Project 3000 Report  
By NIKODEM DRABIK

# Research

## API

### NEW

https://free-apis.github.io/#/categories

### Website

WHOIS API – 500 free calls before I need to change key. No credit card required for this.  
<https://apilayer.com/marketplace/whois-api> - free plan 3000 free requests

<https://whoisjson.com> – free 1000 monthly

<https://docs.virustotal.com> – not allowed commercially 500 per day

<https://urlscan.io/live> - not private high cost to expand

<https://www.mywot.com/developers> - costs

https://openphish.com/academic\_use.html - open source community drive

<https://urlhaus-api.abuse.ch> – key generated check comp 3000

google api safe search

### Verify Email

<https://cleantalk.org/blacklists>

https://www.millionverifier.com/services/email-verifier-api

https://emaildetective.io/pricing -1000 emails per month

<https://rapid-email-verifier.fly.dev/#/default/get_validate> - free and open source

<https://github.com/umuterturk/email-verifier>

### Issues

With this the issue I find is that all API keys are needed to be supplied by a user. Target market cannot do this themselves.

How do I ensure that they are supplying a valid key and how do I ensure that they are able to do so if they are meant to be tech illiterate? I do not have the scope to create my own for these two.

## Languages

### Extension

JSON

Chrome extension

### Standalone App

C++ - main code

js – process data

Node.js – API

# Early Project Identification

## Name

**Phis-Knight.**

Combination of Phishing and Knight. As a knight guards the people, and that is what I want it to be. And the name also allows me to have a theme for the software.

## Style

I want to create a medieval colour scheme and medieval styled UI. So, it looks like a knight. I would like to create a mascot that is a knight that is the face of the app.

### Elevator Pitch:

Phish-Knight a privacy first anti-phishing assistant. Detects phishing attempts and teaches you how to detect them.

## Vision

### Who?

The project is to be aimed at the elderly and young people. People that are tech-illiterate and as a result need assistant.

### Why?

The main problem I want to solve is the large amount of phishing attacks on the vulnerable. According to the NCSC in a study in 2025 they effect 93% of business that were surveyed and 95% of charities that have been breached were accessed through this means.

### The Product?

The project is to scan a screen and possibly screens HTML elements of a website and identifies the suspicious links and notifies the user.   
I would like to also have the app to use a Language Model to analyse the email for possible Scam messages. A few flags:

* Generic and non-direct greetings like Ma’am, Sir, Mr
* Urgently wanting to redirect you.
* Sender’s address does not match the body or domain.

#### Reason For?

The reason for this approach Is this that I want to create a software that act like a trusted and tech literate grandson scanning a grandparent’s device for the same reason. I want software that also teaches the user of possible dangers and make them self-reliant.

### Difference

Main part for this software is its trustworthiness. I want to create a software that has customisable intrusion settings. Based on how intrusive it will be, but less intrusive may mean less effective. However, this is a big part as it will hopefully make the user base more willing to use the software.

Secondly, I want this to be somewhat interactive where the user learns to seek the patterns the software is seeking for and hopefully learn to be independent.

### Risks and Challenges

The phishing scams are one of the main approaches for attacks as a result they are often adapting to be more effective and finding innovative ways to find attack. This will be difficult to keep effective long term without constant modifications and a software that adapts and learns new patterns.

### Personal Agenda.

I have a personal hatred for phishing scams aimed at the more vulnerable as it exploits the people that can’t defend themselves.

# Risk Assessment

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Likelihood | Impact | Mitigation/Response |
| Evolving Phishing Tactics | High | High | Software that can be easily maintained and updated to adapt to changes. |
| False Positives | High | High | Have a clear explanation for each flag. Provide methods to perform their own analysis if they disagree. |
| Data Privacy Concerns | High | Very High | Have all data processed locally where it is possible. If cloud services used or off device processing, use encrypted transmission and have a clear privacy policy |
| Challenges with integration | High | Medium | Start of with a small of integrated software and slowly expand the covered software. |
| Resource Constraints | Medium | High | Create a project schedule and keep to it. Prioritise core functions of the software. Seek feedback on development progress. |
| Legal Compliance and TOS | Low | High | Review GDPR and relevant regulations. Have a EULA and privacy agreement before using software. |
| Dependent API fail | Low | Very High | If a free and opensource API fails or stops working. Paid alternatives may be used incurring additional costs of the project and may result in an unaccounted cost for users |

# Market Solutions

## Software

### Avanan

A software that uses a cloud hosted email service. It evaluates existing relationships between senders and receivers to establish a level of trust. Additional It monitors Teams and Slack as they are emerging attack vectors.

### Cofense PDR

Leverages AI tools and security professionals to identify and mitigate phishing attacks as they happen. They quarantine possible attacks. They also utilise global intelligence on what attacks worked and how to mitigate them

### Mimecast

Several tools protect against phishing attempts. Such features detect suspicious links and redirects, attachments by removing them or rendering them safe via sandboxing. Mimecast prevents code-based attacks via sophisticated methods like QR-codes by opening them in the Mimecast cloud, this simplifies the deployment of their software and ensures prevention tools are always updated.

## Extensions

### PIXM phishing Protection

<https://chromewebstore.google.com/detail/pixm-phishing-protection/flomofhkchlalfciiibgbfcpolhmglai>

Uses AI to scan webpage for Logos and verifies if page is or is not owned by the stated website.

### Anti Phishing

<https://chromewebstore.google.com/detail/anti-phishing/npipcajipdkafdkellochifbfmppdalo>

Warns when accessing blacklisted websites

### Criminal IP: AI-based Phishing Link Checker

<https://chromewebstore.google.com/detail/criminal-ip-ai-based-phis/dhkbjdnlhahnffncheehbnoaecncdpdk>

Checks a link is malicious using who is and blocks access.

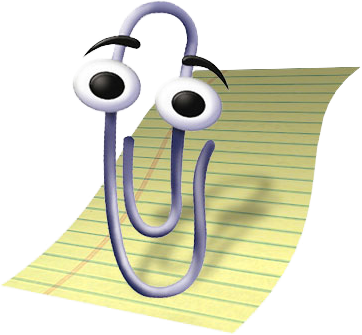
### Blue Arca PhishGuard - Phishing Protection for WebMail

<https://chromewebstore.google.com/detail/blue-arca-phishguard-phis/nbladngkelnbhcinepiogponadjggkcd>

Web mail phishing detector. No idea how it works does not state or have video evidence.

## Mascots

### Clippit

 A windows 97 Icon and mascot. Was designed to offer tips, provide shortcuts, and suggestions while user worked in office applications. Was a simple natural language processor to interpret user actions and provided context-based assistance.

### BonziBUDDY | Villains Fanon Wiki | FandomBonziBuddy

BonziBuddy was a desktop assistant that could tell jokes, talk and sing songs. It aimed to make using the computer more fun by providing web browsing help. The program used text-to-speech technology for this.

A graph with colorful rectangular shapes

AI-generated content may be incorrect.

# Gantt Chart

This is my initial Gant Chart for how long each stage may take. This may Change based on if task take faster or if I encounter Issues. I may also change the order of tasks based on what I see fit with further research and preparation.