4th Year Project Specification

ImageAware+ - Phishing Detection & Educational Platform

Name	**Lorcan Kelly Zazzera**
Student ID	**C00288941**
Course	**Bachelor of Science (Honours) in Cybercrime and IT Security**
Supervisor	**Mark Cummins**

1.0 Introduction

1.1 Background

Phishing attacks continue to be one of the most prevalent and damaging forms of cybercrime. Attackers frequently use deceptive techniques to trick individuals into disclosing personal information, credentials, or financial data. While many organisations employ phishing filters and email-scanning tools, these systems primarily focus on textual content and hyperlinks, leaving embedded images largely unchecked.

Recent trends show a marked increase in image-based phishing, where attackers embed QR codes, copied brand imagery, or realistic login screenshots within emails and websites to deceive users. In the SOC environment, it is common to see emails posing as trusted companies such as PayPal or Amazon that claim to include invoices or order confirmations. These use genuine-looking logos and invoice layouts to convince recipients to click links or open attachments.

Other common examples include emails that appear to be from legitimate providers such as Microsoft, urging the user to sign in or reset their password. The hyperlink or QR code embedded in the image redirects the user to a spoofed Microsoft login page designed to harvest credentials.

While enterprise email gateways often flag these messages as suspicious, they typically cannot interpret or extract the actual threat hidden in the image, such as decoding a QR code or detecting a fake login prompt. This gap not only reduces detection accuracy but also increases the amount of manual investigation time required by analysts. ImageAware+ addresses this by combining computer vision, OCR, and threat intelligence analysis to detect and interpret malicious elements concealed within images, providing analysts with faster, explainable, and proactive threat insights.

1.2 Problem Statement

From my experience working in a Security Operations Centre (SOC), I have observed that while current anti-phishing solutions such as Google Safe Browsing, Microsoft SmartScreen, and enterprise email security gateways are effective at detecting text-based threats, they often struggle when malicious elements are hidden inside images. These systems typically rely on URL reputation and textual content analysis, so when an attacker embeds a QR code, logo, or fake login screenshot within an email image, the underlying threat frequently goes unrecognised.

In practice, many of these tools correctly flag an email as suspicious but cannot interpret or explain the specific reason. For example, they may detect that an attachment looks risky yet fail to extract the QR code linking to a phishing site. This limitation forces analysts to manually

download, inspect, and decode image files, which is time-consuming and inefficient during live investigations.

As phishing campaigns increasingly exploit visual deception, this lack of image-level analysis leaves organisations exposed and places additional strain on SOC analysts who must balance speed with accuracy. There is currently no open-source, explainable framework designed to automatically identify and interpret image-based phishing threats while integrating with existing security workflows.

ImageAware+ seeks to close this detection gap by automating the extraction and interpretation of visual phishing indicators. By analysing images for QR codes, embedded text, and social-engineering cues, it aims to reduce analyst workload, shorten investigation time, and deliver clear, standardised threat reports that can be directly integrated into existing phishing detection and simulation platforms.

1.3 Objectives

The main objective of ImageAware+ is to develop a practical and explainable tool that automates the analysis of images for phishing-related cues, reducing investigation time and providing standardised, transparent risk reports. Specific objectives include:

- Develop an automated image preprocessing and extraction pipeline.
- Implement Optical Character Recognition (OCR) to extract text from images.
- Detect and decode QR and 2D barcodes embedded in phishing images.
- Analyse extracted URLs using threat intelligence APIs (VirusTotal, PhishTank).
- Identify social-engineering language and brand impersonation patterns.
- Generate standardised risk assessment reports highlighting detected threats.
- Integrate the system into an existing phishing simulation platform, King Phisher.

1.4 Project Scope

The project will focus on the detection of phishing indicators within images rather than traditional text-based detection. The scope includes:

- Image preprocessing (noise reduction, contrast enhancement, thresholding).
- OCR for text detection and recognition.
- QR and barcode detection and decoding.
- URL analysis and threat intelligence integration.
- Social-engineering text detection using heuristic methods.
- JSON and PDF-based report generation.
- Integration with the King Phisher platform.

Out of scope: large-scale enterprise email integration, multilingual NLP, and live sandboxing (though these may be considered as future extensions).

2.0 System Overview

The ImageAware+ system is designed to detect and interpret phishing content hidden within images, providing analysts with clear, explainable insights. By combining computer vision, OCR, and threat intelligence analysis, ImageAware+ creates a structured, automated pipeline that identifies and explains malicious elements within visual content.

The system operates as follows:

- Preprocess the input image to enhance detection accuracy.
- Extract text from the image using OCR (Tesseract).
- Detect and decode embedded QR codes or barcodes using OpenCV and Pyzbar.
- Analyse extracted URLs through integrated threat intelligence APIs including VirusTotal and PhishTank.
- Evaluate social-engineering cues by analysing extracted text for manipulative or persuasive language.
- Generate a structured report highlighting detected threats, confidence levels, and annotated visual regions.

By automating these steps, ImageAware+ supports rapid, repeatable, and transparent analysis of phishing images, improving both detection accuracy and investigation turnaround time.

2.1 Image-Based Phishing Detection using ImageAware+

ImageAware+ processes any image whether it comes from an email attachment, phishing campaign, or website capture to uncover embedded phishing indicators. Through its modular design, the system combines OCR-based text extraction, QR code decoding, and threat-intelligence validation to identify malicious elements that would otherwise remain undetected by traditional filters.

This automation allows analysts to focus their efforts on higher-level investigation and response rather than manually inspecting each image. Each component of the pipeline operates independently, ensuring flexibility and ease of integration into existing phishing detection frameworks such as King Phisher.

The outcome is a clear, explainable report that not only flags an image as suspicious but also details why it was flagged, such as a decoded malicious URL, detected login prompt, or suspicious keyword pattern. This explainability enhances both analyst understanding and incident-reporting quality, ultimately improving organisational phishing defence capabilities.

2.2 User Group

The system is intended for the following primary user groups:

- Cybersecurity analysts and SOC (Security Operations Centre) personnel.
- Email security administrators.
- Researchers and educators in cybersecurity.
- End users as indirect beneficiaries through improved detection.

The system is intended for educational, research, and operational environments, assisting in analysing image-based phishing attempts.

3.0 Deliverables

Deliverables represent measurable outputs from the project. They are divided into core (essential) and secondary (optional) components. Together, they ensure that ImageAware+ provides a practical, explainable, and efficient solution that integrates smoothly into existing phishing detection workflows.

3.1 Core Deliverables

System-Specific Deliverables

Image Extraction and Preprocessing Module

Handles file ingestion, normalisation, and noise reduction using OpenCV. This ensures that images are optimised for accurate OCR and QR detection while maintaining processing speed suitable for SOC use.

Text Detection (OCR)

Implements Tesseract OCR to extract and structure text data for further analysis. Automating this step eliminates the need for analysts to manually inspect suspicious images for embedded text.

QR and 2D Code Detection

Uses Pyzbar or OpenCV to identify and decode embedded barcodes and QR codes, allowing ImageAware+ to uncover hidden URLs or data that would otherwise require manual decoding.

URL Analysis Engine

Performs normalisation and validation of extracted URLs using VirusTotal and PhishTank APIs. This automates a common manual SOC process, significantly reducing investigation time and improving accuracy.

Social-Engineering Detection Module

Applies rule-based heuristics and keyword scoring to detect persuasive or manipulative phrases often seen in phishing messages, such as "urgent action required" or "account suspended". This helps analysts quickly identify psychological triggers within phishing content.

Standardised Report Generator

Produces clear, explainable JSON or PDF risk reports containing annotated images, threat summaries, decoded URLs, and confidence scores. These reports streamline SOC communication and documentation by providing ready-to-share analysis outputs.

King Phisher Integration Layer

Integrates ImageAware+ outputs with the King Phisher framework for simulation and testing. This allows organisations to evaluate detection performance within realistic phishing scenarios and enhances training and response workflows.

Non-System Deliverables

- Research Report: Comprehensive documentation of methodology, design, and findings.
- Technical Report: Includes data flow diagrams, system architecture, and evaluation results.
- Testing Report: Summarises evaluation metrics such as precision, recall, and F1 score.

3.2 Secondary Deliverables

- Graphical User Interface (GUI): Optional web-based interface for intuitive, visual interaction with the system, enabling quick uploads and at-a-glance analysis results.
- Cloud Integration (REST API Service): Extends accessibility and enables automated integration with SOC platforms or security dashboards.
- Dashboard and Visualisation Tools: Present real-time statistics, detection trends, and confidence scoring for improved situational awareness.
- Sandboxed URL Resolution: Future feature allowing safer verification of suspicious URLs in a controlled environment.
- Dataset Preparation for Open-Source Release: Curated dataset of phishing images for benchmarking and community research.

4.0 Functionalities

The ImageAware+ system is built with modular and automated functionalities that work together to detect, analyse, and explain image-based phishing indicators. Each function contributes to reducing analyst workload, improving detection accuracy, and producing standardised, easily interpretable reports.

4.1 Backend Functions

- Image Preprocessing: Performs resizing, grayscale conversion, thresholding, and noise reduction using OpenCV to improve OCR and QR detection accuracy.
- OCR Processing: Extracts textual information from images using Tesseract OCR and structures it for further analysis.
- QR and Barcode Detection: Uses Pyzbar and OpenCV to identify and decode embedded QR and 2D barcodes, enabling automatic extraction of malicious URLs.
- Threat Intelligence Integration: Validates extracted URLs through APIs such as VirusTotal and OpenPhish, consolidating threat intelligence directly into reports.
- Risk Scoring and Classification: Assigns confidence scores based on the number, type, and severity of detected phishing indicators.

4.2 Additional Functionalities

- Export Reports: Generates structured, standardised reports in JSON or PDF format for sharing within SOC teams or importing into case management systems.
- REST API Integration: Exposes ImageAware+ as a RESTful API, enabling seamless integration with King Phisher and other security automation platforms.
- Confidence Visualisation: Produces annotated output images highlighting detected QR codes, suspicious text, and regions of interest.

4.3 User Interface Functions

- Web-Based GUI (Planned Extension): A user-friendly dashboard allowing drag-and-drop image uploads, real-time analysis results, and quick access to reports.
- Visualisation Panel: Displays OCR results, decoded QR contents, and URL threat intelligence data in a clear, interactive format.

4.4 Non-Functional Requirements

Requirement	Description
Performance	Each image should be processed in five seconds or less on standard hardware.
Accuracy	Targeting at least 85% detection accuracy on benchmark datasets.
Security	All external API calls use HTTPS encryption and sensitive data is handled securely.
Reliability	The system should maintain consistent results across varied image formats and noise levels.
Usability	Reports must be easy to read, visually annotated, and ready for inclusion in SOC documentation.

5.0 System Design

5.1 System Architecture

The ImageAware+ architecture follows a modular and scalable design, ensuring that each functional component can operate independently or as part of an integrated workflow. This design supports flexible deployment across research environments, testing labs, or active SOC infrastructure.

Data Flow Overview:

- The user uploads or provides an image (email attachment, web capture, or screenshot).
- The image undergoes preprocessing and noise reduction.
- OCR and QR detection modules extract text and embedded data.
- Extracted URLs are validated through integrated threat intelligence APIs.
- The results are processed by the risk-scoring engine.
- A JSON or PDF report is generated with annotated visuals and confidence scores.
- Results can optionally be exported or integrated with King Phisher for simulation and testing.

This modular pipeline ensures each step is transparent and explainable, allowing analysts and researchers to trace exactly how a phishing threat was identified.

5.2 External Tools and Technologies

Tool / Technology	Purpose
Tesseract OCR	Text extraction from phishing images.
Pyzbar / OpenCV	Detecting and decoding QR and 2D barcodes.
VirusTotal / PhishTank APIs	Validating and scoring extracted URLs.
Python 3.x, Flask	Backend service and REST API endpoints.
Scikit-learn, Pandas, NumPy	Analytics, scoring, and data handling.
Matplotlib, Pillow	Visualisation, annotation, and report image generation.

The selected technologies ensure strong performance, reliability, and compatibility with SOC workflows while remaining fully open-source and extensible.

6.0 Use Cases

The following use cases describe how ImageAware+ will be used within a practical environment such as a SOC or cybersecurity research setting. They demonstrate the system's role in automating repetitive investigative tasks, improving detection accuracy, and providing analysts with actionable, explainable outputs.

6.1 Use Case Descriptions

Actors:

- User (Analyst): Initiates image uploads, reviews reports, and interprets results.
- System Backend (ImageAware+): Processes images, extracts data, and generates reports.
- External APIs: Provide threat intelligence and URL validation services.

Use Case	Description	Actor(s)	Precondition	Postcondition
Upload Image	User uploads an image for scanning.	User	Image file available.	Image preprocessing begins.
Extract Text & QR	System extracts embedded text using OCR and decodes any QR or 2D barcodes.	System	Image loaded and processed.	Extracted data stored for analysis.
Analyse URLs	System validates extracted URLs via threat intelligence APIs, assigning threat scores.	System, External APIs	URLs extracted from OCR or QR modules.	Threat data added to results.
Generate Report	System compiles results into a standardised JSON or PDF report with annotated visuals.	System	Analysis complete.	Risk report generated and stored.
Export Result	User exports or downloads the report for further investigation or case documentation.	User	Analysis completed successfully.	JSON or PDF file accessible.

These cases represent the end-to-end workflow from ingestion to report generation, aligning closely with how SOC analysts conduct phishing investigations.

7.0 Project Plan

The project plan outlines the development approach for ImageAware+, from research and design through to implementation, testing, and evaluation. Each phase has defined deliverables that build upon the previous stage, ensuring steady progress toward a functional and validated system.

7.1 Phases of the Project

Phase	Description
Phase 1: Research & Design	Conduct a literature review on phishing detection, collect phishing image datasets, and design the system architecture. Define performance metrics and evaluation criteria.
Phase 2: Prototype Development	Implement the image preprocessing, OCR, and QR detection modules. Establish a modular codebase for future integration and testing.
Phase 3: URL Analysis & Reporting	Integrate threat intelligence APIs (VirusTotal, PhishTank) and develop the standardised reporting engine. Begin generating explainable reports.
Phase 4: Social-Engineering Detection	Implement heuristic and rule-based keyword analysis for detecting persuasive or manipulative text. Begin tuning the risk-scoring algorithm based on early results.
Phase 5: Integration & Testing	Integrate all modules and test the system within controlled phishing simulations through the King Phisher framework.
Phase 6: Evaluation & Documentation	Evaluate detection performance using metrics such as precision, recall, and F1 score. Finalise documentation and project deliverables.

Each phase contributes to a progressively refined system capable of providing automated, explainable phishing detection and supporting real-world SOC workflows.

8.0 Inspiration

The inspiration for ImageAware+ comes directly from my hands-on experience working in a Security Operations Centre (SOC) during my third-year internship, a role I have continued part-time.

While investigating phishing incidents in a live enterprise environment, I repeatedly observed that many detection systems correctly flagged emails containing embedded images as potential threats, yet could not identify the specific malicious element. For example, I often encountered invoice-style scams impersonating PayPal or Amazon, where attackers used authentic-looking brand logos and document designs to appear credible.

I also saw campaigns where emails appeared to originate from Microsoft, asking users to log in or change their password. The links or QR codes embedded in these messages led users to spoofed Microsoft sign-in pages crafted to steal credentials. Although our tools detected that something was suspicious, they lacked the capability to extract the QR code data or analyse the visual components to explain why the message was dangerous.

These recurring challenges revealed a clear blind spot in current phishing detection approaches: the inability to interpret and contextualise visual content. This realization motivated me to develop ImageAware+, a transparent and modular system that automatically analyses images for embedded phishing indicators using computer vision and machine learning.

The goal is to create a solution that enhances threat visibility and investigative efficiency within SOC environments while also contributing to the research and educational community through an explainable, open-source framework for detecting image-based phishing attacks.

9.0 References

IBM. (2024). Cost of a Data Breach Report 2024. IBM Security.

Google. (2024). Google Safe Browsing API Developer Documentation. https://developers.google.com/safe-browsing/

VirusTotal. (2024). Public API v3 Reference. https://docs.virustotal.com

PhishTank. (2024). Open Phishing Database. https://phishtank.org/

Microsoft. (2024). SmartScreen Security Overview.

OpenCV Documentation. (2024). Computer Vision Library. https://opencv.org/

Tesseract OCR Documentation. (2024). https://tesseract-ocr.github.io/

Scikit-learn Developers. (2024). Machine Learning in Python. https://scikit-learn.org/