National College of Ireland

Bachelor of Science Honours in Computing

Cyber Security

Academic Year 2022/2023

Benn Miley

18414856

X18514856@studenst.ncirl.ie

Faux Phish

Technical Report

Contents

Executive Summary

This report provides an in-depth overview of the Faux Phish application developed as part of the BSHCYB4 4th Year Project module. The purpose of this report is to outline and describe the application, its significance, and the process followed for its development. The report serves as the final submission for the project.

The Faux Phish application aims to address the need for improved phishing awareness in companies by enabling them to conduct fake phishing campaigns. The application allows companies to register their email addresses and track the individuals who click on the fake phishing links. When a user clicks on a link, they are redirected to an educational page about the dangers of phishing campaigns.

Throughout the report, the project's background, objectives, and technology used are discussed. The system requirements, including functional, data, user, environmental, and usability requirements, are outlined. The design and architecture of the application are explained, highlighting the chosen technology stack and implementation process.

The graphical user interface (GUI) design considerations are described, focusing on enhancing user experience. The testing approach employed to ensure the application's reliability and functionality is detailed, along with the evaluation process to measure the effectiveness of the application.

Based on the findings and analysis, it can be concluded that the Faux Phish application successfully addresses the objectives of raising phishing awareness and providing companies with a tool for monitoring and educating their employees. The application has shown promising results in tracking phishing link clicks and redirecting users to educational content.

For further development or research, potential enhancements to the application's features, such as more advanced tracking and reporting functionalities, could be explored. Additionally, ongoing evaluation and user feedback collection are recommended to continually improve the effectiveness and usability of the application.

In conclusion, the Faux Phish application serves as a valuable tool for companies to enhance their cybersecurity stature by simulating phishing attacks and educating their employees about the risks. The report provides comprehensive insights into the application's development process, functionality, and future possibilities for improvement.

1.0 Introduction

1.1. Background

Phishing attacks have emerged as one of the most prevalent and damaging cybersecurity threats in recent years. According to Verizon's 2022 Data Breach Report, a staggering 82% of breaches involved the Human Element, highlighting the susceptibility of individuals to social engineering tactics. Additionally, CISCO's 2021 Cybersecurity Threat report indicates that phishing attacks account for over 80% of reported security incidents. These statistics highlight the significant danger and frequency of phishing attacks.

Recognizing the critical need to educate IT users about the dangers of phishing attacks and how to identify them, I decided to develop Faux Phish. As a computing student specializing in Cybersecurity, I saw an opportunity to contribute to mitigating the risks associated with phishing attacks.

1.2. Aims

The primary objective of Faux Phish is to raise awareness of organizations their employees about the threat posed by phishing attacks. By allowing companies to simulate fake phishing campaigns, the application aims to familiarize users with the tactics employed by attackers and provide them with practical knowledge to identify and avoid falling victim to such attacks.

Furthermore, Faux Phish empowers organizations to assess the effectiveness of their current cyber security measures. By tracking and recording clicks on fake phishing links, companies can gain valuable insights into the vulnerability of their systems and identify areas for improvement. This active approach toward greater cybersecurity posture equips organizations with the means to strengthen said posture and reduce the likelihood of successful phishing attacks.

1.3. Technology

To achieve the goals of the Faux Phish application, a combination of technologies were used, including Python, Python Flask, SQLAlchemy, Bitly, JavaScript, HTML, CSS and Visual Studio Code.

Visual Studio Code served as my IDE for developing the Faux Phish application. VS Code provided a robust platform for coding and debugging.

JavaScript (JS) was utilized to add interactivity and dynamic behaviour to the application's frontend. It facilitated dynamic content updates, in particular, it allowed users to delete emails from the address book.

HTML formed the backbone of the application, providing the content for the web pages.

Bootstrap, a popular open-source CSS framework, was utilized for the styling of the Faux Phish application. Bootstrap provided a range of pre-designed CSS components and responsive layout utilities, enabling consistent and visually appealing styling throughout the project.

Python served as the primary programming language for the Faux Phish application, handling server-side logic, data processing, and database interactions.

Python Flask, a lightweight web framework, was used to build the backend of the application. Flask provided tools for handling HTTP requests, routing, and server-side operations

SQLAlchemy, a Python SQL toolkit and ORM library, facilitated seamless interaction with the database. It offered a high-level interface for working with data and ensured efficient data management.

Bitly was integrated into the application to allow for tracking and recording clicks on the links, providing valuable analytics and user interaction monitoring.

Together, these technologies enabled the development of the Faux Phish application. By leveraging the strengths of these technologies, the Faux Phish application provides a platform for conducting fake phishing campaigns, educating users, and empowering organizations to enhance their cybersecurity posture.

1.4. Structure

The report is structured in a way that provides a comprehensive overview of the Faux Phish application, addressing various aspects of the project. It begins with an executive summary, which offers a concise summary of the report's purpose, major points, and key findings.

The introduction section sets the stage by explaining the background behind the project. It highlights the alarming statistics on the prevalence and impact of phishing attacks, emphasizing the need for awareness and education in combating this cybersecurity threat. The objectives and relevance of the Faux Phish application are outlined, emphasizing the significance of developing a tool to educate users about phishing attacks and help organizations assess their security measures.

Moving into the system section, perhaps the largest section of this report, this section delves into the requirements of the Faux Phish application. This includes exploring functional requirements, data requirements, user requirements, environmental requirements, and usability requirements. The design and architecture of the application are then discussed, with a focus on the GUI, the chosen technology stack. The implementation and finally testing approach.

Conclusions are drawn in a dedicated section, summarizing the key findings and outcomes of the project. The report reflects on the achievements of the Faux Phish application in addressing its objectives and offers insights into its effectiveness in educating users and improving organizational cybersecurity.

Further development or research opportunities are discussed, suggesting potential areas for enhancing the application's tracking and reporting functionalities or collecting ongoing user feedback.

The report concludes with a list of references, acknowledging the sources used throughout the document, in the style of Harvard Referencing, followed by the appendices section. The appendices contain supplementary information, such as an archive of my monthly reflective journals, project proposal and additional data, providing additional context and support for the project.

Through this structured approach, the report provides a comprehensive and organized overview of the Faux Phish application, covering its development process, functionality, evaluation, and potential for further improvement.

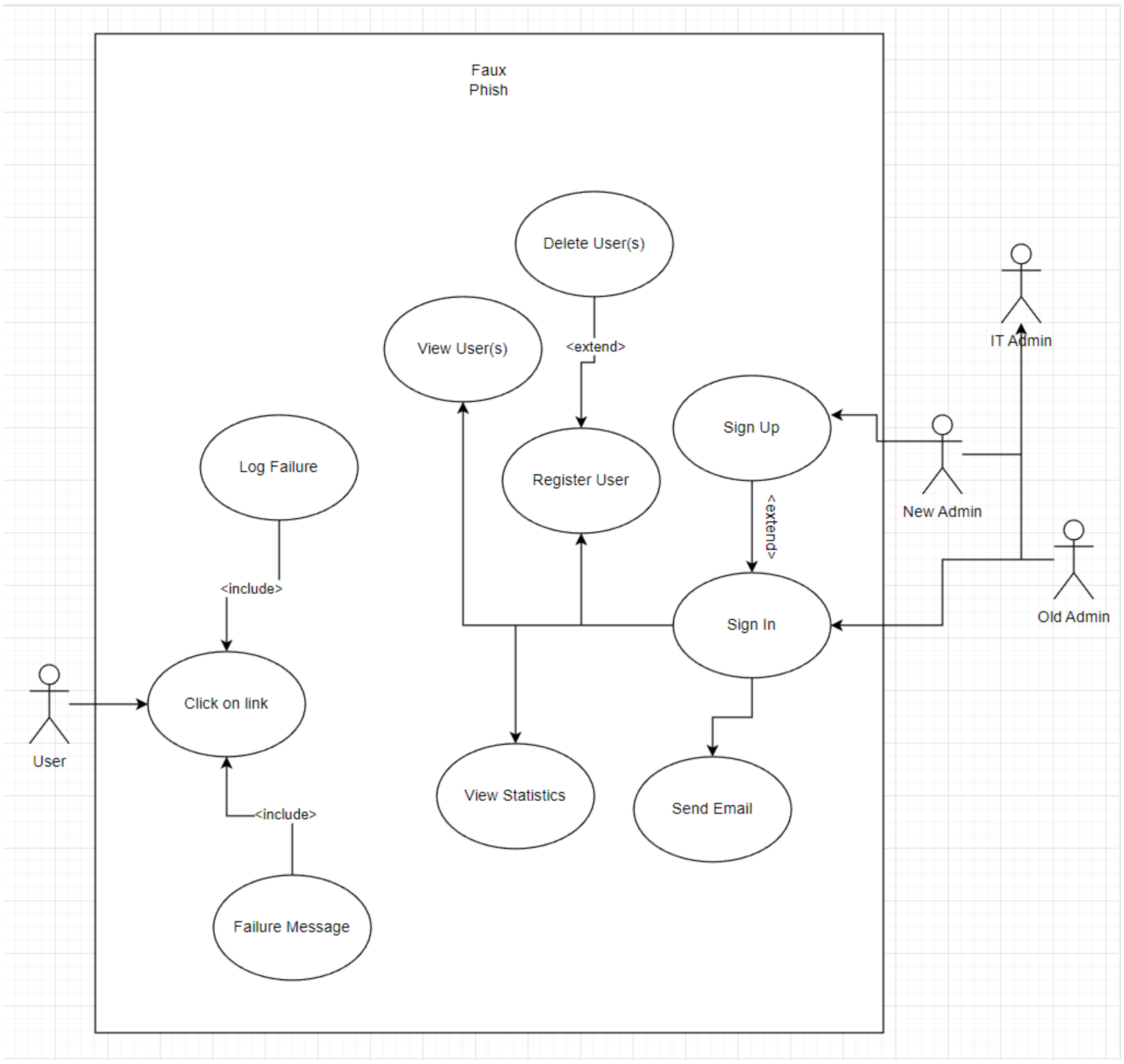
2.0 System

2.1. Requirements

In this section, I will outline the requirements of the Faux Phish application. These requirements are essential to ensure the functionality, usability, and effectiveness of the system. All requirements are verifiable and measurable to provide clear objectives for the development and evaluation of the application.

2.1.1. Functional Requirements

2.1.1.1. Use Case Diagram



2.1.1.2. Requirement 1 Admin Account Management

2.1.1.3. Description & Priority

The requirement for Admin Account Management is essential to the system as it enables the creation, login, and logout functionalities for IT administrators. It holds a high priority to ensure that authorized personnel can access and manage the system effectively.

2.1.1.4. Use Case

Use Case ID: UC- 1.0

Scope

The scope of this use case is to manage the account functionalities specifically for IT administrators.

Description

This use case describes the process of creating an admin account, logging into the account, and signing out from the account.

Flow Description

Precondition

The system is in the initialization mode, and the Sign Up interface is accessible.

Activation

This use case starts when an IT administrator selects the "Sign Up" option.

Main flow

1. The system identifies the request for account management and displays the sign up interface.
2. The IT administrator provides the necessary information to create an account, including email, password, and contact details.
3. The system validates the entered information and creates a new admin account.
4. The IT administrator can then log into their account by entering the email and password.
5. The system verifies the credentials and grants access to the admin dashboard.

Alternate flow

A1 : Login Failure

1. The system detects incorrect login credentials.
2. The IT administrator receives an error message indicating the invalid login attempt.
3. The use case continues at position 4 of the main flow.

Termination

The system presents the admin dashboard, providing access to the various features and functionalities available to the IT administrator.

Post condition

The system goes into a wait state, allowing the IT administrator to perform further actions within their account.

2.1.1.5. Requirement 2 Email Sending

2.1.1.6. Description & Priority

The Email Sending requirement is crucial to the functionality of the system as it enables the application to send phishing emails to recipients. Additionally, it assigns a unique

URL to each email and registers URL clicks for identification purposes. This requirement holds a high priority to ensure the effectiveness of the phishing campaign and tracking of recipient interactions.

2.1.1.7. Use Case

Use Case ID: UC- 2.0

Scope

This use case focuses on the process of sending emails, assigning unique URLs, and registering URL clicks within the Faux Phish application.

Description

This use case describes the steps involved in sending emails to recipients, assigning unique URLs to each email, and recording the click activity associated with those URLs.

Flow Description

Precondition

The system is in the active state, and the IT administrator is logged into their account.

Activation

This use case starts when an IT administrator initiates the email sending process.

Main flow

1. The IT administrator selects the recipients for the phishing email campaign and provides the necessary email content.
2. The system generates a unique URL for each email and embeds it within the email content.
3. The system sends the email to the respective recipients' email addresses.
4. The recipient receives the email and, if clicked, gets redirected to the warning.html page, where the recipient's identity is captured.
5. The system registers the URL click and associates it with the respective recipient's information

Termination

The email sending process is completed, and the system updates the campaign statistics and recipient click data.

Post condition

The system is ready for further actions within the admin dashboard

2.1.1.8. Requirement 3 Address Book Management

2.1.1.9. Description & Priority

The Address Book Management requirement is essential for the Faux Phish application as it enables IT administrators to manage email addresses within the address book. This functionality allows for adding, viewing, and removing email addresses from the address book. The priority of this requirement is considered medium, as it contributes to the overall usability and organization of recipient lists for phishing campaigns.

2.1.1.10. Use Case

Use Case ID: UC- 3.0

Scope

This use case focuses on the process of managing email addresses within the address book, including adding, viewing, and removing addresses.

Description

This use case describes the steps involved in adding new email addresses to the address book, viewing the existing addresses, and removing specific addresses from the address book.

Flow Description

Precondition

The system is in the active state, and the IT administrator is logged into their account.

Activation

This use case starts when an IT administrator selects to add to the address book on the admin dashboard.

Main flow

1. The IT administrator selects the option to add a new email address to the address book.
2. The IT administrator fills out and submits the form, and the system validates and adds the new email address to the address book.
3. The system retrieves and displays the list of email addresses currently stored in the address book.
4. The IT administrator selects the option to remove a specific email address from the address book.
5. The system removes the email address from the address book.

Exceptional flow

E1:Empty Email Address Field

1. If the email address field is empty, the system displays an error message indicating that the email address is required.
2. The use case returns to step 2, allowing the IT administrator to enter a valid email address.

Termination

The address book management process is completed, and the system updates the address book accordingly.

Post condition

The system is ready for further actions within the admin dashboard.

2.1.2. Data Requirements

The Faux Phish application requires specific data to support its functionalities. These data requirements include:

- User information: I will store and manage user profiles, including email addresses, roles, and permissions.
- Campaign data: I will capture details of phishing campaigns, such as campaign settings, target email addresses, and campaign results (e.g., click rates, timestamps).
- Educational content: I will maintain a database of educational materials and resources related to phishing attacks.

2.1.3. User Requirements

The Faux Phish application should meet the following user requirements:

- User-friendly interface: I will provide an intuitive and easy-to-navigate interface for administrators and employees to interact with the system.
- Customization options: I will allow administrators to customize campaign settings & email content to suit their organization's needs.
- Reporting and analytics: I will enable administrators to generate comprehensive reports on campaign performance, click rates, and user engagement.

2.1.4. Environmental Requirements

The Faux Phish application has certain environmental requirements to ensure its smooth operation:

- Supported platforms: The application should be compatible with commonly used web browsers, such as Google Chrome, Mozilla Firefox, and Microsoft Edge.
- Performance: The application should be responsive and performant, capable of handling multiple simultaneous users and processing campaign data efficiently.
- SFW: The application's contents must be safe and appropriate for a workplace environment.

2.1.5. Usability Requirements

The Faux Phish application should adhere to usability standards to enhance user experience and effectiveness:

- Intuitive navigation: I will provide clear and logical navigation paths within the application to allow users to easily access different features and functionalities.
- Error handling: I will display informative error messages when users encounter errors or input invalid data to guide them towards resolution.
- Responsive design: I will ensure the application's layout is scalable and its interface shall adapt to different screen sizes and devices.

2.2. Design & Architecture

The Faux Phish application follows an object oriented design, utilizing Python Flask as the framework for building the application's core functionality. The application adheres to an object-oriented approach, ensuring modularity, maintainability, and scalability. It employs the MVC design pattern to separate its aspects and promote code organization.

System Architecture: The system architecture of Faux Phish is comprised of many components, each playing a role toward contributing to the application's operation:

1. Presentation Layer (View): The presentation layer is responsible for delivering web pages to users. HTML is used to structure and present the pages, and Bootstrap is used to enhance the design and user experience.

2. Application Layer(Controller): The application layer contains the main logic of the Faux Phish application. Its comprised of various components, including controllers and utilities. The controllers, implemented in the **views.py** file, handle the user interactions, process requests, and orchestrate the appropriate actions. The **auth.py** file manages secure pages such as login and sign-up, ensuring proper authentication and authorization.

3. Data Layer (Model): The data layer leverages SQLAlchemy, an ORM library, to interact with the underlying database. SQLAlchemy allows seamless communication between the application and the database. It facilitates the storage and retrieval of data related to, email addresses, user information, and links that have been clicked.

Main Algorithms: The Faux Phish application incorporates the following algorithms:

1. Email Sending Algorithm: The application integrates with Gmail and utilizes Google's SMTP service through the Flask-Mail library. This algorithm follows the standard process of composing and sending emails using Gmail's SMTP server. It allows for customization of email content, attachment of files, and ensures reliable delivery of phishing simulation emails.

2. URL Generation Algorithm: To assign each email a unique URL, the application employs the **secrets.token_urlsafe(16)** function from the Python **secrets** module. This algorithm generates a secure URL-safe string of 16 characters, ensuring its unique. The generated URLs are embedded in the email content and facilitate the tracking of recipients who click on phishing links.

Routing and Design: The routing of web pages is primarily handled through the **views.py** file, which maps URLs to specific functions and views. Secure pages, such as login and sign-up, are managed separately in the **auth.py** file to enforce authentication and protect sensitive information.

HTML is utilized to structure and present the web pages. Bootstrap is utilized for designing and styling the pages, ensuring a responsive and visually appealing GUI.

This design and architecture of Faux Phish creates a solid foundation for the application. It allows for efficient routing, secure authentication, seamless database operations, and visually appealing page presentation.

2.3. Implementation

The implementation of Faux Phish involves the utilization of various algorithms, classes, and functions to achieve its functionality. The following are some of the key components and code snippets that highlight the implementation details:

Email Sending & Url Generation

```
@views.route('/send_email', methods=['GET', 'POST'])
def send_email():
    if request.method == 'POST':
        recipient = request.form['recipient']
        subject = request.form['subject']
        message = request.form['message']

        token = secrets.token_urlsafe(16)

        url = f"http://127.0.0.1:5000/warning/{token}"

        msg = Message(subject=subject, sender='x18414856@gmail.com',
recipients=[recipient])
        msg.body = message + f"Click the link to access the warning page:
{url}"
        mail.send(msg)

        clicked_email = ClickedEmail(user=current_user, email=recipient)
        db.session.add(clicked_email)
        db.session.commit()

        flash('Email Sent!', category='success')

    return render_template("home.html", user=current_user)
```

This code snippet represents the implementation of the email sending functionality in the Faux Phish application.

- A secure token is generated using **secrets.token_urlsafe(16)**. This token will be used to create a unique URL.
- The URL is constructed by adding the token to the base URL

- The Flask-Mail library is utilized to create an email message using the provided subject, sender, recipient, and body.
- The body of the email includes the original message and a clickable URL to access the warning page.

Sign-Up Functionality

```
@auth.route('/sign-up', methods=['GET', 'POST'])
def sign_up():
    if request.method == 'POST':
        email = request.form.get('email')
        first_name = request.form.get('firstName')
        last_name = request.form.get('lastName')
        organisation = request.form.get('organisation')
        password1 = request.form.get('password1')
        password2 = request.form.get('password2')

        user = User.query.filter_by(email=email).first()
        if user:
            flash('Email already exists.', category='error')
        elif len(email) < 4:
            flash('Email must be greater than 3 characters.',
category='error')
        elif len(first_name) < 2:
            flash('First name must be greater than 1 character.',
category='error')
        elif len(last_name) < 2:
            flash('Last name must be greater than 1 character.',
category='error')
        elif len(organisation) < 2:
            flash('Organisation name must be greater than 1 character.',
category='error')
        elif password1 != password2:
            flash('Passwords don\'t match.', category='error')
        elif len(password1) < 7:
            flash('Password must be at least 7 characters.', category='error')
        else:
            new_user = User(email=email, first_name=first_name,
last_name=last_name, organisation=organisation,
password=generate_password_hash(
                password1, method='sha256'))
            db.session.add(new_user)
            db.session.commit()
            login_user(new_user, remember=True)
            flash('Account created!', category='success')
            return redirect(url_for('views.home'))
```

- Inside the view function, the code checks if the request method is POST, indicating that the sign-up form data has been submitted.

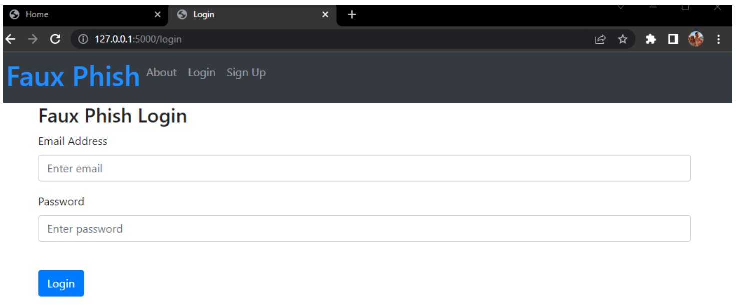
- Various user input fields such as email, first name, last name, organization, password1, and password2 are extracted from the form using the request.form.get() method.
- If all the validation checks pass, a new User object is created with the provided user details.
- The newly created user is automatically logged in using login_user from Flask-Login, with the remember parameter set to True.
- A flash message is displayed to indicate that the account has been created successfully.

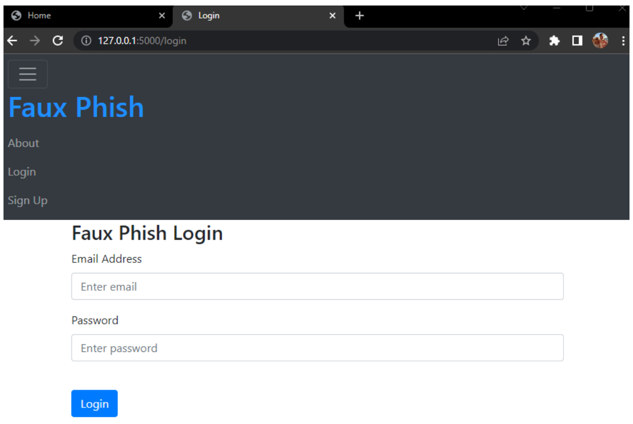
Some Extra Interesting Snippets:

```
app.config['SECRET_KEY'] = 'abcdefghijk lmnopqrst'
```

This line sets the secret key for my Flask Application. The secret key is an important configuration value used to secure the application. It is used for various security-related purposes, such as session management, signing cookies, and protecting against cross-site request forgery attacks.

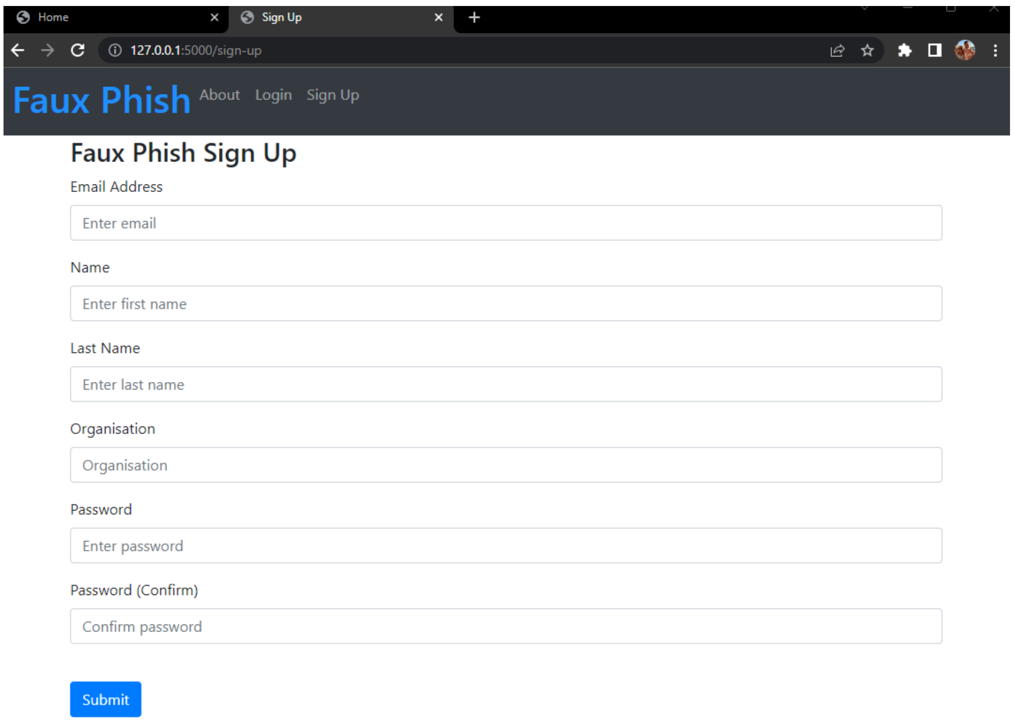
2.4. Graphical User Interface (GUI)





This is the GUI for the login page.

- Note, when shrunk, the about, login and sign up buttons collapse into a collapsible navbar.
- Also note, the navbar will differ from here to the admin dashboard, otherwise, it is the same throughout the application. The navbar can be found in the file base.html



Faux Phish Sign Up

Email Address

Enter email

Name

Enter first name

Last Name

Enter last name

Organisation

Organisation

Password

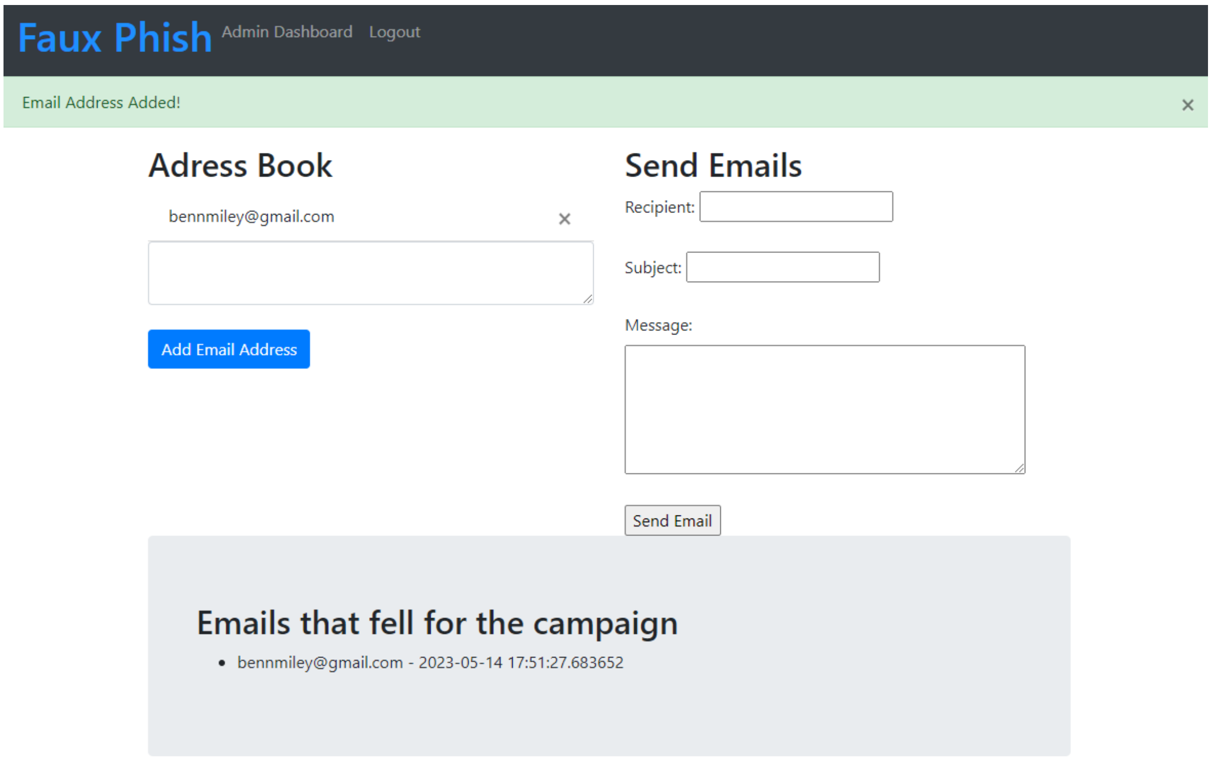
Enter password

Password (Confirm)

Confirm password

Submit

This is the GUI for the sign up page



Email Address Added!

Adress Book

bennmiley@gmail.com

Add Email Address

Send Emails

Recipient:

Subject:

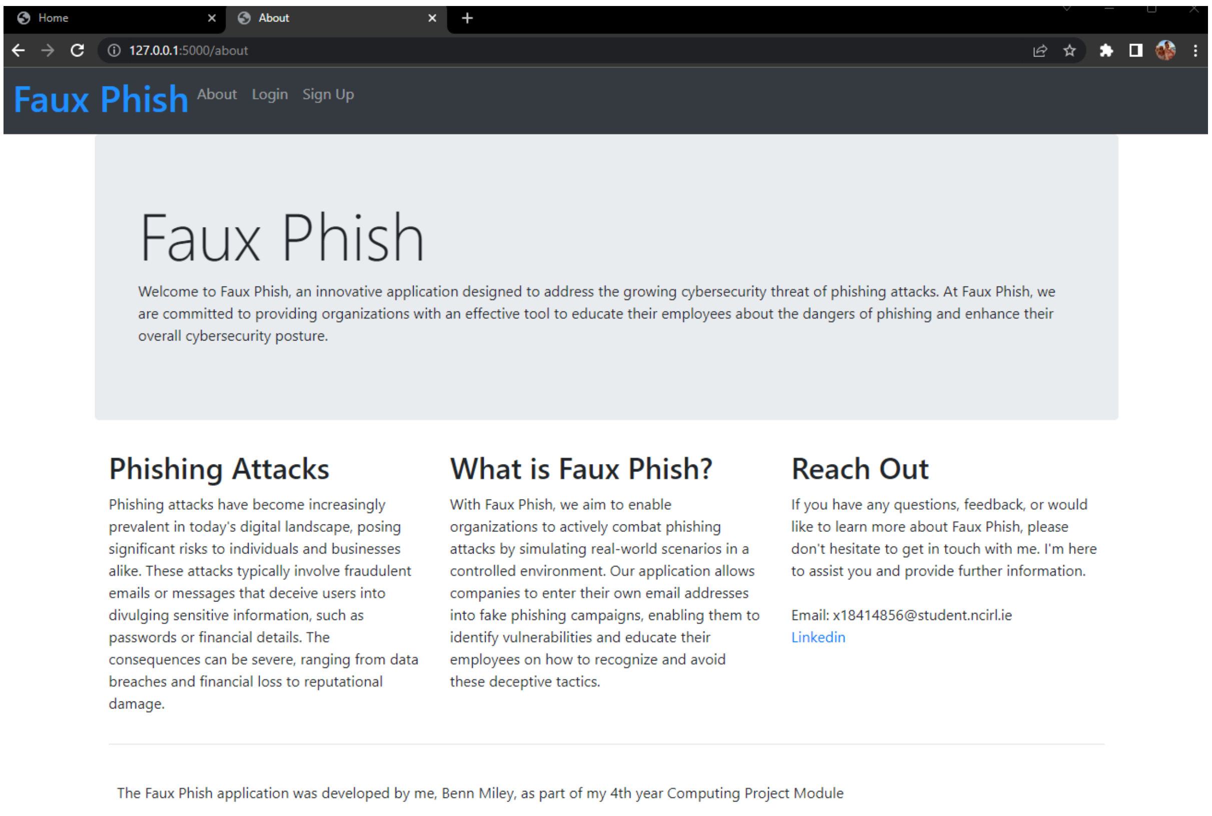
Message:

Send Email

Emails that fell for the campaign

- bennmiley@gmail.com - 2023-05-14 17:51:27.683652

This is the Gui for a populated version of the admin dashboard.


Home | About

127.0.0.1:5000/about

Faux Phish About Login Sign Up

Faux Phish

Welcome to Faux Phish, an innovative application designed to address the growing cybersecurity threat of phishing attacks. At Faux Phish, we are committed to providing organizations with an effective tool to educate their employees about the dangers of phishing and enhance their overall cybersecurity posture.

Phishing Attacks

Phishing attacks have become increasingly prevalent in today's digital landscape, posing significant risks to individuals and businesses alike. These attacks typically involve fraudulent emails or messages that deceive users into divulging sensitive information, such as passwords or financial details. The consequences can be severe, ranging from data breaches and financial loss to reputational damage.

What is Faux Phish?

With Faux Phish, we aim to enable organizations to actively combat phishing attacks by simulating real-world scenarios in a controlled environment. Our application allows companies to enter their own email addresses into fake phishing campaigns, enabling them to identify vulnerabilities and educate their employees on how to recognize and avoid these deceptive tactics.

Reach Out

If you have any questions, feedback, or would like to learn more about Faux Phish, please don't hesitate to get in touch with me. I'm here to assist you and provide further information.

Email: x18414856@student.ncirl.ie
Linkedin

The Faux Phish application was developed by me, Benn Miley, as part of my 4th year Computing Project Module

This is the Gui for the about page

2.5. Testing

Testing is a crucial aspect of the software development process in ensuring the quality and reliability of the application. For Faux Phish, testing tools, test plans, and test specifications were employed to ensure functionality and identify any potential issues. The following sections outline the testing approach and provide evidence and results of the conducted tests.

1. Testing Tools:

   - End User Testing: The application was tested by an end user (me) manually, simulating real-world usage scenarios.
   - Exploratory Testing: Exploratory testing was performed by me, from the outset and throughout the development process to identify any unforeseen issues and assess the user experience.

2. Test Plans and Test Specifications:

   - End User Test Plan: The end user test plan defined various usage scenarios and actions to be performed by end users to ensure the application functions as intended.
   - Exploratory Testing: Exploratory testing was performed by me, from the outset and throughout the development process to identify any unforeseen issues and assess the user experience.

3. End User Testing Results:

    - Evidence: I performed application based on the predefined test plan, following specific usage scenarios listed in the figure below.
    - Results: The end user testing revealed positive feedback and no critical issues were identified. I was able to navigate the application, perform desired actions, and achieve the expected outcomes.

Test Description	Projected result	Actual Result
Open the Faux Phish application by launching main.py.		
Verify that the login page is displayed correctly and all elements are visible.		
Click on the "Sign Up" link to navigate to the sign-up page.		
Fill in the sign-up form with invalid information & Click on the "Sign Up" button to submit the form.		
Verify that a failure message is displayed		
Fill in the sign-up form with valid information, including email, first name, last name, organization, and a strong password. Click on the "Sign Up" button to submit the form.		
Verify that a success message is displayed, confirming that the account was created successfully.		
Log out of the application and verify that the user is redirected to the login page.		
Enter false credentials. Click on the "Log In" button to submit the login form.		
Verify that the user is bot logged in and a error message is displayed.		
Enter the credentials used during sign-up (email and password) into the respective input fields. Click on the "Log In" button to submit the login form.		
Verify that the user is redirected to the dashboard.		
Test the functionality of sending emails by entering recipient email addresses, subject, and message content. Verify that the email is sent successfully.		
Test the address book management functionality by adding, viewing, and removing email addresses from the address book.		
Perform exploratory testing by interacting with the application in various scenarios, such as entering invalid inputs, testing edge cases, and checking error handling and validation.		
Perform cross-browser testing to ensure compatibility with different web browsers (e.g., Chrome, Firefox, Edge)		

The testing phase demonstrated the effectiveness and reliability of application. The end user test confirmed the proper functioning of the implemented features and provided confidence in the application's stability and usability.

2.6. Evaluation

The system was evaluated through various methods to assess its performance, usability, and effectiveness. The evaluation focused on aspects including usage data, performance metrics, and correctness of the system. The following evaluation results were obtained:

Performance Evaluation:

- Scalability: The system was tested under user loads <10 to evaluate its scalability and ability to handle a growing user base without performance implications.

Usage Data:

- Email Sending: The number of emails sent and successfully delivered were monitored to evaluate the effectiveness of the email sending functionality.

Correctness:

- Functional Testing: Extensive functional testing was performed to verify that all system features and functionalities were working as intended.
- Error Handling: The system's error handling capabilities were evaluated by deliberately triggering various error scenarios to ensure appropriate error messages and handling of exceptions.

3.0 Conclusions

In conclusion, the Faux Phish project has successfully achieved its objectives of providing a platform for companies to conduct fake phishing campaigns and raise awareness about the dangers of such attacks. The project has demonstrated its effectiveness in educating users and enabling organizations to assess their security measures.

The advantages Faux Phish include its user-friendly interface, the ability to send customized phishing emails, and the tracking of link clicks for educational purposes. It empowers IT administrators to manage users, send targeted phishing emails, and analyse the effectiveness of their security measures.

By incorporating technologies like Flask, SQLAlchemy, and SMTP email services, Faux Phish demonstrates the use of industry-standard tools and frameworks to ensure reliable performance and functionality.

However, the project also has some limitations. One notable limitation is that the application is not currently deployed to a domain, which restricts certain functionalities such as warning reroutes. This means that users cannot be redirected to the educational page if they click on a phishing link. However, the framework is in place to implement this functionality with minimal effort if the application is deployed to a domain in the future.

Another limitation is its limited scope. While Faux Phish focuses on educating users about phishing attacks, it may not cover all aspects of cybersecurity awareness. It is essential to complement this project with other educational initiatives to address broader security concerns, such as malware and secure browsing practices.

Not so much a limitation, but a privacy and ethical consideration. Faux Phish involves the collection and analysis of user data, including email addresses and interaction. It is essential to handle this data with utmost care, ensuring user privacy and complying with relevant privacy regulations.

Despite these limitations, Faux Phish has provided a solid foundation for companies to conduct internal phishing campaigns and improve their cybersecurity awareness. With future improvements and deployment to a domain, the project has the potential to enhance its functionality and further contribute to combating phishing attacks.

Overall, Faux Phish serves as an important tool for organizations to educate their employees, assess their security measures, and mitigate the risks associated with phishing attacks. It encourages a proactive approach towards cybersecurity and empowers users to recognize and respond to phishing attempts effectively.

4.0 Further Development or Research

Given additional time and resources, the project could be further developed and improved to mitigate the limitations and enhance its overall functionality.

First and foremost, domain deployment. To negate the limitation of some functionalities not being fully implemented due to the lack of domain deployment, one of the immediate steps would be to deploy the application to a domain. This would involve securing a domain and configuring the necessary settings

Secondly, User Interface Refinement. Continuously refining the user interface and user experience can contribute to the overall usability of the application. This includes improving the design elements, streamlining navigation, and incorporating user feedback to make the application and its GUI more intuitive and user-friendly.

Additionally, I would add enhanced phishing simulation capabilities. The project could be expanded to include more sophisticated phishing simulation techniques. This could involve developing advanced email templates, incorporating social engineering techniques, and creating realistic scenarios to test users' response to targeted phishing attacks, rather than having the IT admin choose the body of the email.

By focusing on addressing the existing limitations and optimizing the application's functionality, security, performance, and user experience, the project can evolve into a more robust and reliable solution, offering enhanced value to its users.

5.0 Appendices

5.1. Project Proposal

National College of Ireland

Project Proposal

Faux Phish

14/11/22

BSHCYB4

Cyber Security

2022/2023

Benn Miley

18414856

X18414856@student.ncirl.ie

Objectives

Phishing is a type of cyber-attack in which an attacker attempts to trick a victim into revealing sensitive information, such as passwords or financial information, by sending them a fraudulent message via email.

With my project called Faux Phish. Participants are sent fake phishing emails that are designed to imitate real attacks. The emails contain a link to an educational web page that informs both the organisation IT Manager and the email recipient that they clicked on the possibly fraudulent link and fell for the Faux Phish.

The goals of the project are:

- To teach participants how to identify and avoid these types of attacks by learning to recognize the signs of a phishing email and following best practices for protecting their information.
- To provide individuals within an organisation a safe and controlled environment in which to practice detecting and defending against phishing attacks.
- To assist organizations to assess the effectiveness of their current security measures and identify areas where additional training or improvements may be needed.

Background

According to Verizon's 2022 Data Breach Report, 82% of breaches involved the Human Element. According to CISCO's 2021 Cybersecurity Threat report, Phishing attacks are responsible for more than 80% of reported security incidents.

These two statistics speak volumes of the danger of phishing attacks and the frequency in which they occur.

I believe it is therefore of paramount importance to educate IT users of the dangers of such attacks and how an attack might look.

Its also equally important that organisations are aware of the threat posed by phishing attacks and how effective their current security measures are in combatting such attacks.

It is for the above reasons, that I, as a computing student specialising in Cybersecurity, chose to develop Faux Phish.

State of the Art

There are a number of applications that are similar to mine

1. PhishMe: This is a security awareness platform that includes a phishing simulation tool. It allows organizations to create and send custom phishing emails to their employees and track their responses.
2. Phish Threat: This is a phishing simulation tool that is part of the Microsoft Office 365 suite. It allows organizations to create and send custom phishing emails to their employees and track their responses.
3. Simulated Phishing Attack: This is a phishing simulation tool that is part of Googles G Suite. It allows organizations to create and send custom phishing emails to their employees and track their responses.

My Project is unique in the sense that as well as allowing an organisation to send phishing emails and track their employees response. It will also offer information to those who fall victim to the faux phishing campaign in an attempt to prevent them from re-offending.

Technical Approach

For the development of my project, I will be taking an Agile approach.

I chose an agile approach for the following reasons:

- Agile is the Software Development Lifecycle Methodology which I am most proficient in using.
- Agile affords me great flexibility an allows me to adapt my work for changing circumstances and priorities as I learn more throughout my study of computing.
- Agile encourages regular communication between myself and my project supervisor which encourages me to stay on track in line with the grading rubric
- Agile assists me in focusing on the most important tasks by breaking down my work into smaller, more manageable chunks and prioritising based on functional importance. This assistance avoids me getting caught up in smaller, less important details.
- Agile promotes enhanced productivity by helping me stay organised and focused.

Technical Details

Faux Phish is an object-oriented web application developed in Python 3.11 using the Python Flask web framework.

It uses HTML5 to display webpages. It uses Bootstrap and CSS to handle styling.

It will store data on an SQLite database using SQLAlchemy as its ORM.

To carry out certain functions such as deleting entries to the database. JavaScript will be used.

Microsoft Visual Studio code will be used to code the application.

I am still researching a suitable SMTP sever in order to deliver my faux phishing campaign and am also weighing in the option of using a Transactional email service which would ensure reliable delivery and provide API's which would prove invaluable to my project.

Special Resources Required

- Server capable of handling SMTP

Project Plan

I will use an agile approach and manage said approach with the use of Trello.

Faux Phish Agile Sprint _ Trello.pdf

Testing

Unit (white box) and integration testing will take place throughout the development of my project as part of my use of the Agile Sprint methodology.

System and acceptance testing will take place during my last 2 sprint cycles.

I will not be evaluating the system with an end user, but instead will simulate the evaluation by conducting the tests myself.

5.2. Reflective Journals

Supervision & Reflection Template

Student Name	Benn Miley
Student Number	18414856
Course	BSHCYB4
Supervisor	Cristina Hava Muntean

Month: October

What? In October, I am focusing on project planning and conducting extensive research. I am analysing phishing attacks and their impact on cybersecurity to gain a deeper understanding. I am identifying the key functionalities required for the Faux Phish application
So What?

This initial phase of planning and research is setting a strong foundation for the project. By understanding the scope and requirements, I can define clear objectives and outline the necessary features. The progress made in terms of project planning is a significant success, providing a roadmap for the development process. Challenges remain in terms of selecting the most suitable technologies and frameworks for the application. Further study and understanding of email sending techniques and URL generation are also needed.	
Now What? To address the outstanding challenges, I will research more in-depth, different technologies. This will help me make informed decisions for the development phase. I will allocate more time to learn about email sending techniques and implement secure URL generation platforms.	
Student Signature	Benn Miley

Month: November

What? In November, I have focused on implementing the core functionalities of the Faux Phish application. I have successfully developed the login and sign-up features, allowing users to create accounts and securely access their accounts.	
So What? The completion of these core functionalities is a milestone in the project. It demonstrates progress in turning the project concept into a functional application. Successfully implementing the login, sign-up features showcases my ability to work with databases, and user authentication	
Now What? I will conduct extensive testing of the implemented functionalities to verify the reliability and security of the login, and sign-up, features. I will also gather feedback from Christina. Additionally, I will continue researching and implementing security measures to fortify the application against potential threats and attacks.	
Student Signature	Benn Miley

Month: December

What? In December, I focused on refining the existing functionalities of the Faux Phish application and preparing for the midpoint presentation. I successfully completed the address book management feature, allowing IT admins to add, view, and remove email addresses from the address book. Additionally, I worked on improving the user interface and overall user experience by implementing styling using Bootstrap.
So What? The completion of the address book management feature adds an important component to the application. It enhances the functionality and usability for IT admins. The improved user interface with Bootstrap styling enhances the visual appeal of the application. The preparation for the midpoint presentation offers an opportunity to showcase the progress made so far and receive valuable feedback. This feedback will help in identifying any areas that require further improvement and guide the development process moving forward. However, challenges still exist in terms of ensuring the application's security and scalability. Further testing and validation are needed to identify and address any potential vulnerabilities or performance issues.
Now What? Feedback from the midpoint presentation will be carefully considered to refine and enhance the application based on suggestions.

Student Signature	Benn Miley

Month: January

What? In January, progress on the project have been relatively limited due to other exams and academic commitments.

So What? January highlights the challenge of balancing multiple commitments and maintaining consistent development momentum. Time management and prioritization skills became crucial during this period.	
Now What? To address the outstanding challenges and regain momentum in the project, I will allocate more time and resources in the coming months	
Student Signature	Benn Miley

Month: February

What? In February, I made significant progress. This month was dedicated to in-depth research and exploration of various techniques and technologies related to email sending and phishing simulations. I also reflected on feedback given by the examiners	
So What? The extensive research conducted in February has laid a strong foundation for the project's future success. It allowed me to gain a deeper understanding of email protocols, SMTP servers. This knowledge will prove invaluable in implementing email sending and URL generation functionalities.	
Now What? Building on the knowledge gained from the research phase, I will move forward with implementing the URL generation feature in the coming months Additionally I will implement the email sending functionality.	
Student Signature	Benn Miley

Month: March

What? I successfully implemented the URL generation feature, which adds a crucial layer of functionality to the application. Additionally, I focused on refining the overall user experience, enhancing the application's security measures, and conducting thorough testing	
So What? The successful implementation of the URL generation feature enables the generation of unique URLs for each email sent, providing a reliable mechanism to track and identify recipients' interaction with the simulated phishing emails.	
Now What? With the URL generation feature in place and the application's overall stability and security improved, I will shift my focus towards email sending functionality	
Student Signature	Benn Miley

Month: April

What? I successfully implemented the email sending capabilities, which now allow users to send simulated phishing emails to designated recipients	
So What? The addition of email sending capabilities is a significant milestone in the project. It completes the core functionality of the Faux Phish application, enabling users to create and send realistic phishing emails to assess the cybersecurity awareness and response of their organization.	
Now What? I will now shift my attention towards finalizing the project for the submission.	
Student Signature	Benn Miley