Phishing Audit System

By

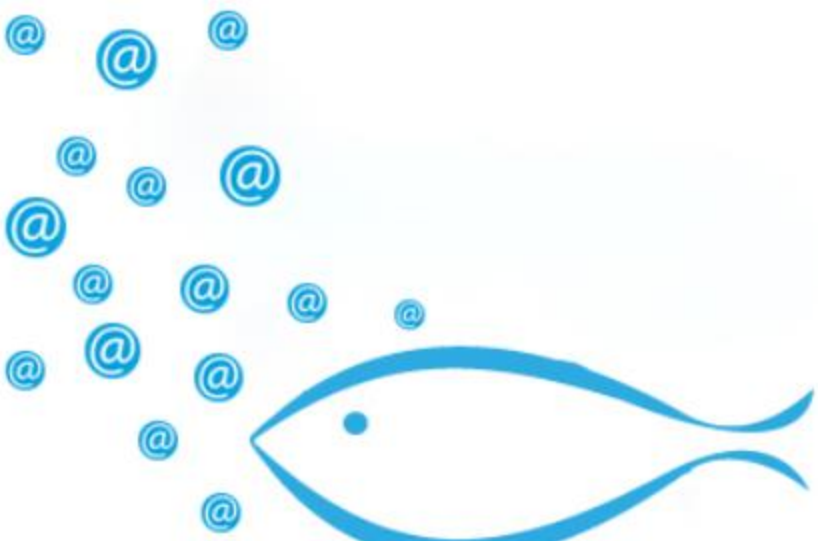
James Combs, Dakota Hampton, Joshua Woody, Tyler Stevens

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

James Combs	***4/15/2019***
James Combs	Date
Dakota Hampton	***4/15/2019***
Dakota Hampton	Date
Tyler Stevens	***4/15/2019***
Tyler Stevens	Date
Joshua Woody	***4/15/2019***
Tyler Stevens	Date
Tyler Hopperton	***4/30/2019***
Tyler Hopperton, Faculty Advisor	

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2019



Phishing Audit System

Prepared by
James Combs, Dakota Hampton, Joshua Woody, and Tyler Stevens

Students of
University of Cincinnati
College of Education, Criminal Justice, and Human Services
School of Information Technology

TABLE OF CONTENTS

List of Figures

Item **Page**

ACRONYMS AND ABBREVIATIONS

RAM Random Access Memory
API Application Programming Interface
MVC Model-View-Controller
URL Uniform Resource Locator

ABSTRACT

The use of Phishing Audit System has the potential to train and educate organizations and their employees to be more proactive when dealing with phishing emails. Phishing is the leading cause of cyber-attacks around the world and is increasing each year. Phishing comes in the form of an email that is made to look like it's coming from a trusted source but contains malware or takes you to an illegitimate website to steal personal information. Our solution to reduce the number of attacks is to create the Phishing Audit System. This system creates a customized, scheduled, and automated phishing simulation. This allows you to track the progress of users understanding and interacting with phishing emails. Additionally, this system provides users with the necessary training material to identify the phishing attacks.

INTRODUCTION

Problem Statement

Although various solutions exist to filter out malicious (phishing) emails before they reach individuals within an organization, training users to determine the legitimacy of an email is still necessary. Many organizations today utilize phishing audits (or simulations) to test people's ability to identify an email with a malicious link or attachment. Conducting these audits is generally very hands-on, time consuming, and not data-driven. Furthermore, many users aren't given proper training on the characteristics of malicious emails. That is why according to the InfoSec Institute that phishing and malware attacks have been the most prevalent form of cybercrime for eight years in a row now. Which cost companies millions of dollars and have become increasingly more expensive each year.

Project Description

Our solution is to create a system that can automate, schedule, customize, and track the progress of phishing simulations within an organization. The front-end of the system would be a user-friendly web interface where IT professionals can view the results of phishing simulations and customize various aspects around how/when they are conducted.

In addition to this, the system will be capable of providing users with training material to properly identify malicious emails and test to confirm their knowledge should they fail an audit. We will be able to track the risk of people's susceptibility to phishing attacks, as well as show the percentage of companies risk to attacks. We can then compare those numbers before and after the training to show the improvement after the training.

Solution

What sets our product apart from others currently out there is the ability for an admin to look at the results of the users interacting with the email. We know that some users will be more susceptible to phishing emails than others. With this in mind our product will allow the admin to customize emails depending on the user's tracked susceptibility. If the users are missing low level phishing emails this will be a clear sign that they need training right away. Once they receive this training and pass the required modules, they can be tested further with increasingly harder emails. If users understand that they are phishing emails and don't interact with them. The admin will then be able to customize emails that are harder to spot such as spear phishing. This customization will allow users to fully understand phishing and educate them to be ready for future attacks.

User Profile

Figure 1: The User Profile details the potential users of the Phishing Audit System. Users will likely be IT professionals with basic technical knowledge. Through our system, users will be able to schedule, target, and customize automated phishing audits. Furthermore, users will be able to customize phishing curriculum for their organization.

PROJECT: Phishing Audit System
POTENTIAL USERS: - IT Professionals/Auditors - Employees or users targeted by the IT Professionals
SOFTWARE, INTERFACE, AND RELATED EXPERIENCE: This project will be primarily targeted towards any IT professionals/auditors. These IT professionals will most likely hold a higher position in the company that will give them the power to select their users and view results.

This project will also be targeted towards any employees/users that the auditors decide on. They will have the capability to choose who their target users are based on security needs.
EXPERIENCE WITH SIMILAR APPLICATIONS: - GoPhish - PhishMe
TASK EXPERIENCE: Using the web interface to schedule, and manage phishing audits
FREQUENCY OF USE: Whenever the auditors need to schedule or manage their phishing audits. They have the capability to use the system as frequently as they deem necessary.
KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS: - Visually fluid and easy to navigate UI - Phishing email examples - Quick start/setup

Figure 1. User Profile

Use Case Diagram

Figure 2: The Use Case Diagram indicates the various tasks each user(s) of the Phishing Audit System will be performing. Each task is separated by the interface users will utilize, e.g. email browsers, the audit website, and the training website.

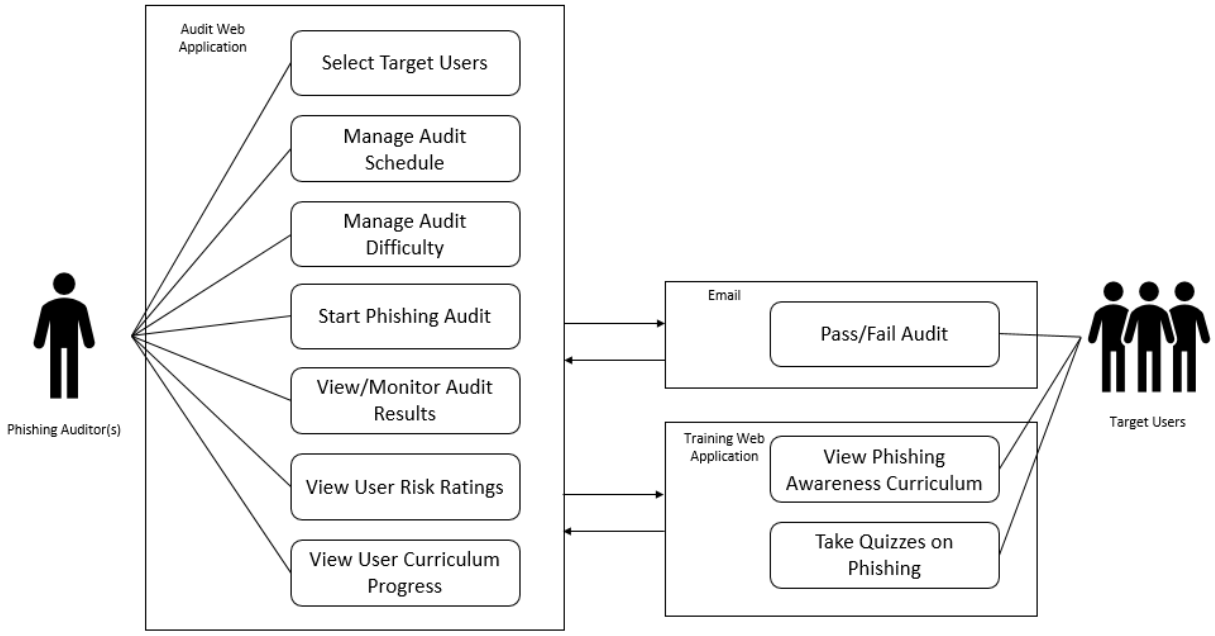


Figure 2. Use Case Diagram

PROJECT MANAGEMENT

Budget

Mostly everything that is being utilized in this project is open-source, currently owned, or provided by the University of Cincinnati. The total money that was used to create this project was $75 while not including labor costs. We spent $65 on servers to support our training program website. At the current moment in the project, there will not be the need to purchase any material to complete this project. Since we are not getting paid and this is for school, labor costs will be calculated if our rate was at $15 an hour. To calculate, we have summed the number of weeks for both semesters for a total of 32 weeks. We averaged the amount of time spent each week between the 4 group members at 20 hours a week. Following these calculations, the estimated labor costs would be $19,200. We break this down below in our Budget Table.

Item	Estimated Cost	Actual Cost
Hardware	$2,000.00	$65.00
Software	$0.00	N/A
Presentation	$20.00	$10.00
Labor	$19,200.00	$0.00
Total	**$21,200.00**	**$75.00**

Figure 10: Budget

Design Objectives/Deliverables

Figure 5: Design Objectives display the core components of the Phishing Audit System that will be present in the final product.

Objective	Description	Category
User-Friendly Front End	An easy-to-use front-end website that can be used to schedule phishing audits.	Front-end Development
User Risk Metrics	Target users will have a tracked risk rating depending on how often they fail phishing audits.	Back-end Development
Risk-Based Phishing Audits	Phishing audits will be able to be automated based on the risk factor of specific users.	Front-end Development
Security	The latest security measures will be applied to the application to maintain the integrity of our system.	Networking
Unique Phishing Curriculum	A complete phishing curriculum that can be personalized for a specific organization and users.	Front-end Development
Phishing Videos and Training	Videos, training material, and quizzes to provide users with the latest phishing methods. While also providing information to combat it and eliminate user interaction with them.	Front-end Development
Data Collection	Real time data collection to provide target users with phishing emails and who don't.	Back-end Development

Data Analytics	Data used to provide scoring system based on the target users outcome when interacting with the phishing email. Score based on the difficulty of the emails sent and users interaction.	Back-End Development
Data Archival	Historical records of past phishing audits and user performance will be available for analysis.	Back-End Development
Variety of phishing attacks	Develop a variety of different phishing attacks ranked by difficulty.	Front-end Development
Email Alerts	Email alerts that will go out when a user has hit a certain risk level based on our risk factor audits	Front-end Development
Active Directory Compatibility	Ability to link product to active directory to pull target user data	Networking

Figure 5: Design Objectives

Design Requirements

We decided to use an agile software development approach for our Phishing Audit System. We came up with a list of objectives we wanted to accomplish during this project. That way we knew exactly the highest priority objectives to finish first. We would work on individual objectives and then meet during the week to conduct compatibility and testing.

Procedures

We would split up the work according to our skill sets. Developers worked with ASP.Net to develop the front end of our system. Then used java script along with ASP.Net to finish the backend. Others worked on email templates and the calendar using HTML and JavaScript. We used an ubuntu server to host our Moodle site. Then created the curriculum content and configured the site so users could complete the necessary quizzes. Lastly we then hosted it so it could be accessible to outside parties.

Project Schedule

Figure 3.1-3.3: The Project Gant chart shows our project schedule, including major milestones, through the Fall and Spring semester.

Figure 4: The Project WBS shows a simplified breakdown of our project timeline.

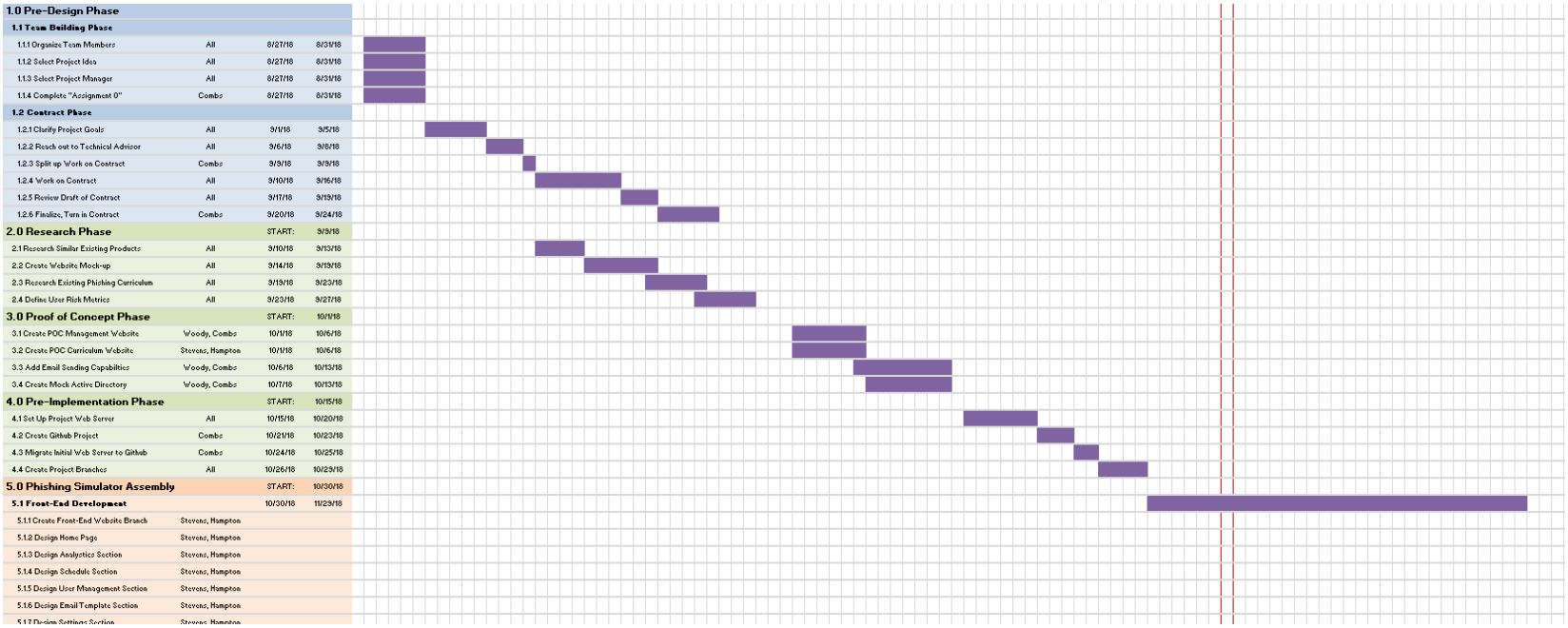


Figure 3.1: Project Gant

5.2.7 Merge Branch with Primary	Woody, Combs		
6.0 Phishing Curriculum Assembly		11/12/18	11/29/18
6.1 Initial Research	Stevens, Hampton	11/12/18	11/18/18
6.1.1 Find most common types of attack:	Stevens, Hampton		
6.1.2 formulate best way to identify	Stevens, Hampton		
6.1.3 research spear phishing techniqu	Stevens, Hampton		
6.1.4 research social engineering tactic	Stevens, Hampton		
6.2 Formulate Curriculim	Stevens, Hampton	11/19/18	11/29/18
6.2.1 Make diagrams	Stevens, Hampton		
6.2.2 Make animations	Stevens, Hampton		
6.2.3 Record videos	Stevens, Hampton		
6.2.4 Make test questions	Stevens, Hampton		
6.2.5 Organize into moduels	Stevens, Hampton		
6.2.6 Create completion tracker	Stevens, Hampton		
7.0 Integration Phase		START:	12/3/18
7.1 Integrate Back-end and Front-end Functionality		12/3/18	12/21/18
7.1.1 Email Controller	Woody, Combs		
7.1.2 Risk Tracker	Stevens, Hampton		
7.1.3 Analytics	Woody, Combs		
7.1.4 Template Manager	Stevens, Hampton		
7.1.5 Settings	Woody, Combs		
7.2 Front-end Smoke Testing	All	12/17/18	12/21/18
7.3 Push to Finalized Branch	All	12/17/18	12/21/18
7.4 Review Finalized Branch Functionalit	All	12/17/18	12/21/18
8.0 Testing, Patching Phase		START:	1/7/19
8.1 Integration Testing	All	1/7/19	1/14/19
8.1.1 Required Patching	N/A		
8.2 System Testing	All	1/14/19	1/21/19
8.2.1 Required Patching	N/A		
8.3 Acceptance Testing	All	1/21/19	1/28/19
8.2.1 Required Patching	N/A		

Figure 3.2: Project Gant Continued

10.0 Project Management		START:	8/27/18
10.1 Assignment 0 Submission	Combs	9/3/18	9/3/18
10.2 Team Contract Submission	Combs	9/24/18	9/24/18
10.3 Project Abstract Prep	All	9/25/18	10/4/18
10.4 Project Scope Rework	All	10/5/18	10/14/18
10.5 Project Abstract Submission	Combs	10/15/18	10/15/18
10.6 Team Contract Resubmission	Combs	10/15/18	10/15/18
10.7 User Profile Report Work	All	10/16/18	10/21/18
10.8 User Profile Table Work	All	10/16/18	10/21/18
10.9 User Case Diagram Work	All	10/16/18	10/21/18
10.10 User Profile Report & Table Submis:	Combs	10/22/18	10/22/18
10.11 User Case Diagram Submission	Combs	10/22/18	10/22/18
10.12 Eleavator Speech Prep	All	10/16/18	10/21/18
10.13 Elevator Speech	All	10/22/18	10/22/18
10.14 Draft Report Review	All	10/24/18	11/4/18
10.15 Draft Report Submission	Combs	11/5/18	11/5/18
10.16 Final Report Review	All	11/12/18	11/25/18
10.17 Final Report Submission	Combs	11/26/18	11/26/18
10.18 Fall Presentation Prep	All	11/7/18	11/18/18
10.19 Fall Presentation	All	11/19/18	11/19/18
10.20 Spring Presentation Prep	All		
10.21 Spring Presentation	All		
10.22 Final Report Draft Assembled	Combs	4/10/19	4/10/19
10.23 Final Report Draft Review	All	4/11/19	4/15/19
10.24 Final Report Submission	Combs	4/15/19	4/15/19
10.25 Tech Expo	All	4/9/19	4/9/19
		START:	2/7/19

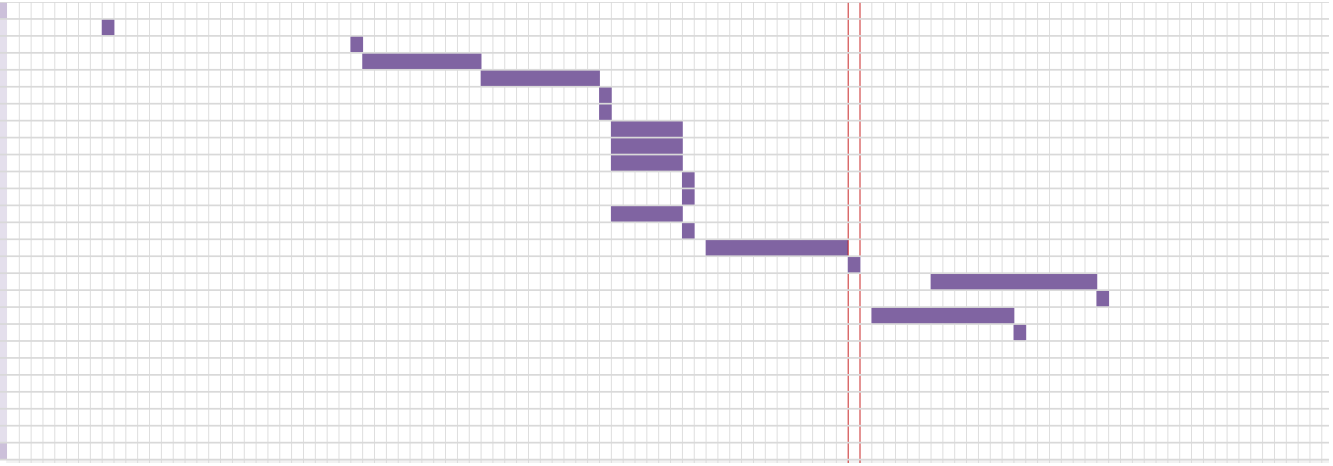


Figure 3.3: Project Gant Continued

Phishing Audit System WBS

TASK	ASSIGNED TO	START	END
1.0 Pre-Design Phase			
1.1 Team Building Phase			
1.1.1 Organize Team Members	All	8/27/18	8/31/18
1.1.2 Select Project Idea	All	8/27/18	8/31/18
1.1.3 Select Project Manager	All	8/27/18	8/31/18
1.1.4 Complete "Assignment 0"	Combs	8/27/18	8/31/18
1.2 Contract Phase			
1.2.1 Clarify Project Goals	All	9/1/18	9/5/18
1.2.2 Reach out to Technical Advisor	All	9/6/18	9/8/18
1.2.3 Split up Work on Contract	Combs	9/9/18	9/9/18
1.2.4 Work on Contract	All	9/10/18	9/16/18
1.2.5 Review Draft of Contract	All	9/17/18	9/19/18
1.2.6 Finalize, Turn in Contract	Combs	9/20/18	9/24/18
2.0 Research Phase		START:	9/9/18
2.1 Research Similar Existing Products	All	9/10/18	9/13/18
2.2 Create Website Mock-up	All	9/14/18	9/19/18
2.3 Research Existing Phishing Curriculum	All	9/19/18	9/23/18
2.4 Define User Risk Metrics	All	9/23/18	9/27/18
3.0 Proof of Concept Phase		START:	10/1/18
3.1 Create POC Management Website	Woody, Combs	10/1/18	10/6/18
3.2 Create POC Curriculum Website	Stevens, Hampton	10/1/18	10/6/18
3.3 Add Email Sending Capabilities	Woody, Combs	10/6/18	10/13/18
3.4 Create Mock Active Directory	Woody, Combs	10/7/18	10/13/18
4.0 Pre-Implementation Phase		START:	10/15/18
4.1 Set Up Project Web Server	All	10/15/18	10/20/18
4.2 Create Github Project	Combs	10/21/18	10/23/18
4.3 Migrate Initial Web Server to Github	Combs	10/24/18	10/25/18
4.4 Create Project Branches	All	10/26/18	10/29/18
5.0 Phishing Simulator Assembly		START:	10/30/18
5.1 Front-End Development		10/30/18	11/29/18
5.1.1 Create Front-End Website Branch	Stevens, Hampton		
5.1.2 Design Home Page	Stevens, Hampton		
5.1.3 Design Analystics Section	Stevens, Hampton		
5.1.4 Design Schedule Section	Stevens, Hampton		
5.1.5 Design User Management Section	Stevens, Hampton		
5.1.6 Design Email Template Section	Stevens, Hampton		
5.1.7 Design Settings Section	Stevens, Hampton		
5.1.8 Review Front-End Layout	All		
5.1.9 Merge Branch with Primary	Stevens, Hampton		

5.2 Back-End Development		10/30/18	11/29/18
5.2.1 Build Email Controller	Woody, Combs		
5.2.2 Build Risk Tracker	Woody, Combs		
5.2.3 Build Template Controller	Woody, Combs		
5.2.4 Build Analytics Controller	Woody, Combs		
5.2.6 Review Back-End Functionality	All		
5.2.7 Merge Branch with Primary	Woody, Combs		
6.0 Phishing Curriculum Assembly		11/12/18	11/29/18
6.1 Initial Research	Stevens, Hampton	11/12/18	11/18/18
6.1.1 Find most common types of attacks	Stevens, Hampton		
6.1.2 formulate best way to identify	Stevens, Hampton		
6.1.3 research spear phishing techniques	Stevens, Hampton		
6.1.4 research social engineering tactics	Stevens, Hampton		
6.2 Formulate Curriculim	Stevens, Hampton	11/19/18	11/29/18
6.2.1 Make diagrams	Stevens, Hampton		
6.2.2 Make animations	Stevens, Hampton		
6.2.3 Record videos	Stevens, Hampton		
6.2.4 Make test questions	Stevens, Hampton		
6.2.5 Organize into moduels	Stevens, Hampton		
6.2.6 Create completion tracker	Stevens, Hampton		
7.0 Integration Phase		START:	12/3/18
7.1 Integrate Back-end and Front-end Functionality		12/3/18	12/21/18
7.1.1 Email Controller	Woody, Combs		
7.1.2 Risk Tracker	Stevens, Hampton		
7.1.3 Analytics	Woody, Combs		
7.1.4 Template Manager	Stevens, Hampton		
7.1.5 Settings	Woody, Combs		
7.2 Front-end Smoke Testing	All	12/17/18	12/21/18
7.3 Push to Finalized Branch	All	12/17/18	12/21/18
7.4 Review Finalized Branch Functionality	All	12/17/18	12/21/18
8.0 Testing, Patching Phase		START:	1/7/19
8.1 Integration Testing	All	1/7/19	1/14/19
8.1.1 Required Patching	N/A		
8.2 System Testing	All	1/14/19	1/21/19
8.2.1 Required Patching	N/A		
8.3 Acceptance Testing	All	1/21/19	1/28/19
8.2.1 Required Patching	N/A		
10.0 Project Management		START:	8/27/18
10.1 Assignment 0 Submission	Combs	9/3/18	9/3/18
10.2 Team Contract Submission	Combs	9/24/18	9/24/18
10.3 Project Abstract Prep	All	9/25/18	10/4/18
10.4 Project Scope Rework	All	10/5/18	10/14/18
10.5 Project Abstract Submission	Combs	10/15/18	10/15/18
10.6 Team Contract Resubmission	Combs	10/15/18	10/15/18

10.7 User Profile Report Work	All	10/16/18	10/21/18
10.8 User Profile Table Work	All	10/16/18	10/21/18
10.9 User Case Diagram Work	All	10/16/18	10/21/18
10.10 User Profile Report & Table Submission	Combs	10/22/18	10/22/18
10.11 User Case Diagram Submission	Combs	10/22/18	10/22/18
10.12 Eleavator Speech Prep	All	10/16/18	10/21/18
10.13 Elevator Speech	All	10/22/18	10/22/18
10.14 Draft Report Review	All	10/24/18	11/4/18
10.15 Draft Report Submission	Combs	11/5/18	11/5/18
10.16 Final Report Review	All	11/12/18	11/25/18
10.17 Final Report Submission	Combs	11/26/18	11/26/18
10.18 Fall Presentation Prep	All	11/7/18	11/18/18
10.19 Fall Presentation	All	11/19/18	11/19/18
10.20 Spring Presentation Prep	All		
10.21 Spring Presentation	All		
10.22 Final Report Draft Assembled	Combs	4/10/19	4/10/19
10.23 Final Report Draft Review	All	4/11/19	4/15/19
10.24 Final Report Submission	Combs	4/15/19	4/15/19
10.25 Tech Expo	All	4/9/19	4/9/19

Figure 4: Work Breakdown Structure

Technical Elements

Audit web application

The audit application will be written in C# using ASP.net. We will be running it on an IIS server. All user data will be serialized and held within the application. The audit system itself will be isolated from the training web application. When needed, the audit system will link users to the training web application.

Training web application

We will be utilizing the open source class manager Moodle. It will be hosted on an Ubuntu server using apache. The content for the curriculum will be created within the Moodle application.

Problems Encountered

The first problem we encountered was when we started setting up the training server it functioned very slow. To fix this we replaced the RAM and added more to increase the speed of the server. Another problem we faced was when tried to connect to the server remotely, we kept getting a connection error. We ended up having to reconfigure Apache to fix this problem. The last problem we encountered was when uploading the training and module files to the server. We had to go into the php.cfg files and change some setting to allow them to be uploaded further.

TECHNICAL DISCUSSION

Technical Diagram

Figure 6.0 represents the interaction between both our Phishing Audit System and our training system. Our solution functions through user interaction with both audit and training systems. The audit system manages the scheduling, customization, and results of phishing audits. The training system leverages data about user risk-rating and audit performance provided by the Phishing Audit System. The training system then provides users with detailed curriculum and testing to ensure their ability to pass future audits.

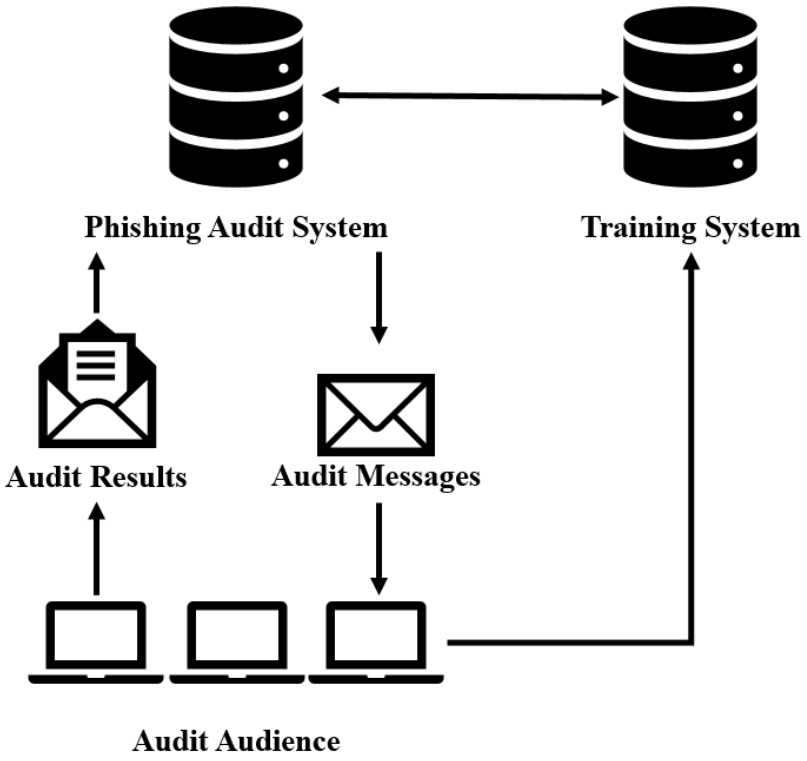


Figure 6.0 *– Technical Diagram*

User Training Dashboard

Figure 7.0 provides a snapshot of what individual users see when logged into our training system. Depending on the training program they are enrolled in, various training courses and quizzes will be available for use.

Figure 7.0 *– Training Dashboard*

Audit System Schedule

Figure 8 shows the schedule page audit administrators will use to schedule, customize, and view the results of phishing audits.

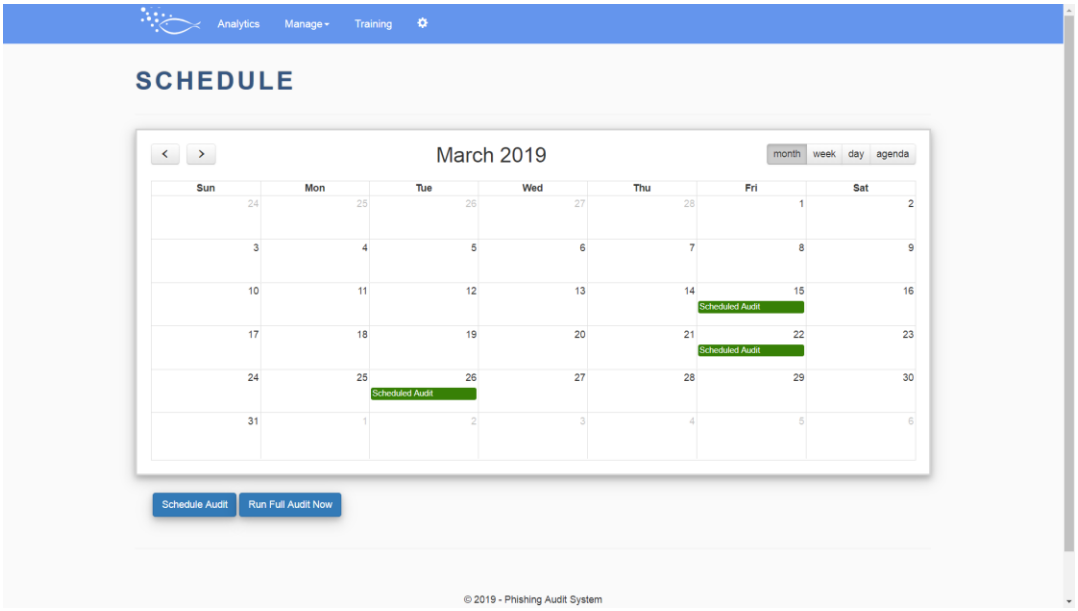


Figure 8.0 – *Audit Scheduling Dashboard*

TESTING OVERVIEW

This section will explain the testing methodology for the Phishing Audit System and should be used as a guide.

Scope

The Scope of testing the operation of the Phishing Audit System to make sure that emails are being sent and received, risk rating are being tracked and recorded, the application and testing website can be accessed on different browsers, new users are able to enroll in the course, audits can be scheduled/customized in advance.

Objectives

The objective of testing is to verify that both the audit system and training course can be accessed via URL. Tests will be carried out on the training course to make sure they complete the training modules in succession. To ensure that the users' performance throughout audits is reflected accurately by their risk rating. Testing will be carried out by the appropriately assigned developers.

Logging Test and Reporting

If an issue is found during testing then, the developer working will create a GitHub issue, a label it depending on its severity. During weekly meeting the team will review the issues and assign it to the relevant developer.

Test Methodology

1. User Interface – This test includes making sure our analytics page displays the risk score and the day audits were scheduled. Displays the correct month on the schedule page, as

well making sure the buttons work to check previous months audits. Buttons work when adding or deleting users. Navigating to the training course is accessible by clicking on the training button.

2. Functionality – This test makes sure when a user fails an audit their risk score changes. Testing that we can pick and send out specific email templates to users. Scheduling an audit for a future date and time. When a user passes a quiz, it checks it as complete and moves them onto the next module.
3. Consistency Test – This test ensures our system will load successfully. Testing to ensure audits that are scheduled ahead of time have the correct template and data assigned.

Test Plan/Results

Procedure	Duration of Test	Pass/Fail	Bugs
Training Test – accessible from the web, checks off modules when completed, cant access other modules until previous one is passed	2/6/2019 – 4/9/2019	Pass	N/A
Audit Test – Ensure that we can send out emails to specified user category	2/6/2019 – 4/9/2019	Pass	N/A
Phishing Email Test – Users receive the emails, the correct email template is sent out to users	2/6/2019 – 4/9/2019	Pass	N/A

Analytics Test – Risk score graph will update once users interact with email, displays the date and number of users who failed during that audit	2/6/2019 – 4/9/2019	Pass	Wouldn't update at first but resolved the issue
Audit Schedule Test – Starts an audit on the date and time specified, can check out previous months audits and number of users failed, audit button works and schedules audit	2/6/2019 – 4/9/2019	Pass	N/A
Audit Risk Rating Test – Ensures once a user's clicks on the phishing email their risk score changes, the risk score is represented on the analytics page	2/6/2019 – 4/9/2019	Pass	N/A

Figure 9.0 *– Testing Procedures and Results*

Problems Faced/Learned

Since beginning testing the Phishing Audit System we have not faced anything with severity high enough to report. Through development bugs have been handled by individual developers prior to their work being combined with others.

CONCLUSION

Spring Semester

Over the two semesters working on this project our team has applied skills gained over years of learning at this University. We created a system we believe is an effective tool to counter one of the largest threats to cybersecurity today. We are proud of our product, the Phishing Audit System, and we believe similar solutions will pave the way towards a more secure cyberspace in the future. While our solution could not leverage the resources a larger software organization would, we believe we applied ourselves well and our efforts are reflected in our project.

A recommendation for us if we did it over again would be to have better communication amongst team members. Our current communication is lacking a little bit because we were all working from the master branch in GitHub. This sometimes caused merge conflicts and although they were simple fixes, it would be beneficial to work off of our own branches in the future. If we had more time for this project we recommend spending more time on the marketing aspect of our project. We didn't spend nearly enough time making our poster, website, and booth stand out in the crowd and I think it hurt the way people viewed our project.

The main recommendation we received from others was to include a template editor, grouping feature of our users, and send data back from the training program to our admin dashboard. We currently have no future plans for this project.

APPENDIX A. ADDITIONAL INFORMATION

Poster

The image below is the layout of the poster we used to present our project at the 2019 IT Expo.

APPENDIX B. REFERENCES

[1] “Phishing is the Leading Cause of Cyber Attacks against Organizations”, InfoSec, Available: https://www.csriskmanagement.co.uk/phishing-is-the-leading-cause-of-cyber-attacks-against-organisations/