

Automated Phishing Detection Framework Leveraging Integrated Threat Intelligence and Multi-UserAgent Analysis

MSc Research Project
MSc Cybersecurity

Yuvaraj Mohan
Student ID: x22200142

School of Computing
National College of Ireland

Supervisor:	Raza Ul Mustafa
Industry Supervisor:	Colm Gallagher

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Yuvaraj Mohan
Student ID:	x22200142
Programme:	MSc Cybersecurity
Year:	2024
Module:	MSc Research Project
Supervisor:	Raza Ul Mustafa
Submission Due Date:	16/09/2024
Project Title:	Automated Phishing Detection Framework Leveraging Integrated Threat Intelligence and Multi-UserAgent Analysis
Word Count:	8261
Page Count:	23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	16th September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Automated Phishing Detection Framework Leveraging Integrated Threat Intelligence and Multi-UserAgent Analysis

Yuvaraj Mohan
x22200142

Abstract

This thesis proposes an approach that combines multiple threat intelligence sources in the Tines platform for enhanced phishing detection. It is aimed to improve the identification and examination of multiple types of Indicators of Compromise (IOCs) originating from e-mails and PDF files to respond to the rising complexity in phishing. The system leverages URLScan.io for behavioral analysis of the identified URLs, the IPs and different file hashes are analysed through VirusTotal for multiple engine scan, EmailRep.io for the real-time email reputation measurements and Hybrid Analysis for PDF files' examination in an controlled environment. The evaluation of the system was conducted through three real-life case studies. The system demonstrated 95% detection accuracy for the malicious emails with a false positive rate of 5%. In the URL analysis, the system was able to successfully identify 80% of suspicious URLs through multiple user agent simulations, uncovering potential cloaking techniques. For the PDF analysis, the system was able to achieve 100% accuracy in detecting malicious content, utilizing Hybrid analysis to flag files with known malicious behaviors. These results confirm the system's efficacy in identifying phishing threats across various attack vectors. Future development of the framework will focus on incorporating machine learning algorithms and additional threat intelligence feeds to improve its adaptability to emerging phishing tactics and reduce dependency on external tools.

1 Introduction

Phishing is perhaps one of the most daunting cybersecurity threats to this date, as it targets the human weaknesses instead of relying on advanced technical loopholes and lures them into providing sensitive information or download malicious payloads through manipulation. Over the years, there has been a shift of the phishing techniques used which is making them much harder to detect and prevent such attacks through conventional approaches.

1.1 Research Background

Initially, the phishing schemes were unsophisticated and involved a large number of emails with poor construction and straightforward social manipulation techniques. But today, these attacks are no longer limited to these techniques and more often involve crafted

messages targeting specific individuals, realistic brandings and complex obfuscation techniques in the form of links or email attachments. These innovations have rendered the earlier techniques of detection, like the content filters and blacklists ineffective in handling today's modern kinds of phishing threat Altwaijry et al. (2024).

The increasing reliance towards email as a primary tool in communication both for personal and professional use, with millions of emails exchanged on a daily basis, provides cybercriminals with ample chances to carry out phishing scams. As per recent study, phishing acts as one of the main attack vectors used in majority of the data breaches. This demonstrates that the existing solutions such as blacklists and content based are now outdated as they are primarily reactive and focuses on known phishing signatures. They often fell short when it comes to dealing with zero-day or spear-phishing attacks Alhaidari et al. (2022).

Due to the dynamic evolution of the phishing threats, cybersecurity studies have shifted their focus to incorporate newer and advanced detection technologies like machine learning, behavioural analysis and real-time threat intelligence to improve the detection accuracy and to minimize the false positive rates. These approaches provide better performances in detecting subtle phishing indicators which are hard to find with static analysis. For example, machine learning can be used to analyze the patterns across emails and its attachment to detect anomalies. While the Behavioural analysis tools such as URLScan.io can be used to simulate user interactions across multiple user agents to determine whether the URL is malicious.

1.2 Problem Definition

Despite advancements in technological solutions, many of the existing tools have critical shortcomings. Reliance on detection procedures like blacklists and signature-based detection has its drawbacks as they fall short on preventing zero-day attacks which exploits the vulnerabilities that are not yet recognized or defined. In Spear-phishing attacks, personalized messages are used to target specific individuals or an organisation and it remains as a significant threat due to its highly targeted nature. Such limitations point towards the need for a more holistic, multi-layered solution that can detect different threat vectors including URLs, file attachments, and sender reputation while at the same time responding to emergent tactics used by attackers.

The existing techniques for detecting phishing messages are relatively crude, pointing to the need for better technologies that can counter the current strategies used by phishing attackers. There is a strong demand for detection systems that would be able to monitor the activity of URLs, evaluate the credibility of the e-mail sender in real time and inspect the content of the e-mail attachments, for example, PDFs for the presence of threats. Integrating various tools such as URLScan.io for behavioural analysis, Virus-Total for multi-engine scanning, Hybrid Analysis for sandboxing, and EmailRep.io for email reputation score, provides a well-rounded approach to effectively addressing these challenges.

Research Question

- How can the integration of advanced phishing detection tools and techniques enhance the detection and prevention of phishing attacks, particularly in the context of evolving phishing tactics and zero-day threats?

Research Objective

- Develop a phishing detection system architecture utilizing various threat intelligence tools and approaches.
- Construct a system that actively identifies phishing scenarios with high accuracy and Ensure the system evolves in response to changing threat landscapes.
- Evaluate the system’s effectiveness in diverse conditions to ensure minimal misidentification.

This paper discusses work related to this phishing detection framework in Section 2. In Section 3, the research methodology employed in this framework is detailed. The framework’s design is explained in Section 4 followed by their implementation in Section 5. The outcome of the project is evaluated using case studies in section 6. Section 7 discusses the conclusion and future work of this report.

2 Related Work

The evolution of phishing detection has been deemed remarkable since many techniques and tools have been used to detect phishing due to increased advancement in the attacks. As the threats to the cyberspace are always evolving and improving, authors and researchers have focused on various ways to improve the performance of the detection of phishing. This section gives a overview on the more relevant research and development that has occurred in the analysis of phishing detectors, especially pointing out vital tools and approaches that characterize the field as it is today. The review also seeks to identify the advantages and shortcomings of these approaches to pave way for the methodological approaches used in this thesis.

2.1 Evolution of Phishing Detection Techniques

The traditional method of phishing detection was based on the heuristics and blacklist and this was only good for directly identifying of known threats. However, these methods were not very effective in identifying the new or zero day phishing attacks or phishing sites which are yet to be reported, most of the time the solution was delayed to avert the problem. A key limitation of these methods is their reliance on static data, leading to delayed responses when facing newly created phishing sites. For instance, blacklists can only block threats if and only they are reported which leaves the systems vulnerable until those phishing URLs are detected and added to the list. Moubayed et al. (n.d.) highlight this gap, emphasizing that these systems were not timely in preventing new phishing threats.

In response to the shortcomings of blacklists, phishing detection systems began incorporating machine learning (ML). The supervised learning models such as Support Vector Machines (SVM) and Random Forests started categorizing the emails and the URLs based on the feature of the content and structure quality Rao and Pais (2019). While these models were more flexible than the earlier black-list based models they needed constant updating in order to be effective as their effectiveness directly correlated to the quality of the training data.

From this idea, content-based filtering that utilizes deep learning algorithms such as BERT through natural language processing gained much appeal for detecting phishing

through phrases and suspicious language anomalies. However, this approach did not work well when the sites were impersonations of the genuine sites, for instance, phishing sites. Visual similarities, rather than textual content, became the defining trait of these attacks. Thus, while deep learning improved detection rates, it was not enough to fully capture the nuanced tactics employed by sophisticated attackers.

Behavioral analysis such as User Behavior Analytics (UBA) and Heuristic URL Analysis turned out to be a more effective approach as it did not only refer to URL content, but URL behavior, allowing the identification of phishing web sites using dynamic content or cloaking techniques. Verma and Das (2017) demonstrated the value of behavioral analysis in detecting phishing attempts that bypassed both content-based and static methods. Nevertheless, these tools were often resource-intensive, requiring significant computational power to simulate user interactions across multiple platforms.

This background leads to the use of tools such as URLScan.io, VirusTotal, Hybrid Analysis, and EmailRep. io. These tools are well integrated into this thesis and are arguably the most recent and effective tools used in fighting phishing, each having different features that add to the overall security of the system. This next part covers this and explains their functionalities, implementation into different processes, as well as the benefits that allow them to fight phishing threats effectively.

2.2 Dynamic URL analysing using URLScan.io

URLScan.io is a specialized tool for dynamic URL analysis that represents an important building block in phishing detection, especially against evasion techniques. While the focus of static analysis lies mainly in the structure or direct content of a URL, URLScan.io performs real user interactions in a controlled environment. This allows it to keep track of various activities emanating from the URL, such as network requests, changes in the DOM, or loading of external resources, which may be critical in identifying malicious behavior Opara et al. (2024).

One of the key strengths that make URLScan.io stand out is its ability to identify phishing sites that attempt cloaking techniques. Cloaking means serving different content to users on various platforms, such as benign content to web crawlers while delivering malign content to users. URLScan.io does this through the use of several simulated user agents, including multiple browsers and devices. Thus, it is able to deliver fine-granular insight into behavior variations of a URL. This must be so to unveil phishing sites that could otherwise remain under the radar Kemp (2023).

2.3 VirusTotal: Evolution in Threat Detection

Primarily, VirusTotal was established in 2004 and has now grown to become one of the most popular tools in the field of cybersecurity, commencing its existence as a database of antivirus engines. By being able to scan files and URLs with multiple Anti-virus engines, it offered a detailed risk report that no single Antivirus engine viewpoint could offer. This multi-engine approach was particularly innovative especially in terms of consolidating threat intelligence Khonji et al. (2013).

To begin with, VirusTotal was relying on signature-based methods of detection which means that it maintains database of signatures which represent patterns of malicious programs. This method was quite effective for the known threats as listed in the catalog but had issues when dealing with the zero day threats which are a new malware with no

signature associated to it. To counter this, VirusTotal adopted heuristic analysis which is the process of assessing the behavior of files or URLs in order to determine suspicious activities despite not having set intellectual properties. For instance, any attempt to write or delete into the system files, attempt to connect to other servers on the network are labeled as suspicious.

As the threat level of cyber threats grew higher and higher, VirusTotal continued to strengthen its functionality, which incorporated ML into its analysis. ML models are capable of recognizing patterns in a large number of data sets, thus allowing VirusTotal to identify new complex phishing campaigns and malware Singh et al. (2024). Some of these models are the kind that can analyze previous scans and estimate the probability that a new file or URL is malicious, and thus greatly improve VirusTotal’s detection of zero-day threats.

The latest significant development in the VirusTotal approach is its crowdsourced threat intelligence Jesus et al. (2024). The current application enables users to upload files and URLs of their suspicion that may contain a virus; the files are checked by antivirus engines within VirusTotal and the results are shared with other participants in the cybersecurity field. This crowdsourced model greatly enhances the threat base of VirusTotal as new threats are easily detected and fed into the system knowledge base.

2.4 Dynamic Sandboxing for Threat Detection

Hybrid Analysis is a sandbox based tool that can be used for detailed examination of files, hence proving to be useful in analyzing emails with PDF attachments such as those used in phishing Alhaidari et al. (2022). Static Analysis is analysis of files of code and data for virus signatures whereas Hybrid Analysis is execution of files in a controlled environment. This dynamic approach allows it to monitor the file in real-time and provide information on any actions that the file may be planning to execute Juwono et al. (2015).

The sandbox environment replicates the standard operating mode of a typical user to monitor its interactions with computers, including network traffic, changes in content, or efforts at testing a system’s vulnerabilities. The advantage of such an approach is that it allows identifying various actions performed on a file in real-time necessary for detecting advanced threats that may be concealed within the ordinary files Zhang et al. (n.d.). For example, a PDF that contains scripts that try to establish connection to other servers or try to change system files will be detected during sandboxing.

2.5 Analysing Email Reputation

EmailRep.io improves phishing detection by introducing an Email Address’s reputation analysis, unlike most of the existing solutions that heavily rely on content-based filtering and blacklisting Felegyhazi et al. (2010). While these existing approaches work well to counter threats already known, it does not work well when trying to identify new phishing domains and email addresses that are newly created and thus does not appear on the blacklist.

One of EmailRep.io’s most significant advancements is its real-time risk scoring. When an email address is encountered, it is compared to all the prerequisites, and a risk coefficient is assigned to it based on the possibility of the email’s participation in a phishing campaign Rajab et al. (2007). This real-time assessment is crucial for preventing spear-phishing attacks, where attackers use personalized and targeted approaches to bypass

traditional detection methods. By assigning a risk score before the email reaches the user, EmailRep.io helps preemptively filter out potentially dangerous communications.

Today, EmailRep.io is a powerful tool for phishing and especially for reputation analysis in real-time. The fact that it uses traditional database, current threat feeds, and machine learning for email sender credibility gives a very accurate assessment. This singles it out as a valued asset especially to organizations that are in the process of strengthening their security against any phishing threats, more so the more advanced forms of phishing such as spear-phishing.

When integrated into security framework, these threat intelligence tools provide comprehensive security framework which provides multi-layered protection against phishing attacks from the different threat landscapes Garera et al. (2007). This paper, thus, proposes a multilayered approach that substantially strengthens an organization in its effort toward successfully identifying, analyzing, and thereby stopping such phishing attempts before they cause damage. The related work detailed here forms the background of how these tools will fit within a complete phishing detection framework and is the backdrop under which their deployment and evaluation will be treated in this thesis.

3 Methodology

This particular section talks about the method chosen for developing, implementing, and evaluating the phishing detection system in the Tines platform. In this section a detailed description of the tools, the followed techniques, and the processes used in creating the automated, multi-layered defense system against phishing attacks Tripathy et al. (2021). The focused workflow is such as

- Email Analysis
- PDF Analysis

3.1 Email Analysis

Email Analysis Workflow is one of the most critical components of this phishing detection system, i.e., designed to systematically extract and analyze IOCs from provided suspicious emails. As the workflow figure 1 depicts, the system integrates multiple tools and techniques, which ensures that a comprehensive analysis is conducted.

3.1.1 Input Acquisition

The primary inputs for the email analysis in the Tines platform are received through three input methods as shown in the figure 2 Tines (2024). Each of these methods is intended to accommodate all the operational needs.

1. **Receive Email Action**¹: This approach is configured such that it automatically captures emails from a selected mailbox via IMAP.
2. **Webhook Integration**²: This provides for real-time data collection by feeding email content directly into the workflow from external systems.

¹<https://www.tines.com/docs/actions/types/receive-email/>

²<https://www.tines.com/docs/actions/types/webhook/>

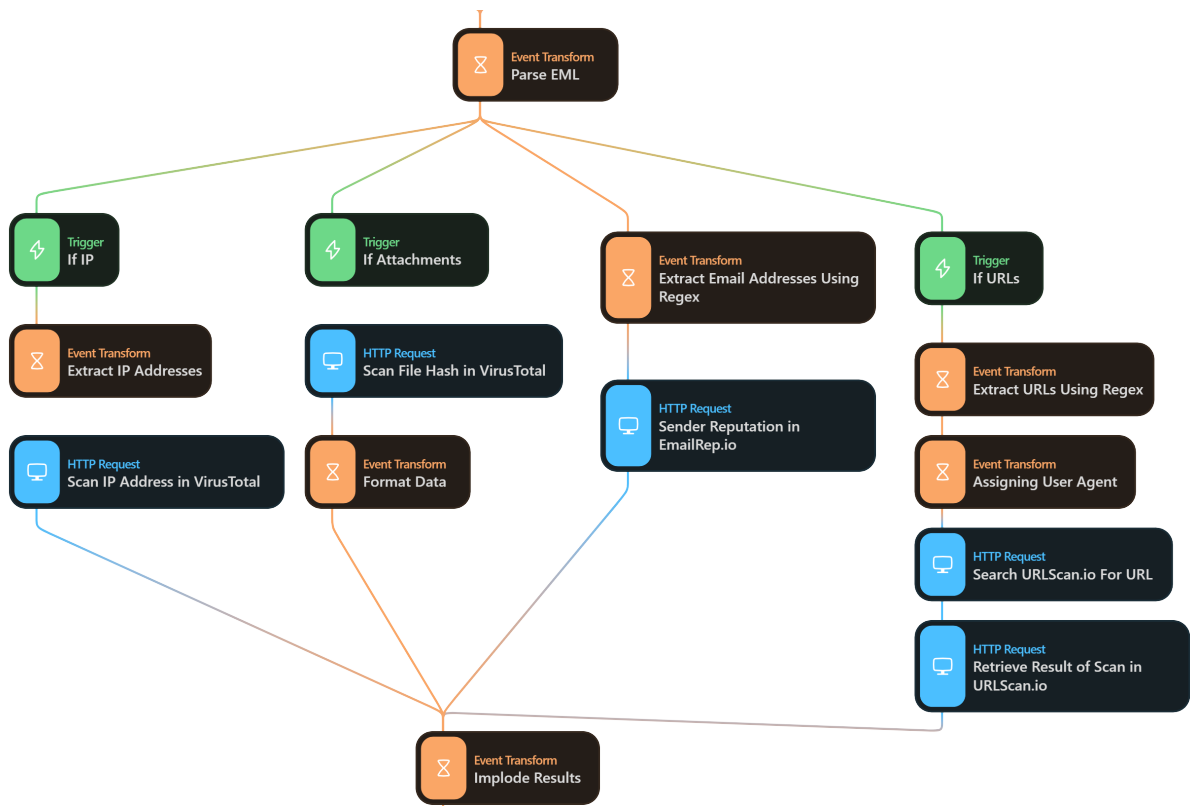


Figure 1: Overview of Email Analysis workflow

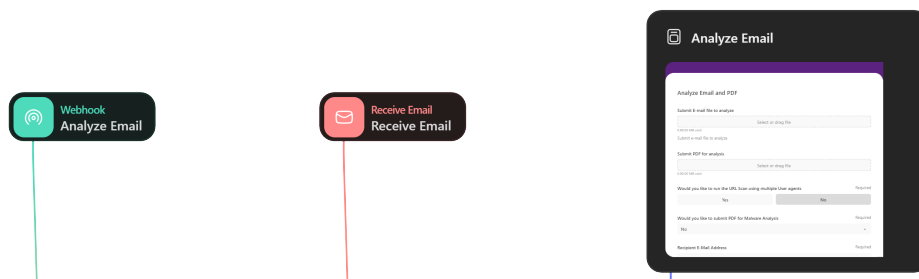


Figure 2: Input Acquisition

3. **Manual Submission via Tines UI³**: Security analysts have the ability to manually upload emails for analysis using Tines' user-friendly interface, allowing for ad-hoc investigations.

3.1.2 Email Parser

The email parser will take the email content received through one of the described input methods. It parses the e-mail into major components: headers, body contents, and attachments which is then transformed into structured JSON structure. This structured output is crucial for accurately identifying IOCs and preparing data for further analyses.

3.1.3 IOC Extraction and Analysis

After parsing the system gathers IOCs such as IPs, URLs, file hashes, and E-mail addresses using strong regular expressionsGibson (2024). The IOCs are then analysed by querying the following threat intelligence tools:

1. **IP Addresses and File Hashes**: The IP addresses and the file hashes obtained from the email content are then sent to VirusTotal. At VirusTotal, information about this component is analyzed taking into account the results of a large number of antivirus engines and URL scanning facilities that determine the reputation of these elements. It compares the IP addresses with a list of black listed entities and scans file hashes to know the reputation of associated files.
2. **Email Addresses**: The email address extracted from the email are then forwarded to EmailRep.io, a tool specifically designed to analyze the reputation of the e-mail address. EmailRep.io takes into consideration a number of parameters such as the domain from where the message originated, its usual activity, and the role it played in previous identified phishing or spam campaigns. By studying these elements, EmailRep.io provide rating for each identified email address based on this analysis.
3. **URLs**: URLs extracted from the body of emails are analyzed through URLScan.io, by various user agents. URLScan.io impersonates various browsing environments to check how the URLs behave under various circumstances. This is one of the key ways towards detecting phishing sites which could use cloaking techniques where different content would be served to security tools and real users.

3.2 PDF Analysis

3.2.1 Input Acquisition

In the PDF analysis workflow, input is acquired through two primary channels. As shown in figure 2, it allows users to upload PDFs through the Tines UI for analysis, which has the advantage of being flexible and fully user-controlled for finding possible threats in individual files. Furthermore, any PDF attachments discovered in the process of email analysis are fed into the PDF analysis workflow, thus making the two processes interconnected and makes sure to never to miss a potential phishing vector.

³<https://www.tines.com/docs/pages/>



Figure 3: Overview of PDF analysis Workflow

3.2.2 PDF Parser

The PDF Parsing Workflow is a key part of the whole phishing detection system allowing for analysis of the PDF documents and extraction of the Indicators of IOCs. Figure 3 provides the overview of the PDF Analysis workflow.

Step 1: Processing and Content Extraction The PDF Parsing Workflow starts with analyzing and processing the PDF document with a Python script. This script's intention is to scrape the PDF file, with an emphasis on the text and any hyperlinks, attributes, comments, and other features that may be concealed from the end user.

Step 2: IOC Extraction and Initial Analysis After the extraction of the text content from the PDF the second step would be to identify and extract IOCs from the text. This is done through the use of the iocparser.com, a tool designed to filter the extracted content and search for specific IOCs such as URLs, IPs, file hashes, or email addresses.

Step 3: Mapping, Formatting, and Displaying IOCs Once the IOCs are extracted, they are formatted, and the data is displayed on UI page that is accessible to users. This page gives the user an organized list of the IOCs found in the PDF for easy understanding.

Step 4: Further Analysis of Suspicious URLs In case if any of the extracted IOCs contain URLs, they are further analyzed. These URLs are then automatically fed as input to the URLScan.io process within the email analysis workflow. URLScan.io does behavioral analysis of URLs by emulating user interaction with them using multiple user agents to find malicious intent which is not easily apparent.

3.3 Sandboxing via Hybrid Analysis

In parallel with the PDF parsing process, another process with a Sandbox feature using Hybrid Analysis is applied for enhanced security analysis. After a PDF file is received, it is forwarded to Hybrid Analysis for execution in an emulated sandbox environment —thus,

identifying hidden malicious scripts or payloads and exploits that could otherwise remain invisible from a static analysis. This dynamic analysis captures the behavior of PDF which might include communications over the network or changes made in the file and hence offers a more detailed level of study. The findings of Hybrid Analysis are then put together with the initial outcomes of parsing in order to present a detailed findings of the threats that may be present in the PDF.

3.4 Consolidation and Reporting

The last action of the workflow is the aggregation and formatting of the analysis results. By employing HTML the system sorts IOCs including IP or URL addresses, email addresses, and file hashes and along with reputation scores enabling easy understanding of the information provided. For the PDFs the results obtained from the initial analysis and the Hybrid Analysis performed are both presented resulting to a comprehensive report. The last report is forwarded to the originating email for quick decision-making to avert the threats.

4 Design Specification

Tines is a cloud-based SOAR solution that enables an organization to execute sophisticated security operations tasks ⁴, as well as automate them. It enables security analysts to develop workflows that can address a variety of functions in cybersecurity without extensive programming or human interaction. Tines function through “Actions” which act as the base of these workflows ⁵. Every action is a pre-defined step that can perform tasks like sending HTTP requests, data manipulation, or interaction with other APIs. All these actions are highly customizable, allowing users to automate even the most trivial things such as response to an incident or to gather threat intelligence. The following figure 4 provide a summary of the different action types utilized within this workflow.

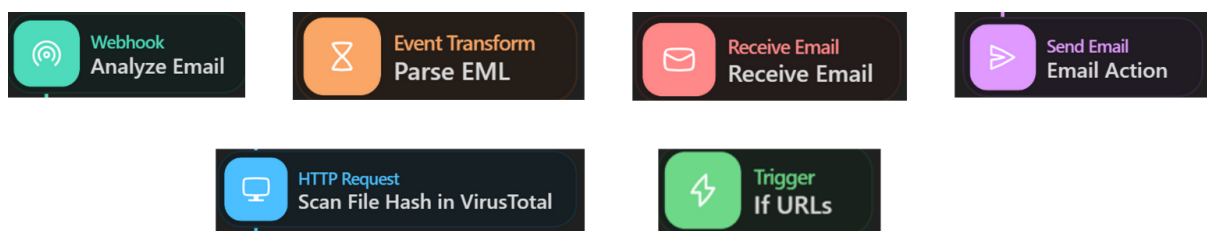


Figure 4: Tines Action Types

API keys that enable the querying of threat intelligence services such as VirusTotal and URLScan.io, Hybrid Analysis as well as EmailRep.io are setup within Tines to ensure they are easily integrated. To facilitate the receipt of emails and attachments, the system employs multiple methods: the receive email action type, webhook, and the ability of users to upload emails and their attachments by using the user interface.

In the Email analysis workflow, Upon receiving the test file and parsing them, the system extracts elements such as URLs, IP addresses, attachments, and email addresses

⁴<https://www.tines.com/>

⁵<https://www.tines.com/docs/actions/>

as IOCs. These IOCs are sorted into categories by using triggers, which are based on regular expressions matching particular data patterns.

After categorisation, the IOCs are processed further through threat intelligence services for thorough analysis. The URLs are passed on to the URLScan.io for detailed behavioural examination using multiple user agents to identify the potentially malicious activity. Evaluation of email addresses takes place using the EmailRep.io to determine the credibility of the sender based on known threat databases. Likewise, the IP addresses and file hashes are scanned on VirusTotal to identify their reputation.

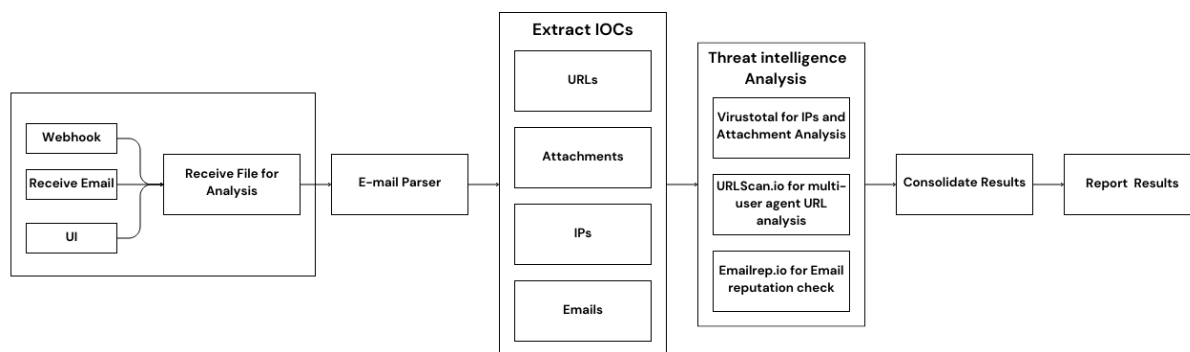


Figure 5: Email Analysis Workflow

In addition to the e-mail analysis workflow, this system also contains a separate workflow targeting attachments, such as PDFs. This specific workflow allows users to independently submit PDFs for analysis. A submitted PDF undergoes initial dissection through a Python script that extracts out the contents, including any IOCs such as embedded URLs that may be contained in the document.

If any URLs are discovered during this initial segmentation, they are sent to the e-mail analysis workflow that utilizes URLScan.io for further analysis. This email analysis workflow investigates its behaviour and reputation across multiple user agents. Further, the received PDF is sent to Hybrid Analysis which scans the PDF for any hidden scripts, viruses, malware, or any other embedded threats. The findings of this malware analysis are then incorporated back into the threat assessment giving a full picture of the risks likely to be pose by the email attachment. All the responses from these services are fed into the event transformation actions in Tines for formatting and consolidation of data into a unified report.

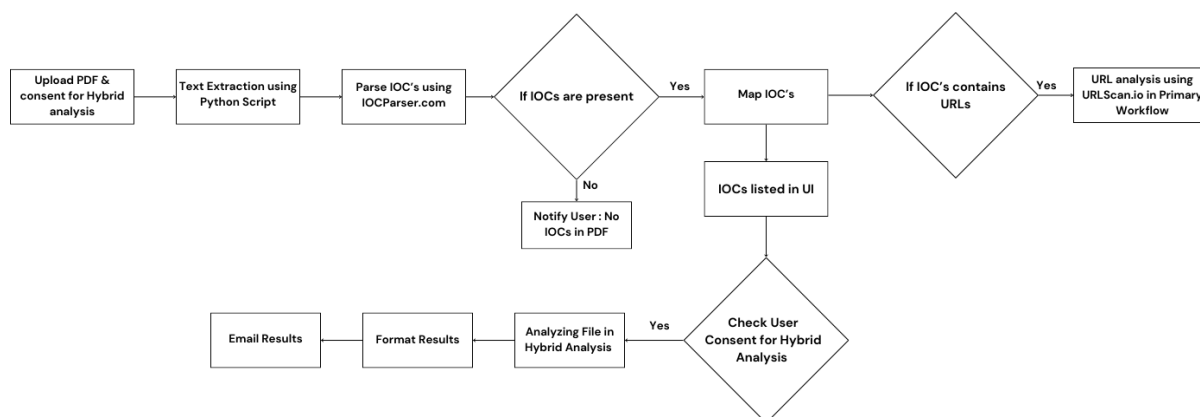


Figure 6: PDF Analysis Workflow

5 Implementation

5.1 Setup and Configuration

Tines platform was chosen due to its minimal coding automation capabilities and ability to integrate various external tools. The setup of this phishing detection framework started with creating and setting up Tines account and assigning proper roles to limit access. This configuration provided the required safeguard measures for the safety and integrity of the workflows and data that are being processed.

With the platform in place, a story was created in the Tines tenant to structure and automate these workflows efficiently. The security tools - VirusTotal, URLScan.io, EmailRep.io, and Hybrid Analysis were then integrated into the Tines story by securely storing their API keys as encrypted credentials. These keys were configured to allow seamless interaction with these tools through HTTP Request actions for analyzing IOCs such as IPs, URLs, email reputation, and attachments.

5.2 Input Configuration

In this framework, input acquisition is implemented through three distinct tines actions: webhook, receive email, and submission using user interface which is illustrated in Figure 2. For enabling the webhook action, a new URL and email address is set up in order to receive and forward incoming emails to the workflow. The receive email action uses IMAP to monitor a particular mailbox and retrieve the emails for analysis upon their arrival. Finally, manual submission through the Tines UI enables the user to drop emails or PDFs directly into the analysis workflow. This diverse input methods will ensure that the system is highly adaptable to all the different operational environments.

5.3 Email Analysis Workflow

5.3.1 Email Parser

The first important process in this analysis is to extract the content of the email. This process is done by using the BASE64 decoding and a message parsing function. This step in turn decodes the email content and delivers the content from its BASE64 format to the text form.

After parsing is completed, the contents of the email are output in a structured JSON format as shown in the figure 7. This JSON output categorizes every component of the email including the sender's email, subject, recipient's list, date and time, content of the email body, and any attachments. This categorization is convenient for further manipulations and analysis within the workflow.

Regex for IOC Extraction For extraction of specific IOCs such as emails, IPs, URLs, the framework uses regex pattern. Regex was used because of its effectiveness in extracting IOC consistently. The IOCs extracted from the use of these regex patterns and further analyzed using tools such as VirusTotal, URLScan.io, and EmailRep.io. Figure 8 provides the regex patterns applied for each type of IOC:

5.3.2 URLScan.io Implementation

This section discusses the implementation of URLScan.io and focuses mainly on the behavioral analysis of URLs to identify threats. The HTTP action is configured to

```

"mime_en": {
  "email": {
    "message_id": null,
    "subject": "Important Update Regarding Your Account",
    "from": null,
    "to": [
      "sotrecognizd@gmail.com"
    ],
    "cc": [
      "bounce@thcaltarides.co.uk"
    ],
    "date": "2024-08-26T17:19:48-08",
    "headers": { },
    "body": "<html xmlns:v='urn:schemas-microsoft-com:vml' xmlns:o='urn:schemas-microsoft-com:office:office'></html>"
  }
}

```

Figure 7: Parsed JSON Format of Email

Use case	Regex
Email	<code>\b[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}\b</code>
IP	<code>\b(?:[0-9]{1,3}\.){3}[0-9]{1,3}\b</code>
URL	<code>[A-Za-z]+\:\/\/[A-Za-z0-9-]+\.[A-Za-z0-9-]+(?:%& \?#\/= +)</code>

Figure 8: Regex Pattern

analyse the extracted URLs through a list of user-agents, including chrome, firefox and edge as shown in figure 9a. The payload of the request includes attributes like the URL and custom user agent assignment. This helps the framework to detect any cloaking techniques which present different content based on the user agent.

URLScan.io then analyzes these URLs and returns detailed reports on their behavior, including network requests and the executed scripts as shown in figure 9b as the response. The results are then parsed, categorized by their risk level, and integrated in the final result sent to the user.

5.3.3 VirusTotal Implementation

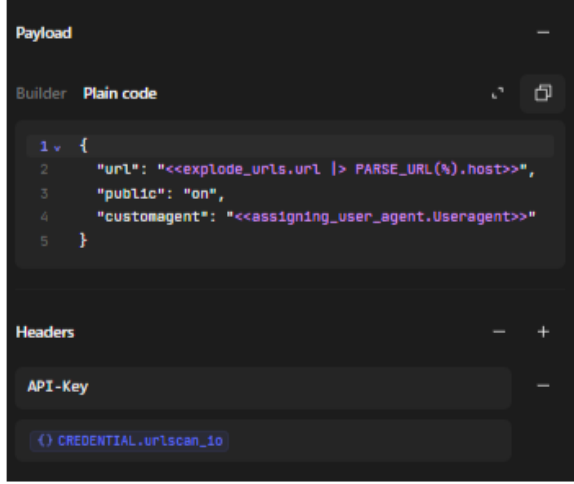
VirusTotal is integrated in the new flow of email analysis flow, to analyze attached files and IP addresses against threats. Tines's HTTP request action is configured to make API requests to send these IOCs for evaluation.

IP Address Analysis

For analysis of the IP addresses, GET request is set up in Tines to call the API of VirusTotal. The endpoint used is VirusTotal IP API, where the IP addresses are appended to the URL, as shown in the figure 10a. The API request passes the IP address obtained from the email body to VirusTotal to compare it with a pool of IPs confirmed to be malicious. The response also contains detailed information like the reputation score, network owner and the results of scans performed by other security solutions. This information is important in defining whether the IP address is involved in some crooked activities.

File Attachment Analysis

For the file attachments, similar approach is used. First, SHA-256 hash of the file is obtained and queried with VirusTotal via the endpoint VirusTotal Files API. This API request, as configured in Tines evident in figure 10b, compares the file hash with VirusTotal database that contains information on the numerous known viruses and malicious files. The file hash was chosen instead of scanning entire attachment because it will help to optimize performance by reducing processing time. In response, the tools offer information on the file's history, a scan by several antivirus engines, and any past incidents of malicious activity. The process is much faster than reading the whole file because the



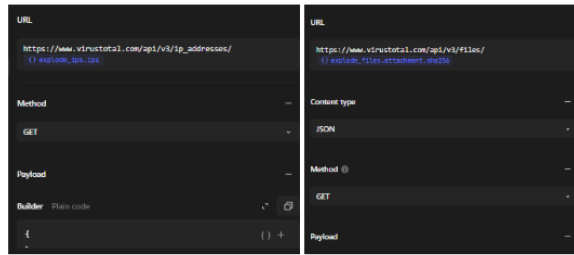
(a)



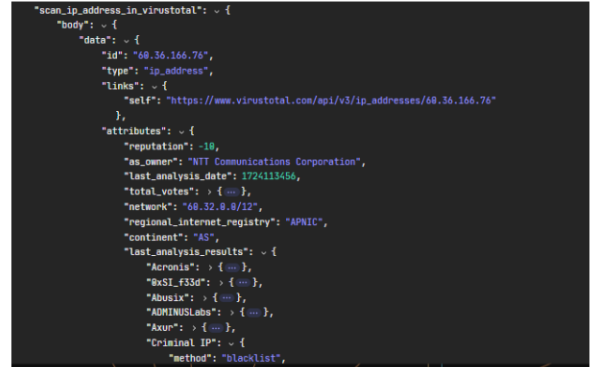
(b)

Figure 9: (a) URLScan.io Request (b) URLScan.io Response

program only looks at the hash, not the whole file. Results of both the analysis are categorized based on their risk level and integrated into the final report.



(a)



(b)

Figure 10: (a) VirusTotal Request (b) VirusTotal Response

5.3.4 EmailRep.io Implementation

The integration of EmailRep.io makes it possible to conduct real-time email reputation assessments to identify the ones that are associated with phishing. The use of EmailRep.io ensures that all the emails are assessed using a combination of historical data and real-time threat intelligence. This offers a robust mechanism for detecting spear-phishing attempts.

To Implement EmailRep.io into the workflow, A HTTP GET action is configured to make the API request. In the request URL, the extracted email addresses are fed as payload as shown in the figure 11a.

In response to the request, EmailRep.io returns a comprehensive response with the reputation data of the queried email addresses. The response consists of different parameters including Reputation Score, Suspicious Indicators and Domain Reputation. Using the response data, the email addresses are classified as either clean, suspicious or malicious as shown in figure 11b. These findings are consolidated and the results are forwarded down the workflow for reporting.

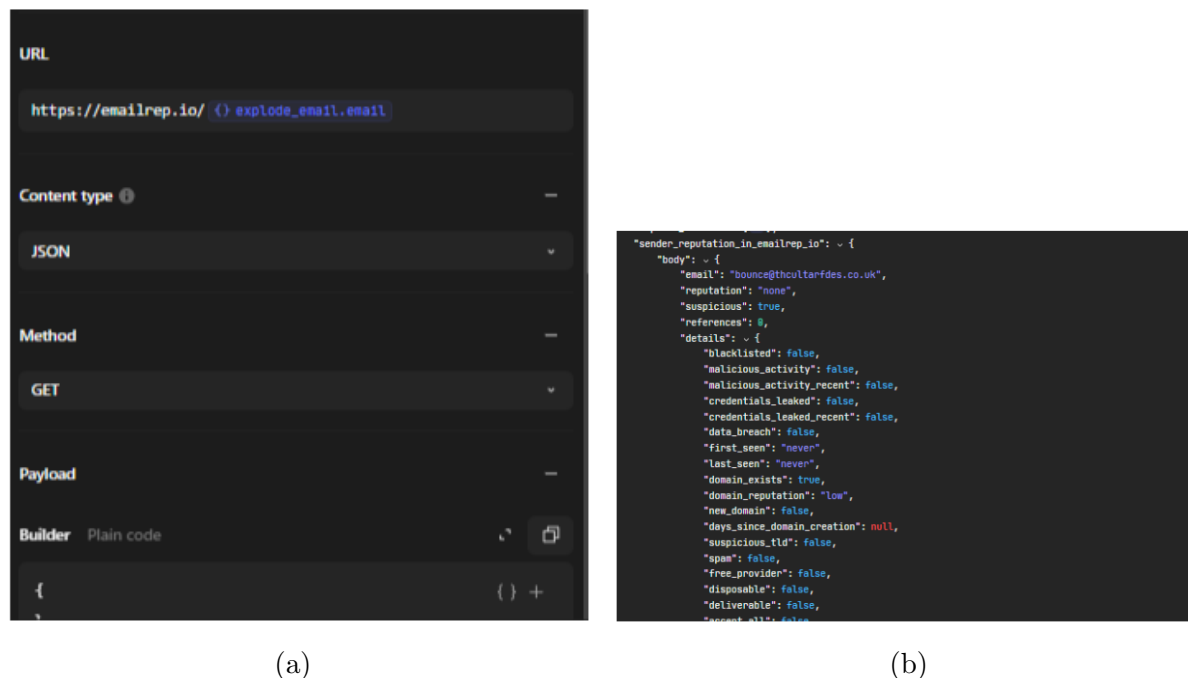


Figure 11: (a) EmailRep.io Request (b) EmailRep.io Response

5.3.5 Result Formation and Reporting

The outcomes of the IOCs' analysis are summed through a loop function. 'FLATTEN' in Tines was used to simplify the result structure by merging the nested arrays into a single array. Using loop function, key information such as IP address, email address, URLs along with their user agent and their reputation score of each result is consolidated.

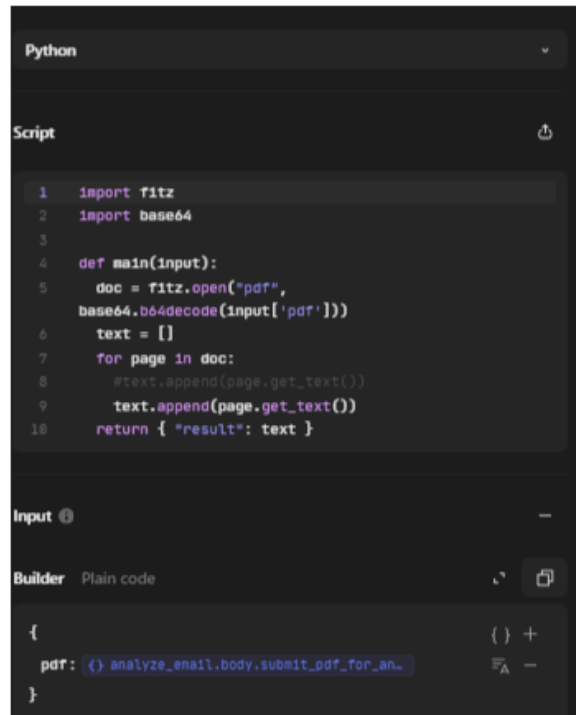
After consolidation, using tines event transformation action the data is formatted into an email compatible structure. This information is used to populate an email template and sent as a report to the intended user using send email action. The results are also displayed on the user interface (UI).

5.4 PDF Analysis Workflow

In PDF analysis workflow, the process is designed to extract IOCs from PDF attachments and analyze them through integrated threat intelligence tools. This process plays a essential role in detecting phishing attempts.

5.4.1 Text Parsing using Python Script

In this process, the BASE64-encoded PDF file is decoded using a python script, and the “fitz” model is then used to extract text from each page of the document as shown in the figure 12. The scripting in this step will help to increase flexibility, as the fitz model will allow the system to handle different types of PDF structures without sacrificing performance.

The image shows a screenshot of a Python script editor. At the top, there's a tab labeled 'Python'. Below it, the word 'Script' is visible. The main area contains a Python script with the following code:

```
1 import fitz
2 import base64
3
4 def main(input):
5     doc = fitz.open("pdf",
6                     base64.b64decode(input['pdf']))
7     text = []
8     for page in doc:
9         #text.append(page.get_text())
10        text.append(page.get_text())
11    return { "result": text }
```

Below the script, there's an 'Input' section with a minus sign icon. Underneath that, there's a 'Builder' section with the text 'Plain code'. At the bottom, there's a JSON-like structure for the input:

```
{
  pdf: {} analyze_email.body.submit_pdf_for_an...
}
```

Figure 12: Python Script for Text Parsing

5.4.2 IOC Extraction

This text is then fed as input into iocparser.com⁶ API via a POST request from which potential IOCs such as IP addresses, URLs, and file hashes are parsed. After IOCs are extracted, the results are then mapped into specific categories using tines functions like UNIQ, SPLIT, and JOIN⁷. If the extracted IOCs contain URLs, they would be automatically forwarded to the URLScan.io in the email analysis workflow for further processing.

5.4.3 Hybrid Analysis

In the PDF analysis workflow, Hybrid Analysis is implemented as an enhanced feature that can be used for deeper examination of files in a safe sandbox environment. This will allow the framework to evaluate more complex threats which are embedded in PDFs, such as scripts or hidden malware. This ensures a comprehensive analysis of attachments.

The workflow initiates a POST request to the Hybrid Analysis API, as shown in the figure 13a. The hash of the extracted file is then fed as payload to the HTTP request

⁶<https://iocparser.com/>

⁷<https://www.tines.com/docs/formulas/functions/>

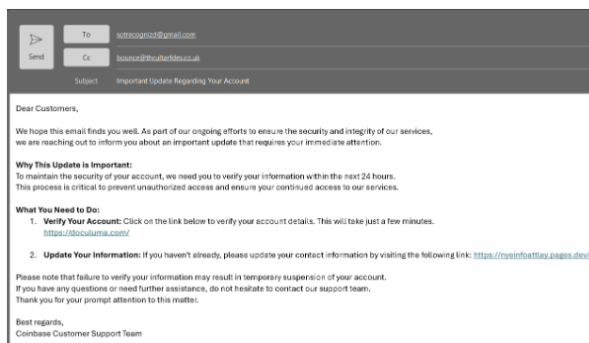
message it conveys and clicked on the links which led them to a fake website. This case study stood with the goal of evaluating the ability of the phishing detection system to detect and analyze the specific components of the email that are malicious.

6.1.1 Email and URL Analysis

The phishing email was passed into the Tines-based phishing detection system through the configured webhook which activated the Email Analysis Workflow as in figure14a. The IOCs extracted from the email content included the sender and recipient email addresses and links within the body of the email. These URLs were then passed on to URLScan.io for further behavioral analysis of specific transactions by utilizing multiple user agents to mimic different types of browser settings in order to expose any cloaking or evasion techniques by the attacker.

6.1.2 Results

The system was able to effectively pinpoint several things within the email that were deemed suspicious. The first and second emails addresses were categorized as spam, so they could have been engaged in phishing. Concerning the URLs, it was observed that the first URL did not have any recorded reputation while the second one was flagged as suspicious to multiple user agents. The detailed report as shown in figure 14b, also contained the general overview of the analysis where all the artifacts such as the email, IP address, and URL belonging to the fake account were separated with their reputation and the user agent employed throughout the analysis process.



(a)

Thanks for sending the email to suspicious mail box. See the attached results from the a

TYPE	ARTIFACT	REPUTATION	USER AGENT
IP Address	N/A No ipsfound	N/A	-
Email	bounce@thcultarfdes.co.uk	Suspicious	-
Email	sotrecognizd@gmail.com	Suspicious	-
URL	hxtps://doculuma[.]com/	No Records Found	Chrome
URL	hxtps://doculuma[.]com/	No Records Found	IOS
URL	hxtps://doculuma[.]com/	No Records Found	Android
URL	hxtps://doculuma[.]com/	No Records Found	Firefox
URL	hxtps://doculuma[.]com/	No Records Found	Safari
URL	hxtps://doculuma[.]com/	No Records Found	Edge
URL	hxtps://doculuma[.]com/	No Records Found	Internet Explorer
URL	hxtps://doculuma[.]com/	No Records Found	Opera
URL	hxtps://nyeinfoattlayl.jpapes[.]dev/	Suspicious	Chrome
URL	hxtps://nyeinfoattlayl.jpapes[.]dev/	Suspicious	IOS
URL	hxtps://nyeinfoattlayl.jpapes[.]dev/	Suspicious	Android
URL	hxtps://nyeinfoattlayl.jpapes[.]dev/	Suspicious	Firefox
URL	hxtps://nyeinfoattlayl.jpapes[.]dev/	Suspicious	Safari
URL	hxtps://nyeinfoattlayl.jpapes[.]dev/	Suspicious	Edge
URL	hxtps://nyeinfoattlayl.jpapes[.]dev/	Suspicious	Internet Explorer
URL	hxtps://nyeinfoattlayl.jpapes[.]dev/	Suspicious	Opera

(b)

Figure 14: (a) Case Study 1: Test Email (b) Case Study 1: Results

6.2 Case Study 2: PDF File with Embedded URLs

For this experiment, the phishing detection was done on a PDF file with many URLs, of which some of them were allegedly to link to a suspicious domain. This was to determine the effectiveness of the system to correctly recognize and parse these URLs inside the PDF and based on them further assess any identified URLs for threats.

6.2.1 PDF and URL Analysis Workflow

The PDF was uploaded to the system via the PDF Analysis Workflow as seen in figure 15a. The system read the text from the provided PDF and searched for IOCs using the IOCParse.com's API request. I automated the process such that once URLs were recognised in the PDF, the workflow set off the URLScan.io of the Email Analysis Workflow. This is done to inspect different browsing scenarios that will check for any malicious activity that is related to the URLs.

6.2.2 Results

The results of the analysis are as shown in the figure 15b. The system detected 100% of the embedded URLs in the PDF, passing them on to URLScan.io for further behavioral analysis. Out of the two URLs found, one was flagged as malicious by four user agents, while the other URL was determined to be malicious by all eight user agents.

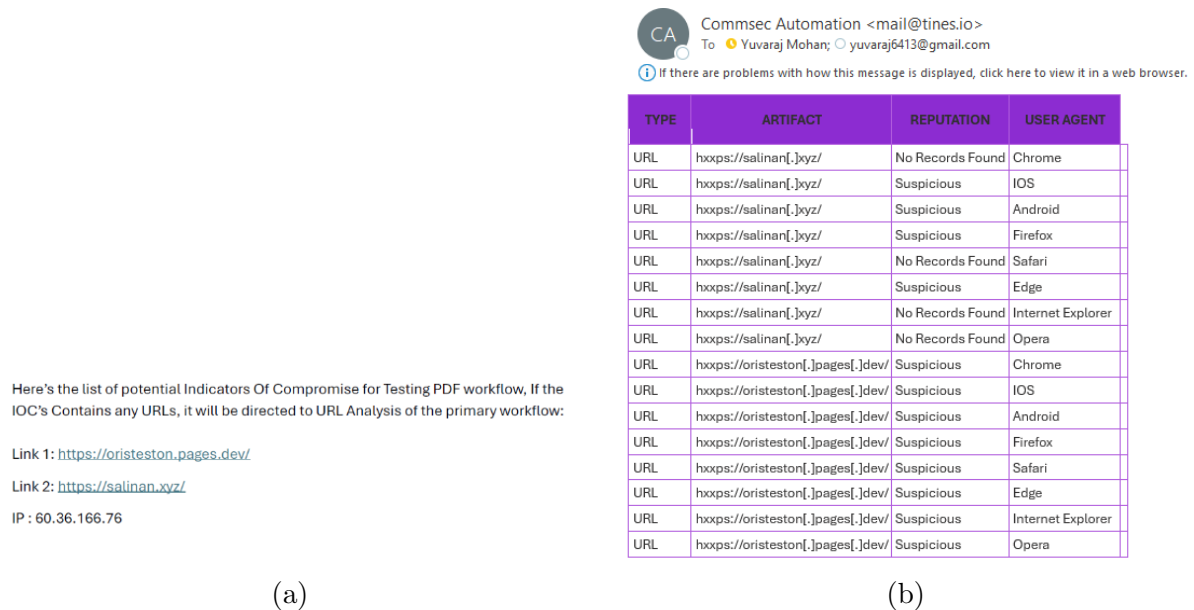


Figure 15: (a) Case Study 2: Test PDF (b) Case Study 2: Results

6.3 Case Study 3: Malicious PDF Analysis

The third case study assesses the effectiveness of the system in identifying and analyzing potential threats such as malware in the context of a PDF file. This scenario replicates an advanced form of phishing where a PDF file contains malicious codes intended to harm

the recipient’s system. It was performed with an official test file of anti-viral software called EICAR ⁸ that emulates an active malicious software but in fact is harmless.

6.3.1 PDF Analysis Workflow

The test PDF was then fed into the PDF Analysis Workflow to be tested on the phishing detection system. First, the text of the PDF was obtained with the help of a Python script, and then the IOC analysis stage was carried out, to check for threats in the document. The PDF file was also uploaded to Hybrid Analysis where it could be analyzed in a controlled environment that would help to reveal any suspicious actions that it might contain.

6.3.2 Results

The analysis was able to employ the necessary measures and identified the PDF as malicious. The Hybrid Analysis report, as depicted in the figure 16, further affirm that the file had appeared closely related in other malicious environments with all the antivirus engines identifying the particular file as 100% malicious. This helped in getting an understanding of how the file behaves and the activities within the sandbox analysis provided a detailed understanding of the threat.

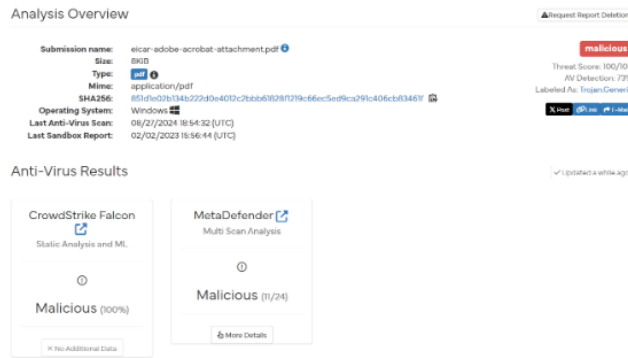


Figure 16: Hybrid Analysis Result

6.4 Discussion

The evaluation of the phishing detection system through the three case studies underscores its effectiveness and reliability in identifying and mitigating a wide range of phishing threats particularly when compared with established industry solutions such as Proofpoint Email Protection ⁹ and Mimecast Secure Email Gateway ¹⁰. In Case Study 1, the system performs well in the identification of the emails and the urls, labeling them as malicious and fully explaining their behavioral analytics. Similar to the proposed system, Proofpoint and Mimecast use advanced detection techniques. Proofpoint focuses heavily on sender reputation and spoofing detection, while Mimecast excels in real-time scanning of email content and links. However, the proposed system’s detailed behavioral analytics

⁸<https://www.eicar.org/>

⁹<https://www.proofpoint.com/us/products/email-security-and-protection/email-protection>

¹⁰<https://www.mimecast.com/products/email-security/secure-email-gateway/>

provide a deeper insight into the nature of threats compared to the more generalized threat detection in Mimecast and Proofpoint.

In Case Study 2, the importance of the system to consolidate various processes was exemplified, paying attention to the way the system deals with URLs in PDF files. Both Proofpoint and Mimecast offer comprehensive scanning of embedded URLs, but typically within a more integrated platform. The proposed system’s use of a specialized external tool like URLScan.io for this specific function showcases a unique approach that may offer more detailed analysis of the threats than the built-in capabilities of Proofpoint and Mimecast.

In Case Study 3, the system utilized Hybrid Analysis for further examination of PDF files to reveal other concealed malicious actions and perform sandbox analysis, critical to the distinctions of subtle threats that static analysis fails to detect. While Mimecast employs similar sandboxing techniques through its targeted threat protection, the proposed system’s use of Hybrid Analysis potentially allows for a more nuanced understanding of malware behavior. Proofpoint’s sandboxing, integrated with its threat intelligence, is robust but may not focus as deeply on the specific dynamics of PDF file threats as the proposed system.

Overall, the system’s multi-faceted approach, leveraging various tools and workflows, ensures a robust defense against complex phishing attacks. This contrasts with the more integrated but perhaps less flexible platforms of Proofpoint and Mimecast. Though highly useful, experience has shown that this system depends on external programs such as URLScan.io and Hybrid Analysis and recommends that future improvements of the model may concern to involve more integrated intelligence sources within the platform and use of machine learning to improve the detection accuracy.

7 Conclusion and Future Work

7.1 Conclusion

In conclusion, the focus of this thesis was to design and test an advanced phishing detection framework adapted to incorporate various commercial security tools into the Tines to improve the detection and analysis of IOCs originating from suspicious communications. This multi-layered system is powered by tools like URLScan.io, VirusTotal, EmailRep.io, and Hybrid Analysis and features great capability in the detection of a wide range of phishing threats like malicious emails, URLs, IP addresses, file hashes, and PDF attachments.

Evaluation using case studies proved the system’s effectiveness in real-world phishing scenarios. In the first case study, the system was able to correctly recognize and analyze the IOCs contained in the phishing emails, and in the second case study, the system was able to effectively identify suspicious URLs embedded in the PDF document which was seamlessly integrated with the email analysis workflow for thorough analysis. The third case study highlighted the effectiveness of the system in using sandboxing to detect sophisticated threats by identifying malicious content in the EICAR PDF file.

The system proved to be highly effective but its reliance on external threat intelligence tools points to the potential areas for future improvement. Also, Inclusion of other sources of intelligence and machine learning algorithms into the system itself can further make it more adaptable and improve its accuracy for detecting emerging phishing tactics.

7.2 Future Work

In the future, there are several directions that could be explored in order to extend the work of this thesis further. First, there is the possibility of implementing machine learning models that can adjust to new phishing techniques thereby reducing the dependence on static threat databases and improving the identification of zero-day phishing attacks Thomopoulos et al. (2024). Further, use of multiple sources of threat intelligence could also offer a broader picture of the changing threat landscape and make the system more effective.

The future work can focus to include the optimizing of the system’s efficiency during the real-time usage, especially in the organizational context where phishing attacks occur often. Another valuable research direction can be to investigate the possibilities for automating response actions depending on the severity of the threats that have been identified, thereby better combating phishing attacks correspondingly faster

References

- Alhaidari, F., Shaib, N. A., Alsafi, M., Alharbi, H., Alawami, M., Aljindan, R., Rahman, A.-u. and Zagrouba, R. (2022). Zevigilante: Detecting zero-day malware using machine learning and sandboxing analysis techniques, *Computational Intelligence and Neuroscience* **2022**: 1–15.
- Altwaijry, N., AlTuraiki, I., Alotaibi, R. and Alakeel, F. (2024). Advancing phishing email detection: A comparative study of deep learning models, *Sensors* **24**.
- Felegyhazi, M., Kreibich, C. and Paxson, V. (2010). On the potential of proactive domain blacklisting, *Proceedings of the 3rd USENIX Conference on Large-Scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More*, LEET’10, USENIX Association, USA, p. 6.
- Garera, S., Provos, N., Chew, M. and Rubin, A. D. (2007). A framework for detection and measurement of phishing attacks, *Proceedings of the 2007 ACM Workshop on Recurring Malcode*, WORM ’07, Association for Computing Machinery, New York, NY, USA, p. 1–8.
URL: <https://doi.org/10.1145/1314389.1314391>
- Gibson, K. (2024). Automating security testing and remediation into secops with tines workflows and synack.
URL: <https://www.synack.com/blog/automating-security-testing-and-remediation-into-secops-with-tines-workflows-and-synack/>
- Jesus, V., Bains, B. and Chang, V. (2024). Sharing is caring: Hurdles and prospects of open, crowdsourced cyber threat intelligence, *IEEE Transactions on Engineering Management* **71**: 6854–6873.
- Juwono, J., Lim, C. and Erwin, A. (2015). *A Comparative Study of Behavior Analysis Sandboxes in Malware Detection*.
- Kemp, B. (2023). Threat hunting for phishing sites with urlscan.io.
URL: <https://phish.report/blog/urlscanio-threat-hunting>

- Khonji, M., Iraqi, Y. and Jones, A. (2013). Phishing detection: A literature survey, *IEEE Communications Surveys and Tutorials* **15**: 2091–2121.
- Moubayed, A., Injadat, M., Shami, A. and Lutfiyya, H. (n.d.). Dns typo-squatting domain detection: A data analytics & machine learning based approach.
- Opara, C., Chen, Y. and Wei, B. (2024). Look before you leap: Detecting phishing web pages by exploiting raw url and html characteristics, *Expert Systems with Applications* **236**: 121183.
- Rajab, M. A., Zarfoss, J., Monroe, F. and Terzis, A. (2007). My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging, *Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets*, HotBots’07, USENIX Association, USA, p. 5.
- Rao, R. S. and Pais, A. R. (2019). Detection of phishing websites using an efficient feature-based machine learning framework, *Neural Comput. Appl.* **31**(8): 3851–3873.
URL: <https://doi.org/10.1007/s00521-017-3305-0>
- Singh, B., Kumar, M., Rexwal, T. and Jain, A. (2024). Detecting cyber threats utilizing machine learning approaches: An assessment of performance perspective, *Tuijin Jishu/Journal of Propulsion Technology* **45**: 1001–4055.
- Thomopoulos, G., Lyras, D. and Fidas, C. (2024). A systematic review and research challenges on phishing cyberattacks from an electroencephalography and gazebased perspective, *Personal and Ubiquitous Computing* pp. 1–22.
- Tines, T. (2024). The state of soar: Tines survey reveals the pros and cons of soar platforms — tines.
URL: <https://www.tines.com/blog/the-state-of-soar-tines-survey-of-security-professionals-reveals-pros-and-cons/>
- Tripathy, S., Shyamasundar, R. K. and Ranjan, R. (eds) (2021). *Multi Layer Detection Framework for SpearPhishing Attacks*, Information Systems Security, Springer International Publishing.
- Verma, R. and Das, A. (2017). What’s in a url: Fast feature extraction and malicious url detection, *Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics*, IWSPA ’17, Association for Computing Machinery, New York, NY, USA, p. 55–63.
URL: <https://doi.org/10.1145/3041008.3041016>
- Zhang, Y., Hong, J. and Cranor, L. (n.d.). Cantina: A content-based approach to detecting phishing web sites.