# PHISH-AWARE

Functional Specification

SEAN DOWLING
C00246571

# Contents

## Introduction

This functional specification will explain the tool's design philosophy as well as a thorough description of some of its most important features, such as its main objective, target audience, and operational details. The purpose of this tool is not only to detect the presence of a phishing email, but also inform the user on why it is a phishing email, this will allow the user to learn and expand their knowledge of phishing. This will be particularly good for college students within SETU, as we all use our emails daily, whether it be for talking to lecturers, or getting newsletters on the latest event taking place on campus. The tool will be presented as an Outlook Add-In which will analyse users' emails with a click of a button.

## Inspiration or Motivation

The lack of tools that specifically prohibit users from visiting dangerous URLs and, more importantly, inform them of why they have been blocked, is the fundamental driving force for this project. It is important not only to protect users from phishing attacks, but also train users to detect phishing themselves. While on work placement in Security Risk Advisors, I seen the lack of knowledge when it comes to what is a phish and what isn't, so designing a tool which will teach users, alongside analysation of headers, attachments and URLs, would be of a great benefit to all. Auto detect tools similar to this tool have already been created in the past, but due to machine learning restrictions and how sophisticated attackers are becoming, phishing attacks can still bypass these tools, leaving the user susceptible to these attacks.

## System Overview

An automatic phishing detection tool is a type of technology designed to identify and prevent phishing attempts. Phishing, in simple terms, is a type of cyber-attack that is constructed by using fraudulent emails or websites to trick users into revealing sensitive information, such as login credentials or financial information. These attacks are often difficult to detect, as they can be made to look like legitimate communications from trusted sources.

This tool will have access to the users incoming email and with a click of a button will begin analysing if it is a phishing attempt or not. The system will extract links and attachments and run them through known APIs and will compare them to other security vendors, while this is happening, the system will also be extracting the headers to analyse them also, to determine the source of the email, the domain name, see if the DKIM, DMARC and SPF were passed or not, and other relevant characteristics.

Once the determination is made, feedback will be presented to the user on whether the email, that was analysed, is a phishing email or not. It will also show where the problems arose, such as the link or attachment was deemed as malicious, or the email headers were suspicious.

This will, in turn, significantly reduce a successful phishing attack, while also informing and educating the user.

## Main Users

The main users of this tool will be college students and staff within SETU Carlow. Although most students may have a basic knowledge of computers and emailing, they may not know the risks that come hand in hand, staff on the other hand may have never gotten formal computing education over the years, so this tool will provide that. As I said, my tool will be used as a teaching tool for students looking to be protected online while also learning a thing or two along the way. This phishing tool will teach students what to look out for while reading emails to prevent successful phishing attacks. In my four years as a student at this college, there has been very limited training or workshops put in place to protect against phishing attacks. Around 25% of all data breaches involve phishing and 85% of data breaches involve a human element (Jones, 2022). From these statistics, it shows it's never been a better time to expand our knowledge and understanding about these attacks, and maybe one day protect the college from a serious problem. Throughout my research, I have been looking at other similar add-ins to this tool and none have provided stats on their success rates, also, most of the tools include an "additional purchase" for URL and attachment checks.

The tool will primarily be focused for SETU students and staff but will also be available to any other individual also. Since the tool will be available as a free Outlook Add-In, anyone can use it whenever they want. This has been done as any tool I have found during my research phase, has been designed specifically for large corporations, and the companies

running these tools are asking for a lot of money, not taking individuals into account. Some examples of these tools are:

- Avanan
    - Protect - $3.60 per mailbox/per month.
    - Advanced Protect - $4.60 per mailbox/per month.
    - Complete Protect - $6.00 per mailbox/per month.
- IRONSCALES
    - Email Protect - $6.00 per mailbox/per month.
    - Complete Protect - $8.33 per mailbox/per month.
- GreatHorn
    - Only works for companies with 250+ users.
- Mimecast Email Security with Threat Protection
    - Starts from $485 per month for up to 49 users.

From the prices shown above, it is obvious that these top tools are only really designed for larger companies, and tools that do consider individuals are priced very highly. So, a tool that is primarily focused on college students, paying for safety, shouldn't be a concern.

## Risks Involved

There are many risks that come with designing a tool like this, firstly attackers could send malicious emails to a fake email, run this programme, and see the results. If the results say, this is a phishing attempt, the attacker can manipulate the email and keep on testing until the results show that it is a clean email with no harm.

Another risk is misclassification of an email. Phishing attacks are constantly evolving, and it can be difficult for the systems underlying detection tools (APIs) to keep up with the latest tactics and techniques used by attackers. This means that the system may fail to identify some phishing attempts, potentially allowing them to succeed and putting the user at risk.

Another risk is that the system may generate a high number of false positives, incorrectly identifying legitimate emails as phishing attempts. This can result in a significant amount of inconvenience and frustration for users and may cause them to become less trustful of the system over time.

# Functional Specifications

## I. Core Functions

The main function of my system/add-in will be to determine whether an email is a phishing attack or not. The system will utilize APIs to determine if attachments and links are legitimate or malicious. I will be utilizing many different APIs to get a better perspective on an URL or attachment. The APIs I will be using are:

- IsItPhish
- CheckPhish
- VirusTotal
- Kaspersky
- MetaDefender Cloud

This tool will also be designed to extract header information and analyse this data to determine the legitimacy. The results sent back to the user will also include information, on two different levels, a brief overview, and then an in-depth analysis on all checks that were run and determinations that were made, this will be done so the user can gain a greater understanding of what attacks may look like and what information to look out for in the future.

## II. Additional Functions

If everything goes according to plan, I would like to implement additional features such as an excel spreadsheet that outlines all the results congregated into one area and allow the user to view this and analyse the data. This will again be a good learning tool for the user and all the data will be broken down into sections.

Another function I would like to implement is a content analyser. This would extract the text from the email and look for key words that appear in most phishing attempts, such as “urgent” or “confidential”.

## Use Case Diagram

A misuse case diagram is a type of visual representation that shows how a system or software can be misused. It is typically used in software engineering to identify potential security risks and vulnerabilities. Misuse case diagrams typically include the following elements:

- The system or software being evaluated.
- The potential attacker, who is attempting to misuse the system.
- The intended use cases, which are the legitimate ways in which the system is intended to be used.
- The misuse cases, which are the ways in which the attacker is attempting to misuse the system.

Below is an example of how attackers may misuse my automatic phishing detection tool.

A quick overview on the steps:

- Fetch Emails – Email is sent to the user's inbox available for them to open and read.
- Extract URLs – Take out URLs within an email and do rudimentary checks on it.
- Extract Attachments – Take out Attachments and do rudimentary checks on it.
- Extract Header Information – Take out Header Information and do checks on it.
- Make Determination on Legitimacy – Determines whether all three are legitimate or malicious.
- Make Classification – Return to user with whether the email is deemed safe or not.
- Receive Feedback and next step information – Receives info on all checks done and what is recommended to do with that email.

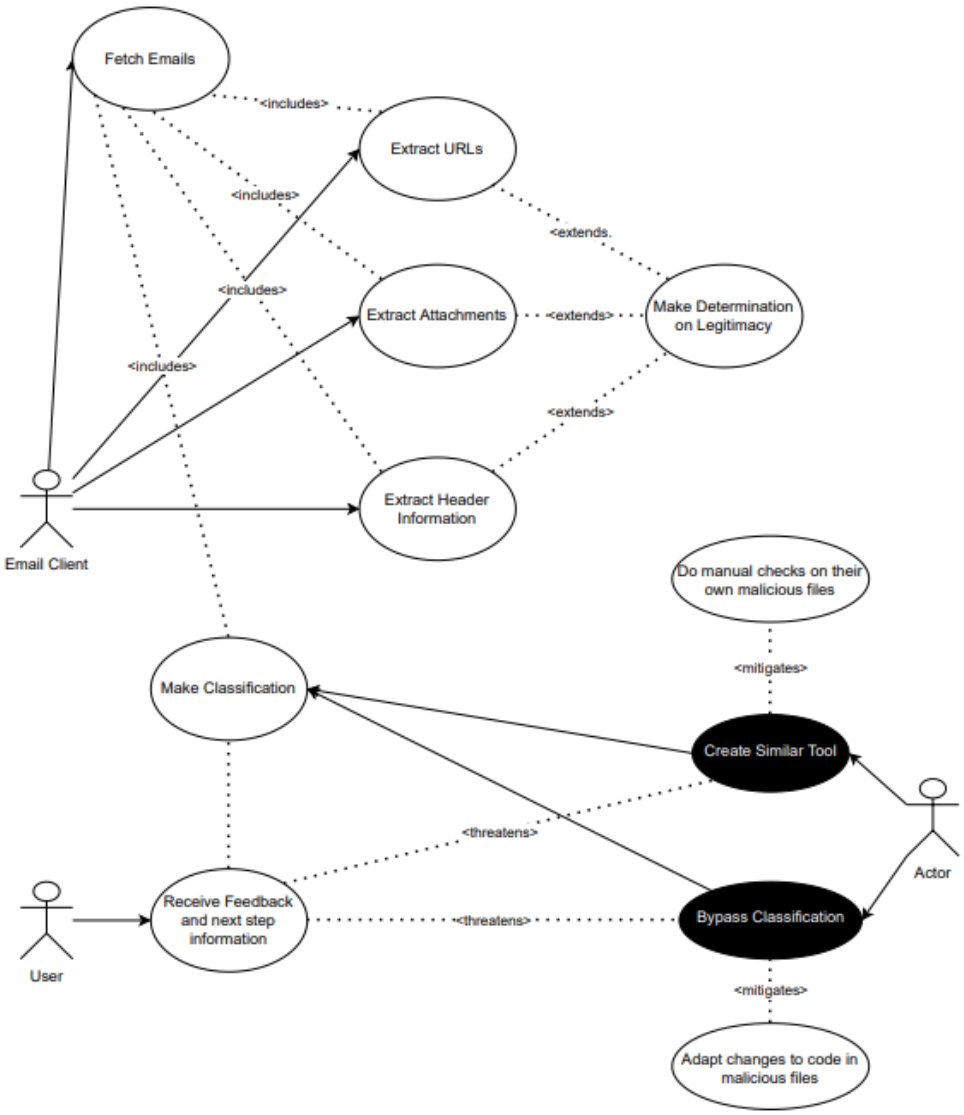


*Figure 1: Misuse Case Diagram*

## Use Cases

A use case describes how the person who will actually use the process or system will accomplish a goal. It's typically associated with software systems, but can be used in reference to any process (Study.com, 2022). The use cases below show each step and function of the tool.

### I. Receive Email for Analysis

Use Case – 1 highlights the first port of call for this tool. The Email is received by the client and will be placed within the user's inbox.

| Use Case - 1 | Receive Email for Analysis |
|---|---|
| **Description** | Email received by the Email Client (Outlook) |
| **Actor(s)** | Outlook |
| **Stakeholder(s)** | User and Attacker(s) |
| **Trigger** | Incoming Email |
| **Precondition(s)** | Email is received |
| **Postcondition(s)** | Extract URLs, Header Information and Attachments, Make determination, Send Feedback to User |
| **Main Success Scenario** | Email is received and is in Users inbox |

### II. Analyse Header Information

Use Case – 2 highlights what will be required for Header Information analysis.

| Use Case - 2 | Analyse Header Information |
|---|---|
| **Description** | Extract Header Information and analyse |
| **Actor(s)** | Outlook |
| **Stakeholder(s)** | User and Attacker(s) |
| **Trigger** | Incoming Email |
| **Precondition(s)** | Email information is located |
| **Postcondition(s)** | Make Determination on Legitimacy, Send Feedback to User |
| **Main Success Scenario** | Header Information is analysed correctly and fed back to User. |

### III. Analyze URLs

Use Case – 3 highlights what will be required for URL Analysis.

| Use Case - 3 | Analyse URLs |
|---|---|
| **Description** | Extract URLs and analyse |
| **Actor(s)** | Outlook |
| **Stakeholder(s)** | User and Attacker(s) |
| **Trigger** | Incoming Email |
| **Precondition(s)** | URLs are located within an email |
| **Postcondition(s)** | Make Determination on Legitimacy, Send Feedback to User |
| **Main Success Scenario** | URLs are analysed correctly and fed back to User. |

## IV. Analyze Attachments

Use Case – 3 highlights what will be required for Attachment Analysis.

| Use Case - 4 | Analyse Attachments |
|---|---|
| **Description** | Extract Attachments and analyse |
| **Actor(s)** | Outlook |
| **Stakeholder(s)** | User and Attacker(s) |
| **Trigger** | Incoming Email |
| **Precondition(s)** | Attachments are located within an email |
| **Postcondition(s)** | Make Determination on Legitimacy, Send Feedback to User |
| **Main Success Scenario** | Attachments are analysed correctly and fed back to User. |

## V. Feedback to User

Use Case – 4 highlights what information will be needed to send feedback to user.

| Use Case - 5 | Feedback to User |
|---|---|
| **Description** | Gather Header Information, URL Analysis and Attachment Analysis and present information back to user. |
| **Actor(s)** | Outlook |
| **Stakeholder(s)** | User and Attacker(s) |
| **Trigger** | Email Determination |
| **Precondition(s)** | Header Information, URLs and Attachments were analysed |
| **Postcondition(s)** | Feedback is presented to user |
| **Main Success Scenario** | Feedback is presented to user in a simple way for anyone with any technical knowledge will be able to understand |

## System Configurations

An overview of all the steps required to configure the prospective system.

### I. Executable

- Configure an executable to interact with an email client, preferably Outlook. This will be done using “pyOutlook” library for Python.

### II. Extract Information

- Within that executable, create functions to extract header information, attachments and URLs. This will be done using either “pywin32” or “win32com” Python libraries.

### III. Analyse Information

- Utilize APIs to run each function through them. This will be done using various APIs like VirusTotal or IsItPhish

### IV. Make Classification

- Make informed classification of the nature of the email. This will be done by a scoring system.

### V. Gather Information

- Gather all relevant information in one area.

### VI. Feedback to User

- Relay results back to user, this will be done in levels:
    - Level 1 – Clean email, no phish attempt detected.
    - Level 2 – Warning, this may be a phishing attempt.
    - Level 3 – This is very clearly a phishing attempt.
- Each level will have all relevant information shown to user.

### VII. Additional Information

- An additional information tab will be presented too. This is for a more in-depth analysis of all checks that were complete.

## Non-Functional Requirements

Non-functional requirements are a type of requirement that specifies the desired performance, reliability, scalability, and other quality attributes of a system or software. These requirements focus on how the system should work and its overall behaviour, rather than specific functions or features. Below are the non-functional requirements of my project.

### I. Functionality

- The application should allow the user to analyse their emails.
- The application should analyse all information within the email and output results for the user to view in full.
- The application should output recommendations for the user on what to do with the email.

### II. Usability

- The user interface should be easy to understand and self-explanatory to allow users of varied technical backgrounds to use it to its full extent.

### III. Reliability

- The application will perform all the tasks it was designed or intended to do.
- The application will only output reliable results.

### IV. Performance

- The application will perform its tasks in an orderly fashion, not leaving the user waiting for long periods of time.

### V. Security

- The application will protect all data analysed from unauthorized access and threats.

### VI. Supportability

- The application will be available for use within Outlook Add-ins.

## External Tools / Requirements

Below are the tools and services I will be availing of to complete this project. I will be using multiple APIs to get the best determination for the URLs and Attachments.

- **Outlook** – A free Email client used to host the tool.
- **VSCode** – An open-source application which facilitates software development.
- **IsItPhish API** – Used to evaluate malicious URLs and attachments.
- **CheckPhish** - Used to evaluate malicious URLs and attachments.
- **VirusTotal** - Used to evaluate malicious URLs and attachments.
- **GoPhish** – Used for testing the application.
- **King Phisher** – Used for testing the application.
- **Python Libraries/Modules** – Open to additional resources that may be needed.
    - Win32
    - PyOutlook
    - Requests

## Example of GUI

Below is an example of what the GUI may look like, this will obviously change overtime but getting a better understanding of what will be needed to go into the GUI is helpful. All relevant information will be shown first, laid out nicely for the user to read easily, this will be done in a way that users with minimal technical knowledge will be able to understand the data that is being shown.

| **Automatic Phishing Detection System** | | | |
|---|---|---|---|
| | **URL Analysis** | **Header Analysis** | **Attachment Analysis** |
| **Analysis** | URL Name<br>VirusTotal Score<br>IsItPhish Score<br>CheckPhish Score | SPF: Pass/Fail<br>DMARC: Pass/Fail<br>DKIM: Pass/Fail<br>Sender<br>Return Path | Attachment Name<br>VirusTotal Score<br>IsItPhish Score<br>CheckPhish Score |
| **Scores** | Overall Score<br>Additional Information | Overall Score<br>Additional Information | Overall Score<br>Additional Information |

## Project Milestones

Below are the approximate dates I have set out for myself to have each function of the specification up and running. Of course, these are open to change as I may run into problems in some areas.

| Task at hand | Start | Finish | Duration |
|---|---|---|---|
| Initial Research | 14-Oct | 25-Nov | 43 Days |
| Functional Specification | 25-Nov | 16-Dec | 21 Days |
| Tool Development | 16-Dec | 28-Feb | 89 Days |
| Gather Email Information (URL, Headers, Attachments) | 16-Dec | 01-Feb | 47 Days |
| Analyse Information | 01-Feb | 10-Feb | 10 Days |
| Feedback | 10-Feb | 17-Feb | 7 Days |
| Extra Information | 17-Feb | 21-Feb | 4 Days |
| Test tool | 21-Feb | 28-Feb | 7 Days |
| Create as Email Add-In | 28-Feb | 07-March | 7 Days |
| Additional Functions | 07-March | 21-March | 14 Days |
| Final Changes | 21-March | 31-March | 10 Days |

## GANTT Chart

Below are all project milestones shown in a GANTT Chart, showing an approximated progression schedule of the overall project.

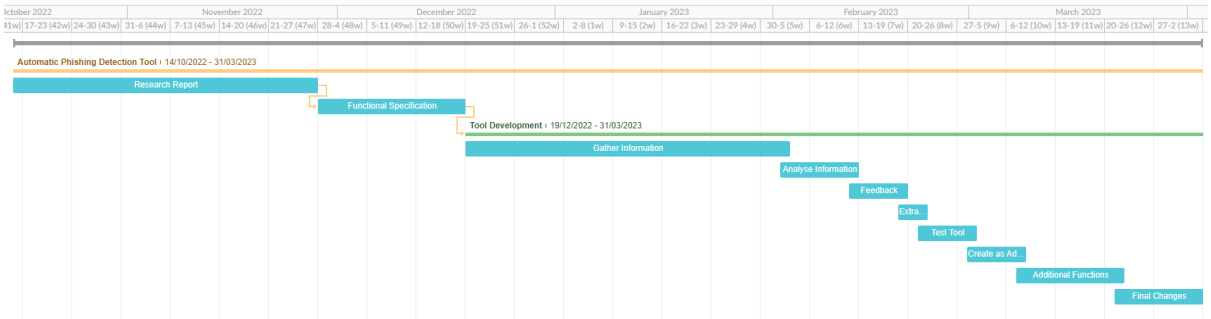


*Figure 2: GANTT Chart*

# Bibliography

Jones, C., 2022. *50 Phishing Stats You Should Know In 2022.* [Online]
Available at: https://expertinsights.com/insights/50-phishing-stats-you-should-know/
[Accessed 05 December 2022].

Study.com, 2022. *What is a Use Case? - Definition & Examples.* [Online]
Available at: https://study.com/academy/lesson/what-is-a-use-case-definition-examples.html
[Accessed 11 December 2022].