level. Figure 7.3 provides a view of the Main page.

Figure 7.3. Main Page View

7.6. MySQL Server

The MySQL server stores all player information for the game. The player login is stored in the MySQL server with the password hashed as well as a salt for the password. There is also a login_attempts table used along with the PHP login script. This table is used to check for brute force attacks. The table is used to store user_id numbers and the time of the inserted entry. PHP uses this table to check if there have been five failed login attempts in the last two hours. If so, the player is locked out until two hours have passed or the administrator for the game has removed these entries from the table.

Each player is given a unique ID number in the database that is automatically incremented each time a user is added to the database. This ID is used as a Primary key in the database. Each level has its own table as well. The tables for the levels have the foreign key as the player's ID number. That way, it can be referenced when entering scores into the level for the player. The level table also has a *Played* column, which is used to indicate if the player has played the specific level or qualifies to play the level in question. This determination is given to the PHP frontend to grant or deny the player access to the specific level. To help secure the MySQL database, different user-types are created between players and administrators. Each user is given the minimal permissions as the user needs to complete their tasks. Figure 7.4 illustrates the MySQL architecture used.

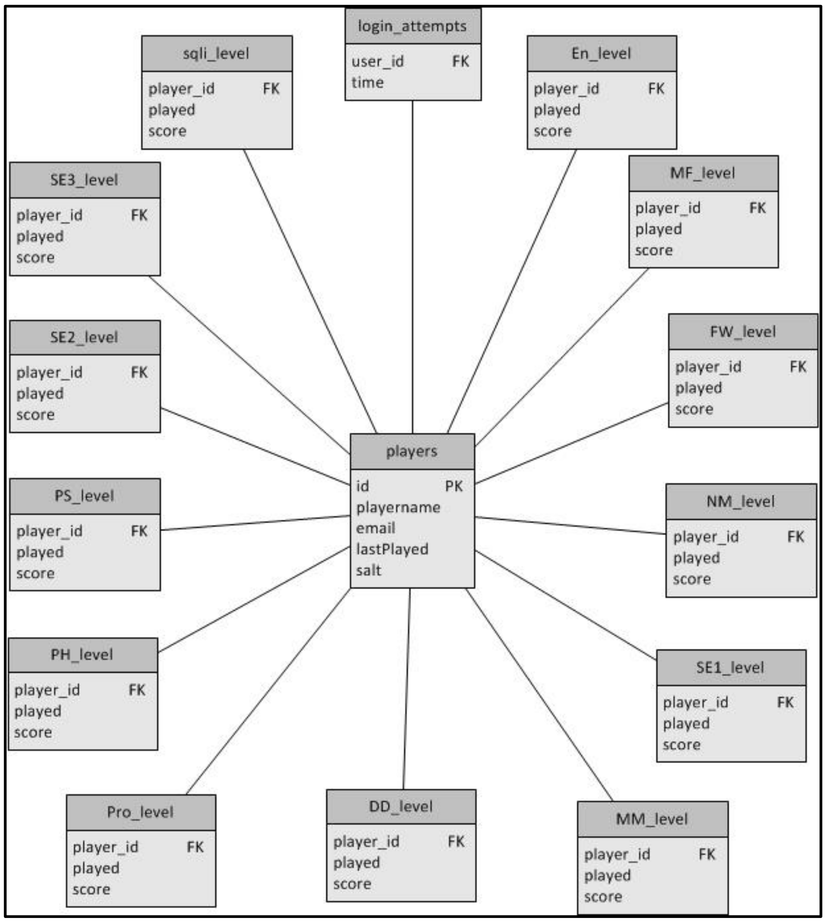

Figure 7.4. MySQL Architecture

7.7. WebSocket Server

The WebSocket server, which was built on top of the scoring server, was built using Tornado version 3.1 as the backend. Early attempts were made to use version 2.0, however it was never successfully implemented. Tornado is a Python Web framework and asynchronous networking library. Once installed, the server code, built in Python, was modified to accept connections on different ports and used a specialized "handler" to be embedded in the JavaScript on the frontend. For this configuration, port 8888 and "/ws" was used. An example of the WebSocket configuration file used to score the defense of a Distributed Denial of Service (DDoS) attack module can be seen in Figure 7.5.

The WebSocket has three main functions: creating the connection, sending a message, and closing the connection. For simplicity purposes, we designed the button *Grade Me* to perform all of these functions. This was implemented by referencing the button id *Connect* for all three functions in our echo JavaScript file.

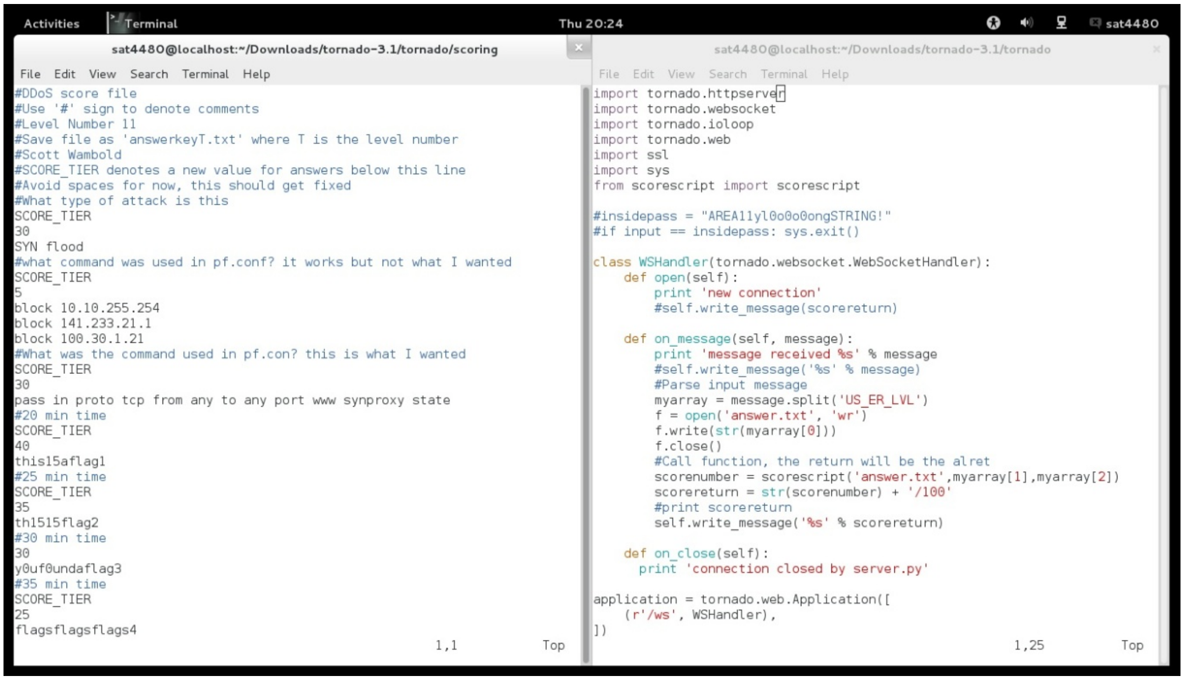

Figure 7.5 WebSocket Configuration File

For troubleshooting purposes, additional lines of code were added to the server1.py file, sending a message to the console when a successful connection was made, which text was sent, and when a connection was closed. These lines are currently commented out to avoid them from showing up, but can be uncommented to troubleshoot any problems. For security purposes, an additional line of code was added to immediately close the connection once the message was received. When the server code finishes, and the WebSocket server is started, successful connections from other virtual machines to the WebSocket server are made. It should be noted however, we experienced a "connection refused" when the first attempt was made to send a message from the Web interface to the WebSocket server. Upon troubleshooting, it was determined a firewall rule allowing TCP connections on port 8888 was needed. Once this change was made, the Websocket server was able to successfully receive transmissions from the Web frontend.

Once the WebSocket configuration and functionality was successfully implemented, the next step was to determine how to handle the message received. To tackle this problem, it was necessary to place the received message into a text file on the server for scoring and analyzing purposes, which is described in Scoring section.

8. SCORING

The Scoring server comprises of a Fedora operating system which hosts a MySQL database and WebSocket server. The database tracks user login attempts, as well as module level and user's scoring. The WebSocket server securely sends and receives information from Web clients. The scoring is completed by comparing the results to entries in a MySQL database. If the entry matches, then the corresponding points in each row are given to the individual. Comparisons are also done using the "AND" and "OR" operations to ensure entries are given fair evaluation. Scoring was designed, permitting all levels were able to score using the same method. Players' answers are sent to a scoring server virtual machine using the Tornado Websocket. The answer is stored in a string along with their user id and number. The string is then parsed and the answer is saved while the user id and number are used to start the scoring script. The scoring script will compare the answer file to an answer key configured by the level designer.

The answer key allows designers to choose keywords an answer should contain, as well as each keyword's value, and penalized guessing. The script will then perform error checking, preventing users from scoring outside of the 0 to 100, and store the total values in the database. The score is then output to the user, as well as the option to *try again* or *continue*. In future releases, we plan to add more abilities to the scoring script, including the use of or statements for similar answers, along with testing security and stability.

9. ADDITIONAL SECURITY MEASURES

To protect the game's integrity and accuracy, enforcing security within the game is critical. NoVNC was used as the main connection between the Web clients and the virtual machines and provides one of the most vulnerable areas for attacking the game itself. The NoVNC connection uses SSL encryption to protect against tampering. It also provides a flash-based WebSocket emulator for any browsers that do not fully support WebSockets.

The WebSocket is used to securely provide a socket for communication to and from our module virtual machines, as well as the scoring server. The benefit of WebSockets is that the handshake only needs to be completed once, for the communication to continue. This results in less overhead traffic, as well as less checking of credentials on each connection. WebSockets use the same TLS/SSL security as HTTPS to protect against attacks trying to mimic the module virtual machines, and pass them off as authentic. This encryption also prevents replay attacks from "copying" another user doing the exercises.

Each module virtual machine is secured by enabling only the WebSocket proxy port outside of the firewall. This allows only the secured wss:// connection to be the only way to interact with this virtual machine. If players alter a machine's configuration or render a virtual machine unusable, the machine will be brought back to its original configuration using the VMware workstation's snapshots of VM. The scoring server itself is protected by only allowing wss:// connections that use a certificate signed by the root Certificate Authority (CA).

Logins to the game is conducted using a PHP script that checks the MySQL server for user login information. User passwords are stored using SHA512, along with a hashed salt value. The login script also checks for brute force attacks against a user's account. Just like HTTPS, a certificate was created and installed on the scoring server, using openssl, to encrypt the WebSocket communication. This prevented the player's transmission from being sent in plain text to the scoring server. In order for the WebSocket server to know to encrypt the data, two lines were added that point to the location of the installed certificate.

10. PRELIMINARY RESULTS

The purpose of this project was to develop an interactive and fun computer and network educational game for students targeting a specified age range. Although we are still developing the game, we wanted to gather feedback regarding the approach and design we have taken. To do this, we let fifteen students within the ages of 14-18 years of age play the game. The average results from our questionnaire are shown in Table 10.1. These early results are positive, especially considering the level of students' interest in the field of computer and network security before trying the game and the increased interest after playing the game. There remains room for improvement within the game's user interface, graphics, and response time of the game. We plan to address these areas and build a better experience for the user in the next development cycle.

Table I0.1 Average Response from 15 Students from Player 14-18 years of age

Computer & Network Security Game Survey	Scale from 0-5 (0=No Interest or Poor to 5=Very Good or Great)
Your interest in the computer & network security field before the game.	2.7
Your interest in the computer & network security field after the game.	4.3
Your level of interest in interactive games.	3.9
I would continue to use this game.	4.0
Quality of the computer and network security content.	3.4
Quality of the user interface and graphics.	3.1
I was satisfied with the response time of the game.	3.8
The overall challenge of the game's levels was adequate.	4.2
I found the education and hint sections helpful.	4.1
I learned from each level played.	4.2

A combination of ten teachers who instruct computing courses for students within the targeted age range were also asked to play the game and provide evaluation and feedback. The average responses are summarized in Table 10.2. Results illustrate teachers of the targeted age range would use the computer and network security game as a learning tool within their computing and networking courses.

Average scores for both student and faculty/teachers were positive, however, improvement in areas such as the game's graphical content and overall response time is needed. Open-ended responses from teachers described how they enjoyed the safe environment of the game for the students to play and the education sections within each module, however, many had concerns regarding the lack of computer and network security coursework content within their schools. In essence, they feel the game provides a good introduction, awareness, and platform for students to learn about computer and network security, yet most schools do not have textbooks or other coursework to help reinforce these concepts. Another concern was the training or education of the teachers in the continuously evolving subject of computer and network security. Some teachers felt they did not have adequate knowledge within the game's subject areas if the students had follow-up questions. Each of these two concerns are addressed in the next section.

Table I0.2 Average Response from 10 Teachers Instructing Computing Courses for Targeted Age Range

Computer & Network Security Game Survey	Scale from 0-5 (0=No Interest or Poor to 5=Very Good or Great)
Quality of the game content for computer and network security.	4.1
Quality of the User Interface and graphics.	3.3
Your satisfaction with the response time of the game.	3.1
The overall challenge of the game's levels was adequate.	4.1

I feel the education and hint sections were helpful.	4.8
The education section was useful in my teaching of the respective computer and network security subjects.	4.3
I would use this interactive game as a learning tool for my students.	4.3
I enjoyed using this tool.	4.6

11. CONCLUSIONS AND FUTURE WORKS

Our objective of creating a prototype interactive computer and network security game, which attempted to captivate students while providing rich-education through a fun experience, was met with positive and encouraging results. Both targeted students and teachers who instruct computer-related courses and found the game overall beneficial and would use it as a learning tool for computer and network security education. The major improvements needed consist within the graphical user-interface and enhancement of the response-time of the game. We also plan to explore the development of supplemental coursework in this area for teachers to add to their existing curriculums helping to better meet the teacher's needs while producing content to help reinforce the concepts learned within the game.

Moving forward we will continue to make improvements with the game and continue to test both students and teachers. We also plan to extend the storyline and create additional modules containing more computer and network security educational tools, while we look at options of deploying the produce in a cloud system capable of hosting the virtual machines. Overall, we are pleased with the game's progress and its potential to help educate students in this critical field, and we are excited to continue its development to offer a unique security education platform in the area of system and network administration.

REFERENCES

CHAPMAN, P., BURKET, J. and BRUMLEY, D. "PicoCTF: A Game-Based Computer Security Competition for High School Students". 2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14), San Diego, CA. 2014. USENIX Association.

CHECK POINT SOFTWARE TECHNOLOGIES LTD. "The Risk of Social Engineering on Information Security: A Survey of IT Professionals". Dimensional Research. September 2011. www.checkpoint.com/press/downloads/social-engineering-survey.pdf

CISCO SYSTEMS,INC. "Cisco 2014 Annual Security Report", Cisco Systems, Inc. January 2014. https://www.cisco.com/Web/offer/gist_ty2_asset/Cisco_2014_ASR.pdf

CSOMY, L. "Careers in the growing field of information technology services," Beyond the Numbers: Employment & Unemployment, Vol. 2, No. 9 (U.S. Bureau of Labor Statistics, April 2013), http://www.bls.gov/opub/btn/volume-2/careers-in-growing-field-of-information-technology-services.htm

HACKER HIGH SCHOOL - ISECOM. "About Hacker High School: http://www.hackerhighschool.org/about.html, 2000 [Dec. 12, 2013].

JORDAN, C., KNAPP, M. AND MITCHELL, D. "Countermeasures: An Interactive Game for Security Training" http://Web.cs.wpi.edu/~claypool/mqp/counter-measures/final-paper.pdf, March 2011 [November, 2013].

MARTIN, J. "NoVNC" http://kanaka.github.io/noVNC/ [Dec. 12, 2013].

PHP VIRTUAL BOX - imoore76 "phpVirtualBox" http://sourceforge.net/p/phpvirtualbox/wiki/Home/ [Dec. 12, 2013].

RealVNC - RealVNC.com http://www.realvnc.com/, 2002 [Dec. 12, 2013].

SRIKWAN, S. "Using Cartoons to Teach Internet Security" http://markus-jakobsson.com/papers/jakobsson-cryptologia08.pdf, July 2007 [Dec. 12, 2013].

VMware WSX - VMware "VMware Workstation 9.0.1 Release Notes" https://www.vmware.com/support/ws90/doc/workstation-901-release-notes.html, Nov 2012 [Dec. 12, 2013].

VMware vSphere Web client - VMware "Install and Start the vSphere Web Client" http://pubs.vmware.com/vsphere-50/index.jsp?topic=%2Fcom.vmware.vsphere.install.doc_50%2FGUID-74AA3EF1-BDF3-4752-89DB-A522CDE30A66.html [Dec. 12, 2013].

Wang, X., Hembroff, G. C., Bai, Y., 2013 USENIX Summit for Educators in System Administration (SESA '13), "ITSEED: Development of Instructional Laboratories for IT Security Education" USENIX, the Advanced Computing Systems Association, Washington, D.C., USA. (November 5, 2013).