ASTRO-NACSA CODING CHALLENGE

2024

Title:

SecureNet: Advanced Mobile Threat Detection

Group:

lyjYYDS

Prepared By:

Harraz Amsyar Khairul Amin

Koy Chang Wei

Goh Hong Xuan

Lee Yung Jie

Lim Jia Shin

Submission Date:

30th June 2024 5:00 PM

TABLE OF CONTENTS

INTRODUCTION

In this modern age, the interaction between humans and the internet has become more interconnected than ever. Although this facilitates tasks for us, it raises a lot of alarms regarding cyber attacks and threats that may cause detrimental damage. It is revealed in a study conducted by Surfshark that Malaysia is ranked as the 8th most infiltrated country in 2023. To illustrate, it is recorded that a shocking 144% increment of compromised user accounts, with a devastating total of 494699 in the third quarter compared to the previous quarter. Day by day, the online activities performed by the user increase exponentially, therefore requiring the need to solve this issue. Additionally, it is proved that there is a significant relationship between illegal video streaming and malware. Users who are involved with illegal streaming sites are 3.5 times more likely to get infected, in comparison to other domains.

Our mobile application is developed in response to emerging cybersecurity threats and is developed to identify malicious software and risky URLs. We aim to provide a seamless, user-friendly interface on both Android and iOS platforms, while processing and analyzing URLs and files to detect potential threats, ensuring users receive timely and accurate information.

In Astro-NACSA Coding Challenge 2024, our team aims to address a lot of worrying cybersecurity issues that may harm users in any shape, way, or form. Besides safeguarding end-users against malware, our mobile application also raises awareness regarding the potential risks that are related to illegal streaming and web activities. This initiative is a call to action for university students to apply their coding skills to confront and mitigate the growing cyber threats in our digital landscape.

LANGUAGES

Mobile Apps Frontend

We have selected Flutter as our frontend solution for malicious URLs and malware detection mobile applications. Flutter is an open-source UI software development kit created by Google for building natively compiled applications across mobile, web, and desktop from a single codebase. Having this capability cross-platform is a massive benefit for us, helping to improve our development speed and ensuring that users have the same experience on both Android and iOS platforms.

Flutter comes with a rich set of pre-built, and high-quality widgets that we can customize for an interactive UI. It is particularly essential in any cybersecurity application as it requires anything from the screen to be clear enough for use. Hot reload in Flutter is a very powerful feature that helps us develop and test applications quickly by showing changes within a fraction of a second to update our running app, hence speeding up the iteration process. Moreover, Flutter offers top-tier performance, providing a smooth, responsive, and seamless user experience while using our mobile application.

Server Backend

We opted to use Python for the backend of our malicious URL and malware detection implementation. Python is one of the most versatile and powerful programming languages we have today, celebrated for its simplicity, and readability, as well as an extensive array of libraries and frameworks that speed up development. In this particular example, we are building out our server using Flask, a lightweight WSGI web application framework in Python. Since we are going to be developing RESTful APIs for our Flutter frontend, Flask is the perfect choice as it provides a simple and flexible means of making communication between our Flutter frontend and the backend.

We use Python for building because it has an extensive set of libraries like requests, and TensorFlow which allow us to use complex threat detection algorithms as well as analyze data. The backend will process requests made within the mobile application, as well as data processing and responses about whether a particular URL is safe or malicious to provide our end-users with the most timely correct information on how to thoroughly protect its devices.

Integration

Combining Flutter and Python in our project will result in a powerful as well as scalable solution for discovering malicious URLs plus malware. The Flutter frontend provides a seamless user experience for users to interact with it and the Python backend processes complex security checks and analytics. This mix utilizes the benefits of both technologies to develop a high-performance application with an emphasis on security and efficiency. With this combo of Flutter and Python, we are able to deliver a comprehensive cybersecurity tool that is both user-friendly and highly effective in detecting threats, thereby enhancing security and peace of mind for our users.

FUNCTIONALITIES

i) Malicious IP address detection

Equips our mobile application with the capability to assess the reputation and safety of IP addresses. Users can input an IP address through our Flutter interface, prompting our backend system to conduct comprehensive checks. Through integration with AbuseIPDB's API, our application queries databases containing historical data on IP address behavior. This includes identifying IPs associated with malicious activities such as spamming, phishing, or malware distribution. By leveraging these external resources, our application provides users with insights into the trustworthiness of IP addresses, thereby mitigating potential security risks associated with connecting to suspicious or compromised hosts

ii) Malicious URL detection

This functionality aims to safeguard users against accessing potentially harmful websites. Upon receiving a URL input from the user via our intuitive Flutter interface, the application initiates a series of security checks. These checks include examining the URL structure, performing domain reputation analysis, and querying external threat intelligence databases such as VirusTotal. Through integration with VirusTotal's API, our application fetches real-time threat assessments, including malware detection, phishing risks, and suspicious behavior associated with the provided URL. This functionality ensures that users can browse the internet securely by being informed of potential threats before accessing suspicious links.

iii) Malware detection

Our feature empowers users to verify the safety of files before opening or downloading them onto their devices. Leveraging advanced scanning capabilities embedded within our backend server, the application accepts file uploads from users via the Flutter interface. Upon receiving a file, our system initiates thorough malware analysis using VirusTotal's API. This analysis encompasses signature-based scanning, heuristic analysis, and behavioral monitoring to identify any malicious code or known threats present within the uploaded file. By providing users with immediate feedback on potential malware risks, our application ensures that file handling remains secure and prevents inadvertent exposure to harmful software.

iv) Phishing IP address reporting

Our Report Phishing IP Addresses feature empowers users to contribute to the collective effort of combating online threats. If a user encounters a suspicious IP address during their online activities, they can swiftly report it through our mobile application's intuitive interface. Upon submission, the application securely transmits the reported IP data to our backend server. The server logs these reports and may take further actions, such as notifying relevant security authorities or organizations tasked with combating cybercrime. By facilitating user-driven reporting of phishing IP addresses, our application not only enhances individual awareness of cybersecurity threats but also contributes to the broader community's efforts to maintain a safer digital environment.

ARCHITECTURE

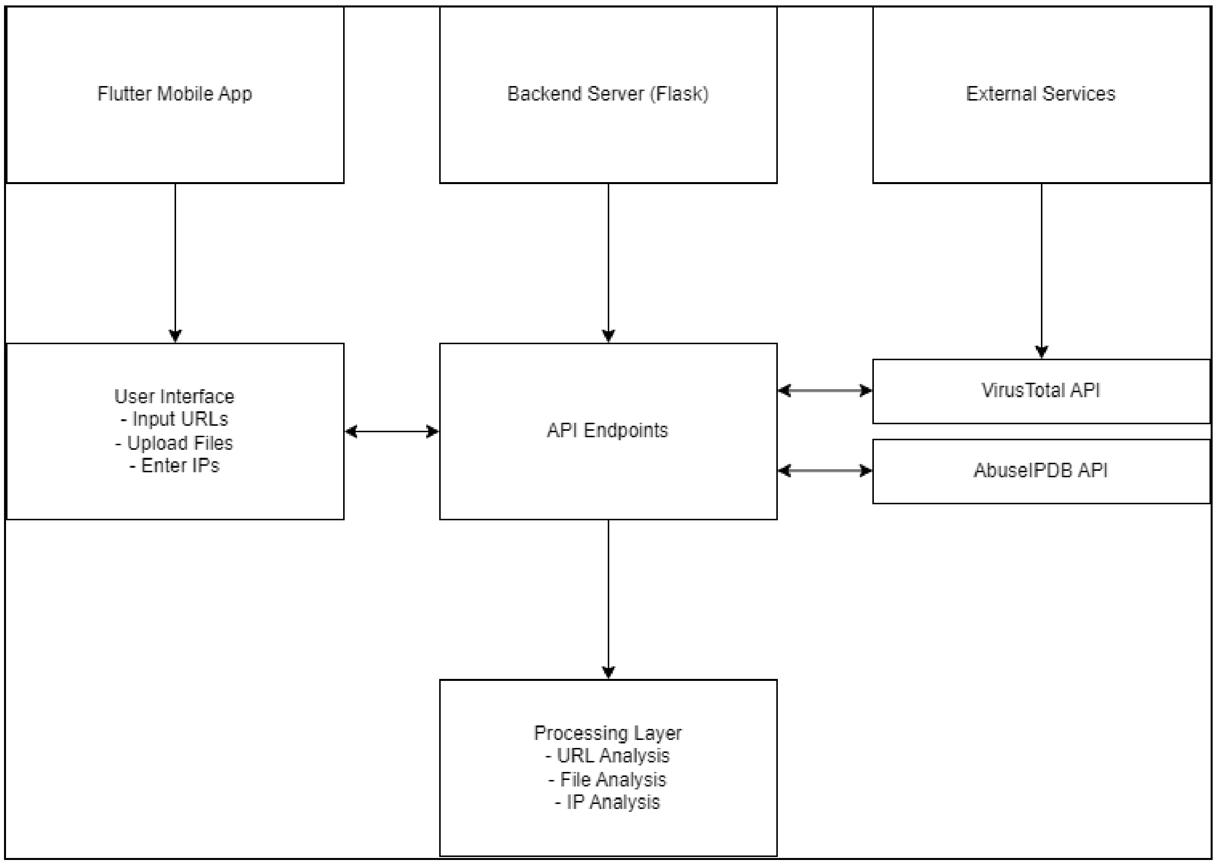


Figure 1.0 Architecture Diagram

Here is a further explanation of the architecture:

i) Flutter Mobile App

Users input potentially malicious URLs, upload malware, and put in IP addresses. The communication layer will then send requests to our backend server and also receive responses from it.

ii) Backend Server (Flask)

There are 4 API endpoints to detect malicious URLs, malware, and IP addresses, as well as report IP addresses. The processing layer will utilize libraries and tools in order to analyze if malicious content.

iii) External Services

- VirusTotal API
    - This scans IPs, URLs, and files.
- AbuseIPDB API
    - This reports IP addresses that are associated with phishing.

The flow of interaction is demonstrated below:

1. Users input IP addresses/URLs or upload files through the Flutter app.
2. The app sends a request to the designated API endpoints.
3. The requests are processed by the backend server.
4. After processing, results are returned to the Flutter app and users will know if whatever they submitted is safe or marked as malicious.

There are several benefits of our architecture such as:

i) User-friendly

We provide an easy-to-use interface for people to access cybersecurity functionalities.

ii) Reliability

Our results are reliable as threat intelligence and reputation checks that we use are established; VirusTotal and AbuseIPDB.

iii) Scalability

New features or additions can be made as our architecture is designed to be modular and scalable in the future.

INTERFACES

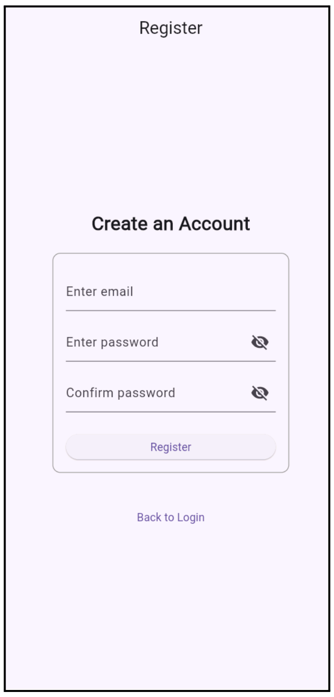


Figure 2.0 Register

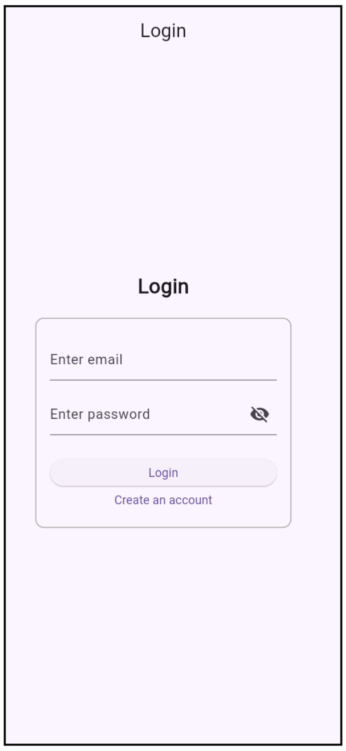


Figure 3.0 Login Page

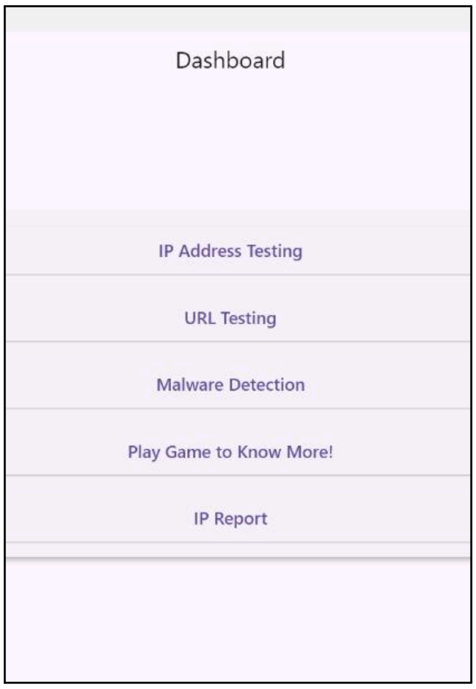


Figure 4.0 Dashboard

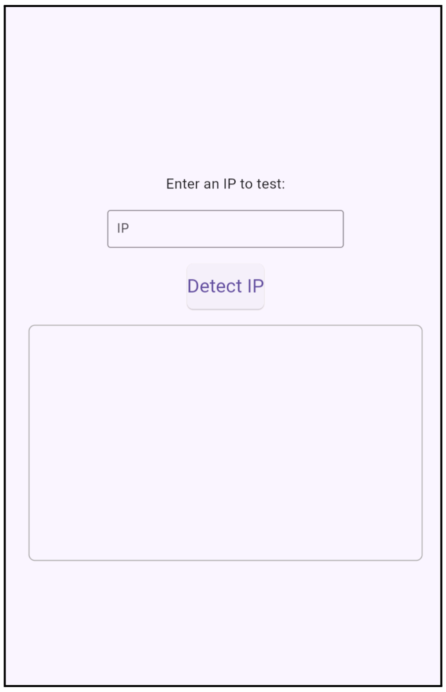


Figure 5.0 IP Address Testing

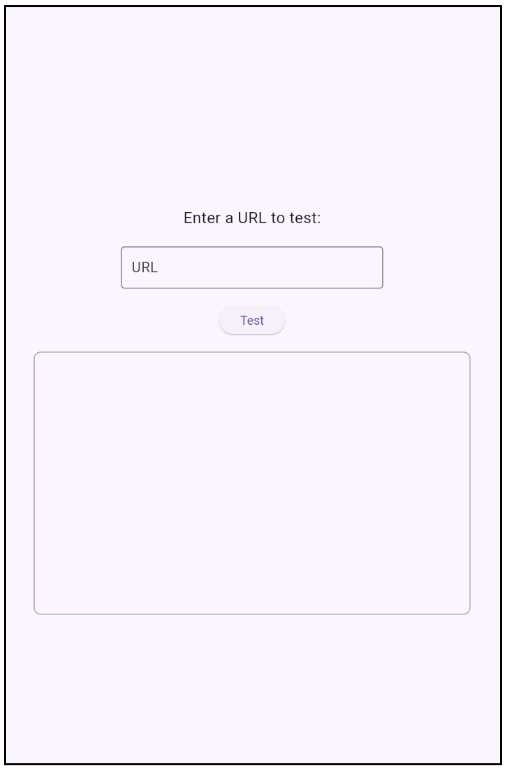


Figure 6.0 URL Testing

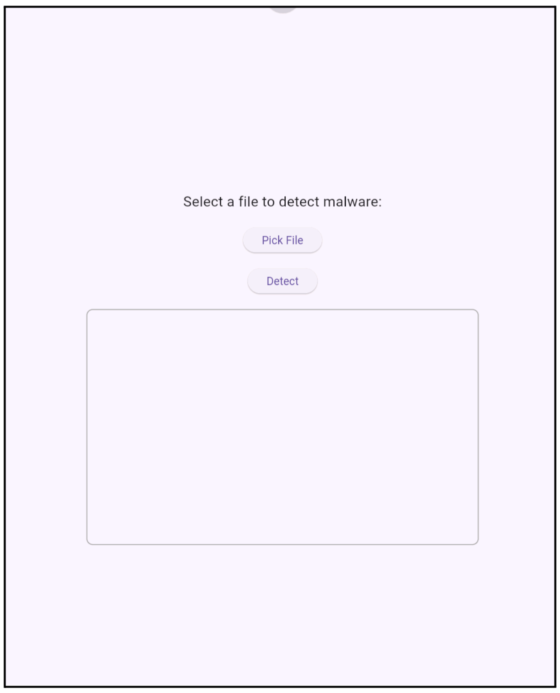
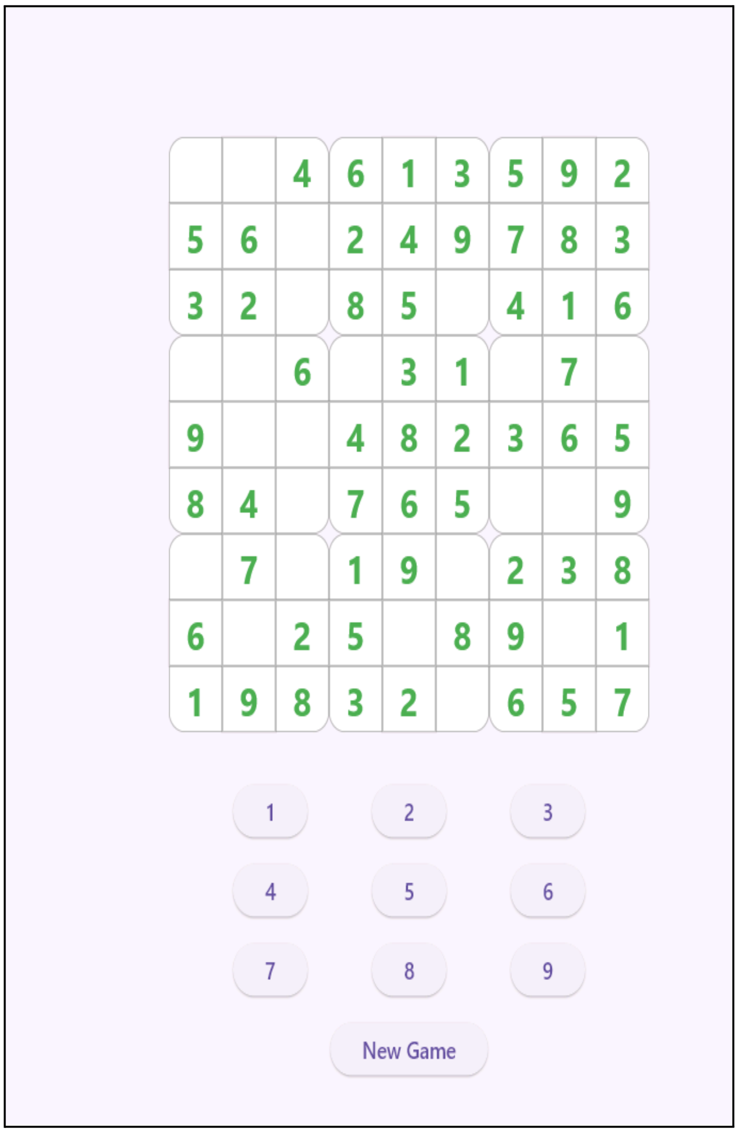


Figure 7.0 Malware Detection



		4	6	1	3	5	9	2
5	6		2	4	9	7	8	3
3	2		8	5		4	1	6
		6		3	1		7	
9			4	8	2	3	6	5
8	4		7	6	5			9
	7		1	9		2	3	8
6		2	5		8	9		1
1	9	8	3	2		6	5	7

Figure 8.0 Play Game

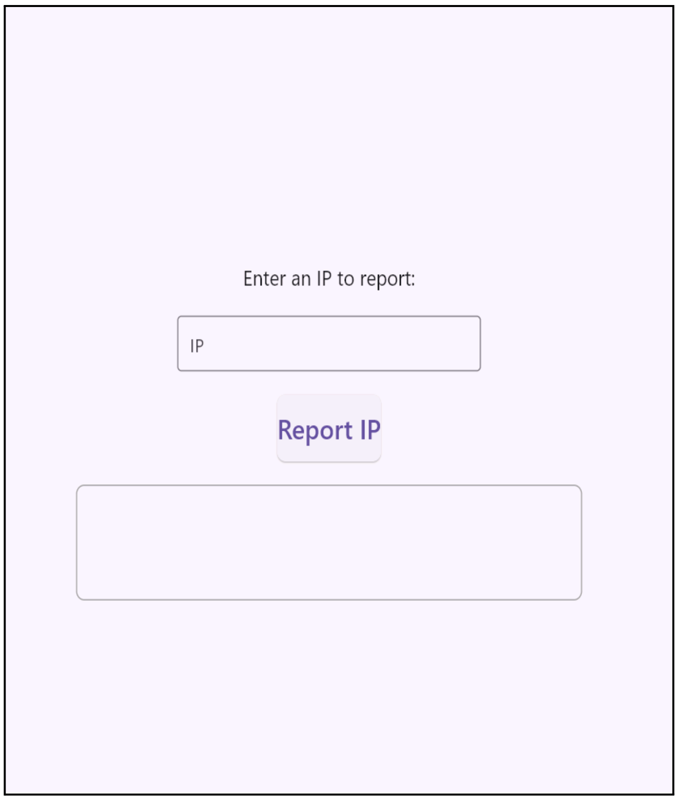
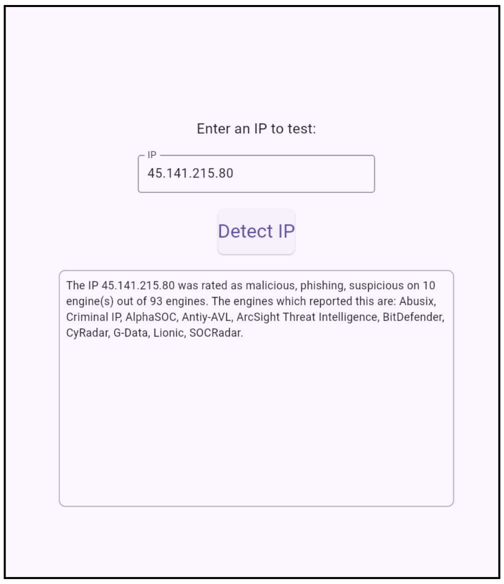


Figure 9.0 IP Report

TEST CASES

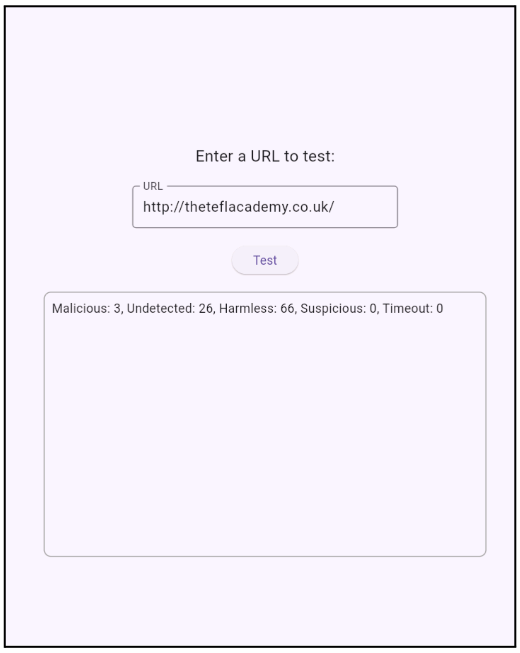
To validate our solution, we have included a few test cases for each of the functionality of our mobile application.



45.141.215.80

The IP 45.141.215.80 was rated as malicious, phishing, suspicious on 10 engine(s) out of 93 engines. The engines which reported this are: Abusix, Criminal IP, AlphaSOC, Antiy-AVL, ArcSight Threat Intelligence, BitDefender, CyRadar, G-Data, Lionic, SOCRadar.

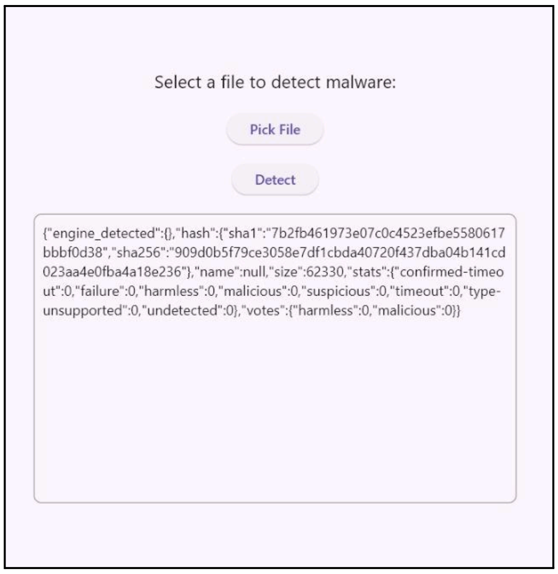
Figure 10.0 Malicious IP Address Detection



http://theteflacademy.co.uk/

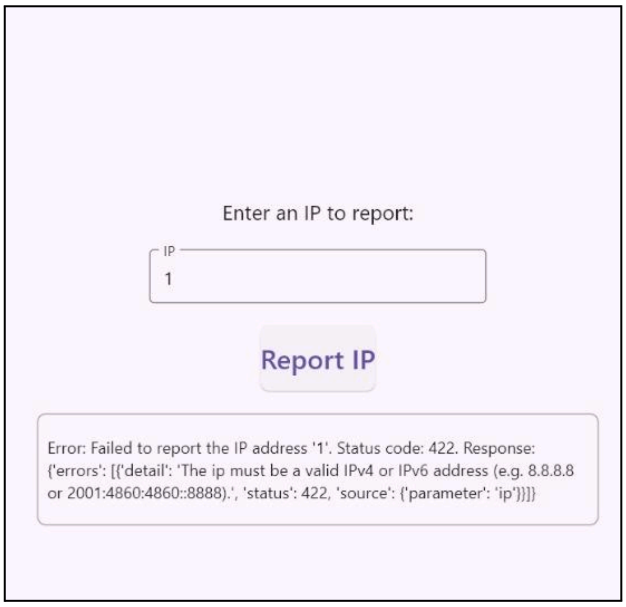
Malicious: 3, Undetected: 26, Harmless: 66, Suspicious: 0, Timeout: 0

Figure 11.0 Malicious URL Detection



```
{"engine_detected":{},"hash":{"sha1":"7b2fb461973e07c0c4523efbe5580617
bbbf0d38","sha256":"909d0b5f79ce3058e7df1cbda40720f437dba04b141cd
023aa4e0fba4a18e236"},"name":null,"size":62330,"stats":{"confirmed-timeo
ut":0,"failure":0,"harmless":0,"malicious":0,"suspicious":0,"timeout":0,"type-
unsupported":0,"undetected":0},"votes":{"harmless":0,"malicious":0}}
```

Figure 12.0 Malware Detection



IP: 1

```
Error: Failed to report the IP address '1'. Status code: 422. Response:
{'errors': [{'detail': 'The ip must be a valid IPv4 or IPv6 address (e.g. 8.8.8.8
or 2001:4860:4860::8888).', 'status': 422, 'source': {'parameter': 'ip'}}]}
```

Figure 13.0 Phishing IP Address Reporting