

**AMERICAN INTERNATIONAL UNIVERSITY – BANGLADESH**

**Malicious Link Detection Using Web Analytics**

**A thesis submitted in partial fulfillment of the requirements for the degree of**

**Bachelor of Science in Computer Science and Engineering**

**Submitted by**

|  |  |
| --- | --- |
| **Name** | **ID** |
| Islam, Nuhan | 18-37898-2 |
| Fahim, Abu Talha | 18-38125-2 |
| Faahim, Mohammad Zaahirul Islam | 18-38163-2 |
| Mazumder, Mohammad Sakib Islam | 19-39290-1 |

**Department of Computer Science**

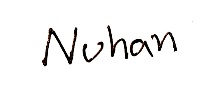
**Faculty of Science and Information Technology**

**American International University – Bangladesh**

**April 16, 2022**

# DECLARATION

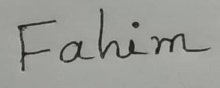
We hereby declare that this thesis is our original work and has not been submitted in any form for another degree or diploma at any university or institution of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of reference is provided.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Islam, Nuhan**

18-37898-2

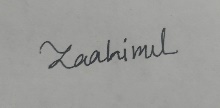
Computer Science and Engineering

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Fahim, Abu Talha**

18-38125-2

Computer Science and Engineering

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Faahim, Mohammad Zaahirul Islam**

18-38163-2

Computer Science and Engineering

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Mazumder, Mohammad Sakib Islam**

19-39290-1

Computer Science and Engineering

# APPROVAL

The thesis titled “Malicious Link Detection Using Web Analytics” has been submitted to the following respected members of the board of examiners of the department of computer science in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science on 16 April 2022 and has been accepted as satisfactory.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Dr. Md. Sohidul Islam**

Supervisor, Assistant Professor

Department of Computer Science

*American International University – Bangladesh*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Dr. Md. Abdullah – Al - Jubair**

Assistant Professor and Head-in-charge

Department of Computer Science

*American International University – Bangladesh*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Dr. M M Manjurul Islam**

External, Assistant Professor

Department of Computer Science

*American International University – Bangladesh*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Professor Dr. Tafazzal Hossain**

Dean

Faculty of Science and Information Technology

*American International University – Bangladesh*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Dr. Carmen Z. Lamagna**

Vice Chancellor

*American International University – Bangladesh*

# ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to Almighty for enabling us to complete the report on “Malicious Link Detection Using Web Analytics”. For the successful completion of this project, we have taken help from our respected faculty members, seniors and graduates. We convey our sincere gratitude to our supervisor, Dr. Md. Sohidul Islam sir, for taking us under his wing and guiding us to conduct research on this project, thereby gathering knowledge in the field of networking and cyber security.

# TABLE OF CONTENTS

[DECLARATION i](#_Toc101024355)

[APPROVAL ii](#_Toc101024356)

[ACKNOWLEDGEMENTS iii](#_Toc101024357)

[TABLE OF CONTENTS iv](#_Toc101024358)

[TABLE OF FIGURES vi](#_Toc101024359)

[Abstract viii](#_Toc101024360)

[Chapter 1: Introduction 1](#_Toc101024361)

[1.1 Introduction 1](#_Toc101024362)

[1.2 Background Study 2](#_Toc101024363)

[1.3 Problem Statement 2](#_Toc101024364)

[1.4 Goal of this Thesis 2](#_Toc101024365)

[Chapter 2: Literature Review 4](#_Toc101024366)

[2.1 DNS 5](#_Toc101024367)

[2.1.1 What is DNS 5](#_Toc101024368)

[2.1.2 A brief history of DNS 5](#_Toc101024369)

[2.1.3 How DNS Works 6](#_Toc101024370)

[2.2 DNS Filtering 6](#_Toc101024371)

[2.3 Domain Greylisting 7](#_Toc101024372)

[2.4 Phishing 8](#_Toc101024373)

[2.4.1 What is Phishing 8](#_Toc101024374)

[2.4.2 History of Phishing 8](#_Toc101024375)

[2.4.3 Types of Phishing 9](#_Toc101024376)

[1. Deceptive phishing 9](#_Toc101024377)

[2. Spear phishing 9](#_Toc101024378)

[3. Whaling 10](#_Toc101024379)

[4. Vishing 10](#_Toc101024380)

[5. Smishing 11](#_Toc101024381)

[6. Pharming 11](#_Toc101024382)

[2.5 API (Application Programming Interface) 12](#_Toc101024383)

[Chapter 3: Web Analytics 14](#_Toc101024384)

[3.1 Bounce Rate 14](#_Toc101024385)

[3.2 Page Per Visit 14](#_Toc101024386)

[3.3 Time on Site 15](#_Toc101024387)

[3.4 Total Visits 15](#_Toc101024388)

[Chapter 4: Malicious Link Detection Model 16](#_Toc101024389)

[Chapter 5: Web Analytics Tests and Result Analysis 17](#_Toc101024390)

[5.1 Testing Bounce Rate 17](#_Toc101024391)

[5.2 Testing Page Per Visit 17](#_Toc101024392)

[5.3 Testing Time on Site 17](#_Toc101024393)

[5.4 Testing Total Visits 17](#_Toc101024394)

[Chapter 6: Structural Overview of the Application 18](#_Toc101024395)

[Chapter 7: Benefits of using the Malicious Link Detection System 21](#_Toc101024396)

[Chapter 8: Proposal 22](#_Toc101024397)

[8.1 Motivation 22](#_Toc101024398)

[8.2 Proposal 22](#_Toc101024399)

[Chapter 9: Web Application Testbed 23](#_Toc101024400)

[Chapter 10: Future Work 28](#_Toc101024401)

[Chapter 11: Discussion & Conclusion 29](#_Toc101024402)

[11.1 Discussion 29](#_Toc101024403)

[11.2 Conclusion 29](#_Toc101024404)

[Chapter 12: References 30](#_Toc101024405)

[Appendix A: Code 32](#_Toc101024406)

[A.1 Website Structure 32](#_Toc101024407)

[A.2 Website Functionality 39](#_Toc101024408)

# TABLE OF FIGURES

[Figure 4.1: Malicious Link Detection Model 16](#_Toc101024328)

[Figure 6.1: Use Case diagram of the system 18](#_Toc101024329)

[Figure 6.2: Class diagram of the system 18](#_Toc101024330)

[Figure 6.3: Activity diagram of the system 19](#_Toc101024331)

[Figure 6.4: Sequence diagram of the system 20](#_Toc101024332)

[Figure 6.5: Entity Relationship diagram of the system 20](#_Toc101024333)

[Figure 9.1: Secure Site demonstration 23](#_Toc101024334)

[Figure 9.2: Malicious Site demonstration 24](#_Toc101024335)

[Figure 9.3: Web Interface 25](#_Toc101024336)

[Figure 9.4: Call API 26](#_Toc101024337)

[Figure 9.5: Scan Analytics 27](#_Toc101024338)

# Abstract

Safe web browsing and avoiding contact with any sort of malicious web content is of utmost priority in today’s technology-oriented world. Mitigating online based attacks is an intermediate step but identifying and nullifying them is preliminary. Through the use of Firewalls and extensive protocols including artificial intelligence and machine learning, an integrated safety system can help to identify and prevent attacks from happening. Safeguarding oneself in the virtual world is becoming more and more difficult as technology and their counterparts continue to evolve rapidly. This is where the concern of limited resources come under consideration. A trusted web-based application to check for malicious links can serve as a prerequisite before visiting an unknown website. Doing so can effectively bring down the severity of the risks one might have to undertake if fallen victim to cyber-attacks and loss of personal data. Inexperienced users can be introduced with the learning vectors regarding how they can protect themselves online.

# Chapter 1: Introduction

## 1.1 Introduction

Web surfing is one of the most prominent uses of technology. People require gathering and exploring the internet for information regarding all sorts of materials and topics. Among them, there are both good and bad. While the good ones are practically harmless, the ones which are not, can bring about adverse effects to the user’s confidential data along with their medium of access, which is their device.

Websites which carry threat concerns are termed as malicious. Malicious websites often appear as normal and trustworthy. They often hide behind the disguise of popular websites such as search engines and social media websites. One way of identifying them is to understand how Universal Resource Links (URLs) work, which is the link to a website. A valid URL is well structured and well defined following the nomenclature rules. These include Hyper Text Transfer Protocol Secured (HTTPS) initials, country extensions and domain names.

Malicious websites have their way of going around these rules and still appear valid. This is why we have come up with a simple yet robust solution to the problem. We have implemented Web Analytics against which a website is measured and tested. A website scanned against those analytics can significantly narrow down whether a website is safe to visit or malicious.

The measures for analytics which we have used are Bounce Rate, Pages Per Visit, Total Visits and Time on Site. Each of these metrics are run against a certain website to test its validity. The result is then viewed on our web application.

## 1.2 Background Study

While conducting our research, we have run tests across a wide range of variety of websites. Unusual behavior, anomalies and luring for possible theft of data were common among malicious websites. Often it is found that inexperienced users fall into the trap of malicious sites. Starting from data theft all the way to compromising a user’s system with malware, malicious sites possess a great threat.

There are several ways through which malicious sites can be detected and blocked. Most prominent of them is through the use of a Firewall. However, due to limited resources, this research provides a simple yet robust solution through the use of API. Basic analytics are run against a website and tested to check whether the site is safe or malicious. A major advantage is that due to the simple interface, the system is fast.

## 1.3 Problem Statement

* Real life situations such as malware attacks in organizations require more complex tools to detect and block malicious web content.
* The project is limited use in personal and minimal threshold browsing systems.
* There is no alternative to gathering knowledge about online threats and scams to which users may fall victim.
* Current stage of project is usable and effective but further development is possible.

## 1.4 Goal of this Thesis

This thesis aims to provide a simple, fast and straightforward tool to check for malicious sites. Secondary goals include overview of various threats a user can face while visiting malicious sites, familiarize with the threats and using this tool as a prerequisite to visit unfamiliar sites.

# Chapter 2: Literature Review

Phishing is a cybercrime in which a target or targets are contacted by email, telephone or text message by someone posing as a legitimate institution to lure individuals into providing sensitive data such as personally identifiable information, banking and credit card details, and passwords. Phishing attacks are the practice of sending fraudulent communications that appear to come from a reputable source. It is usually performed through email. The goal is to steal sensitive data like credit card and login information or to install malware on the victim's machine. Phishing is a common type of cyber-attack that everyone should learn about in order to protect themselves. Phishing is a type of social engineering where an attacker sends a fraudulent message designed to trick a human victim into revealing sensitive information to the attacker or to deploy malicious software on the victim's infrastructure like ransomware. there are four types of phishing attacks. Spear Phishing: Spear phishing targets a specific group or type of individual such as a company’s system administrator. Whaling: A whaling phishing email might state that the company is facing legal consequences and that you need to click on the link to get more information. The link takes you to a page where you are asked to enter critical data about the company such as tax ID and bank account numbers. Vishing: Vishing has the same purpose as other types of phishing attacks. The attackers are still after your sensitive personal or corporate information. This attack is accomplished through a voice call. Email Phishing: Email phishing is the most common type of phishing, and it has been in use since the 1990s. Hackers send these emails to any email addresses they can obtain. The email usually informs you that there has been a compromise to your account and that you need to respond immediately by clicking on a provided link. There are many ways to prevent phishing attacks. But it can not be stopped hundred percent. As this is not any virus or similar thing so it is quite impossible to get rid of this. But we have discovered a unique way to reduce these phishing attacks. We are planning to filter phishing sites so that it will not be able to get reach though target user. Our proposed method is to detect phishing link by checking its visitor number. We will detect it also by monitoring its hosting time. As we know a phishing link is quite new website which are recently hosted by the attacker. Here we are developing a system which will monitor the phishing sites, and then it will make them accessible to the user if the website is safe.

## 2.1 DNS

## 2.1.1 What is DNS

DNS is a machine for computer systems and offerings related to the Internet that resolves domains to IP addresses. It converts human readable domains into internet protocol addresses. Computers will solely communicate mistreatment series of numbers; thus, DNS was developed as a kind of phonebook that interprets the domain, we enter in our browser into a machine-readable IP.

### 2.1.2 A brief history of DNS

Thirty years ago, once the net was still in its infancy, once we wished to go to an internet site had to understand the informatics address of that site. That’s as a result of computers are and were solely ready to communicate victimization numbers. we wanted the simplest way to translate computer-readable data into one thing human readable. It had to be fast, lightweight, and scalable. within the early 1980’s, Paul Mockapetris came up with a system that mechanically mapped IP addresses to domain names and therefore the DNS was born. This same system still is the backbone of contemporary internet. Yet, only a tiny low subset of the planet is aware of that it exists, and an excellent smaller cluster perceive what it does. the important drawback is that the those who got to knowledge it works and will truly enjoy this information don’t take the time to learn.

### 2.1.3 How DNS Works

DNS resolution converts a hostname into a computer compatible IP address. Every device on the Internet is assigned an IP address, and that address is needed to find that particular Internet device, much like a street address is used to find a specific house. When a user wants to load a web page, a translation must be performed between what a user types into their web browser and the machine compatible address required to locate the web page. To understand the process behind DNS resolution, it is important to understand the different hardware components between which a DNS query must pass. For the web browser, DNS lookup happens behind the scenes and requires no interaction from the user's computer other than the initial request. [1]

## 2.2 DNS Filtering

DNS filtering is the process of restricting access to certain websites for a specified reason, most commonly content screening. If a site, or a group of sites, is judged a threat, its IP address is banned by a DNS filter, and access is denied. Adult, gambling, productivity sinks, and sites known to represent a major virus risk are examples of sites that may be restricted. DNS filtering is critical for organizations because it may drastically minimize the number of threats that a network is exposed to, reducing the remediation workload for MSPs and IT professionals. Effective DNS filtering can prevent up to 88 percent of internet borne malware from ever reaching the network. [2]

## 2.3 Domain Greylisting

The process of addressing sites that have not yet been classified as safe or dangerous is known as domain greylisting. Domain greylisting is a technique for blocking access to an unknown domain for a set period of time. This implies that users will have to return to the site at a later time. Because most spammers only attempt to send traffic to harmful domains once, requiring a second attempt screens out many malicious sites. Greylisting is effective since it assumes that harmful actors aren't persistent. This is a common occurrence in phishing scams. By definition, phishing assaults are low-effort. All a criminal has to do is concoct a cunning enough scheme to fool someone, somewhere. Phishers frequently use a wide net in the hopes of catching a few unwitting victims. Many of their attempts are blocked by spam filters, but enough get through that it's worthwhile. These con artists don't come up with more sophisticated strategies for re-applying once they've been rejected the first time. We have safeguarded ourselves by creating a simple roadblock in the form of greylisting, which temporarily denies access. Malicious domains and phishing sites are expensive and widespread. A new phishing site is launched every 20 seconds, according to a 2020 mobile threat landscape assessment. With new domains being formed on a daily basis, the speed and volume make it hard to prevent each new malicious domain that is created. DNS filters provide security solutions that, among other things, use domain greylisting to keep our company safe. If our DNS software identifies a domain as a phishing website, it will prevent us from seeing the page. All newly registered domains are restricted for 30 days for your security, because we shouldn't have to worry about malicious undertows when we surf the web. [3]

## 2.4 Phishing

### 2.4.1 What is Phishing

Phishing attacks are forgeries that look to come from a reliable source yet compromise a variety of data sources. Attackers can gain access to our online accounts and personal information, gain permission to modify and compromise connected systems such as point-of-sale terminals and order processing systems, and in some cases, hijack entire computer networks until a ransom is paid. For financial benefit, hackers are sometimes satisfied with obtaining our personal data and credit card information. In other circumstances, phishing emails are sent to collect employee login information or other details for use in more aggressive assaults against a small group of people or a single firm. Phishing is a sort of cyber assault that everyone should be aware of in order to protect themselves and ensure that email security is maintained across a business. Phishing begins with a phony email or other kind of communication intended to entice a victim. The communication is disguised to appear as if it was sent by a reliable source. If the victim is duped, he or she will typically be persuaded to provide personal information on a fake website. Malware is sometimes installed on the computer of the target. [4]

### 2.4.2 History of Phishing

In the mid-1990s, hackers began employing phishing emails to "fish" for information from unsuspecting users, coining the word "phishing." Because the early hackers were known as "phreaks," the name "phishing," with a "ph," was coined. Phishing emails attempt to entice recipients to take the bait. And once they've been hooked, the user as well as the company are in big trouble. The history of phishing, like that of many other popular hazards, dates back to the 1990s. When AOL was a popular content system with internet access, attackers pretended to be AOL workers and utilized phishing and instant messaging to mislead customers into giving their passwords, allowing them to hijack accounts. [5]

### 2.4.3 Types of Phishing

### 1. Deceptive phishing

The most common sort of phishing fraud is deceptive phishing. Fraudsters pose as a real company in order to obtain people's personal information or login passwords. Threats and a sense of urgency are used in these emails to terrify recipients into doing what the attackers want. The success of a deceptive phish is determined by how much an attack email looks like legitimate correspondence from a faked firm. Recognizing this, users should carefully examine all URLs to check if they lead to an unfamiliar or questionable website. They should also be aware of generic salutations, grammatical flaws, and spelling issues.

### 2. Spear phishing

Spray and pray strategies aren't used in all phishing scams. Some ruses rely on a human touch more than others. They do it because they wouldn't be able to succeed if they didn't. Fraudsters personalize attack emails with the target's name, position, company, work phone number, and other information in order to fool the receiver into thinking they have a connection with the sender. It tricks the user into visiting a malicious URL or opening an email attachment in order to obtain their personal information. Given the quantity of data required to build a convincing attack effort, it's no wonder that spear phishing is widespread on social media sites like LinkedIn, where attackers may combine data from several sources to craft a targeted attack email. Organizations should conduct continuing staff security awareness training that, among other things, discourages users from disclosing sensitive personal or corporate information on social media to protect against this type of scam. In addition, businesses should invest in technologies that scan inbound emails for known harmful links or attachments. This solution should be able to detect indicators of known malware as well as zero-day threats. [6]

### 3. Whaling

Fraudsters can choose to undertake CEO fraud if their attack is successful. CEO fraud is the second step of a business email compromise (BEC) scam, in which attackers exploit a CEO's or other high-ranking executive's compromised email account to authorize fraudulent wire transfers to a financial institution of their choosing. Alternatively, they can use the same email account to conduct W-2 phishing, in which they seek W-2 information for all employees in order to submit false tax returns on their behalf or post the information on the dark web. Because bosses frequently fail to participate in security awareness training with their workers, whaling attacks are successful. To combat the concerns of CEO fraud and W-2 phishing, businesses should require that all employees, including executives, get ongoing security awareness training. Multi-factor authentication (MFA) channels should also be injected into financial authorization processes so that no one can authorize payments alone through email.

### 4. Vishing

This sort of phishing attempt foregoes sending an email in favor of making a telephone call. An attacker can carry out a vishing campaign by setting up a Voice over Internet Protocol (VoIP) server to imitate numerous entities in order to steal sensitive data and/or payments, according to Comparitech. The FBI discovered that in 2020, malicious actors employed such approaches to ramp up their vishing attempts and target remote workers. Users should avoid taking calls from unknown phone numbers, never give out personal information over the phone, and use a caller ID app to protect themselves from vishing attacks.

### 5. Smishing

Vishing isn't the only sort of phone-based phishing. Smishing is a term used to describe what they can do. This approach uses fraudulent text messages to get consumers to click on a dangerous link or provide personal information. Users can help protect themselves from smishing attacks by looking up strange phone numbers and calling the company mentioned in suspicious SMS messages if they have any questions. [7]

### 6. Pharming

As people become more aware of classic phishing schemes, some con artists are abandoning the practice entirely. Rather, they're turning to pharming. The domain name system used by the Internet to transform alphabetical website names to numerical IP addresses so that it can locate and guide users to computer services and devices, is used in this phishing tactic. A pharmer targets a DNS server and alters the IP address associated with an alphabetical website name in a DNS cache poisoning attack. As a result, an attacker can drive people to any malicious website they want. This is true even if the victim types in the right URL. Organizations should encourage employees to enter login credentials only on HTTPS protected sites to avoid pharming attacks. Antivirus software should be installed on all company devices, and virus databases should be updated on a regular basis. Finally, they must keep up with security updates given by a reputable Internet Service Provider. [8]

## 2.5 API (Application Programming Interface)

In today's commercial world, software has become a need. Without software, company success has become difficult to imagine. As a result, breakthroughs in the field of software development have been made in terms of how software is built and tested. Over the years, software development has been viewed as a demanding process that necessitates numerous actions. As a result, developers are constantly looking for new ways to make the software development process easier. Application Programming Interfaces (APIs) are one of these software development advances. APIs are a necessary part of the software ecosystem. These software ecosystems have shown to be an excellent approach to build massive software solutions based on a shared technical platform. APIs have existed since the invention of the personal computer. APIs were originally designed to facilitate communication between two or more programs. APIs, on the other hand, began to appear on the web around the year 2000. APIs have piqued the interest of practitioners and scholars since then, to the point where some analysts believe that we now live in the API economy. This viewpoint is bolstered by the reality that we have never been more linked, and APIs are principally responsible for this interconnectedness of people, apps, and systems. APIs are thus becoming the backbone of the digital ecosystem, which aims to connect businesses and economies in order to produce value and expand possibilities. APIs are widely used in the mobile application business, which is one of the fastest expanding areas of information technology. APIs are generally used by mobile app developers to create trustworthy and compatible applications. Despite these advancements, academics have yet to assess the current status of API research in order to provide guidance for future study. As a result, the current study aims to give an overview of existing API research in order to assess the current state of the field and guide future research. APIs have been defined by a number of authors from various angles. While some authors provide a specific definition for APIs, others prefer to provide hints by describing API properties or traits. These definitions might be seen from a technical or sociotechnical perspective. A deeper understanding of APIs, in my opinion, extends beyond the technical definition and encompasses a broader idea that takes into account the perspectives of users and practitioners. APIs make it easier to reuse code and increase software development productivity. APIs are defined by other authors as a collection of programs packed with interfaces that make it easier for other developers to use them. Similarly, APIs facilitate software reuse by providing pre-implemented capabilities, decreasing programmers' effort and time spent designing software. Interfaces are extremely important in the software development process. Interfaces have grown increasingly important in modern software development, according to previous study. In general, the above answers appear to be more technical and may be less understandable to a layperson. Other, less technical definitions of APIs claim that APIs provide a common platform for software to communicate with one another. This common ground allows various software to communicate with one another. They can create value by exchanging information and services across and within enterprises. [9]

# Chapter 3: Web Analytics

## 3.1 Bounce Rate

In web investigation, 'bounce rate' alludes to the level of meetings that have just a single site visit. Bounce sessions are generally viewed as disappointments by advertisers and web examiners. While certain bounces are to be sure disappointments, many are significant, including long visits that just happen not to incorporate any extra site hits. The primary justification for misconception bounces as disappointments is that, in examination, there is no marker to signify the end of the session. Thus, long single site hit meetings are inappropriately estimated and look equivalent to short sessions through the analytics dashboard. Also, single site hit sessions are on the ascent because of the expansion in portable perusing, as well as the capacity of web indexes to place the right content before searchers, decreasing the need to demand another page. [10]

## 3.2 Page Per Visit

Pages per visit is a Web investigation proportion of the number of bits of content (Web pages) a specific client or gathering of clients sees on a solitary site. Pages per visit is normally shown as a normal, which is determined by isolating the all-out number of site hits by the absolute number of guests. It tends to be additionally separated by nation, area and demographic in complex Web analytics programs.

Pages per visit is an expansive proportion of how convincing visitors observe the substance on a site and how well it is sorted out for route.

A bigger number of pages per visit by and large implies that the content on a given Web page is great and the connected content - different connections on the page - is important and intriguing to the point of justifying a second, third or fourth snap. A low number of pages per visit might propose that either the content is missing or the capacity to explore effectively between content is hampered by the site's general plan/association.

## 3.3 Time on Site

Time on Site is a sort of visitor report that gives information on how much time (in minutes or seconds) visitors have spent on a site. While reviewing the time on site report in the Web investigation program it is critical to recollect that the outcomes can be deluding on the grounds that at times the visitors might have been cooperating with the pages and website content or they might have left the program window open and were not really seeing the page.

## 3.4 Total Visits

Total Visits is the amount, all things considered, to a site and is utilized as an estimation of site traffic and site guests. The total site visitors can be investigated for single site or contender set of sites to benchmark execution.

A visit (session) for a site is determined when a visitor gets to at least one page. Ensuing site views are remembered for a similar visit until the visitor is inert for over 30 minutes. Assuming a visitor becomes dynamic again following 30 minutes, that considers another visit. Another meeting will likewise begin at midnight.

Total Visits is utilized to get the general number of connections with a site. Total Visits shows the number of chances altogether a site needs to guarantee that visitors complete the client venture. Visits can be additionally dissected utilizing nation and time-frame channels.

# Chapter 4: Malicious Link Detection Model

The built web application takes in input from the user. A website is entered into the search bar. Then, it is run against web analytics in the background through the API. The metrics calculated are Bounce Rate, Page Per Visit, Time on Site and Total Visits. If all metrics are found to be valid, the output shown is that it is a safe site. If the metrics are mismatched, the message shown that it is malicious.

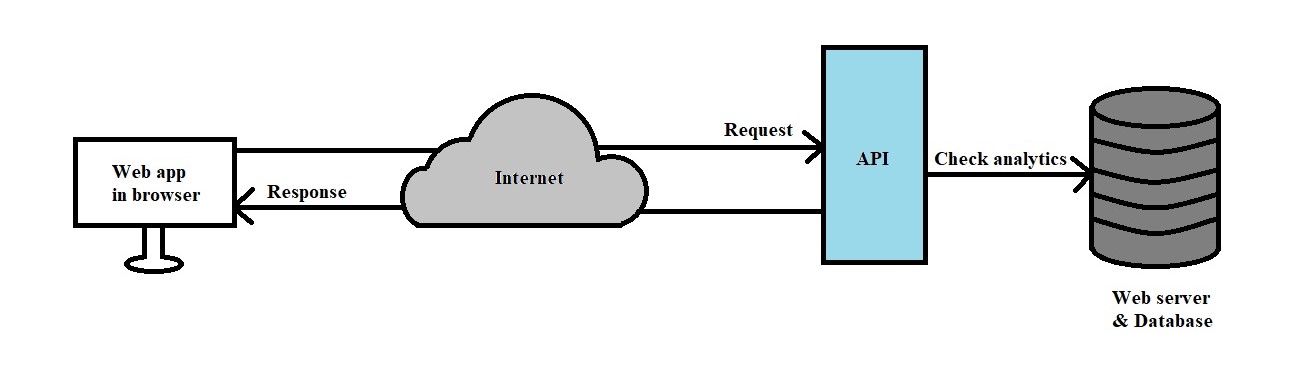


Figure 4.1: Malicious Link Detection Model

# Chapter 5: Web Analytics Tests and Result Analysis

The following metrics are tested for a website to check whether it is safe or malicious:

## 5.1 Testing Bounce Rate

The ratio corresponding to which a site is visited primarily on the first page and then left is bounce rate. The threshold taken by the API is greater than or equal to 0.15 and less than or equal to 20.

## 5.2 Testing Page Per Visit

Pages Per Visit by a user is taken ranging from greater than or equal to 1 and less than or equal to 15 for a valid website.

## 5.3 Testing Time on Site

Time spent surfing on the website is taken greater than or equal to 30 and less than or equal to 800 for a website safe to visit.

## 5.4 Testing Total Visits

For a website to be deemed safe, Total Visits has to be greater than or equal to 999. Less traffic on a website usually signifies that the site is invalid or has malicious content.

# Chapter 6: Structural Overview of the Application



Figure 6.1: Use Case diagram of the system

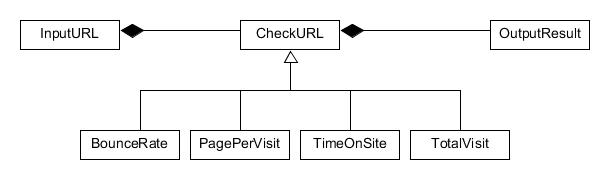


Figure 6.2: Class diagram of the system

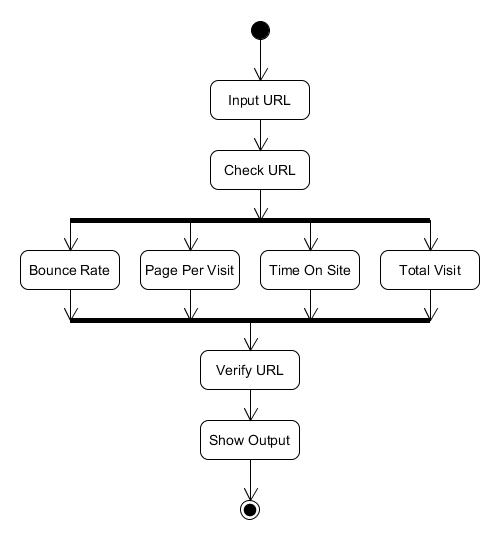


Figure 6.3: Activity diagram of the system

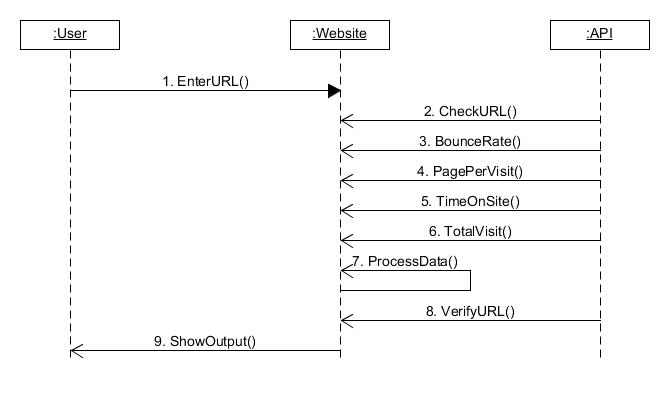


Figure 6.4: Sequence diagram of the system

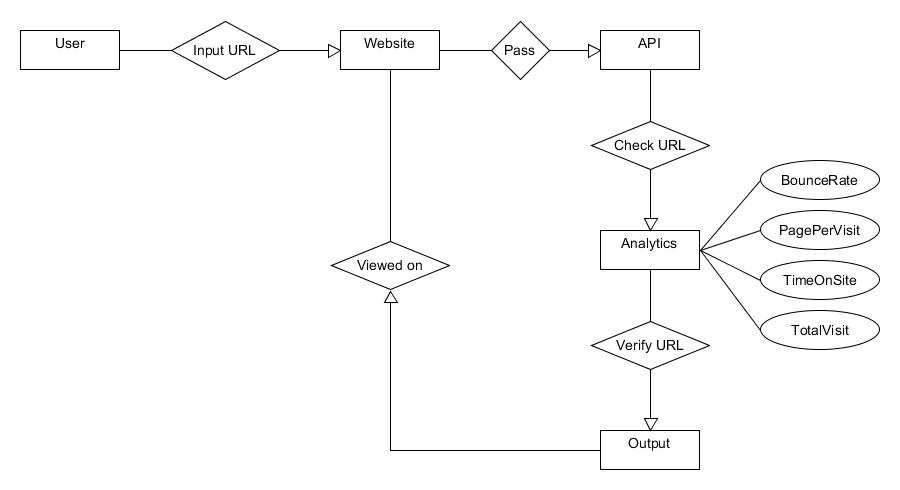


Figure 6.5: Entity Relationship diagram of the system

# Chapter 7: Benefits of using the Malicious Link Detection System

The Malicious Link Detection System is easy to use. It has a simple interface that can be operated by anyone. The program is also designed dynamically which enables it to work and respond fast. This is essential for users to check for possible threats ahead before moving forward to a website. The system is flexible to future changes and further development. Integration with other applications and systems is possible with minor alterations to the source code. The system can serve as a proving ground for those new to the metrics of cyber security.

# Chapter 8: Proposal

## 8.1 Motivation

Access to the internet is a common phenomenon in the modern world. However, unfiltered browsing can lead to landing in websites that can possess great threats to inexperienced users. These include malware attacks and theft of data. Web contents can be misleading and direct users to believing they are true. Our aim was to develop a simple yet robust system that will be fast and easy to use. The system can serve as a testbed for detecting malicious links and a prerequisite to check unfamiliar websites before visiting them.

## 8.2 Proposal

Our proposal is to find whether or not a website is safe to visit. Instead of complex and expensive approach, we focused on a simple and inexpensive route to detect malicious links. This can provide accurate results in minimum time along with a simple interface of operation.

Our work is targeted at inexperienced users who often fall prey to malicious web content. The system can be used by them for simple and secure web navigation.

# Chapter 9: Web Application Testbed

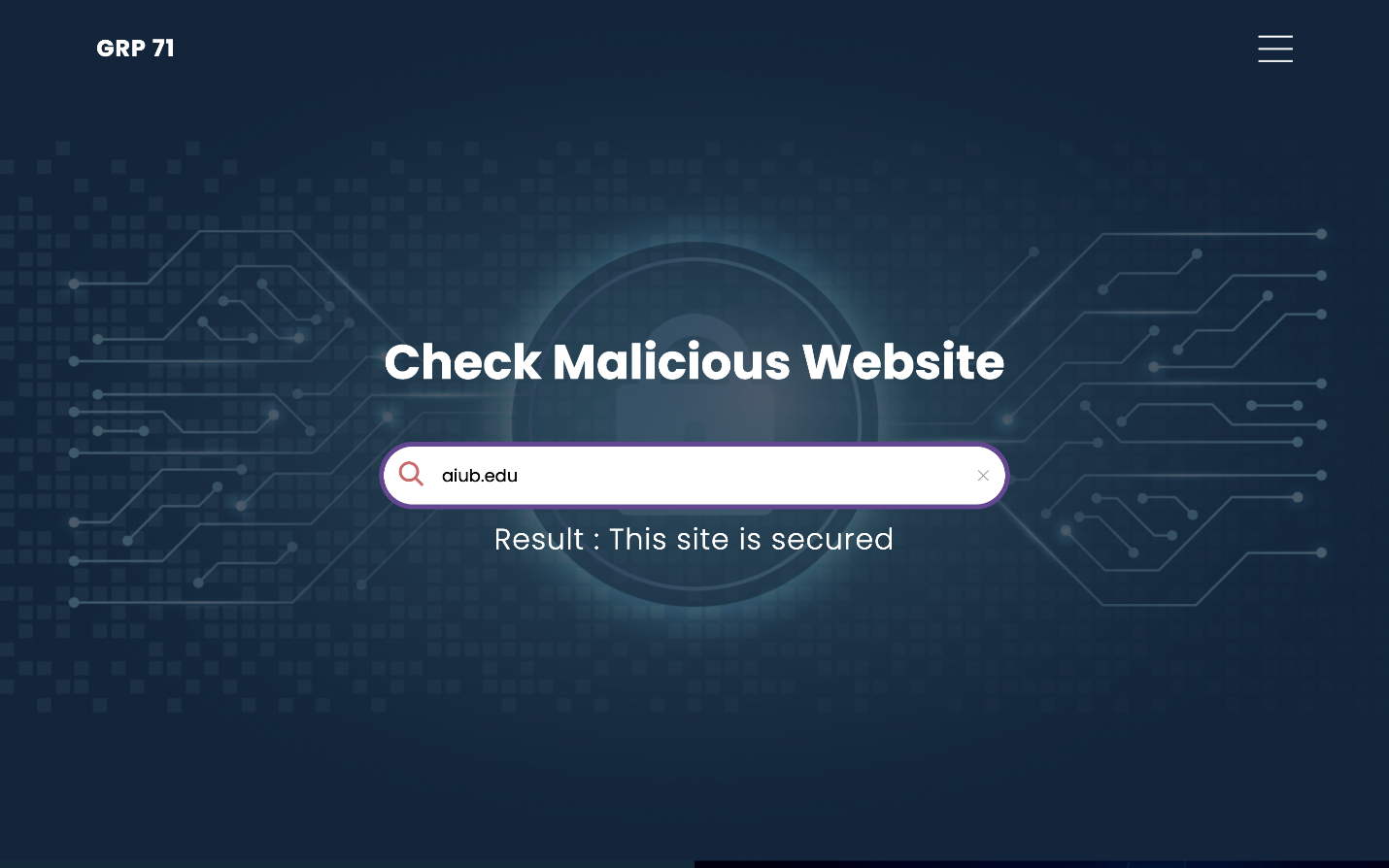


Figure 9.1: Secure Site demonstration

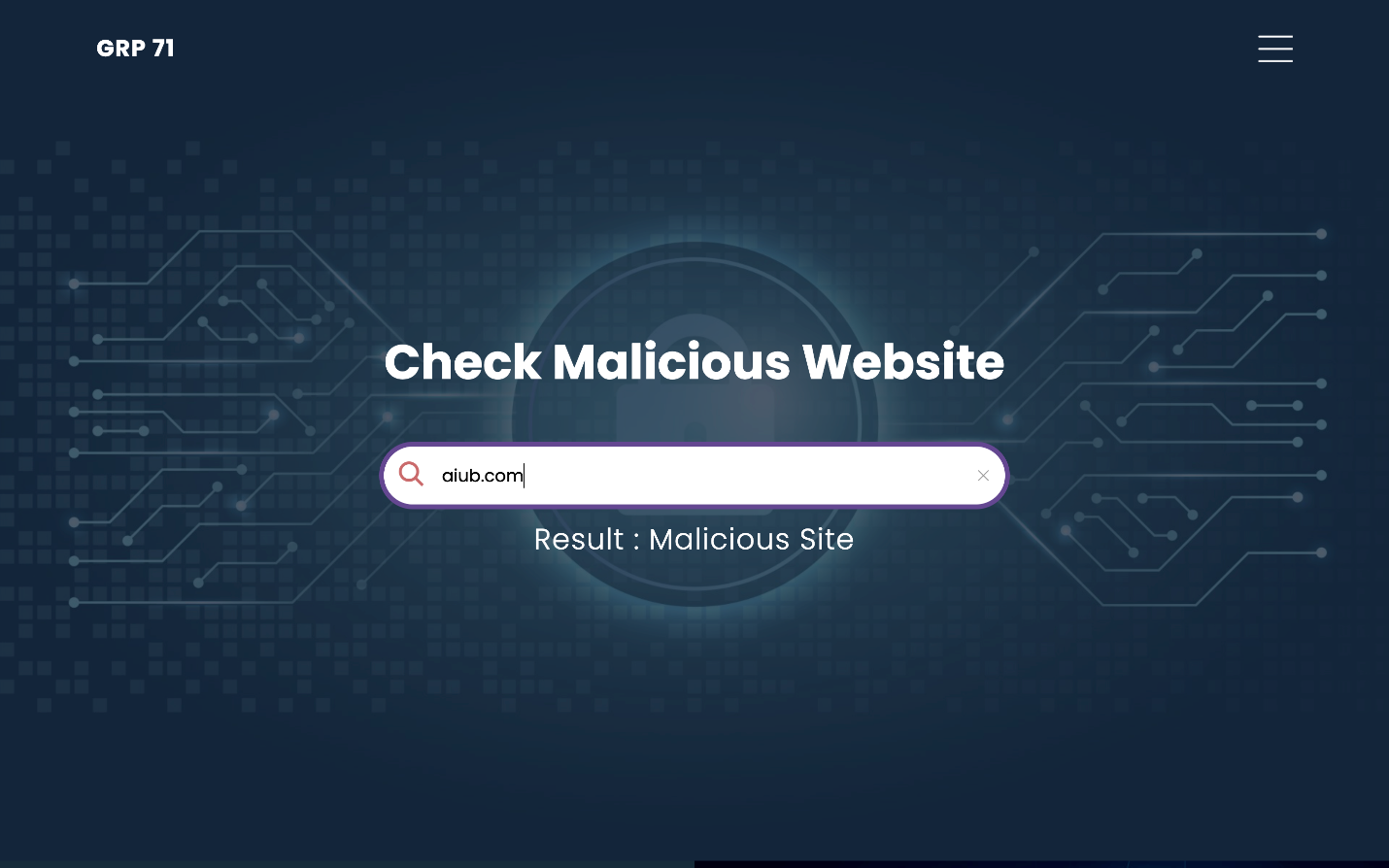


Figure 9.2: Malicious Site demonstration

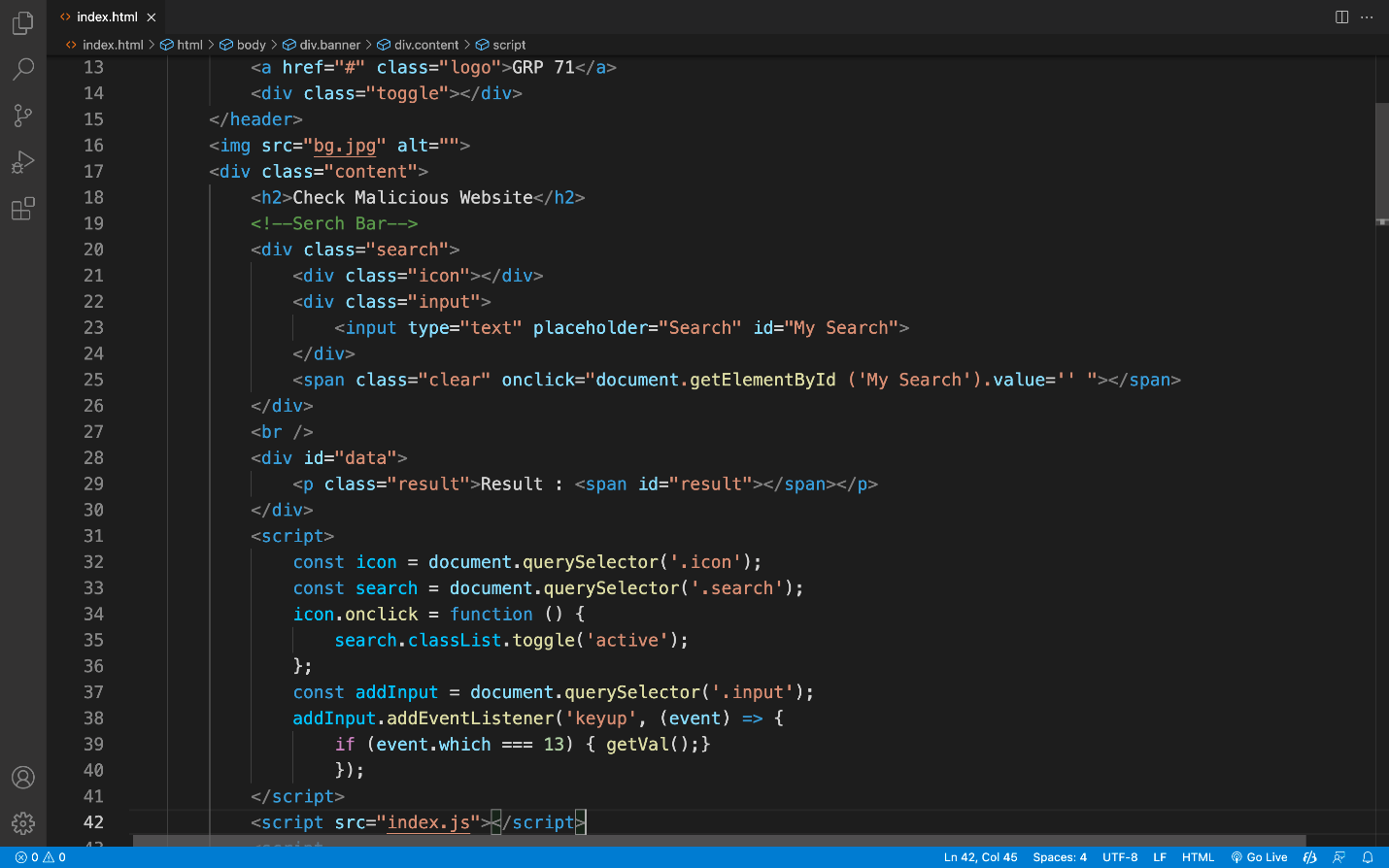


Figure 9.3: Web Interface



Figure 9.4: Call API

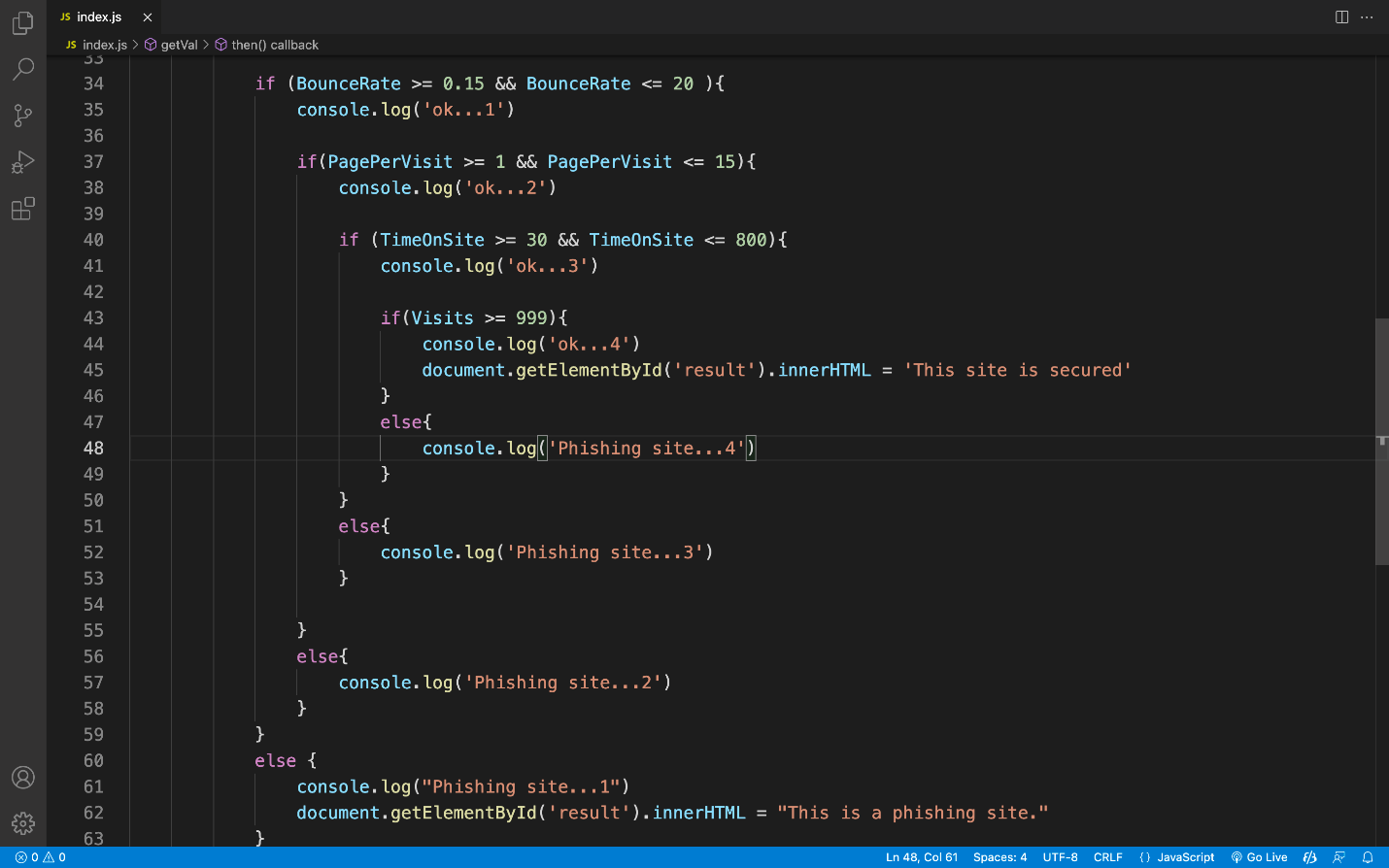


Figure 9.5: Scan Analytics

# Chapter 10: Future Work

Conducting further research and development over the existing solution can help to bring about advanced solutions to the simple solution we have provided. Integrating Machine Learning techniques and implementing advanced solutions such as Firewall, Firmware Security and Network Security can help detect and block malicious web content in real time. Pseudo malware detection can be the next step of prospective research of this thesis.

# Chapter 11: Discussion & Conclusion

## 11.1 Discussion

Malicious Link Detection System is a prospective project with wide scopes of development. Primary stages of internet security can be achieved through web analytics. However, considering the fast-paced world and ever-changing technology, it mandates the ongoing development of security systems. This is where the prospect of cyber security comes into action.

## 11.2 Conclusion

This paper documents our work on Malicious Link Detection using Web Analytics. The system can be beneficial to both inexperienced and experienced users. Through the use of API in the backend, we are able to run analytics of a website by inputting that in the front page i.e., the frontend. The system shows input readily from calculated data against the set metrics threshold in the code.

Program functionality is flexible and integrable in cross platforms since it is web based. Analytical tests are overlooked towards complex solutions but we tried to prove that this is not the case. A simple yet robust solution can help solve or warn against online threats.

# Chapter 12: References

[1] Bilge, L., Kirda, E., Kruegel, C. and Balduzzi, M., 2011, February. Exposure: Finding malicious domains using passive DNS analysis. In *Ndss* (pp. 1-17).

[2] Crocker, Steve, David Dagon, Dan Kaminsky, DKH Danny McPherson, and Paul Vixie. "Security and other technical concerns raised by the DNS filtering requirements in the PROTECT IP Bill." *White Paper* 6 (2011): 1.

[3] Dolmans, Ralph. "Preventing DNS Amplification Attacks using white-and greylisting." *Unpublished dissertation in partial fulfillment of the requirements for the degree of Master of System and Network Engineering, University of Amsterdam, Amsterdam, Netherlands* (2013).

[4] Khonji, Mahmoud, Youssef Iraqi, and Andrew Jones. "Phishing detection: a literature survey." *IEEE Communications Surveys & Tutorials* 15, no. 4 (2013): 2091-2121.

[5] Dhamija, Rachna, J. Doug Tygar, and Marti Hearst. "Why phishing works." In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 581-590. 2006.

[6] Tandale, Kiran D., and Sunil N. Pawar. "Different types of phishing attacks and detection techniques: A review." In *2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC)*, pp. 295-299. IEEE, 2020.

[7] Chiew, Kang Leng, Kelvin Sheng Chek Yong, and Choon Lin Tan. "A survey of phishing attacks: Their types, vectors and technical approaches." *Expert Systems with Applications* 106 (2018): 1-20.

[8] Gupta, Surbhi, Abhishek Singhal, and Akanksha Kapoor. "A literature survey on social engineering attacks: Phishing attack." In *2016 international conference on computing, communication and automation (ICCCA)*, pp. 537-540. IEEE, 2016.

[9] Ofoeda, Joshua, Richard Boateng, and John Effah. "Application programming interface (API) research: A review of the past to inform the future." International Journal of Enterprise Information Systems (IJEIS) 15, no. 3 (2019): 76-95.

[10] Kamerer, D., 2020. Reconsidering bounce rate in web analytics. Journal of Digital & Social Media Marketing, 8(1), pp.58-67.

# Appendix A: Code

## A.1 Website Structure

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Web Analyze</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<!--Banner-->

<div class="banner">

<header>

<a href="#" class="logo">GRP 71</a>

<div class="toggle"></div>

</header>

<img src="bg.jpg" alt="">

<div class="content">

<h2>Check Malicious Website</h2>

<!--Search Bar-->

<div class="search">

<div class="icon"></div>

<div class="input">

<input type="text" placeholder="Search" id="My Search">

</div>

<span class="clear" onclick="document.getElementById ('My Search').value='' "></span>

</div>

<br />

<div id="data">

<p class="result">Result : <span id="result"></span></p>

</div>

<script>

const icon = document.querySelector('.icon');

const search = document.querySelector('.search');

icon.onclick = function () {

search.classList.toggle('active');

};

const addInput = document.querySelector('.input');

addInput.addEventListener('keyup', (event) => {

if (event.which === 13) { getVal();}

});

</script>

<script src="index.js"></script>

<script

src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"

integrity="sha512-894YE6QWD5I59HgZOGReFYm4dnWc1Qt5NtvYSaNcOP+u1T9qYdvdihz0PPSiiqn/+/3e7Jo4EaG7TubfWGUrMQ=="

crossorigin="anonymous"

referrerpolicy="no-referrer"

></script>

</div>

</div>

<!--About-->

<section class="about">

<div class="contentBx">

<h2 class="heading">About Us</h2>

<p class="text">The project is a web-based detection tool for malicious websites. Websites are evaluated against web analytics in the background in real time. Visits, Page Per Visit, Time On Site, and Bounce Rate are examples of these metrics.</p>

</div>

<div class="imgBx"></div>

</section>

<!--Services-->

<section class="services">

<h2 class="heading">Our Services</h2>

<p class="text">We are using a variety of approaches to provide network security to our clients. Our main objective is to protect users from harmful websites.</p>

<div class="container">

<div class="serviceBx">

<div>

<img src="icon1.png" alt="">

<h2>Design</h2>

</div>

</div>

<div class="serviceBx">

<div>

<img src="icon2.png" alt="">

<h2>Development</h2>

</div>

</div>

<div class="serviceBx">

<div>

<img src="icon3.png" alt="">

<h2>Branding</h2>

</div>

</div>

</div>

</section>

<!--Technology-->

<section class="technology">

<div class="contentBx">

<h2 class="heading">Edge Technology</h2>

<p class="text">The API receives data from the user via the website's search bar. Then, in the back end, it performs an analysis of various data and generates a report for that website. It then makes a determination as to whether the requested site is safe or harmful.</p>

</div>

<div class="imgBx">

<img src="tech.png" alt="">

</div>

</section>

<!--Client-->

<section class="client">

<h2 class="heading">Our Media Portal</h2>

<p class="text"></p>

<div class="imgBx">

<img src="brand1.png" alt="">

<img src="brand2.png" alt="">

<img src="brand3.png" alt="">

<img src="brand4.png" alt="">

</div>

</section>

<!--Testimonials-->

<section class="testimonials">

<h2 class="heading">Client Experience</h2>

<div class="container">

<div class="contentBx">

<div>

<p>Opinion</p><br>

<h3>Anonymous User</h3>

</div>

</div>

<div class="contentBx">

<div>

<p>Opinion</p><br>

<h3>Anonymous User</h3>

</div>

</div>

</div>

</section>

<!--Contact-->

<section class="contact">

<h2 class="heading">Contact Us</h2>

<p class="text"></p>

</section>

<section class="about">

<div class="contentBx redbg">

<div class="form">

<div class="inputBx">

<input type="text" name="" placeholder="Full Name">

</div>

<div class="inputBx">

<input type="text" name="" placeholder="Email Address">

</div>

<div class="inputBx">

<input type="text" name="" placeholder="Mobile No.">

</div>

<div class="inputBx">

<textarea placeholder="Write Your message here"></textarea>

</div>

<div class="inputBx">

<input type="submit" name="" value="send">

</div>

</div>

</div>

<div class="imgBx2"></div>

</section>

<!--Footer-->

<section class="footer">

<p class="text">Design & Development By Nuhan Islam</p>

<ul>

<p class="text">Follow us on : </p>

<li><a href="#"><img src="facebook.png"></a></li>

<li><a href="#"><img src="twitter.png"></a></li>

<li><a href="#"><img src="linkedin.png"></a></li>

</ul>

</section>

</body>

</html>

## A.2 Website Functionality

function getVal() {

const val = document.querySelector('input').value;

console.log(val);

let result = '';

let requestOptions = {

method: 'GET',

redirect: 'follow',

};

if (data) {

document.getElementById('data').style.display = 'block';

}

fetch(

`https://data.similarweb.com/api/v1/data?domain=${val}`,

requestOptions

)

.then((response) => {

if (response.ok) {

return response.json();

} else {

throw document.getElementById('result').innerHTML = 'Malicious Site';

}

})

.then((data) => {

console.log(data);

BounceRate = data.Engagments.BounceRate;

PagePerVisit = data.Engagments.PagePerVisit;

TimeOnSite = data.Engagments.TimeOnSite;

Visits = data.Engagments.Visits;

if (BounceRate >= 0.15 && BounceRate <= 20 ){

console.log('ok...1')

if(PagePerVisit >= 1 && PagePerVisit <= 15){

console.log('ok...2')

if (TimeOnSite >= 30 && TimeOnSite <= 800){

console.log('ok...3')

if(Visits >= 999){

console.log('ok...4')

document.getElementById('result').innerHTML = 'This site is secured'

}

else{

console.log('Phishing site...4')

}

}

else{

console.log('Phishing site...3')

}

}

else{

console.log('Phishing site...2')

}

}

else {

console.log("Phishing site...1")

document.getElementById('result').innerHTML = "This is a phishing site."

}

})

.catch((error) => console.log('FETCH ERROR:', error));

}