**EVENT-DRIVEN MALICIOUS URL EXTRACTOR**

2021-085

Project Proposal Report

Renu Harshatha A.

B.Sc. (Hons) Degree in Information Technology Specialized in Cyber Security

Department of Computer Science and Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

February 2021

**EVENT-DRIVEN MALICIOUS URL EXTRACTOR**

2021-085

Project Proposal Report

B.Sc. (Hons) Degree in Information Technology Specialized in Cyber Security

Department of Computer Science and Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

February 2021

# **Declaration**

I declare that this is my own work and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

|  |  |  |
| --- | --- | --- |
| Name | Student ID | Signature |
| Renu Harshatha A. | IT18034400 | Text, letter  Description automatically generated |

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

..…………………….……. ………………………… Signature of the Supervisor Date

(Mr. Amila Seneviratne)

# **Abstract**

As people do not have a proper guidance and identification method to detect malicious URLs, there is a rapid increase in cybercrimes along with the technology advancement and transformation. This research proposes to identify malicious URLs based on events which happens both locally and globally using ensemble model which notifies the user real time with as a solution to this problem. In this research, the major components, the web application, and feature extraction will be used to identify the malicious URLs. Training the model with keywords, identifying the malicious URLs based on the trending events and incorporating scalability and ease of use will be used as components to identify, detect, and proactively block malicious URLs based on an event. For extraction process this research uses a well-trained ensemble model using Machine Learning and Deep Learning algorithm, which results accurate detection of malicious URLs. Creation of malicious URLs proportionally increases with a given occurred event. This results a delay in identification process. To overcome this challenge, in this research we will be incorporating a scalable model using Nomad. This results to achieve the main output of the entire study, which is extracting malicious URLs with better accuracy and efficiency.

Key words: - machine learning and deep learning, ensemble model, scalability, Nomad, malicious URLs

# **Table of Content**

[**Declaration** i](#_Toc67493974)

[**Abstract** ii](#_Toc67493975)

[**Table of Content** iii](#_Toc67493976)

[**List of figures** iv](#_Toc67493977)

[**List of tables** v](#_Toc67493978)

[**List of abbreviations** vi](#_Toc67493979)

[**List of appendices** vii](#_Toc67493980)

[**1.** **INTRODUCTION** 1](#_Toc67493981)

[**1.1 Background & Literature Survey** 1](#_Toc67493982)

[**1.2 Research Gap** 4](#_Toc67493983)

[**1.3 Research Problem** 6](#_Toc67493984)

[**2.** **OBJECTIVES** 8](#_Toc67493985)

[**2.1 Main Objective** 8](#_Toc67493986)

[**2.2 Specific objectives** 9](#_Toc67493987)

[**3.** **METHODOLOGY** 10](#_Toc67493988)

[**3.1 System Architecture** 14](#_Toc67493989)

[**3.1.2 Commercialization** 16](#_Toc67493990)

[**4.** **PROJECT REQUIREMENTS** 17](#_Toc67493991)

[**4.1 Functional requirements** 17](#_Toc67493992)

[**4.2 User requirements** 18](#_Toc67493993)

[**4.3 System requirements** 18](#_Toc67493994)

[**4.4 Non-functional requirements** 19](#_Toc67493995)

[**4.5 Use cases** 20](#_Toc67493996)

[**4.6 Wireframes** 21](#_Toc67493997)

[**5. GANTT CHART** 23](#_Toc67493998)

[**5.1 Work Breakdown Structure (WBS)** 24](#_Toc67493999)

[**6.** **BUDGET AND BUDGET JUSTIFICATION** 25](#_Toc67494000)

[**REFERENCES** 26](#_Toc67494001)

[**APPENDICES** 27](#_Toc67494002)

# **List of figures**

Page

[Figure 1.1: Malicious Activity increasing over the given period iv](file:///C:/Users/Acer/Desktop/IT18034400-2021-085.docx#_Toc67492726)

[Figure 1.2: Malicious Activity increasing over the given period 3](file:///C:/Users/Acer/Desktop/IT18034400-2021-085.docx#_Toc67492727)

[Figure 1.3: Percentage of Phishing Attacks hosted on HTTPS 4](#_Toc67492728)

[Figure 1.4: Sudden spike after an event has occurred 6](file:///C:/Users/Acer/Desktop/IT18034400-2021-085.docx#_Toc67492729)

[Figure 3.1: Activity diagram of pre-trained plugin 13](file:///C:/Users/Acer/Desktop/IT18034400-2021-085.docx#_Toc67492730)

[Figure 3.2: High Level Diagram of the system 14](file:///C:/Users/Acer/Desktop/IT18034400-2021-085.docx#_Toc67492731)

[Figure 3.3: High Level Diagram of the pre-trained plugin 15](file:///C:/Users/Acer/Desktop/IT18034400-2021-085.docx#_Toc67492732)

[Figure 4.1: Watcher web application wireframe 21](file:///C:/Users/Acer/Desktop/IT18034400-2021-085.docx#_Toc67492733)

[Figure 4.2: Plugin wireframe: generate report 21](file:///C:/Users/Acer/Desktop/IT18034400-2021-085.docx#_Toc67492733)

[Figure 4.3: Plugin wireframe: malicious URLs 21](file:///C:/Users/Acer/Desktop/IT18034400-2021-085.docx#_Toc67492733)

[Figure 5.1: Gannt Chart 23](file:///C:/Users/Acer/Desktop/IT18034400-2021-085.docx#_Toc67492734)

[Figure 5.2: WBS 2](file:///C:/Users/Acer/Desktop/IT18034400-2021-085.docx#_Toc67492734)4

# **List of tables**

[**Table 1.1: Comparison of previous work 5**](#_Toc67493390)

**Table 1.2: Comparison of Nomad vs Kubernetes 11**

**Table 1.3:Required technologies, techniques for research study 12**

**Table 1.4:Use case fro admin generating report 20**

# **List of abbreviations**

Abbreviation Description

CNN Convolutional Neural Network

CoVID Corona Virus Disease

DL Deep Learning

HTTPS Hyper Text Transfer Protocol Secured

ML Machine Learning

NLP Natural Language Processing

RNN Recurrent Neural Network

URL Uniform Resource Locator

WHO World Health Organisation

WWW Word Wide Web

# **List of appendices**

Appendix Description Page

[**Appendix - A : Plagiarism Report** 27](https://mysliit.sharepoint.com/sites/CDAPSubmissionCloud/2021RegCloud/2021-041-Students/1.%20Project%20Proposal/Report/IT18002102.docx#_Toc67340062)

# **INTRODUCTION**

## **1.1 Background & Literature Survey**

As the technology environment grows, we become more dependable on Information Technology and its services. One such service that we depend so much is the World Wide Web which is also known as Internet. At present a world without internet cannot be imagined. Almost 4.66 billion people were active internet users as of October 2020 [1]. This technology advancement has caught the eyes of cybercriminals to lure the internet users and propagate cyber-attacks.

Although there are various other attack mechanisms available to lure the victims, cybercriminals propagate the attack through malicious URLs. Internet users or the target victim tend to click on these malicious URL as they look legitimate. During the CoVID-19 pandemic, the entire globe shifted to online platforms from usual traditional platforms. While the number of internet users increased, cybercriminals took this opportunity to lure them for their personal and/or business sensitive information by sending malicious URLs with attracting titles.

Uniform Resource Locator (URL) is an address of a resource in World Wide Web (WWW) which helps a user to access websites [2]. An URL consists of three main parts protocol, hostname, and path namely. A cyber attacker modifies this URL for attack purpose. Thus, these modified URLs are called malicious URLs. In order to identify these malicious URLs, researchers and developers started building models and applications with the help of Machine Learning (ML) and Deep Learning (DL). Using Machine Learning & Deep Learning researchers were able to come up with various models and detection system for malware detection, network intrusion detection, spam detection and malicious URL detection [3].

We witness that machine learning is explicitly growing in the technological industry in terms of detection systems. Detection systems which uses pronunciations, characters, and shapes stays constant over the period [3] ,but applications used to detect malicious content changes and evolves over the time which makes the detectors unusable. This has become one of the greatest challenges for malicious URL detectors.

Many research works have been explored in the past to identify and differentiate malicious URLs. In addition, they used machine learning approaches to detect the malicious URLs and give better solutions. These research works were focusing on increasing the accuracy and blocking the detected URLs.

However, these models do not incorporate scalability dominantly. This prevents a system to scale according to the processing power given to it. A malicious URL is identified with a delay due to lack of scalability. Additionally, the existing solutions for detecting malicious URLs takes extra time which reduces the speed and efficiency of identifying them [4].

The proposed system for this problem is a web application that takes URL as input from the user and detects, extracts malicious URLs related to the keyword in real-time. To improve the efficiency and accuracy of the results, this research uses ensemble model which incorporates ML & DL. With excess creation of malicious URLs depending on the event that has occurred, the accuracy level is questionable, and time taken to provide the result is high. During this stage, the trained model should be able to detect the number of malicious even if there is an increase in number of links. In order to achieve this, the ensemble model should be able to scale in and out depending on the event given. For this process, TensorFlow, Azure Machine Learning and ML and DL algorithm will be used.

According to APWG 4th Quarter report, throughout the COVID-19 outbreak malicious attacks have become more pervasive and damaging than ever [5]. During the course of CoVID-19 pandemic there was a rapid increase in phishing activity, by end of December 2020 the number of phishing sites reached to 200,000. (Figure 1.1).

Figure 1.1: Malicious Activity increasing over the given period

Figure 1.2: Malicious Activity increasing over the given period

Several years ago, to protect humans from malicious websites and URLs, users were requested to check for “HTTPS” and a padlock sign. Recently, the Anti-Phishing Working Group discovered that attackers use HTTPS encryption protocol to fool humans and lure them to gather sensitive information [5].

The figure below represents the percentage of phishing attack which were hosted using HTTPS over the last couple of years starting from 2017. (Figure 1.2). Most of the attack has been held in the last quarter of 2020 and it is depicted that over the years, attackers using encryption protocol has increased although there has been decreased values.

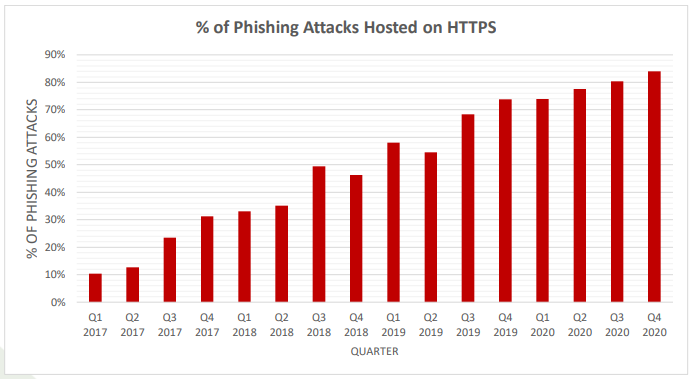


Figure 1.3: Percentage of Phishing Attacks hosted on HTTPS

In the past, various research studies were performed to identify and detect malicious websites and URLs. These research studies have used various detecting methods using ML, DL algorithms like lexical analysis, NLP, CNN, RNN etc. [6] [7] [8].

Most of the research studies either showed ML, DL algorithm used for detecting the malicious URLs. One research study introduced ensemble model to detect malicious URLs [9]. Ensemble model is a process where multiple diverse models are created to predict an outcome, either by using different algorithm or using different training data set.

## **1.2 Research Gap**

There are several research studies and technological methodologies to identify malicious URLs and to improve the output’s degree of accuracy level. According to Research A [7], malicious URLs were detected using ML lexical analysis methodology. This research study focused on detecting the malicious URLs effectively and efficiently by using a lightweight approach which is lexical analysis.

Secondly, Research B [8], uses seven different classification algorithm and NLP. This research focuses on real-time system. Using these two-algorithm methodologies the detection system was able to give its best performance and better accuracy level. Thus, it is understandable that this research study has focused on performance and accuracy to detect the malicious URLs and blacklist them.

Research C [9] has used ensemble model to detect malicious, benign URLs for detection process, the proposed system in this research study replaces the simple traditional approach of blacklisting and reduces the false positive rate. This improves the efficiency of the model and helps to detect malicious URLs from benign one.

**Table 1.1: Comparison of previous work**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Solution** | **Considers accuracy of the model** | **Considers efficiency of the model** | **Categories URLs based on events** | **Building scalable system** |
| Research A [7] | **û** | **ü** | **û** | **û** |
| Research B [8] | **ü** | **û** | **û** | **û** |
| Research C [9] | **ü** | **û** | **û** | **û** |
| Proposed system | **ü** | **ü** | **ü** | **ü** |

Previous research studies indicate that used only accuracy, efficiency and performance are dominantly focused. This research study proposes to classify the URLs based on a specific event and scale accordingly by maintaining accuracy, efficiency, and the system’s performance in real-time

## **1.3 Research Problem**

Cyber-attacks have been leveraging as the world digitally transforms. Numerous malicious URLs are spawned at an instant. Cybercriminals come up with different ways to make the site look legit as possible. During the pandemic period, the COVID-19 malicious URLs had keywords like covid-19, WHO, vaccine etc. These keywords change with the trending words related to COVID-19 as the global environment changes.

The number of the malicious URLs are higher than the registered legitimate URLs. [10] This proves that if any trending event occurs around the globe, there are high chances that number of malicious links will get increased. In order to detect these numerous links, there should be a scalable model which will scale in and scale out according to the weightage of the event. If a malicious URL is identified with a delay due to lack of scalability, the internet users are prone to cyber-attacks easily.

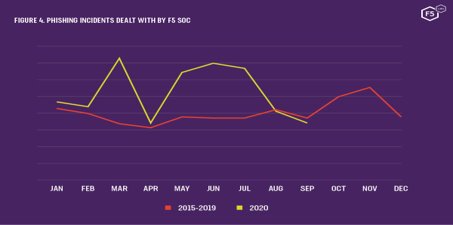
Figure 1.3 depict a sudden increase in the malicious URL incident for the year 2020. The sudden spike in trend lines were witnessed after an event occurred. A solution for timely detecting malicious URLs after an event is still not proposed. A rapid increase in number of malicious URLs might slow down the detecting system as they are mostly focused on efficiency and accuracy dominantly.

Figure 1.4: Sudden spike after an event has occurred

There is a need of a system which could detect malicious URLs depending on the event that has occurred and being trending locally and globally. This system model should be able to scale in and out depending on the number of keywords and malicious URLs that could be born. Thus, the research problem for this study would be “A feasible way to deploy scalable model which is comaptible with the ensemble model selected to detect malicious URLs.”

# **OBJECTIVES**

## **2.1 Main Objective**

The main objective of this research study is to identify a suitable ensemble model using deep learning and machine learning algorithm for accurate analysis and faster detection. This ensemble model should be able to scale in and scale out according to the relevant hot events that is happening around the globe. This will include both local and global events.

The research component scalability will be tested on other three components in the research. The three components are,

* Web Application with suitable system architecture
* Keyword token to corelation with malicious URLs database
* Feature reduction

The web application should be able to be deployed in the both mobile devices and desktop devices. Our research study is focused on that the web application which should be highly performable even when there are more inputs than usual. This shall make the application to work efficiently and effectively.

When training the Machine learning, Deep learning system, which is known as feeding the data, the system should be able to leverage horizontally. Thus, the time, cost and performance of the system will improve.

Scaling the feature reduction is one of the challengeable tasks in the research. The study will take effort to scale this, however this shall be kept for future research work.

Another main objective in the context of scalability is the model or service which will be chosen should be compatible with the ensemble model. Therefore, the scalability algorithm needs to be studied and understand.

## **2.2 Specific objectives**

In order to achieve the main objective, the specific objectives that need to be attended are as follow,

* Identify a solution for scalability which can be compatible with Ensemble Model. Compare available solutions for scalability, in order to do this a deep analysis on how each model works need to be studied and their algorithms needs to be compared.
* Integrate the chosen scalable solution (Nomad) to scale the proposed system in this research.
* Find potential methodologies to containerise the proposed system conducted in this research.
* Test the container’s compatibility with Ensemble Model. This scalable solution should be able to expand workload when the scope increases and shrink when the scope decreases.
* Improving the scalability of the system using both machine learning and deep learning techniques to enhance the efficiency and reliability of the system.
* Develop a browser extension for the proposed solution, which analyse, detects and extracts malicious URLs real time.
* Budget analysis should be conducted to finalise the scalability service chosen and for hosting purposes.

# **METHODOLOGY**

As stated in the research gap, the main aim of this proposing model is to enhance the degree of accuracy of the result. With the advancement made in artificial intelligence, the need for an accurate model is justified. The domain adopts new approaches every year and previous URL classifiers become less efficient as time goes on. It has been proven that the ensemble model provides high classification accuracy with low false positive.

The figure 3.2 depicts the overall system diagram to identify the malicious URLs. As the system uses ML and DL algorithm methodologies, deploying the ensemble model system in a less complex method. To solve the research problem mentioned above, this research study will containerise the created application and deploy it in a workload orchestrator.

Kubernetes has been used by most of the ML research studies when the containerization and orchestration is taken under consideration for the proposed solution. This research study use Nomad instead of Kubernetes. Table 1.2 provides a comparison between Nomad and Kubernetes which states why this research uses Nomad as orchestrator.

Nomad is an orchestrator which helps to deploy and manage containerized and non - containerized application both on premises and clouds at scale with flexible and less complex workload.

**Table 1.2: Comparison of Nomad vs Kubernetes**

|  |  |  |
| --- | --- | --- |
| **​** | **Nomad​** | **Kubernetes​** |
| **Simplicity**​ | * Architecturally much simpler. * Distributed, * highly available * Operationally simple. ​ | * Operationally complex to setup​ |
| **Flexible Workload Support**​ | * Supports virtualised, containerised and standalone​ | * Focused on Docker​ |
| **Consistent Deployment**​ | * Can be deployed in local dev, productions, on premises and in clouds. * Ease of use across all environments ​   ​ | * Installation for production environment is time consuming. ​   ​ |
| **Scalability**​ | * Easy to scale apps deployment across multiple data centres, regions, and clouds with no complexity. ​ | * Significant challenges of managing the system at scale. ​ |

Following are the technologies and techniques that will be used to accomplish the aforementioned process (Table 1.3).

**Table 1.3 Required technologies, techniques for the research study**

|  |  |
| --- | --- |
| Technologies | **Azure Machine Learning**  **Python**  **TensorFlow**  **Nomad**  **HTML**  **CSS**  **Javascript** |
| Techniques | **Containerisation**  **Orchestration** |

Along with the system created in this research study, a pre-trained plugin will be created for users. Considering the prevailing pandemic situation around the globe, remote working has become the new norm of working in the industry. For ease of use and access to the users this pre-trained plugin will be in handy.

Figure 3.1, The activity diagram gives a basic idea on how the plugin works. The pre-trained plugin should be installed in the users’ web browser. Once the user visits the URL page, the pre-trained plugin sends the user visited URL to the ensemble model which detects whether the URL is malicious or benign.

If the URL is considered to be malicious the plugin, parallelly performs the following tasks:

* Blocks the user from accessing the URL
* Records the event in the database
* Notifies system administrator as an Alert

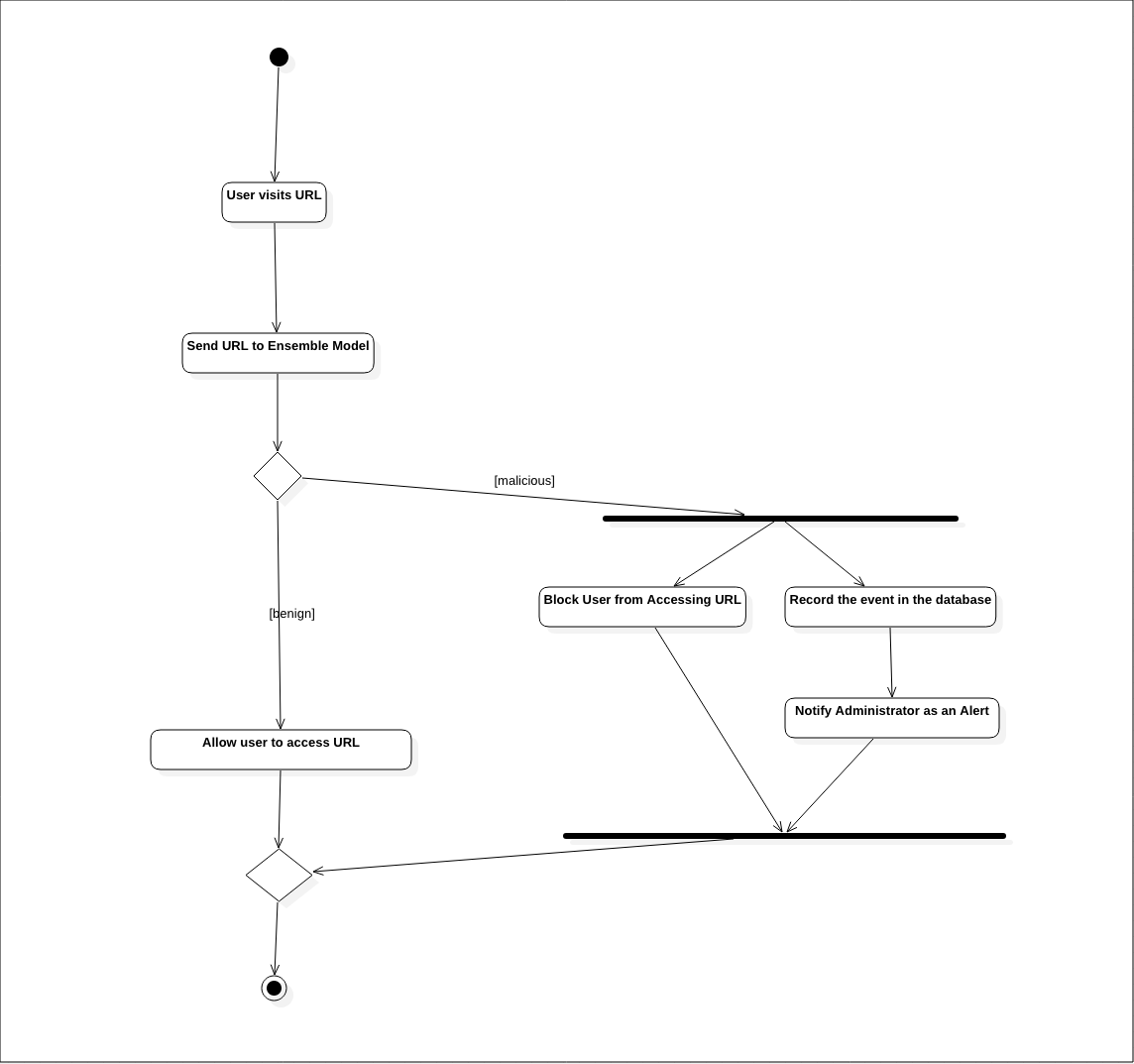


Figure 3.1: Activity diagram of pre-trained plugin

## **3.1 System Architecture**

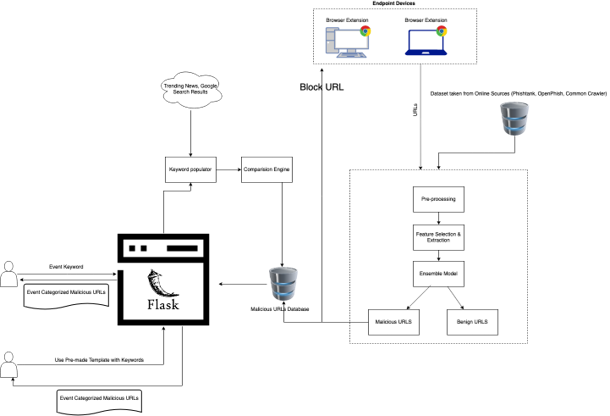


Figure 3.2: High Level Diagram of the system

System overview diagram for the pre-trained plugin

Graphical user interface, application

Description automatically generated

Figure 3.3: High Level Diagram of the pre-trained plugin

### **3.1.2 Commercialization**

Since a part of the userbase for this system would be small-medium enterprises (SME), this system can be commercialized with the note of providing basic security. Even though, at this stage the system cannot compete with state-of-the-art endpoint management system, this will be suitable for SMEs due to the lower cost and higher usability.

Two versions of this system can be implemented.

* A free version that SMEs and Researchers can use to collect URL lists based on events with limited export capability.
* A paid version that will provide seamless export capability in addition to the basic endpoint protector using the browser plugin.

|  |  |
| --- | --- |
| **Free Version** | Rate Limit on event-based malicious URL list and restricted export capabilities. |
| **Paid Version** | Browser plugin to protect users from malicious pages with basic reporting to the administrator  No limit on export and event-based malicious URLs |

The low-cost barrier should be a key aspect to attract SMEs. Researchers/ Investigators investigating certain events can make use of the data to correlate with the incident.

# **PROJECT REQUIREMENTS**

## **4.1 Functional requirements**

1. The system should be able to identify when to scale in and scale out depending on the number of keywords and user inputs.

As a functional requirement of this research, the system should be able to detect the trending hot events. During this event the proposed system should scale and sustain its performance and the output’s accuracy level. This is detected and trained using TensorFlow. When the trending event fades out or has less user inputs and the malicious links are identified the system should scale in back. Thus, the resources will not be wasted.

2. The system should be able to generate a report on demand of the pre-trained plugin administrator and provide accurate result real-time.

Considering individuals adapting to remote working, the use of deploying the system on premises is less required. In such scenarios, users can install the pre-trained real-time plugin which should be able to detect and differentiate the malicious and benign URLs. On demand of the system administrator, the plugin should be able to generate report of all the incidents, alert messages sent for a given period of time.

## **4.2 User requirements**

This system will consist of two sets of users.

1. Small/Medium Enterprises – They can use this system to generate event-based blocklists to stop a phishing attack at the early stages of the lifecycle. In addition, with the pre-trained plugin installed in their employee’s browsers will assist in preventing them from visiting malicious pages and help the organization receive alerts.

2. Security Researchers/Investigators – They can utilize this system to investigate URLs related to any events in a short span of time. This tool will assist them in determining the gravity of the event and help them decide to follow the investigation path or not.

## **4.3 System requirements**

System requirements are crucial when specifying the software resource requirements and prerequisites which must be implemented on a device in order for the designed system to run properly. Following are requirements which needed for this research study.

1. Web application, Web browser plugin

As a final outcome of this research study, Web application and Browser plugin is developed. In this study we will focus that the developed systems run successfully in computer system. For future work, this system should be able to work in mobile devices as well.

2. Nomad Orchestration & Containerisation

Nomad is an orchestrator which helps to deploy and manage containerized and non - containerized application both on premises and clouds at scale with flexible and less complex workload. This helps the system to be elastic whenever required.

## **4.4 Non-functional requirements**

1. Scalability

The system’s performance relies on how well it meets its requirement for timeliness. This product’s utmost priority is to detect and extract malicious URLs given upon an event has occurred. In order the give the results with less time taken to process, the system should be elastic. In other words, it should be able to scale in and out when necessary.

2. Availability

The system should be available when it is called upon. In other words, the system should not be down or be under maintenance when it is needed by the user. The availability of this system during a trending hot event is a must to identify and detect the malicious URLs input by the user. When system is not available for users, there is a high chance of users accessing malicious URLs.

## **4.5 Use cases**

Table 3. 1: Types of versions provided through the proposed application

Table 3. 2: Types of versions provided through the proposed application

Figure 5.4.1: Gannt ChartTable 3. 3: Types of versions provided through the proposed application

**Table 1.4: Use case for admin generating report**

|  |  |
| --- | --- |
| Use case name | User demanding for generating report |
| ID | S03 |
| Description | The user will click “generate report” icon with a given period and the system should generate a report with all the malicious URLs users tried to access within the given periods along with alert messages sent to admin. |
| Actors | Organisation administrator |
| Triggers | Clicking on the generate report icon |
| Pre – condition | User must be organisation administrator; or a privileged user. |
| Main course | 1. Organisation administrator must log in to the system with valid credentials. 2. Organisation administrator must click on “generate report” icon for system to generate report. 3. Organisation administrator must provide a period of time for system to retrieve data to create the report. 4. System should retrieve malicious URLs accessed or tried to access logs from the database and generate a pdf report. |
| Exceptions | 1a. Only 3 attempts are allowed. In case of forgotten password code will be sent to email address or phone number chosen by the user.  3a. Generate popup message to enter time period, If user does not enter period.  4a. In case of null data, banner stating no malicious URLs were accessed during this period of time should be showcased. |

## **4.6 Wireframes**



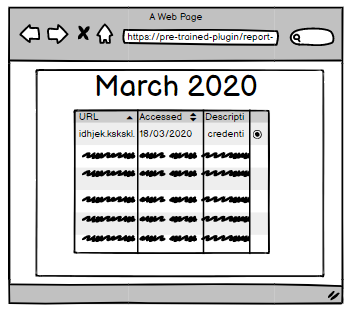


Figure 4.2: Plugin wireframe: generate report

Figure 4.1: Watcher web appication wireframe

Figure 4.2: Watcher web appication wireframe

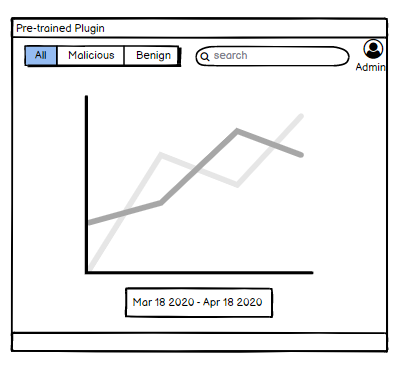


Figure 4.3: Plugin wireframe: malicious URLs trend

# **5. GANTT CHART**

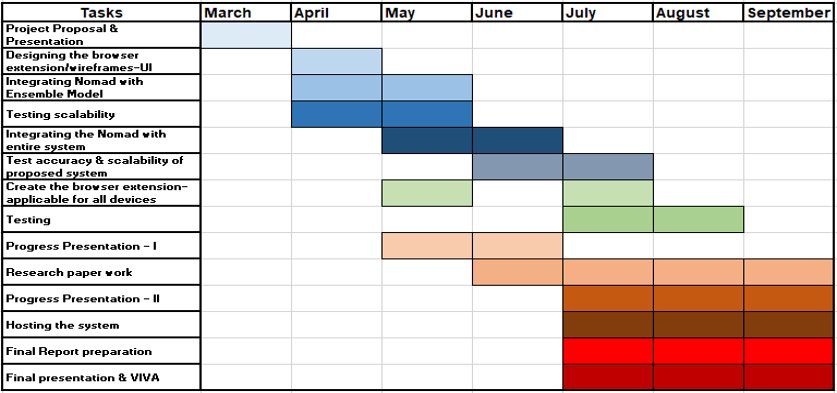


Figure 5.1: Gannt Chart

## **5.1 Work Breakdown Structure (WBS)**

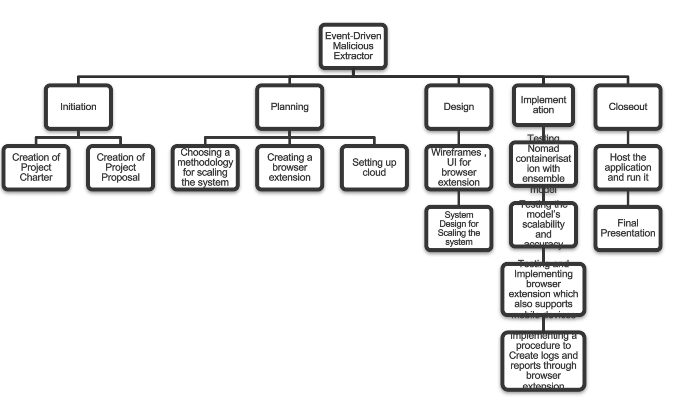


Figure 5.2: WBS

# **BUDGET AND BUDGET JUSTIFICATION**

|  |  |  |
| --- | --- | --- |
| **Task** | **Cost($)** | **Cost(Rs.)** |
| Azure Machine Learning Studio (Implementation and Cloud Deployment) | 100 (Monthly Fee) | **19,900** |
| Web Application Hosting | 72 | **14300** |
| Database – Mongo DB | 57 | **11400** |
| Plugin upload to Chrome Web store | 5 (One-time fee) | **995** |
|  | Total Cost | **46,595** |

To cover these costs, we can use the pricing models mentioned in the commercialization section.

# **REFERENCES**

|  |  |
| --- | --- |
| [1] | 27 January 2021. [Online]. Available: https://www.statista.com/statistics/617136/digital-population-worldwide/. [Accessed 15 February 2021]. |
| [2] | M. SIDDHARTHA, “Wisdom ML,” 19 August 2020. [Online]. Available: https://wisdomml.com/detecting-malicious-urls-using-machine-learning-approach/. [Accessed 14th February 2021]. |
| [3] | B. A. Miller, “Scalable Platform for Malicious Content Detection Integrating Machine Learning and Manual Review,” p. 1, 2015. |
| [4] | Z. M. X. L. J. A. J. Z. K. K. .. Basit A., “ A comprehensive survey of AI-enabled phishing attacks detection techniques,” 2020 . |
| [5] | “APWG Trend Report,” 2020. [Online]. Available: https://docs.apwg.org/reports/apwg\_trends\_report\_q4\_2020.pdf. . [Accessed 25 February 2020 ]. |
| [6] | C. G. M. V. P. D. S. Kurt Thomas, “Design and Evaluation of a Real-Time URL Spam Filtering Service,” *IEEE Symposium on Security and Privacy,* 2011. |
| [7] | M. A. R. H. L. N. S. A. A. G. Mohammad Saiful Islam Mamun, “Detecting Malicious URLs Using Lexical Analysis,” *Springer International Publishing,* 2016. |
| [8] | E. B. O. D. B. D. Ozgur Koray Sahingoz, “Machine learning based phishing detection from URLs,” *Expert Systems With Applications,* 2018. |
| [9] | S. T. Lekshmi A R, “The Kozinec-SVM Model for Detecting Malicious URLs,” *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT),* vol. 08, no. 07, 2019. |
| [10] | “APWG,” 2020. [Online]. Available: https://docs.apwg.org/reports/apwg\_trends\_report\_q3\_2020.pdf. |
|  |  |

# **APPENDICES**

**Appendix - A : Plagiarism Report**

