VULNERABILITY DETECTION FOR HTTPS SPOOFING AND EMAIL HIJACKING ATTACKS OF WEB APPLICATION USING BOYER MOORE STRING MATCHING ALGORITHM

THAZIN EAINDRA BO

M.C.Sc. **DECEMBER, 2022**

VULNERABILITY DETECTION FOR HTTPS SPOOFING AND EMAIL HIJACKING ATTACKS OF WEB APPLICATION USING BOYER MOORE STRING MATCHING ALGORITHM

By

Thazin Eaindra Bo

B.C.Sc.

A dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science (M.C.Sc.)

University of Computer Studies, Yangon

DECEMBER , 2022

ACKNOWLEDGEMENTS

Firstly, I would like to show my appreciation to all teachers who examined and helped to success my thesis during the studying at the University of Computer Studies, Yangon (UCSY).

Secondly, I would like to say a special thank you to principal **Prof. Dr. Mie Mie Khin,** Rector, University of Computer Studies, Yangon who gave me a chance to develop my thesis for her kindly guidance during the studying of thesis period.

And I would like to special thank you to **Prof. Dr. Mie Mie Thet Thwin,** former Rector, University of Computer Studies, Yangon who gave me a chance to develop my thesis for her kindly guidance during the studying of thesis period.

I would like to show my admirable gratitude to **Prof. Dr. Si Si Mar Win** and **Prof. Dr. Tin Zar Thaw**, Dean of the Master 26th batch, University of Computer Studies, Yangon, for both excellent and kindly guidance.

I would like to deeply thankful and gratitude to former Course Coordinator, **Dr. Mie Mie Su Thwin** who told me to understand about my thesis field, "Cyber Security" and the piratical applying businesses about this field.

My sincere thanks and respect also go to my supervisor, **Dr. Zin Thu Thu Myint**, Associate Professor of Faculty of Information Science (FIS) department, the University of Computer Studies, Yangon, for her valuable advice, giving me detailed and kindly guidance, support and comments throughout the thesis process period. Especially, I admired for her excellent guidance on her all candidates and helped me for my thesis topic, "Vulnerability Detection for HTTPS Spoofing and Email Hijacking Attacks of Web Application Using Boyer Moore String Matching

Algorithm".

In addition, I would like to thank all of my thesis’s board examiners who gave the precious comments and corrections about my thesis for getting good result. To reading the committee teachers, especially **Daw Mya Hnin Mon**, Associate Professor, English Department, for her valuable supports and editing my thesis book from the language point of view.

Moreover, I am deeply thankful to my parents and my family for supporting, caring and encouragement to me from the childhood to the present time. Finally, I am extremely thankful to all my teachers, my colleagues and all of my friends for their valuable and precious help and general advice.

STATEMENT OF ORIGINALITY

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

--------------------- --------------------

Date Thazin Eaindra Bo

ABSTRACT

Nowadays, many people use the internet for more than one purposes. Among these purposes, they mostly apply the web application which is one of the internet usage technologies. A web application is composed of a web server and web browser in other terms client-side and server-side. When people access a web application from any one browser, firstly they send a request to the web server and then this web server responds this request to the web application server and processing continued tasks Today, web applications are popular for people because these have many advantages: easily use and cost effective for users. Maintaining web application security is the important case for users because web application may have vulnerabilities. Web application vulnerabilities are weakness of this web application and can find many kinds of reasons. For example, application developer errors within coding, application design weakness and so on. So, attackers can be tried by using these vulnerabilities to exploit system for getting privileges and personal information. In this paper, the proposed algorithm can find two types of vulnerabilities included in Man-In-The-Middle (MITM) attack which are HTTPS Spoofing and Email Hijacking attacks. For cyber-security field, MITM attack is well-known attack and HTTPS Spoofing and Email Hijacking are kinds of attacks in MITM types. In this thesis, Boyer Moore string matching algorithm uses to search including vulnerabilities with attacked datasets. The proposed algorithm compares pattern and text and then shift more than one position at a time and it save time consuming. This proposed system used Python programming language. Finally, the evaluation results show that how results accurate based on having false negative and false positive rate.

Keywords: HTTPS Spoofing Attack, Email Hijacking Attack, Boyer Moore String Matching Algorithm, Man-In-The-Middle Attack, Web Application

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1
INTRODUCTION

As a trend of developing technology, most people use web applications for their respective fields such as education, business and communication field and so on. Using web applications, people process using personal data for their needs and store this data on them. Some attackers try to get unauthorized access into user's accounts and steal sensitive information for using malicious purposes. And they can be used this information for many malicious purposes such as identity theft, security violation, other illegitimate processes and so on. Vulnerabilities are web applications' drawbacks that can be caused illegal processes by attackers. In this system, vulnerabilities can also be checked for information security before using.

There are many kinds of vulnerabilities in web applications. Among them, Man-In-The-Middle(MITM) attack is widespread attack and this attack target connections between two. MITM attack is one of the most popular top 10 common types of Cyber-security field. MITM attack has many types of attack. They are HTTPS Spoofing, Email Hijacking, IP Spoofing, DNS Spoofing, Wi-Fi Eavesdropping and others. Today, people are using HTTPS protocols for their searching process from browsers. HTTPS extension refers to Hypertext Transfer Protocol Secure. HTTP protocol is a protocol for communication and that protocol encrypted http data over the internet for both client and server sides. In HTTPS spoofing, an attacker uses similar domain that look like domain of targeted website. Moreover, people widely use emails for their working and communication fields. In Email Hijacking, attackers try to spoof target's email address, access target's account of email using this target's information to perform malicious purposes.

At present, people are using web application for many reasons. So, they need web application security to secure information and protect stealing information. Today, many people are using vulnerability detection tools for web application security. In vulnerability detection, there are many methods such as true positive rate and true negative rate, false positive rate and false negative rate. Some people occurs a high rate of false negative and false positive rate. A false positive defines where the detected results show that the web application exist vulnerability but the vulnerability actually does not exist. And a false negative defines opposite to a false positive where the detecting results show no vulnerability but in actually, vulnerability exists.

Thus, the system proposed vulnerability detection for this two kinds included in MITM attack by using Boyer Moore String Matching algorithm with a low rate of false negative and false positive. For searching process, this algorithm is well-known algorithm. Moreover, there are many others string matching algorithm but some matching algorithm compare each character by character. So, they require many times for comparison. The proposed algorithm is reliable for string matching cases within the system and it can save time for comparison because it compare pattern and then shift more than one character. In summary, this proposed method protects the web application from the accessing attacks by searching the vulnerability. In this thesis, the following section discusses some of the related works.

1.1 Related Works

In this thesis, the system is aimed to provide the administrators who require to get secure web applications and to protect any attacks from the attackers. This section discusses the former studies of detecting web application vulnerabilities associated with attack.

The first study is that "an approach for detecting Man-In-The-Middle attack using DPI and DFI" is implemented by Argha Ghosh and A. Senthilrajan. DPI and DFI are techniques.[2] For DPI, this is used to manage and analyze for real-time network traffic with high-speed networks and DFI is also used to classify network flow feature for traffic of incoming network. This paper used a technique for detection (MITM) attack using Deep Packet Inspection and Deep Flow Inspection. Between DPI and DFI, a co-ordinate module is used for maintaining data transmission and identifying incoming, outgoing traffic affected by MITM attack or not. In this experimental evaluation section, Wireshark is used to analyze for the system's performance of the proposed approach.

In the other study, Robert A. Sowah, Kwadwo B. Ofori-Amanfo, Godfrey A. Mills, and Koudjo M.Koumad implemented about "detection and prevention of Man-in-the-Middle Spoofing attacks in MANETs using in Artificial Neural Networks (ANN)".[7] In this paper, a Mobile Ad-Hoc Network (MANET) meaning is a convenient wireless infrastructure and displays advantage for network settings. For intrusion detection, ANN classification algorithm use for attack detection, reconfiguration, isolation and measured on dataset with mobility patterns and network-varied traffic. The final detection rate of this paper is 88.235%. To perform MITM attack on simulation platform, the working of this paper is not only offered productive and less expensive way but also identified time as an important factor in determining.

A. Zubaidah MohdSaleha, N. A. Rozalia, A. G. Bujaa, K. Abd. Jalila, F. H. M. Alia, T. F. Abdul Rahmana proposed a paper about "a detection method for web application vulnerabilities using Boyer Moore String Matching Algorithm".[1] In this paper, the proposed method displayed that ability to detect vulnerabilities based on no false positive and have false negative with minimum processing time. This paper tested four common vulnerabilities and they are SQL Injection, Buffer Overflow, Cross Site Scripting and Cross Site Request Forgery. This method processes only detecting web pages. Finally, this paper recommendation should be combined hybrid string matching for better detection ability.

U. Rahamathunnisa A.P, Vellore N. Manikandan, V. U. S. Kumaran, V. C. Niveditha analyzed a paper about "preventing from phishing attack by implementing url pattern matching technique in web".[11] This paper described that identification of phishing websites using uniform resource locators (URLs) matching of webpages. This paper used TF-IDF algorithm and this algorithm measured appearance of particular term in the whole document. This paper has been tested phishing attack according to user URL request by matching with a blacklist and whitelist of database. Finally, this process runs browser backend and validates for each and every request.

1.2 Objectives of the Thesis

The main objectives of the thesis are as follows:

- to know the targets of attacks which can be intruded via web application vulnerability
- to protect information that can be used for many purposes, including identity theft, unapproved fund transfers or an illicit password change
- to provide the security of web application
- to assist web developers for detecting many web application's weaknesses
- to support information for https spoofing and email hijacking attacks detection

1.3 Organization of the Thesis

In this thesis, there are five chapters. Chapter 1 is the introduction section and this section including the introduction to web application vulnerability, the related works, the objectives, and the organization of the thesis.

Chapter 2 presents the background theory related to this thesis such as web application vulnerability and types of web application vulnerability, Man-In-The-Middle attack and types including HTTPS Spoofing attack and Email Hijacking attack and Boyer Moore string matching algorithm that are described in detail.

Chapter 3 describes the design of the proposed system describing system flow and flowchart diagrams for HTTPS Spoofing and Email Hijacking, the detail explanations with algorithm and the evaluation of the output resulting from the detection.

Chapter 4 presents the implementation of the proposed system in detail and the experimental result.

Finally, Chapter 5 includes the conclusion of this thesis that are the benefits, limitations and further extensions of the proposed system.

CHAPTER 2
BACKGROUND THEORY

In this research work, the related background theory is displayed in this chapter. This chapter explains web application vulnerability and string matching algorithm. In this chapter of the first section, the contents of web application vulnerability and the types of web application vulnerability are described. Next part, Man-In-The-Middle attack and its types including HTTPS Spoofing and Email Hijacking attacks are described. Next, string matching algorithm and the most common string matching algorithm are expressed in detail.

2.1 Web Application

A Web application is an application of program that has client side and server side applications. It is stored on a remote server and delivered over the Internet through any browser interface. Today, people use many types of web browser such as Google Chrome, Microsoft Edge, Mozilla Firefox, Internet Explorer, Safari and Opera. When people access a web application from any browser, firstly they send a request to the web server and this web server responds the request to web application server and process required tasks. Web services mean Web apps. Many websites can be contained Web apps. People use web applications for a wide variety of reasons such as e-commerce shops, online banking, online shopping etc.

A web application requires a web server, application server, and a database to operate its processes. Web servers manage the requests that come from a user while the server completes the requested task. A database can be used to store information. Many web apps are written in JavaScript, HTML5, or Cascading Style Sheets (CSS). These languages are utilized in Client-side programming language which helps to build an application front-end. Server-side programming is done to create the scripts for use in a Web app. Server-side programming language utilizes many languages such as Python, Java and Ruby and so on.

A web application needs a database, web server and application server. If a user sends HTTP requests through a web browser, HTTP is commonly used to send GET or POST method to the web server. The web server processes the request and sends the requested response to the web browser. And the browser accepts the response and shows the requested web page.

2.2 Web Application Vulnerability

Web application vulnerabilities can be classified with various terms. These vulnerabilities are security flaws, bugs, weaknesses, flaws in the software, and vulnerabilities in the computer program. Hackers can be exploited to compromise the application's security from weaknesses in a web application. This exploitation can lead to data leakage, data corruption, or tampering with data. This can become a high risk for organizations.

Many web applications are vulnerable to access for attackers. As more and more vulnerabilities being discovered in web applications every year, people are required to keep their security from attackers. To protect web applications from attackers, people should apply web application testing, analyzing their web applications from different places.

Web application security detection is a process of assessing the web application’s security from flaws, vulnerabilities, and loopholes to protect malware, data breaches, and other cyber-attacks. To test the security of a web application using security detection techniques, web developers and security administrators used web security testing. The objective for Web application security testing is to define any vulnerabilities or threats or integrity of the Web application. This testing process is included the identified vulnerabilities, possible threats and recommendations for overcoming the security failure.

Web application vulnerability assesses from known vulnerabilities in web applications. And they are measured the risk of exposure to security weakness of a website. Web application vulnerability assessment consists of two results:

- findings, risks and recommendations in a non-technical expression as an “overall” expression of the current security of the web applications tested;
- An extensive technical report includes the detailed findings, risks and recommendations. It is aimed at web application developers and hosting parties.

These reports are accurate and allow the user to identify vulnerabilities in a targeted manner. This allows the security level of web application to be improved by actions, based on priorities.

2.3 Types of Web Application Vulnerability

There are many web application vulnerabilities in cybersecurity field. Among them, the following are common vulnerabilities:

- Broken Authentication and Session Management
- SQL Injection
- Cross Site Scripting
- Cross Site Request Forgery
- Insecure Direct Object References
- Security Misconfiguration
- Invalidated Redirects and Forwards
- Insecure Cryptographic Storage
- Insufficient Transport Layer Protection
- Failure to restrict URL Access

2.3.1 Broken Authentication and Session Management

Many websites need users to login to enter their accounts with usernames or emails and passwords. If a user signs into a website, this site uses a proprietary algorithm to produce a unique session ID. A site will assign and send a unique session ID to log in for a user. A unique session ID can be cookies, URL parameters, authentication tokens and others that permit for communication between the user and web app for the valid session. The user's device uses that session ID as a key to their valid identity. A valid user can be impersonated to access the user's account by a cybercriminal. This leads to a broken authentication and session management attack.

With exploiting a broken authentication, an attack become by taking advantages of badly managed credentials and login sessions to masquerade as valid users. For example, someone can try to intercept a user's session ID or credentials to impersonate that look like a valid user. This case occurs when using on a public network (like public Wi-Fi) or a public computer that anyone can be accessed and intercepted.

2.3.2 SQL Injection

SQL Injection means structured query language injection and is known as SQLI. It is a common attack that uses malicious SQL code to access information for backend database manipulation. This information may include sensitive company data, private business information, user information or private customer details. Users use SQL queries to execute commands in database such as data retrieval, updates, and delete records. A successful attack may process unauthorized viewing of user information, the deletion tables and gaining administrative rights to a database for an attacker. SQL injections can be divided into three categories: In-band SQL Injection (Classic), Blind SQL Injection (Inferential) and Out-of-band SQL Injection.

For In-band SQL injection, attacker uses same channel of communication to process their attacks. In-band SQL injection include two categories. They are Error-based and Union-based SQL injection. In Error-based SQL injection, the attacker processes actions using the data from error messages in database to gather information about the database structure. In Union-based SQL injection, UNION SQL operator combines multiple select statements that produced from the database to get a HTTP response. The attacker supports data for this response.

In Blind (Inferential) SQL injection, using data payloads, the attacker sends to the server and reconstructs the database structure by observing behavior of the server and application. In-band SQL injection can be classified into two categories: Boolean based and Time based SQL injection. In Boolean based SQL injection, the attacker sends a SQL query to the database and the application returns a result based on query result (true or false). The information in the HTTP response may be change or unchanged based on result. The attacker can be processed on this generated result. In Time based SQL injection, the attacker can know from the time taken in database to respond a query result (true or false).

In Out-of-band SQL injection to start the attack and gather information, the attacker cannot use the same channel.

2.3.3 Cross Site Scripting

Cross-site scripting (XSS) is a type of web security vulnerability. This attack can be caused to compromise the interactions from a vulnerable application by an attacker. In this attack, an attacker process to masquerade as a targeted user, to process the user's performance, and to access the user's information. If the targeted user has privileged access with the application, the attacker might be gained full control over the application's data and function. Cross-site scripting processing returns malicious JavaScript to users by manipulating a vulnerable web site. When malicious code executes within targeted user's browser, XSS attack occurs from this user visiting the web page or web application.

2.3.4 Cross Site Request Forgery

Cross-site request forgery (CSRF) is a kind of web application vulnerability and this exploit a website or web application with unauthorized commands. CSRF exploits the trusted user having a particular site in a user's browser. CSRF attack tricks the targeted user into clicking a malicious URL, unauthorized request for a Web application. This malicious request sends to a targeted Web application from user's browser. The request includes credentials associated to the particular website. An attacker has many ways to try and exploit the CSRF vulnerability.

2.3.5 Insecure Direct Object References

When an application supports direct access to objects based on user provided input, Insecure Direct Object References (IDOR) can be found. This vulnerability can result attackers to bypass authorization and access resources directly in the system. Resources are database records or files belonging to other users in the system. For example, the attacker retrieves information in the database such as personal information and other credentials data. The user is not used hashing or encryption algorithms to store this credentials data. So, the attackers will be easily accessed in the system.

2.3.6 Security Misconfiguration

Security misconfigurations mean security controls. These controls are not accurately configured or left insecure, putting systems and data at risk. For example, poorly presented configuration changes, default settings, or a technical issue in your system could lead to a misconfiguration. Security misconfiguration can be caused by the misconfiguration of application server, web server and database server. The most common security misconfigurations are missing headers, out-of-date software, operating systems, and other reasons. An attacker can exploit these vulnerabilities by the application errors, potentially gaining access via default credentials, or getting SQL injection or a web shell via outdated software.

2.3.7 Invalidated Redirects and Forwards

When the user visits a link located on a trusted website, this link is be redirected to an untrusted site for a user. This condition can be invalidated redirect vulnerabilities. This vulnerability is also known as Open Redirect. Web developers frequently use redirect and forward method for identified purpose with web application. This can harm the users by redirecting malware included sites or phishing, forwarding unauthorized parts of that site and reputation of web sites. The most common case of invalidated redirects and forwards can be lead to phishing attacks or others that involved in Social Engineering.

2.3.8 Insecure Cryptographic Storage

Insecure Cryptographic Storage vulnerability finds when sensitive data is not stored securely. In a Secure Software Development Lifecycle, protecting sensitive data by encrypting should include as a step. Insecure Cryptographic Storage is a collection of vulnerabilities. When it needs to be, the most important data is encrypted. In Insecurity Cryptographic Storage, an attacker can be processed to steal sensitive information like credit cards, passwords, authentication tokens, or login credentials by stealing cookies. If this sensitive information is not stored properly by using encryption algorithm or hashing, the attackers will be easily controlled and accessed this information.

2.3.9 Insufficient Transport Layer Protection

Insufficient Transport Layer Protection is a kind of security weakness. In Insufficient Transport Layer Protection, an insecure encryption layer is used to transmit data across a network without the advantage of cryptography. For example, if the user sends data to the server without using strong algorithms, valid or unexpired SSL or certificate, this can happen data compromising. So, many companies have introduced a higher layer of transport security. This security layer detects any possible regardless of encryption being used on TLS packets.

2.3.10 Failure to restrict URL Access

In Ethical Hacking, failure to restrict URL access means a kind of mistake in which a user can access data in a system but they have without permission to view, possibly resulting in data loss, fraud, or other violations of security policies. An application protects sensitive functionality to unauthorized users by preventing the display of links or URLs. By accessing those URLs directly, attackers can use this weakness to access and perform unauthorized operations. For example, when an error in access-control settings results in users are able to access pages that are restricted or hide, this condition occurs failure to restrict URL access.

All of the above vulnerabilities are the most common vulnerabilities in cyber-security field. Continuously, next part is studying types of attacks and their working in cyber-security.

2.4 Types of Attacks in Cybersecurity

Cybercriminals launch different cyber attacks for many purposes. These purposes are to steal sensitive data, such as credit card numbers or personal information. Using the victims' identities, the cybercriminals can be processed unauthorized access to victims' accounts. There are top 10 common types of cyber attacks in following:

- Denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks
- Man-in-the-middle (MitM) attack
- Drive-by attack
- Password attack
- SQL injection attack
- Cross-site scripting (XSS) attack
- Phishing and spear phishing attacks
- Birthday attack
- Eavesdropping attack
- Malware attack

2.4.1 Denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks

A denial-of-service attack is an attack that is started from a large number of other host machines on system's resources. This DoS attack is infected by malicious software controlled by the attacker. A DDoS attack can be disrupted the normal traffic of a targeted server, service, network or its surrounding infrastructure with a flood of Internet traffic. In DDoS attacks, this can be achieved by utilizing multiple compromised computer systems that are sources of attack traffic. The common types of DoS and DDoS attacks are TCP SYN flood attack, smurf attack, teardrop attack, ping-of-death attack and botnets.

2.4.2 Man-in-the-middle (MitM) attack

A man-in-the-middle (MITM) is a type of cyber-attack. In MITM, attackers can be intercepted an existing conversation or data transfer by eavesdropping or pretending to be a legitimate participant. The purpose of a MITM attack is to get confidential data such as credit card numbers, bank account details or login credentials. These data can be used to process other crimes like identity theft or illegal fund transfers. This attack target e-commerce sites, SaaS businesses, the users of financial applications and other websites where logging in is required.

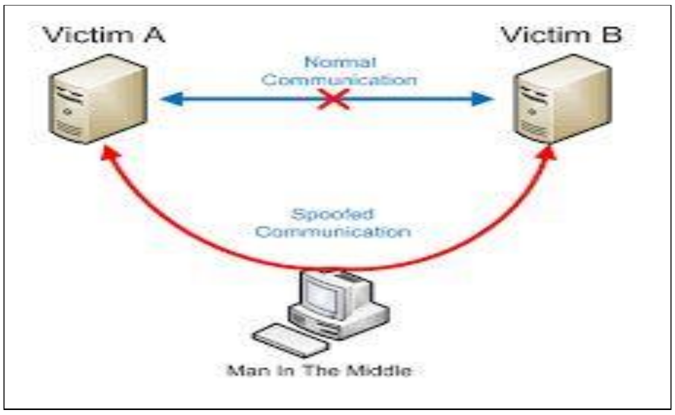


Figure 2.1 Man-in-the-middle (MitM) attack

2.4.3 Drive-by attack

A drive-by attack is a cyber-attack and also known as a drive-by download attack. In this attack, a malicious script causes a program to download and install itself on a user device, without permission from the user. Drive-by attacks use malicious parts of software created by hackers to exploit and launch the automatic download. Drive-by attacks can be processed to steal information, infect devices and cause damage to data.

2.4.4 Password attack

Password attack is a type of cyber-attack and hackers try to access a file, account, folder or computer secured with a password. This attack can be processed successful attack with the help of software by cracking or guessing passwords. Password attacks including broken authorization vulnerability combined with automatic password attack tools in the system. To avoid this attack, people should not be used easy passwords such as your pet’s name, nicknames, address, birthday, NRC card number etc.

2.4.5 SQL injection attack

SQL injection means a type of an injection attack that executes malicious SQL statements. A database server can be controlled these statements behind a web application. To bypass application security measures, attackers use SQL Injection vulnerabilities. They try to retrieve the contents of the entire SQL database, exploit authentication and authorization of a web page or web application, add, modify and delete records in the database. An SQL Injection can be affected website or web application used an SQL database such as MySQL, Oracle, SQL Server, or others.

2.4.6 Cross-site scripting (XSS) attack

Cross-site scripting (XSS) allows an attacker to compromise the interactions that user has with a vulnerable application. A victim user can be masqueraded by an attacker to process any actions that user is able to perform and access the user's data. If the targeted user has privileged access in the application, the attacker can get to control fully over all of the application's functionality and data.

2.4.7 Phishing and spear phishing attacks

When phishing attacks target people who might click, spear phishing attacks try to fool people who working at businesses to access the target**.** Phishing and spear phishing are designed to trick user by performing a specific action such as clicking on malicious link or attachment. By clicking the link and malware, it might be downloaded on user device and go to a fake website. Then, this website can be asked to enter user name, address, and social-security number. This information can be used for fraud or identity theft. In a spear phishing attack, attackers can use a mixed email spoofing, drive-by downloads and dynamic URLs to bypass security controls. In advanced spear phishing attacks, this attack can be exploited zero-day vulnerabilities in browsers, applications or plug-ins.

2.4.8 Birthday attack

A birthday attack includes as a type of brute force attack. A birthday attack is a cryptographic attack form and it cracks mathematical algorithms by looking for matches in the hash function. The attack is used to manipulate communication between the two parties and determine by attacking between random attacks and the degree of permutation. In birthday problem, this considers the probability in a set of *n* randomly chosen of some paired people who will have the same birthday.

2.4.9 Eavesdropping attack

When an attacker intercepts, deletes, or modifies data that is transmitted between two devices, an eavesdropping attack occurs. This attack is known as sniffing or snooping and this based on unsecured network communications to access data in movement between devices. In eavesdropping attack, a user connects to a network in which traffic is not secured or encrypted and sends sensitive business data to other user. While the data is transmitted across an open network, an attacker tries to get the chance by exploiting vulnerability and intercept it via various methods.

2.4.10 Malware attack

Malware attack is a type of malicious software. This attack is processed to cause harm or damage to a computer, server, client or computer network and infrastructure. Attackers create malware for many different reasons. These reasons are used to steal personal, financial or other business information. Attackers focus techniques and procedures on getting access to privileged credentials and accounts to do their mission. Most malware types can be classified into many categories. They are virus, worm, trojan, hybrid malware, adware, malvertising, spyware and ransomware.

2.5 Types of Man-In-The-Middle (MITM) Attack

Cybercriminals use the many ways to launch MiTM attacks. There are many types of Man-In-The-Middle(MITM) attack. They are:

- Internet Protocol spoofing
- Domain Name System spoofing
- HTTPS spoofing
- Secure Sockets Layer hijacking
- Email hijacking
- Wi-Fi eavesdropping
- Session hijacking
- Cache poisoning

2.5.1 Internet Protocol spoofing

When data is transferred on the internet, internet protocol (IP) is used. The packet headers are included the identity of the sender and the receiver of IP addresses. When an attacker alters the sender's IP address on the data header, it looks like coming from the spoofed IP address. Internet protocol is a protocol and data from the previous sessions are not retained. An IP spoofing attack is used to create a backdoor in the target's IT systems by getting root access to the host. This attacks target on the trust between two devices and it can be used as part of a scheme to start a cyber-attack.

2.5.2 Domain Name System spoofing

Domain Name System spoofing is a kind of man-in-the-middle attack that tricks a user to a fake website rather than the real one the user wants to visit. Then, cybercriminals change domain names to redirect traffic to fake websites. Users may think this website as a secure and trusted website when they are actually interacting with an attacker. The attacker's goal is to redirect traffic to a fake website or to capture user login credentials.

2.5.3 HTTPS spoofing

HTTPS Spoofing is a type of man in the middle attack. This HTTP extension means Hyper Text Transfer Protocol and it is widely used on the Internet. In HTTPS, S means secure and this protocol encrypts the data exchange between the client and the website that user interacting with that website. In HTTPS spoofing, an attacker creates a fake HTTPS website by spoofing the address of a legitimate website. Then, attacker sends a link to users who visiting this fake website. When user opens this link, this will be launched attack. Cybercriminals can steal shared personal information and monitor user interactions through the redirection.

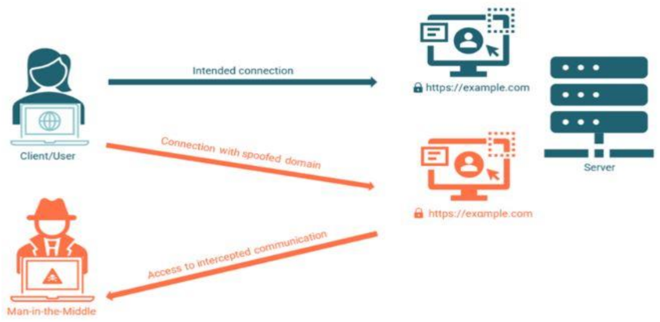


Figure 2.2 Working of HTTPS Spoofing attack

2.5.4 Secure Sockets Layer hijacking

Secure Sockets Layer (SSL) hijacking attack is a type of man in the middle attacks and criminal hijacks a user’s legitimate session by masquerading as that user. SSL is a protocol and it can be established an encrypted connection between browser and web server. In SSL hijacking, a cybercriminal can use other computer and a secure server to intercept by traveling all information between the server and the user's computer. SSL hijacking attack is known as session hijacking attack and this includes stealing session ID or session key to get unauthorized control on the victim’s session.

2.5.5 Email hijacking

Email hijacking includes as a kind of man-in-the-middle attack. E-mail security is a part of data security that contains securing the privacy and accessibility of mail frameworks and the information. In email hijacking attack, attackers try to compromise and gain access to a user’s email account and watch communications about that account. Then, the attacker is silently monitors this communication between the client and the provider and uses for malicious purposes. This attack targets on mail rather than websites.

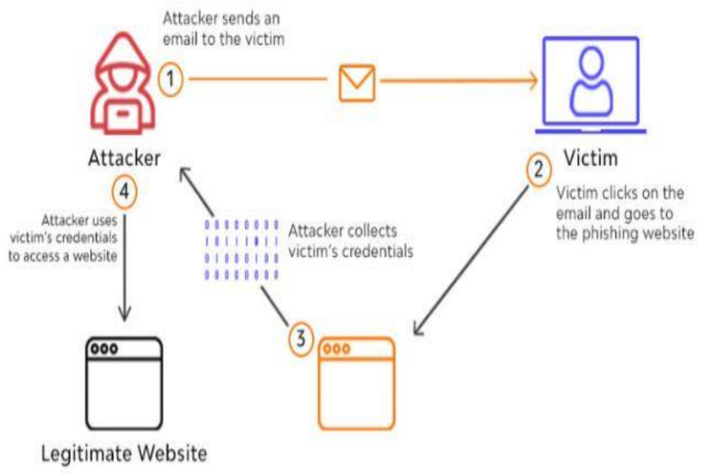

Attacker sends an email to the victim
1
Attacker
Victim
2
Victim clicks on the email and goes to the phishing website
4
Attacker uses victim's credentials to access a website
Attacker collects victim's credentials
3
Legitimate Website

Figure 2.3 Working of Email Hijacking attack

2.5.6 Wi-Fi eavesdropping

Wi-Fi eavesdropping is another type of man-in-the-middle attack and a hacker can steal data on a public, unsecured wifi network. A cybercriminal can start wi-fi eavesdropping attack when the victim connects via Wi-Fi to a network. For example, an attacker can create a Wi-Fi with the name of a company, a store, a restaurant or a place that are legitimate. Then, attacker will try to believe that the victims are processing with a legitimate connection. Finally, the attacker can take advantage from this bogus network and steal the credentials.

2.5.7 Session hijacking

Session hijacking means stealing browser cookies and cybercriminals steal personal information and passwords stored inside the cookies of a user's browsing session. In session hijacking attack, hackers try to gain access to a target's computer and take control of a user's browsing session. Session hijacking can be processed by using malware to infect the user's computer and direct access to the machine by hijacking any active sessions.

2.5.8 Cache poisoning

Cache poisoning is known as address resolution protocol or ARP cache poisoning. This is a kind of man-in-the-middle attack. ARP poisoning is stored a fake address in the ARP cache. In ARP cache poisoning attack, attackers try to inject wrong information into local area network traffic by redirecting connections to their device. Attackers can use ARP poisoning attacks for many purposes such as denial of service attacks (DOS) and distributed denial of service attacks (DDOS).

2.6 String or Pattern Matching Algorithm

Pattern matching is a part of data processing and basic problem of computer science. String (or text) matching is a special process of pattern matching. String matching algorithms can try to find places where one or many strings that are called patterns found within a larger string. In this case, this pattern is indicated by a finite sequence of symbols (or alphabet) Σ. Σ can be an alphabet such as the letters a through z or Greek letters. It includes the finding one or all occurrences of a pattern P with length m in a pattern database T that including on n patterns where m and n > 0. Today, pattern or string matching algorithms are commonly used in web application vulnerability detection, plagiarism detection, bioinformatics processing, natural language and image-processing, intrusion- detection, digital forensics and others.

There are many types of string or pattern matching algorithms. The most common algorithms are:

- Naive Pattern Matching Algorithm
- Rabin Karp Pattern Matching Algorithm
- Knuth-Mooris-Pratt (KMP) Algorithm
- Boyer-Moore String Matching Algorithm

All of the above algorithms have owned advantages, disadvantages and their working flow.

2.6.1 Naive Pattern Matching Algorithm

Naive algorithm is the simple algorithm. This algorithm easy and understand to implement than other pattern matching algorithms. In finding patterns, it can compare text and pattern from left to right and compare character by character until a match is found. Naive algorithm shifts the validation by using a loop that tests a situation P [1...m] = T [s+1 ...s+m] for each of the "n – m + 1" possible values of s. T means text and the symbol P means pattern. This algorithm is commonly used in string matching, matrix multiplication and pattern searching.

The loop testing decides whether the current shift of pattern with the text is valid or invalid. This loop includes implicit loop for testing the corresponded character positions until all of positions mismatch is found or all positions match successfully. If a match is found, it returns the start index of the pattern finding and shifts once again to test the subsequent matches of pattern. The running time of Naive algorithm takes O(mn) in worst case and searching the normal text takes O(m+n) where m means length of the pattern and n is the text of length.

Naive algorithm has advantages and disadvantages. The advantages are

- Pre-processing phase is not required because the running time of Naïve String Matcher is equal to its matching time.
- Extra spaces are not needed.
- The comparisons can be processed in any order.

The disadvantages are

- Naive method is inefficient method because of a shift of information is not using again.
- Naive algorithm shifts pattern string in one position to the right after each comparison.
- This algorithm is efficient brute force approach. If a matching case, pointers in text and pattern strings are advanced. If not a matching case, the pointer to text is incremented and pointer of the pattern is reset. This process is repeated until the end of the text.

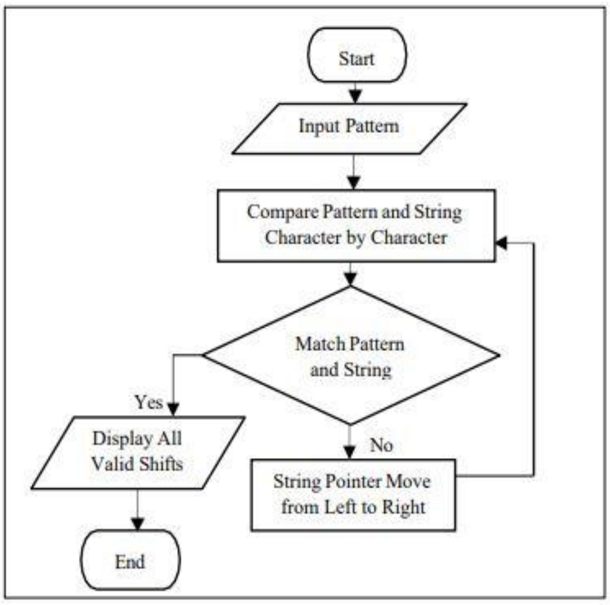

Start
Input Pattern
Compare Pattern and String Character by Character
Match Pattern and String
Yes
No
Display All Valid Shifts
String Pointer Move from Left to Right
End

Figure 2.4 Flowchart of Naive pattern matching algorithm

2.6.2 Rabin Karp Pattern Matching Algorithm

Rabin Karp algorithm is an algorithm that is used for patterns searching/matching process in the text using a hash function. In this algorithm, the hash value is calculated in the pre-process state and compared this value rather than the characters of the string. Rabin Karp algorithm does not travel every character in the initial phase and it filters that characters that do not match and performs the comparison. In Rabin Karp algorithm, the worst-case complexity is O(mn) in the spurious hits and average complexity is O(m+n). A spurious hit means the hash value of the pattern is matching the hash value of text that this text actually does not exist.

The advantage of Rabin Karp algorithm is

- When the hash values match, this results only one comparison per text sub-sequence and brute force is only needed.

The drawback of Rabin Karp algorithm is

- Hash function based on the created table and sometimes even hash code is matched of both pattern and Text but the characters of the Pattern do not match the Text.

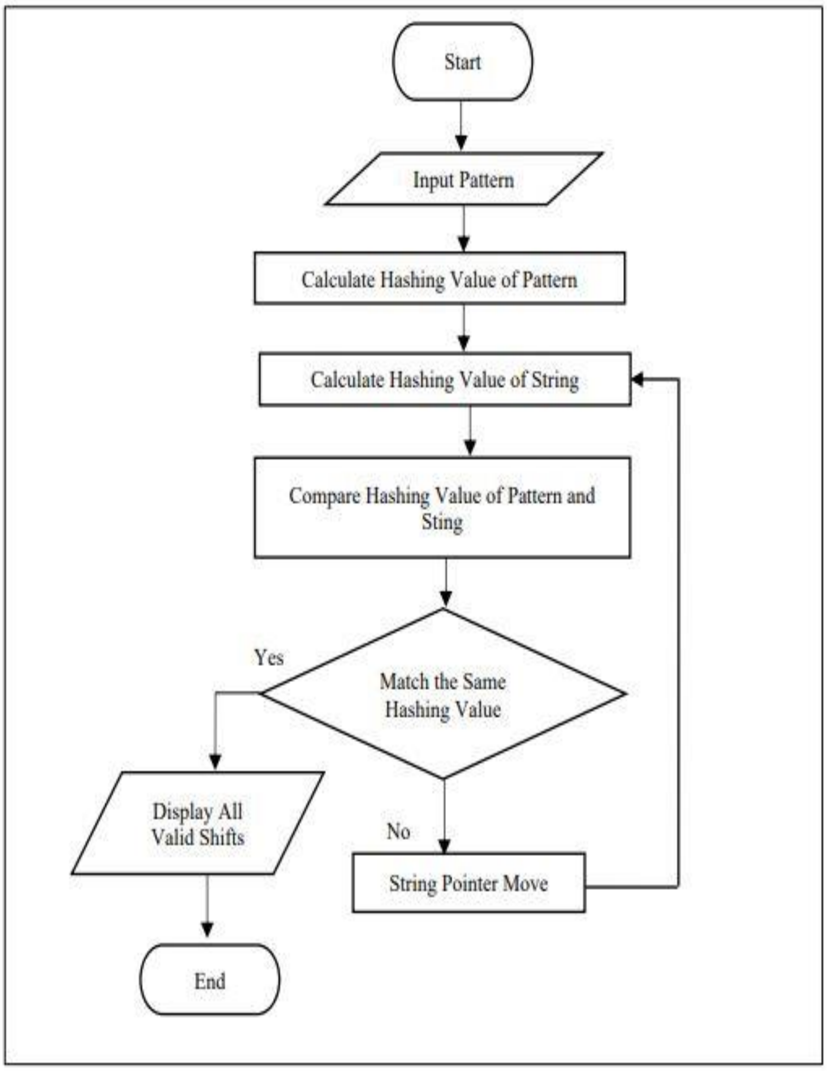

Start
Input Pattern
Calculate Hashing Value of Pattern
Calculate Hashing Value of String
Compare Hashing Value of Pattern and Sting
Match the Same Hashing Value
Yes
No
Display All Valid Shifts
String Pointer Move
End

Figure 2.5 Flowchart of Rabin Karp pattern matching algorithm

2.6.3 Knuth-Mooris-Pratt (KMP) Algorithm

The Knuth-Mooris Pratt Algorithm is an efficient string matching algorithm and similar to Naive algorithm but KMP algorithm requires preprocessing phase. In this preprocess phase, it calculates how many characters to be skipped. This algorithm is a linear time algorithm and exploits the observation in every time a match or mismatch happening. In this process, the pattern itself contains enough information to command where the new examination should begin from. In comparison process, if the pattern does not match the current index of string, then it works the preprocess phase and shifts the index of text until match again between the characters of the text and pattern. In KMP algorithm, the time complexity of this matching is O(n) and the space complexity is O(m).

The advantages of the KMP algorithm are

- It's time complexity is fast because it's very fast as compared to other exact string matching algorithm.
- No worse cases or accidental inputs exist.

The disadvantage of the KMP algorithm is

- it is very complex to understand.

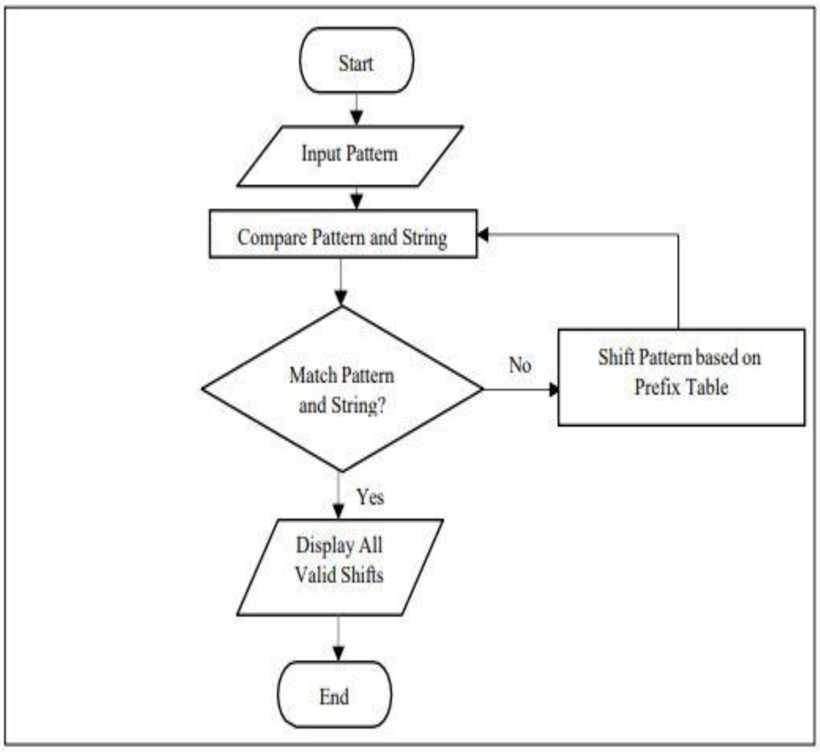

Start
Input Pattern
Compare Pattern and String
Match Pattern and String?
No
Shift Pattern based on Prefix Table
Yes
Display All Valid Shifts
End

Figure 2.6 Flowchart of Knuth-Mooris Pratt pattern matching algorithm

2.6.4 Boyer-Moore String Matching Algorithm

The Boyer-Moore algorithm is the most efficient string matching algorithm. During the preprocessing state, Boyer-Moore algorithm gathers information to skip the length of shifts. Boyer Moore algorithm is known as index search algorithm such as Knuth Morris Pratt (KMP) algorithm. Unlike other pattern matching algorithm, Boyer Moore algorithm scans the characters of the pattern from right to left beginning with the rightmost character. In the mismatch or match cases, it uses two pre-computed functions. These two shift functions are the Good Suffix Heuristics (matching shift) and the Bad Character Heuristics (occurrence shift).

The Bad Character Heuristics are the followings. In searching case, the first character that does not match the pattern is called the Bad Character.

Case I: Shift the pattern until the mismatch becomes a match

In this case, it starts checking the pattern occurrence from the rightmost character and shifts the pattern in a way that it gets aligned with the mismatched character in the text.

Case II: Shift the pattern until it moves past the bad character

In this case, it begins matching the pattern from the right end if it matches one by one with all the characters of the given string. If the rightmost character is not matching with the given string of the rightmost character, it shifts the whole pattern ahead of the last character of the string.

The Good Suffix Heuristics are the followings.

Case I: Shift the pattern until the next occurrence is found

In this case, when pattern is not matching with the given string, it shifts the pattern to the right for checking whether the pattern is present in the further characters of the string.

Case II: Shift pattern until prefix of pattern matches with a suffix of text

In this case, it tries to match the prefix of the pattern with a possible suffix of the given text by shifting the pattern that are many times as the length of prefix.

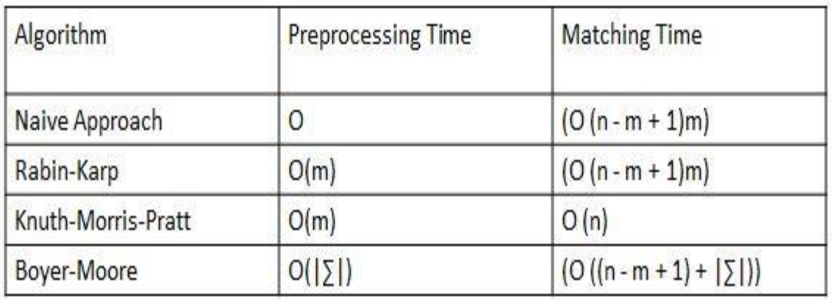
The worst case complexity of Boyer Moore algorithm is O(m+n) if pattern does not appear in the text or O(mn) if pattern appears in the text. In Boyer Moore Algorithm, the time taken during pattern matching in the worst case is O(m*n). m means the length of the pattern and n means the length of the string provided.

The advantages of the Boyer Moore algorithm are

- In most searching, it takes O(m+n). So it is faster.
- It is Quick searching algorithm.

The disadvantages of the Boyer-Moore algorithm are

- Runs in time O(mn) in the worst case.
- It is only suitable for smaller strings



Algorithm	Preprocessing Time	Matching Time
Naive Approach	O	(O (n - m + 1)m)
Rabin-Karp	O(m)	(O (n - m + 1)m)
Knuth-Morris-Pratt	O(m)	O (n)
Boyer-Moore	O(|∑|)	(O ((n - m + 1) + |∑|))

Figure 2.7 Comparison of the Boyer-Moore Algorithm with other methods

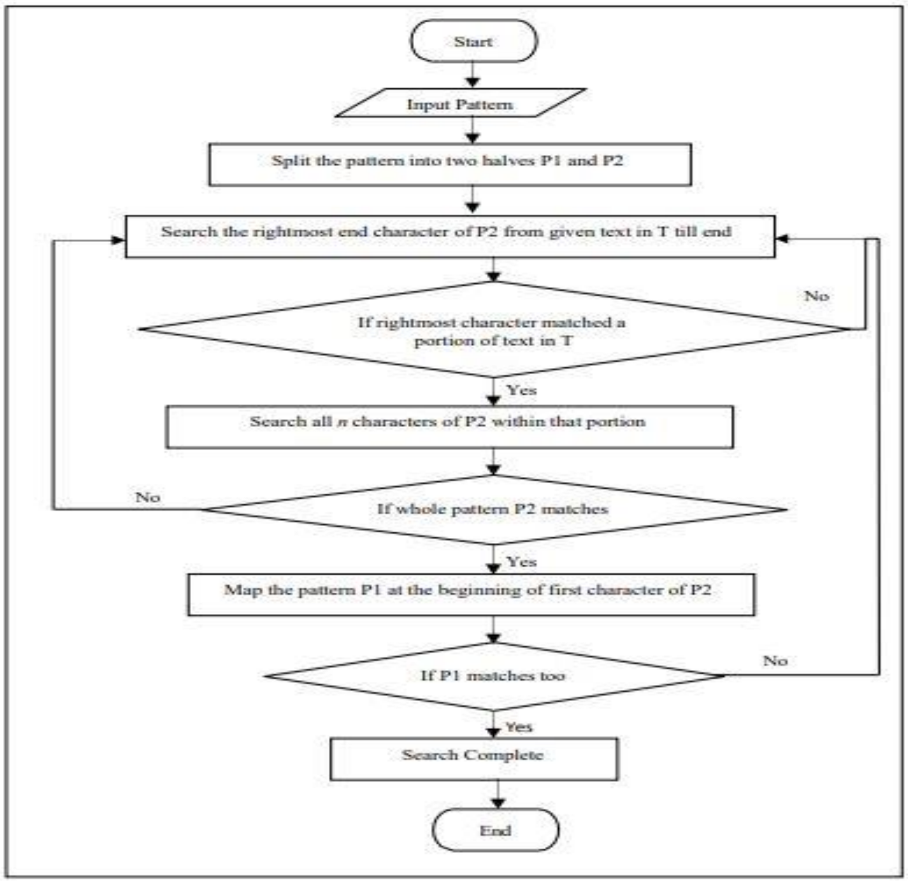

Start
Input Pattern
Split the pattern into two halves P1 and P2
Search the rightmost end character of P2 from given text in T till end
If rightmost character matched a portion of text in T
No
Yes
Search all n characters of P2 within that portion
If whole pattern P2 matches
No
Yes
Map the pattern P1 at the beginning of first character of P2
If P1 matches too
No
Yes
Search Complete
End

Figure 2.8 Flowchart of Boyer-Moore pattern matching algorithm

CHAPTER 3
DESIGN OF THE PROPOSED SYSTEM

This thesis aims to detect the web application vulnerability with the vulnerability searching method. In this thesis, this system focuses on the detection of HTTPS Spoofing and Email Hijacking vulnerabilities from web application using Boyer Moore string matching algorithm. This system can be processed to search attack signatures in URL and Email links. Firstly, the overview of the proposed system of system architectures for HTTPS Spoofing and Email Hijacking are described. This chapter mainly implements on the designs of the system.

3.1 Overview of the Proposed System

In this part, the system shows the detection process of HTTPS Spoofing and Email Hijacking attack.

3.1.1 Overview of the Proposed System for HTTPS Spoofing

In this proposed system, the detection process of vulnerabilities for HTTPS Spoofing attack is described the following. This includes the detailed explanation for working processes. The proposed architecture for this system is described in Figure 3.1.

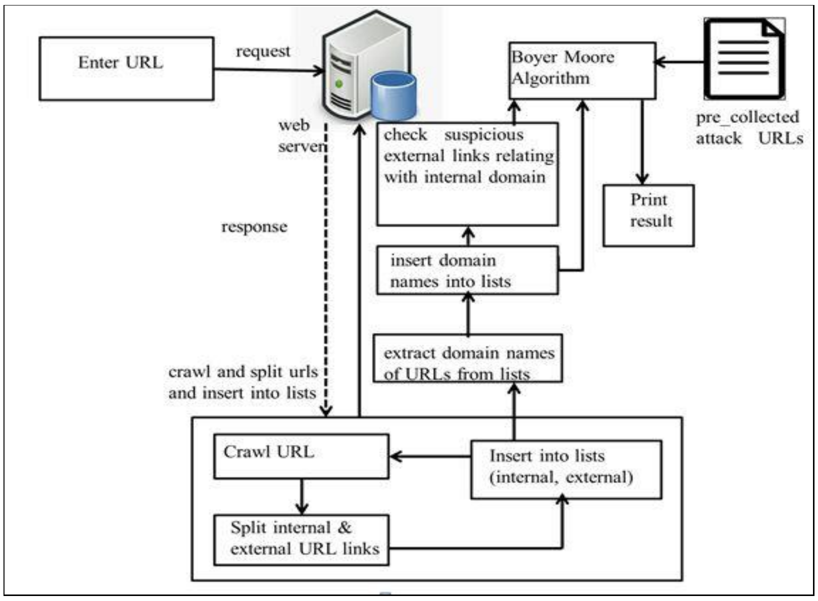

Enter URL
request
web server
response
crawl and split urls and insert into lists
Boyer Moore Algorithm
pre_collected attack URLs
check suspicious external links relating with internal domain
Print result
insert domain names into lists
extract domain names of URLs from lists
Crawl URL
Insert into lists (internal, external)
Split internal & external URL links

Figure3.1 Vulnerability Detection Architecture of HTTPS Spoofing Attack

The figure 3.1 shows the system architecture of vulnerability detection for HTTPS Spoofing attack. These vulnerabilities detect to look for attack signatures in input URL of a website. For detecting vulnerability, the proposed system uses a Boyer Moore string matching algorithm and analyzes the behavior of the web application response and delay time. Boyer Moore string matching algorithm helps to search attack signatures in the response by using pre-collected attacked URLs. The proposed system which discovers HTTPS Spoofing vulnerabilities and then the detailed explanations is described with the following steps.

In Step 1: In this step, enter URL in web application and request to the server. Firstly, to start the web application, users enter respective URLs to detect HTTPS Spoofing vulnerabilities.

In Step 2: In this step, the server responses the inserting website of URL. Then, the system is crawling into this URL with the corresponded each web page, splitting into internal and external URL links and inserting into respective lists. Then, domain names of these URLs are extracted from respective lists and this domain names inserted into lists.

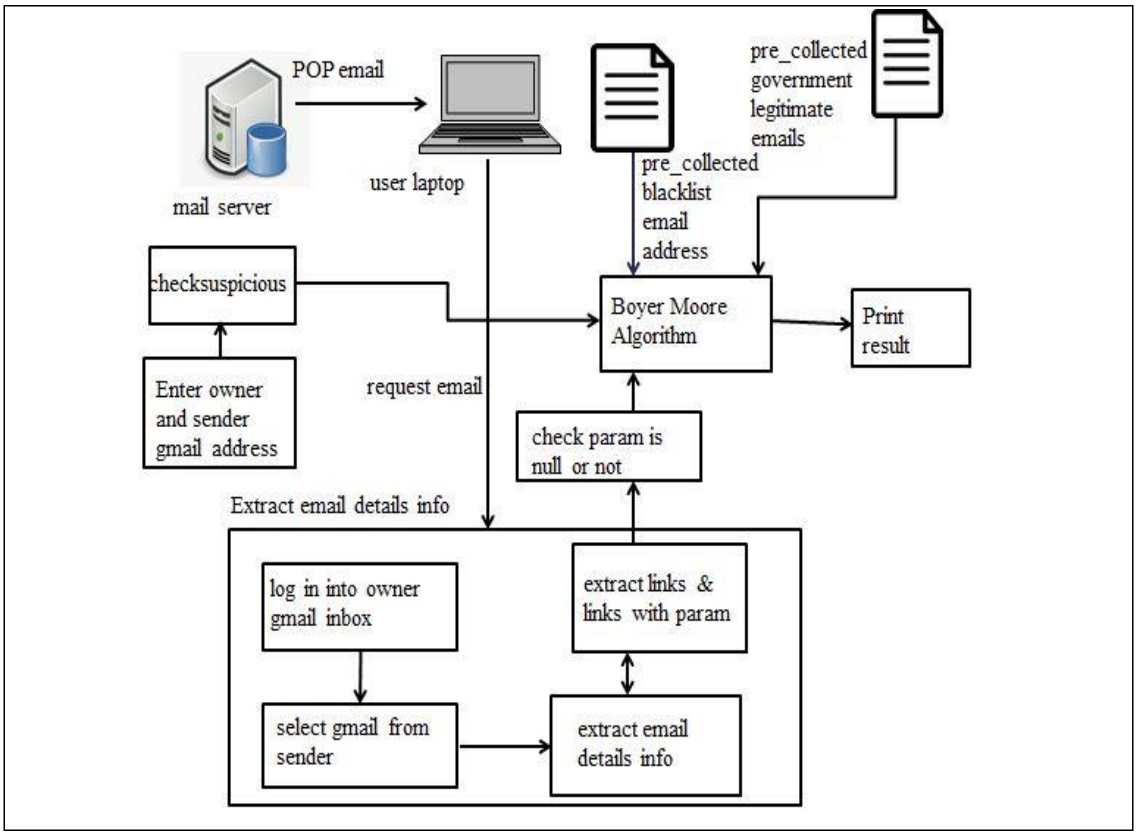
In Step 3: In this step, domain names included lists are getting. The system is included pre-collected attack URLs that existing HTTPS Spoofing attack. Then, the system is checked with pre-collected attack URLs to search vulnerabilities by using Boyer Moore algorithm.

In Step 4: And then, after processing the above steps, the system is decided existing vulnerability or not. In this step, if not exit vulnerability, external URL links are checked again according to the relating with internal domain by using algorithm. This means the system is checking external domains including fake domains or not.

Finally, the system can decide the website that has included attacked links or not. HTTPS Spoofing vulnerability can check using this proposed system. Checking this vulnerability is important thing because this can cause unsecure information security and lose sensitive information and money. Attackers may create the attacked links within website. When users visit this website, users may reach the redirected website that created by attackers. At last, attackers steal accessed users information to this website such as bank's credit card number, email password and other important information. By using this proposed system, the advantages are_ users protecting their important information, knowing which website is secure and avoiding the websites included HTTPS Spoofing vulnerability.

3.1.2 Overview of the Proposed System for Email Hijacking

In this proposed system, the detection process of vulnerabilities for Email Hijacking attack is described the following. This includes the detailed explanation for working processes. The proposed architecture for this system is described in Figure 3.2.



POP email
mail server
user laptop
pre_collected government legitimate emails
pre_collected blacklist email address
checksuspicious
Enter owner and sender gmail address
request email
Boyer Moore Algorithm
Print result
check param is null or not
Extract email details info
log in into owner gmail inbox
extract links & links with param
select gmail from sender
extract email details info

Figure3. 2 Vulnerability Detection Architecture of Email Hijacking Attack

The figure 3.2 describes the system architecture of vulnerability detection for Email Hijacking attack. These vulnerabilities detect to look for attack signatures in email for gmail users. For detecting the vulnerability, the proposed system uses a Boyer Moore string matching algorithm and analyzes the processing of the web application response and delay time. Boyer Moore string matching algorithm helps to find attack signatures in the response by using pre-collected blacklists of email. The proposed system which discovers Email Hijacking vulnerabilities and then the detailed explanations is described with the following steps.

In Step 1: In this step, enter owner and sender's gmail address in the system. Firstly, the system checks sender's mail address with pre-collected blacklist email addresses using Boyer Moore Algorithm.

In Step 2: In this step, read the specified email in user laptop that is downloaded from Mail Server using post office protocol (POP). Then, system can login into owner's gmail inbox and select sender's mail using sender's gmail address. Continuously, the system processes by extracting email detailed information and links within email.

In Step 3: In this step, all links included in sender email are getting. The system is included pre-collected attack links that existing Email Hijacking attack. This extracted email links makes checking with pre-collected attack links to find vulnerability by using Boyer Moore algorithm.

In Step 4: And then, after processing the above steps, the extracted links test again including parameter or not parameter. If these URL links include parameter, system next detects sender's email address with pre-collected legitimate email lists. The system has pre-collected legitimate email lists that come from officially trusted website. When sender's email includes in this legitimate email lists, this email cannot assume the vulnerability that existing Email Hijacking attack. If not, this email can assume the vulnerability that existing attack.

Finally, the system can decide the incoming email that has included attacked links or not. Email Hijacking vulnerability can check using this proposed system. Checking this vulnerability is important thing because this can cause stealing email account passwords, not securing information security and losing sensitive information and money. By stealing email accounts, attackers can create many malicious purposes using this email account. Sample malicious purposes are phishing money and credit card numbers and other credential information by masquerading legitimate users. Attackers can create the attacked links within email. When users visit this email and click the attacked links, users can reach the redirected website or page that created by attackers. At last, attackers can steal accessed user’s information to this website such as bank's credit card number, email password and other important information. By using this proposed system, the advantages are_ users protecting their important information, knowing which email is secure and avoiding the email links included Email Hijacking vulnerability.

3.2 Flowcharts of the Proposed System for HTTPS Spoofing and Email Hijacking

In this part, the detection process of vulnerabilities for HTTPS Spoofing and Email Hijacking attacks are described the following. This includes flowcharts that are the detailed explanation to detect these two vulnerabilities for working processes. The expected architecture for this system is shown in figure 3.3, figure 3.4 and figure 3.5.

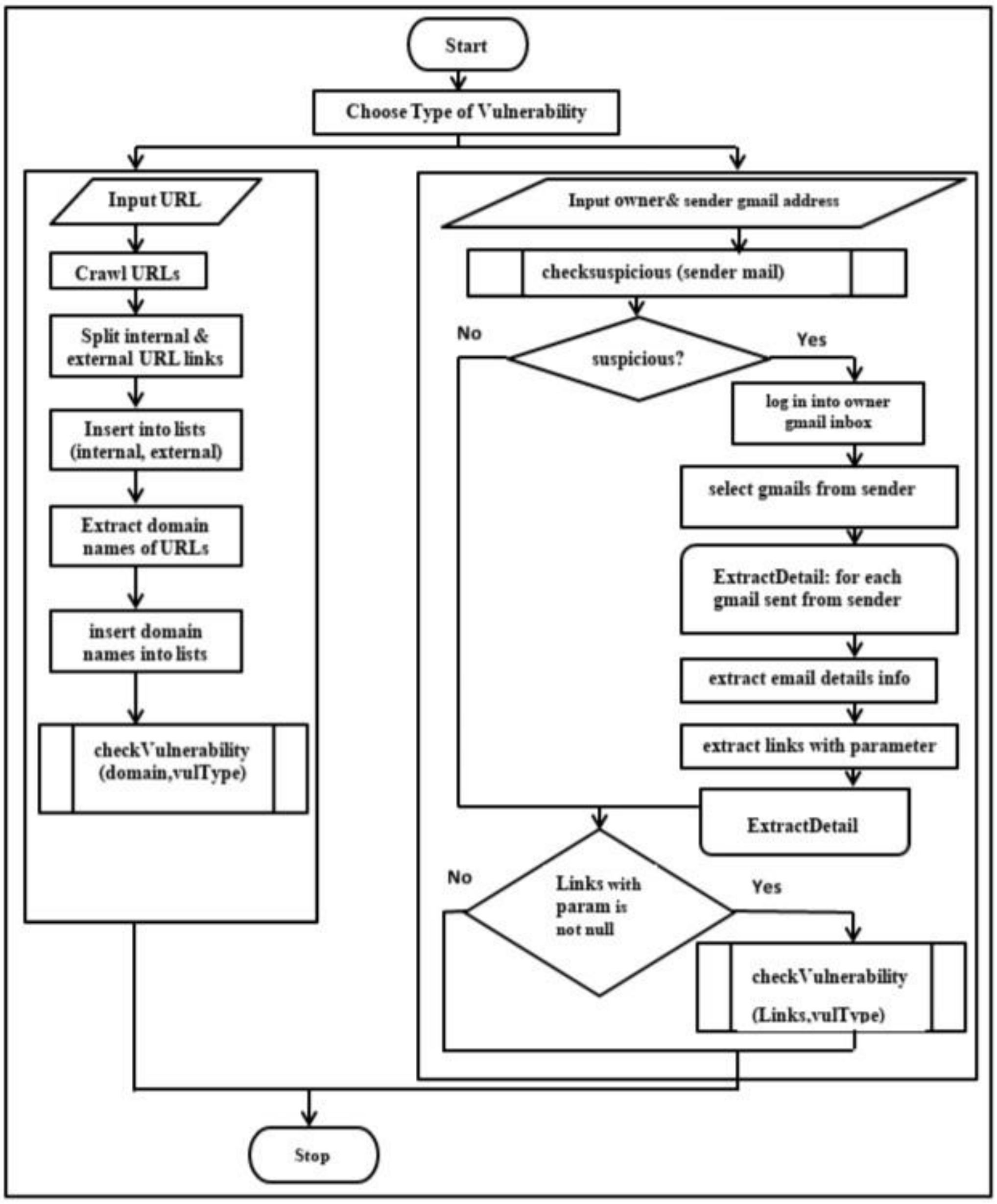

Start
Choose Type of Vulnerability
Input URL
Crawl URLs
Split internal & external URL links
Insert into lists (internal, external)
Extract domain names of URLs
insert domain names into lists
checkVulnerability (domain,vulType)
Input owner& sender gmail address
checksuspicious (sender mail)
suspicious?
No
Yes
log in into owner gmail inbox
select gmails from sender
ExtractDetail: for each gmail sent from sender
extract email details info
extract links with parameter
ExtractDetail
Links with param is not null
No
Yes
checkVulnerability (Links,vulType)
Stop

Figure3.3 Flowchart of the proposed system

In Figure 3.3, the flowchart shows the detailed steps of the proposed system for vulnerability detection of HTTPS Spoofing and Email Hijacking attacks. Firstly, this system must choose types of vulnerability. These types are HTTPS Spoofing or

Email Hijacking. If users choose HTTPS Spoofing type, users enter URL as input. Then, crawling this URL for the whole website, splitting into internal & external URL links, and inserting into corresponding lists. After that, the system extracts domain names of URLs from list and insert into corresponding lists. Finally, the system checks domain and vulnerability type. The next steps for this system express in Figure 3.4.

If users choose Email Hijacking type, users enter owner and sender gmail address as inputs. Then, system checks sender mail that mail is suspicious or not. If mail is suspicious, firstly, system processes log in into owner Gmail inbox and selects gmails from sender. The system processes each gmail sent from sender such as extracting email detailed information and links with parameter by performing looping process. Extracted links check parameters that existing or not. . If mail is not suspicious, the system checks links and vulnerability type. If links included parameters exist, the system checks links and vulnerability type. If links included parameters not exist, this has not Email Hijacking vulnerability. The next steps for this system express in figure 3.4.

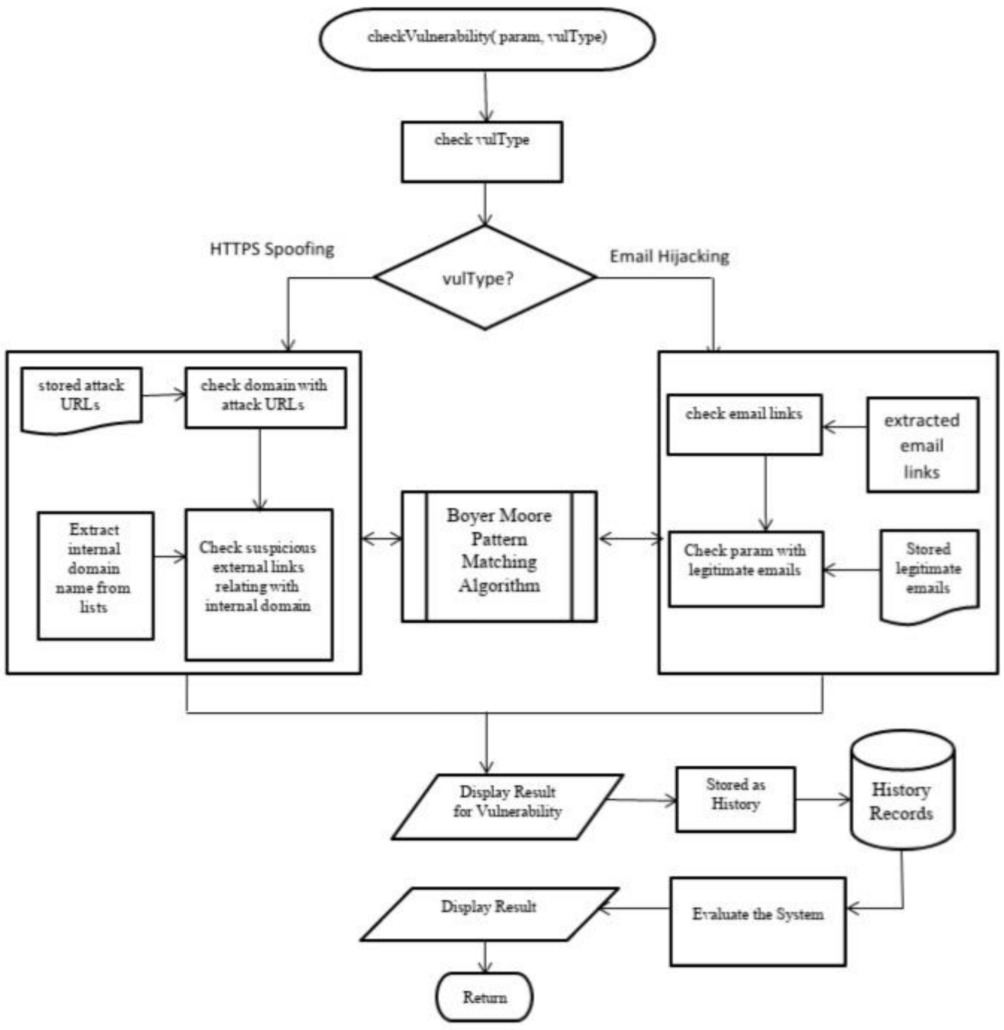

checkVulnerability(param, vulType)
check vulType
vulType?
HTTPS Spoofing
Email Hijacking
stored attack URLs
check domain with attack URLs
Extract internal domain name from lists
Check suspicious external links relating with internal domain
Boyer Moore Pattern Matching Algorithm
check email links
extracted email links
Check param with legitimate emails
Stored legitimate emails
Display Result for Vulnerability
Stored as History
History Records
Display Result
Evaluate the System
Return

Figure3.4 Flowchart of the proposed system

In Figure 3.4, the system checks parameters and vulnerability types which parameters are domain or links. Then, system checks vulnerability types which types are HTTPS Spoofing or Email Hijacking. If vulnerability type is HTTPS Spoofing, the system included pre-collected attack URLs. This attack URLs check domains of parameters by using Boyer Moore pattern matching algorithm. The system includes a list of internal domain names. The system next checks suspicious external links relating with internal domain by using Boyer Moore algorithm.

If vulnerability type is Email Hijacking, the system included pre-collected attack links. Then, the system checks email links and this attack links by using Boyer Moore pattern matching algorithm. Next, this system included pre-collected legitimate emails. The system next checks parameter links of email and legitimate emails by using Boyer Moore pattern matching algorithm.

Finally, the system displays result for vulnerability, stores as history in history records of database, evaluates the system and displays final result.

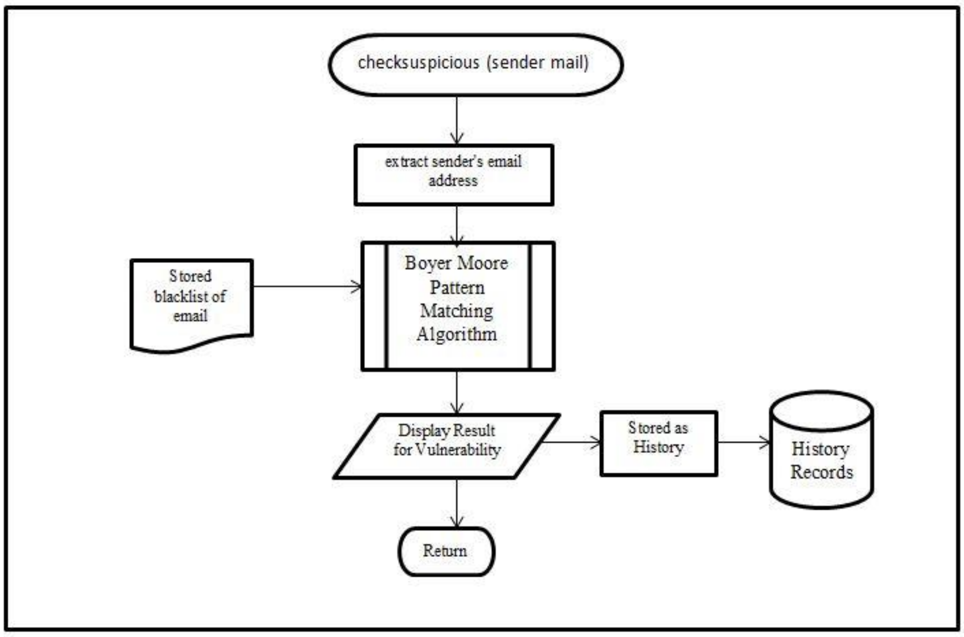

checksuspicious (sender mail)
extract sender's email address
Stored blacklist of email
Boyer Moore Pattern Matching Algorithm
Display Result for Vulnerability
Stored as History
History Records
Return

Figure3.5 Flowchart of the Proposed System

In Figure 3.5, the system checks sender mail whether mail is suspicious or not. Firstly, the system extracts sender's email address. The system included pre-collected blacklists of email addresses. Then, the system checks this blacklist emails and sender's email address by using Boyer Moore algorithm. Finally, the system displays result for vulnerability and stores as history in history records of database.

The above explanations are the detailed steps of 'Vulnerability Detection for HTTPS Spoofing and Email Hijacking Attacks of Web Application using Boyer Moore String Matching Algorithm'.

3.3 Performance Evaluation

The proposed system was evaluated in results of true positive and true negative, false positive and false negative rates. The accuracy of the detection method describes the results by comparing with attacked datasets. The accuracy was depended on the correctness of vulnerability detection or it describes that the tested results are nearly similar to the correct value. It is also based on the situation of false positive and negative rates. To evaluate the performance of the proposed system, the contingency table is used as described in Table 3.1.

Table3.1 Specification of true positive, true negative, false positive and false negative rates

	Normal	Attack
Normal	True Negative (TN)	False Positive (FP)
Attack	False Negative (FN)	True Positive (TP)

True positive, true negative rate and false positive, false negative rate are expressed confusion matrix equations.

False Positive Rate (FPR)	= FP/(FP+TP)
False Negative Rate (FNR)	= FN/(TP+FN)
True Positive Rate (TPR)	= TP/(TP+FN)
True Negative Rate (TNR)	= TN/(TN+FP)

The descriptions of confusion matrix equations are as follows.

True Positive (TP): The system can actually detect that there is no vulnerability when the website or email is not vulnerable.

True Negative (TN): The system can accurately detect the vulnerability when the website or email is actually vulnerable.

False Positive (FP): The system can detect that there has no vulnerability of while the website or email has vulnerability.

False Negative (FN): The system can detect that there has vulnerability but actually, there is no vulnerability in the website or email.

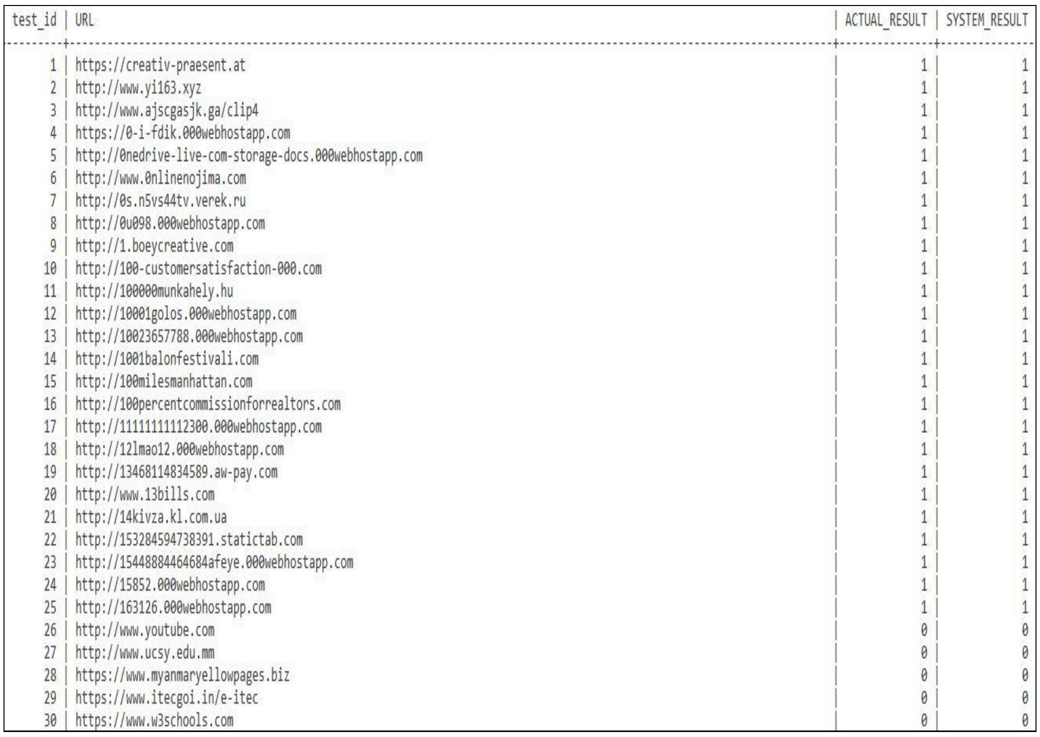
The following equation was calculated the accuracy of the proposed system.

Accuracy = ((TPR+TNR) / (TPR+FPR+TNR+FNR)) *100

Equation 3. 1

3.4 URLs and Emails Used in the Experiment

To evaluate the performances of the system, the 113 URLs and Emails are used as case study. Firstly, these links and Emails were collected and examined by their choosing type of vulnerabilities using Boyer Moore String Matching Algorithm. In this checking, the system has used datasets as attacked URLs and Email datasets to decide including vulnerability or not. This sample URLs and Email links can be described in Figure 3.6.



```
 test_id | URL                                                         | ACTUAL_RESULT | SYSTEM_RESULT
---------+-------------------------------------------------------------+---------------+--------------
       1 | https://creativ-praesent.at                                 |             1 |             1
       2 | http://www.yi163.xyz                                        |             1 |             1
       3 | http://www.ajscgasjk.ga/clip4                               |             1 |             1
       4 | https://0-i-fdik.000webhostapp.com                          |             1 |             1
       5 | http://0nedrive-live-com-storage-docs.000webhostapp.com     |             1 |             1
       6 | http://www.0nlinenojima.com                                 |             1 |             1
       7 | http://0s.n5vs44tv.verek.ru                                 |             1 |             1
       8 | http://0u098.000webhostapp.com                              |             1 |             1
       9 | http://1.boeycreative.com                                   |             1 |             1
      10 | http://100-customersatisfaction-000.com                     |             1 |             1
      11 | http://100000munkahely.hu                                   |             1 |             1
      12 | http://10001golos.000webhostapp.com                         |             1 |             1
      13 | http://10023657788.000webhostapp.com                        |             1 |             1
      14 | http://1001balonfestivali.com                               |             1 |             1
      15 | http://100milesmanhattan.com                                |             1 |             1
      16 | http://100percentcommissionforrealtors.com                  |             1 |             1
      17 | http://11111111112300.000webhostapp.com                     |             1 |             1
      18 | http://12lmao12.000webhostapp.com                           |             1 |             1
      19 | http://13468114834589.aw-pay.com                            |             1 |             1
      20 | http://www.13bills.com                                      |             1 |             1
      21 | http://14kivza.kl.com.ua                                    |             1 |             1
      22 | http://153284594738391.statictab.com                        |             1 |             1
      23 | http://15448884464684afeye.000webhostapp.com                |             1 |             1
      24 | http://15852.000webhostapp.com                              |             1 |             1
      25 | http://163126.000webhostapp.com                             |             1 |             1
      26 | http://www.youtube.com                                      |             0 |             0
      27 | http://www.ucsy.edu.mm                                      |             0 |             0
      28 | https://www.myanmaryellowpages.biz                          |             0 |             0
      29 | https://www.itecgoi.in/e-itec                               |             0 |             0
      30 | https://www.w3schools.com                                   |             0 |             0
```

Figure3.6 Sample URLs and Email links Collection

CHAPTER 4
IMPLEMENTATION OF THE PROPOSED SYSTEM

4.1 Experimental Setup

This chapter includes the parts that show the implementation, system design, and performance evaluation of the proposed system. This web application testing system aims to detect the vulnerability of the web application. The system mainly detects two vulnerabilities that are existing HTTPS Spoofing and Email Hijacking attacks. In this system, it is implemented by writing Python Programming Language and MySQL server is used to store the detecting results for the performance evaluation.

4.2 System's Design Implementation

Firstly, the following figures are the design forms of the system's application. When the system starts, the user can find the main design form of the system as described in Figure 4.1. This main design includes the input box for entering the URL or Email, input box for evaluation result which enters existing vulnerability or not, next input box for sender's email address which is used for Email Hijacking vulnerability, the two main options namely HTTPS Spoofing and Email Hijacking, and the six panels namely, Crawl Page, Extract Domain Names, Testing, Extract Email Info, Testing, Extract Links and test accuracy button respectively. The first three panels are used for HTTPS Spoofing vulnerability and next three panels are used for Email Hijacking vulnerability.

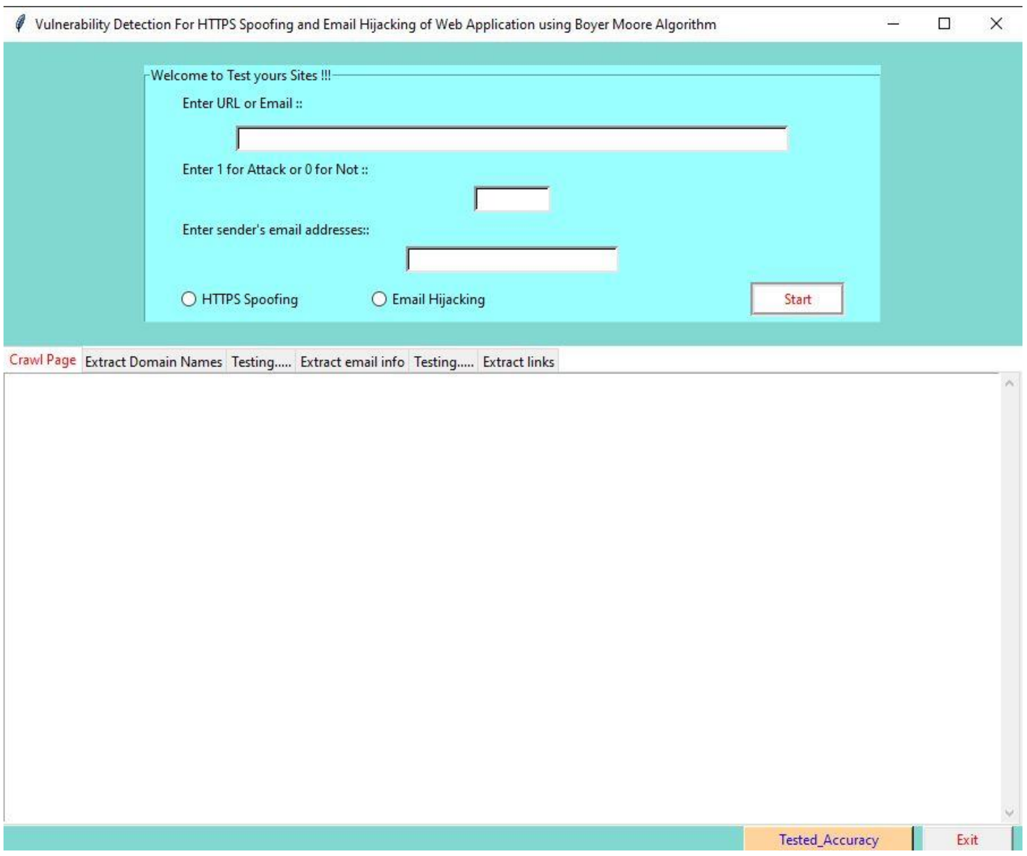
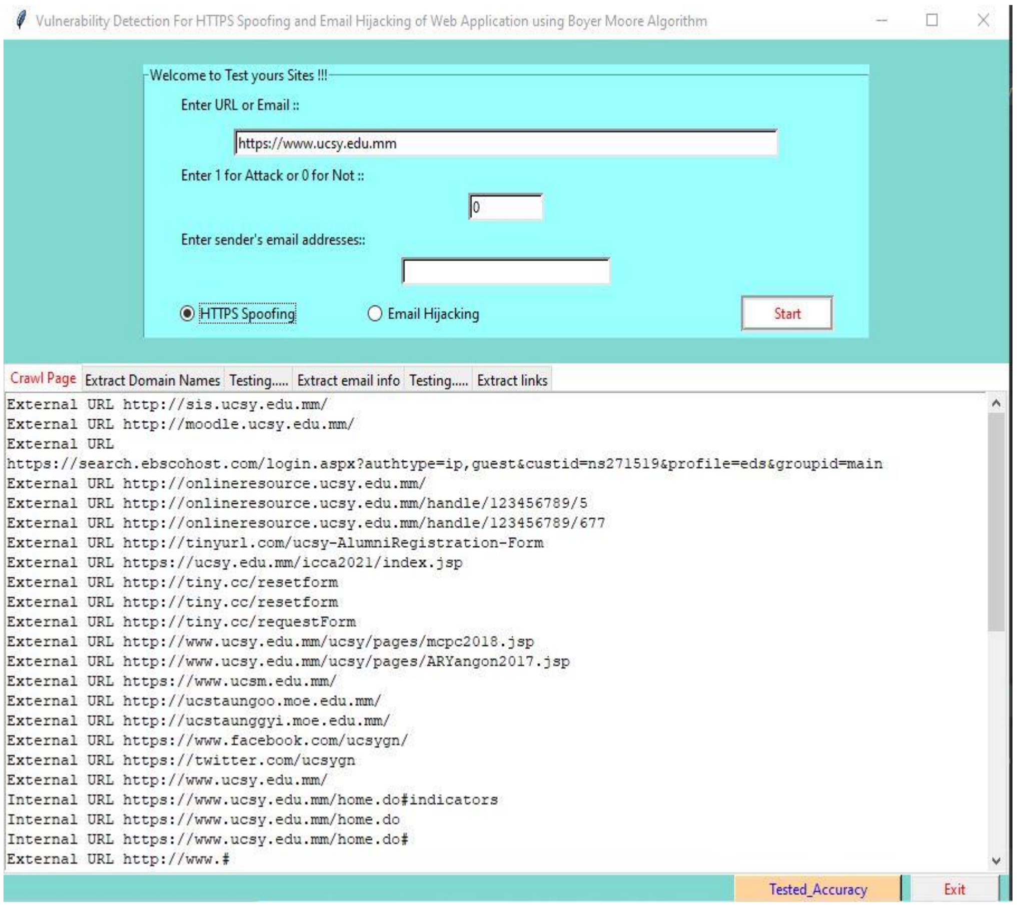

Vulnerability Detection For HTTPS Spoofing and Email Hijacking of Web Application using Boyer Moore Algorithm
Welcome to Test yours Sites !!!
Enter URL or Email ::
Enter 1 for Attack or 0 for Not ::
Enter sender's email addresses::
HTTPS Spoofing
Email Hijacking
Start
Crawl Page
Extract Domain Names
Testing.....
Extract email info
Testing.....
Extract links
Tested_Accuracy
Exit

Figure 4.1 Main form of the proposed system

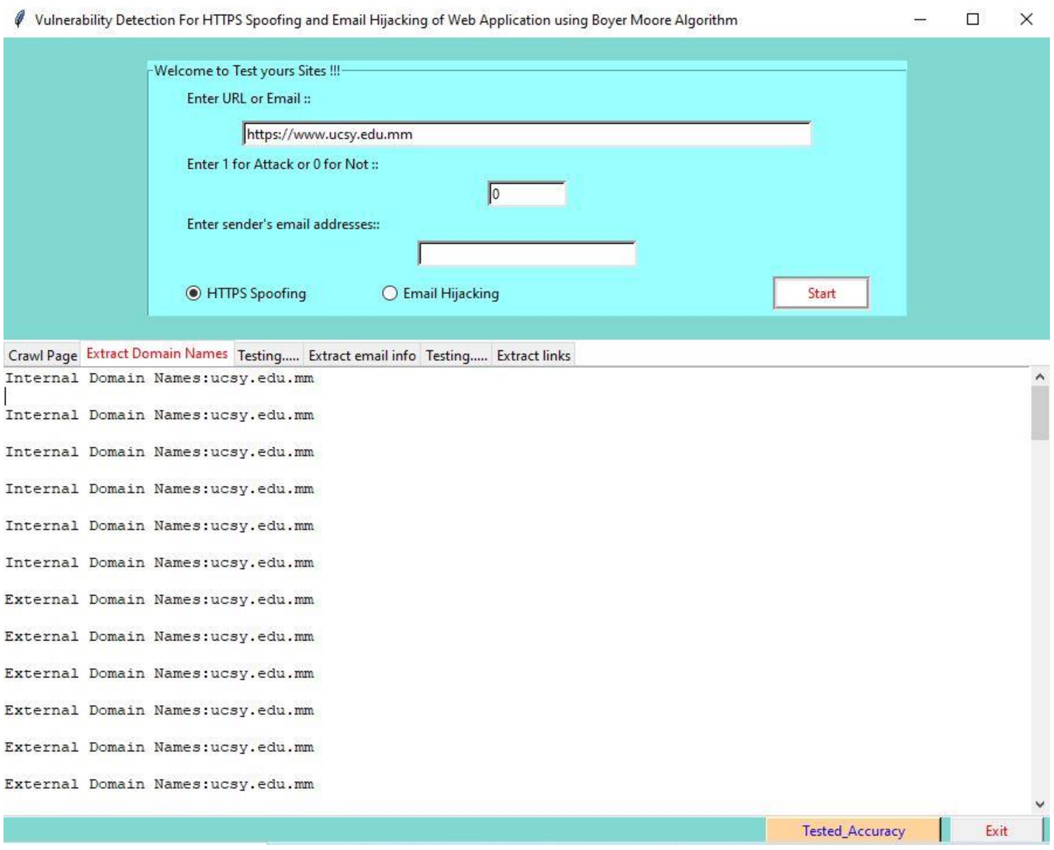
In this main form, the input box is used for the required URL or Email data to test in the system. The user enters the URL or Email in this input box. The second input box is used for the evaluation result of the system which user fills entered URL or Email is existing attack or not such as 1 is defined for attack and 0 for not attack. The next input box for sender's email address which is only used for checking Email Hijacking vulnerability. The two options are used to detect the HTTPS Spoofing vulnerability and Email Hijacking vulnerability. The user needs to select one of them to detect the desired vulnerability. The first three panels are used for HTTPS Spoofing which shows the detail description and analysis results of the proposed system about this vulnerability. The next three panels are used for Email Hijacking which shows the detailed description and analysis results of the proposed system about this vulnerability. They appeared step by step when testing the URL or Email. In HTTPS Spoofing, the first panel of Crawl Page is shown in Figure 4.2. And, Tested Accuracy button is a button which allows for viewing the comparison result and the accuracy calculation of the proposed system.


Vulnerability Detection For HTTPS Spoofing and Email Hijacking of Web Application using Boyer Moore Algorithm
Welcome to Test yours Sites !!!
Enter URL or Email ::
https://www.ucsy.edu.mm
Enter 1 for Attack or 0 for Not ::
0
Enter sender's email addresses::
HTTPS Spoofing
Email Hijacking
Start
Crawl Page
Extract Domain Names
Testing.....
Extract email info
Testing.....
Extract links
Tested_Accuracy
Exit

```
External URL http://sis.ucsy.edu.mm/
External URL http://moodle.ucsy.edu.mm/
External URL
https://search.ebscohost.com/login.aspx?authtype=ip,guest&custid=ns271519&profile=eds&groupid=main
External URL http://onlineresource.ucsy.edu.mm/
External URL http://onlineresource.ucsy.edu.mm/handle/123456789/5
External URL http://onlineresource.ucsy.edu.mm/handle/123456789/677
External URL http://tinyurl.com/ucsy-AlumniRegistration-Form
External URL https://ucsy.edu.mm/icca2021/index.jsp
External URL http://tiny.cc/resetform
External URL http://tiny.cc/resetform
External URL http://tiny.cc/requestForm
External URL http://www.ucsy.edu.mm/ucsy/pages/mcpc2018.jsp
External URL http://www.ucsy.edu.mm/ucsy/pages/ARYangon2017.jsp
External URL https://www.ucsm.edu.mm/
External URL http://ucstaungoo.moe.edu.mm/
External URL http://ucstaunggyi.moe.edu.mm/
External URL https://www.facebook.com/ucsygn/
External URL https://twitter.com/ucsygn
External URL http://www.ucsy.edu.mm/
Internal URL https://www.ucsy.edu.mm/home.do#indicators
Internal URL https://www.ucsy.edu.mm/home.do
Internal URL https://www.ucsy.edu.mm/home.do#
External URL http://www.#
```

Figure 4.2 Splitting into External and Internal URL links

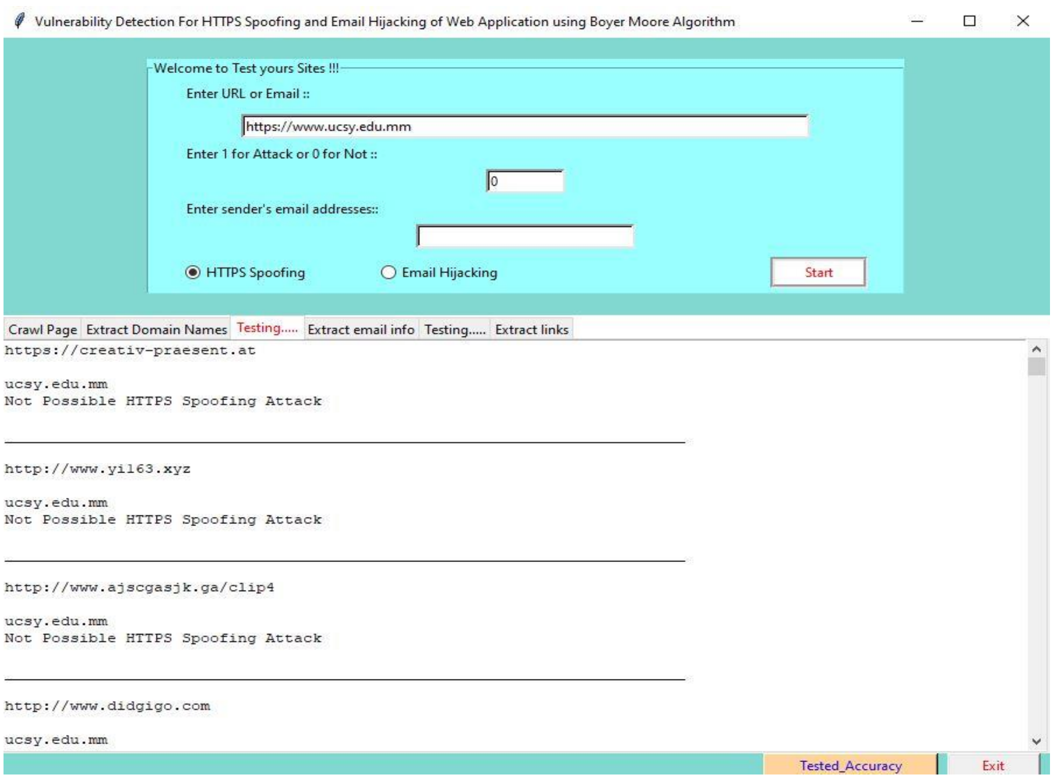
Firstly, the user can consider the desired vulnerability type which is HTTPS Spoofing or Email Hijacking. If user chose HTTPS Spoofing, user must enter URL in first input box. If user chose Email Hijacking, user must enter user's Gmail in first input box. After filling the input URL or Email, the user next enters 1 or 0 which is attack or not for evaluation result in the second input box. In HTTPS Spoofing, the user is not need to enter sender's email address in third input box. Next, the user requires for choosing the desired options. By choosing the desired option from the two radio button and clicking start button. If user chose HTTPS Spoofing, the first three panels is working. In the first panel, all possible of internal and external URLs of the detected web application can be shown in Figure 4.2. Start button is used to operate this detecting process. And then the extracting domain names process appears on the next panel. This panel allows the user to view the domain names of the URLs that extract the internal and external URL links from first panel. The domain names result of filtering is shown in Figure 4.3.


Vulnerability Detection For HTTPS Spoofing and Email Hijacking of Web Application using Boyer Moore Algorithm
Welcome to Test yours Sites !!!
Enter URL or Email ::
https://www.ucsy.edu.mm
Enter 1 for Attack or 0 for Not ::
0
Enter sender's email addresses::
HTTPS Spoofing
Email Hijacking
Start
Crawl Page
Extract Domain Names
Testing.....
Extract email info
Testing.....
Extract links
Tested_Accuracy
Exit

```
Internal Domain Names:ucsy.edu.mm

Internal Domain Names:ucsy.edu.mm

Internal Domain Names:ucsy.edu.mm

Internal Domain Names:ucsy.edu.mm

Internal Domain Names:ucsy.edu.mm

Internal Domain Names:ucsy.edu.mm

External Domain Names:ucsy.edu.mm

External Domain Names:ucsy.edu.mm

External Domain Names:ucsy.edu.mm

External Domain Names:ucsy.edu.mm

External Domain Names:ucsy.edu.mm

External Domain Names:ucsy.edu.mm
```

Figure 4.3 Extracting Internal and External Domain Names

After filtering second panel has finished, testing process is found in the third panel. In third panel, firstly the extracted domain names from second panel are check with the pre-collected attacked URLs. This pre-collected attacked URLs including blacklist domain names. If the extracted domain names include in the pre-collected attacked URLs, this decides HTTPS Spoofing vulnerability. If not, this is not HTTPS Spoofing vulnerability. Figure 4.4 shows the testing results in third panel.


Vulnerability Detection For HTTPS Spoofing and Email Hijacking of Web Application using Boyer Moore Algorithm

Welcome to Test yours Sites !!!
Enter URL or Email :: https://www.ucsy.edu.mm
Enter 1 for Attack or 0 for Not :: 0
Enter sender's email addresses::
● HTTPS Spoofing ○ Email Hijacking Start

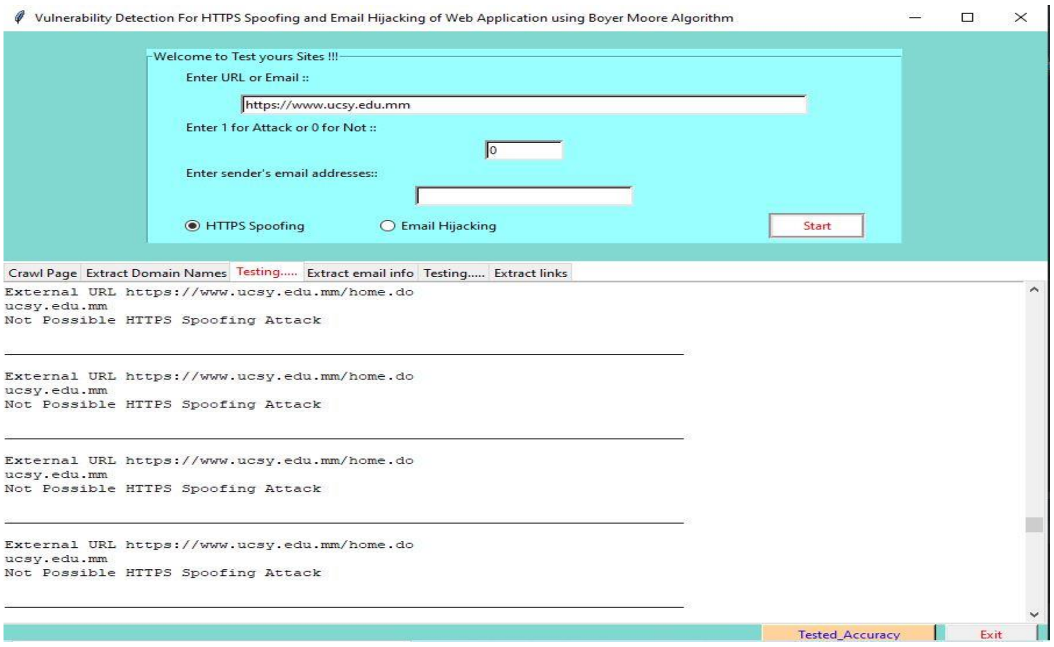
Crawl Page | Extract Domain Names | Testing..... | Extract email info | Testing..... | Extract links

```
https://creativ-praesent.at

ucsy.edu.mm
Not Possible HTTPS Spoofing Attack

________________________________________

http://www.yil63.xyz

ucsy.edu.mm
Not Possible HTTPS Spoofing Attack

________________________________________

http://www.ajscgasjk.ga/clip4

ucsy.edu.mm
Not Possible HTTPS Spoofing Attack

________________________________________

http://www.didgigo.com

ucsy.edu.mm
```

Tested_Accuracy | Exit

Figure 4.4 compare domain names with pre-collected attack URLs

Then, the system next checks in third panel. In this next check, the external URL links check about relating with internal domain names. This checks internal domain name included part of external domain names. If include as part of external domain names, this can decide HTTPS Spoofing vulnerability. If not, this is not HTTPS Spoofing vulnerability. This testing result appended in third panel in Figure 4.5.


Vulnerability Detection For HTTPS Spoofing and Email Hijacking of Web Application using Boyer Moore Algorithm

Welcome to Test yours Sites !!!
Enter URL or Email :: https://www.ucsy.edu.mm
Enter 1 for Attack or 0 for Not :: 0
Enter sender's email addresses::
● HTTPS Spoofing ○ Email Hijacking Start

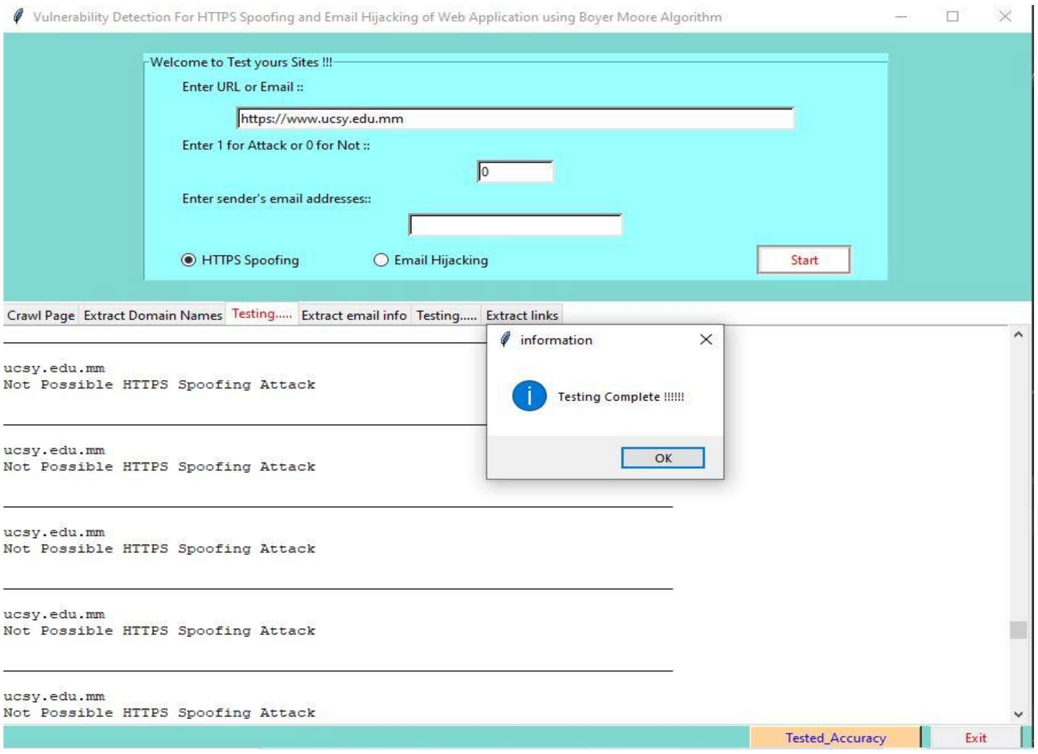
Crawl Page | Extract Domain Names | Testing..... | Extract email info | Testing..... | Extract links

```
External URL https://www.ucsy.edu.mm/home.do
ucsy.edu.mm
Not Possible HTTPS Spoofing Attack

________________________________________

External URL https://www.ucsy.edu.mm/home.do
ucsy.edu.mm
Not Possible HTTPS Spoofing Attack

________________________________________

External URL https://www.ucsy.edu.mm/home.do
ucsy.edu.mm
Not Possible HTTPS Spoofing Attack

________________________________________

External URL https://www.ucsy.edu.mm/home.do
ucsy.edu.mm
Not Possible HTTPS Spoofing Attack

________________________________________
```

Tested_Accuracy | Exit

Figure 4.5 Check external links relating with internal domain

After detecting process of HTTPS Spoofing vulnerability is finished, the tested results are described in the third panel. Then, the information type of message box is appeared when the testing process of HTTPS Spoofing is finished as described in Figure 4.6.



Vulnerability Detection For HTTPS Spoofing and Email Hijacking of Web Application using Boyer Moore Algorithm

Welcome to Test yours Sites !!!

Enter URL or Email :: https://www.ucsy.edu.mm

Enter 1 for Attack or 0 for Not :: 0

Enter sender's email addresses::

(●) HTTPS Spoofing ( ) Email Hijacking

information

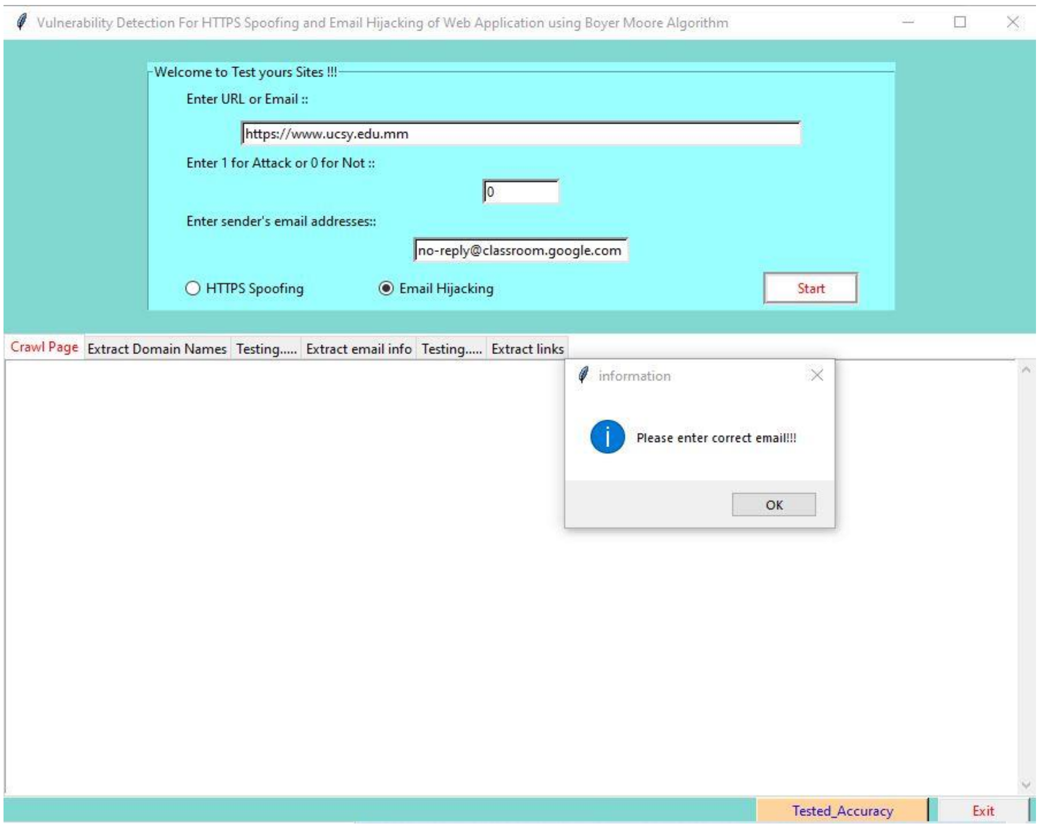
Testing Complete !!!!!!

```
ucsy.edu.mm
Not Possible HTTPS Spoofing Attack

ucsy.edu.mm
Not Possible HTTPS Spoofing Attack

ucsy.edu.mm
Not Possible HTTPS Spoofing Attack

ucsy.edu.mm
Not Possible HTTPS Spoofing Attack

ucsy.edu.mm
Not Possible HTTPS Spoofing Attack
```

Figure 4.6 Testing Complete State of HTTPS Spoofing

In Email Hijacking, user must enter user's Gmail in first input box. After filling the input Email, the user next enters 1 or 0 which is attack or not for evaluation result in the second input box. Then, the user needs to enter sender's email address in third input box. Next, the user chooses the desired option from the radio button and clicks start button. After start button is clicked, the system checks user's Gmail validation. This checking result is shown in Figure 4.7 and 4.8.

In Figure 4.7, user enters incorrect email in first input box. The user needs to enter the correct Gmail. When the user clicks start button by entering incorrect email, the system shows the information text box as Figure 4.7.



Vulnerability Detection For HTTPS Spoofing and Email Hijacking of Web Application using Boyer Moore Algorithm

Welcome to Test yours Sites !!!

Enter URL or Email :: https://www.ucsy.edu.mm

Enter 1 for Attack or 0 for Not :: 0

Enter sender's email addresses:: no-reply@classroom.google.com

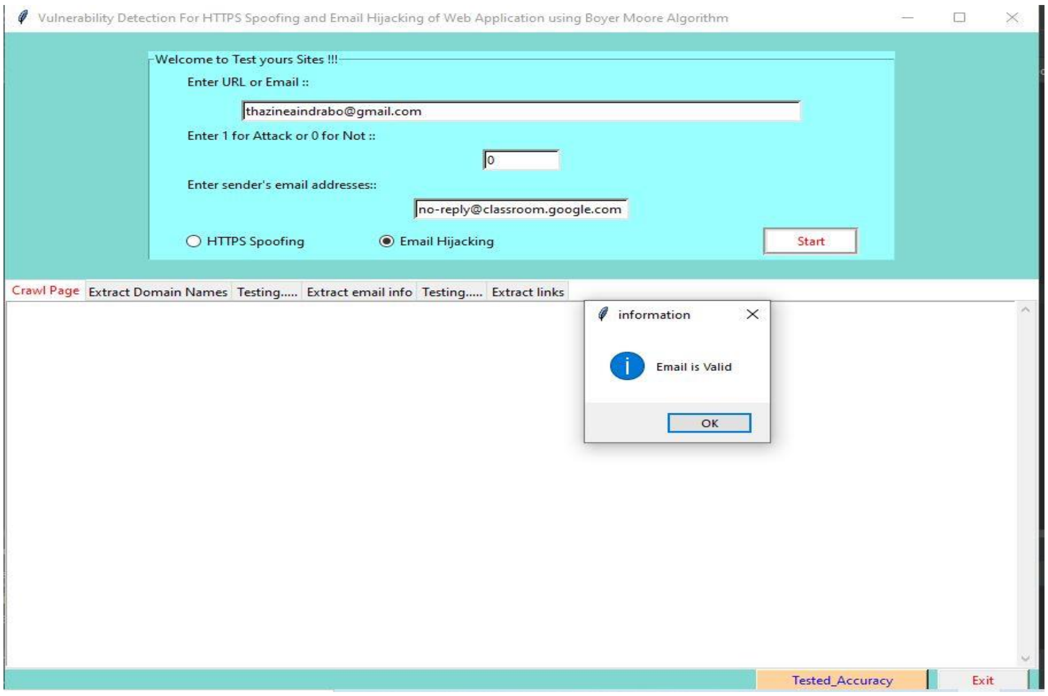
( ) HTTPS Spoofing [✓ Email Hijacking]

information

Please enter correct email!!!

Figure 4.7 Showing information type of message box for incorrect Email

In Figure 4.8, the user enters correct Gmail in first input box. When the user clicks start button by entering correct Gmail, the system shows the information text box as Figure 4.8.



Vulnerability Detection For HTTPS Spoofing and Email Hijacking of Web Application using Boyer Moore Algorithm

Welcome to Test yours Sites !!!

Enter URL or Email :: thazineaindrabo@gmail.com

Enter 1 for Attack or 0 for Not :: 0

Enter sender's email addresses:: no-reply@classroom.google.com

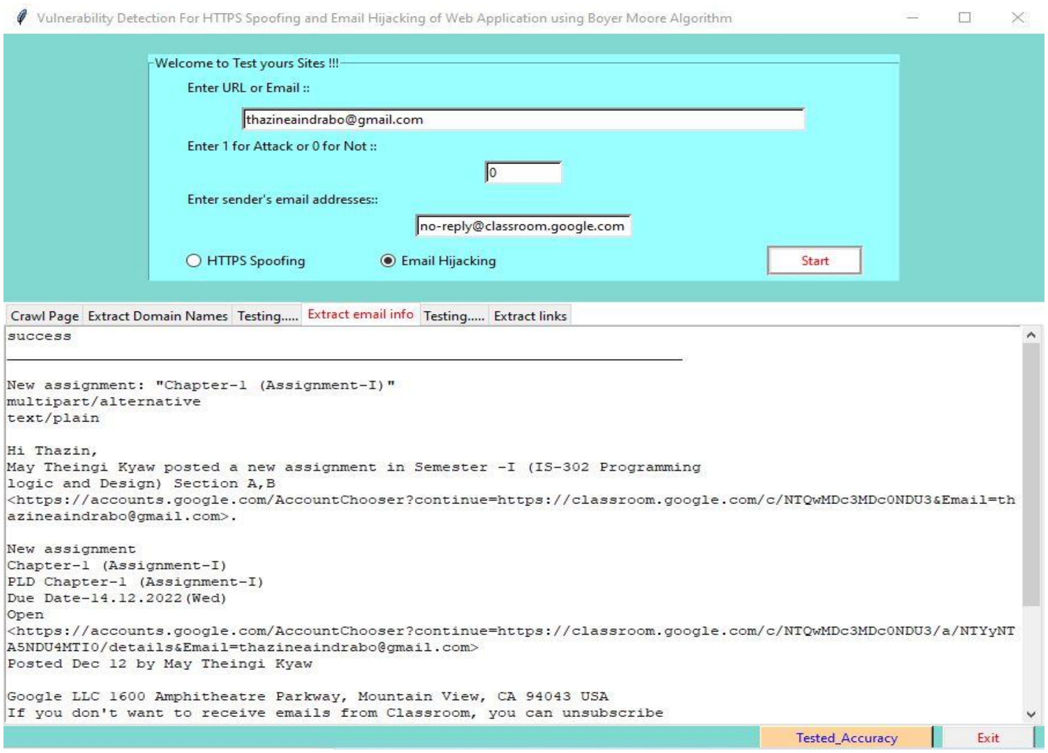
( ) HTTPS Spoofing [✓ Email Hijacking]

information

Email is Valid

Figure 4.8 Check email validation

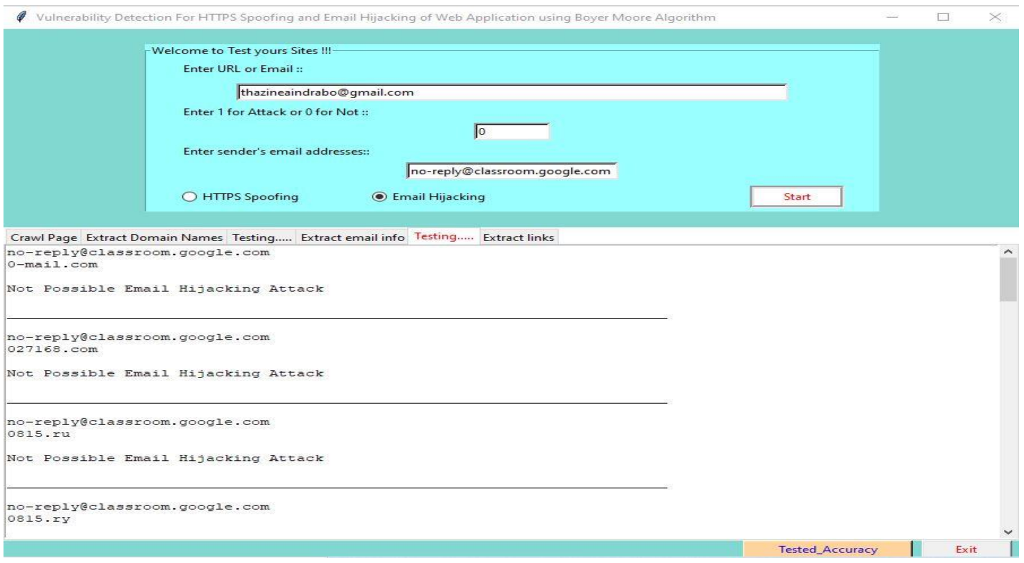
If the user chose Email Hijacking, the last three panels is working. In the fourth panel, the detailed email information is extracted according to sender's email address. This email information can be seen in Figure 4.9.


Vulnerability Detection For HTTPS Spoofing and Email Hijacking of Web Application using Boyer Moore Algorithm
Welcome to Test yours Sites !!!
Enter URL or Email ::
thazineaindrabo@gmail.com
Enter 1 for Attack or 0 for Not ::
0
Enter sender's email addresses::
no-reply@classroom.google.com
HTTPS Spoofing
Email Hijacking
Start
Crawl Page
Extract Domain Names
Testing.....
Extract email info
Testing.....
Extract links
Tested_Accuracy
Exit

```
success
________________________________________________________________

New assignment: "Chapter-1 (Assignment-I)"
multipart/alternative
text/plain

Hi Thazin,
May Theingi Kyaw posted a new assignment in Semester -I (IS-302 Programming
logic and Design) Section A,B
<https://accounts.google.com/AccountChooser?continue=https://classroom.google.com/c/NTQwMDc3MDc0NDU3&Email=th
azineaindrabo@gmail.com>.

New assignment
Chapter-1 (Assignment-I)
PLD Chapter-1 (Assignment-I)
Due Date-14.12.2022(Wed)
Open
<https://accounts.google.com/AccountChooser?continue=https://classroom.google.com/c/NTQwMDc3MDc0NDU3/a/NTYyNT
A5NDU4MTI0/details&Email=thazineaindrabo@gmail.com>
Posted Dec 12 by May Theingi Kyaw

Google LLC 1600 Amphitheatre Parkway, Mountain View, CA 94043 USA
If you don't want to receive emails from Classroom, you can unsubscribe
```

Figure 4. 9 Extracting email detailed information

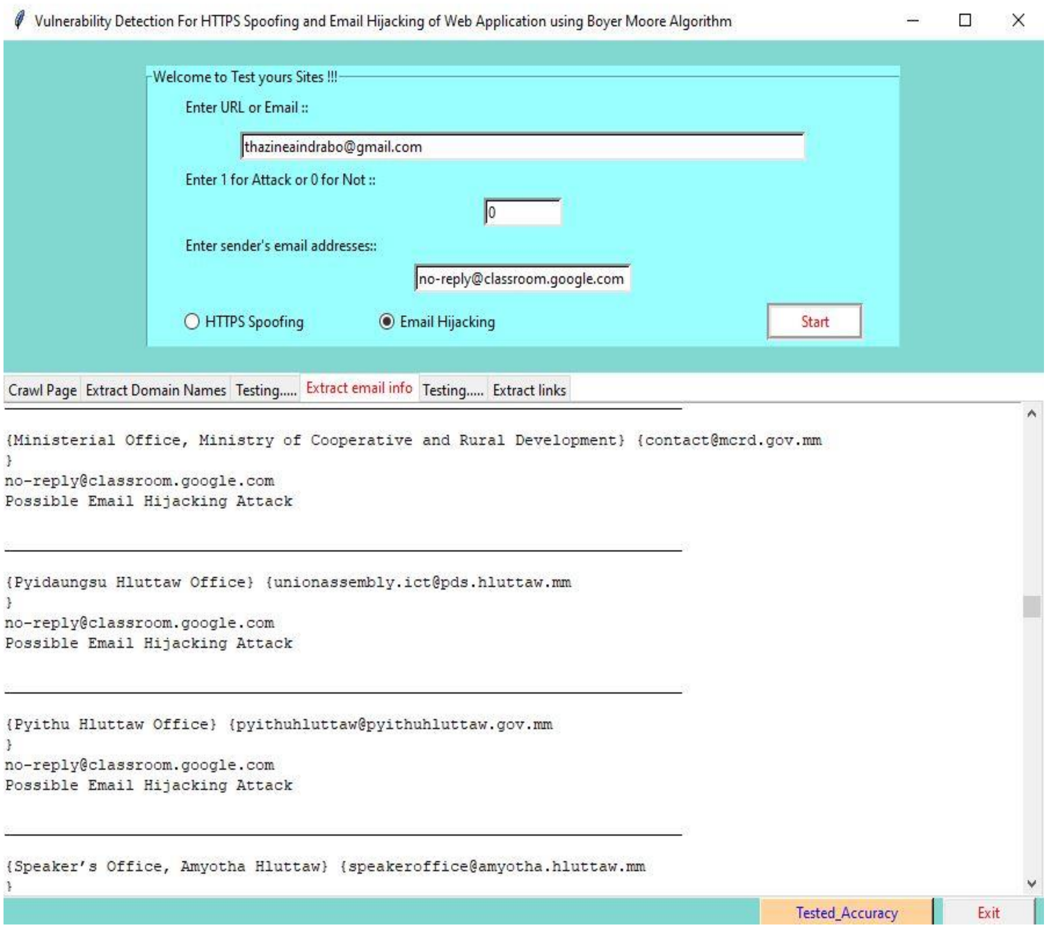
After checking email validation, the system checks sender's email address with pre-collected attack email domains in fifth panel. If sender's email includes this attack email domains, this email is possible Email Hijacking. If not include, this email is not possible Email Hijacking. This testing result can be seen in Figure 4.10.


Vulnerability Detection For HTTPS Spoofing and Email Hijacking of Web Application using Boyer Moore Algorithm
Welcome to Test yours Sites !!!
Enter URL or Email ::
thazineaindrabo@gmail.com
Enter 1 for Attack or 0 for Not ::
0
Enter sender's email addresses::
no-reply@classroom.google.com
HTTPS Spoofing
Email Hijacking
Start
Crawl Page
Extract Domain Names
Testing.....
Extract email info
Testing.....
Extract links
Tested_Accuracy
Exit

```
no-reply@classroom.google.com
0-mail.com

Not Possible Email Hijacking Attack

________________________________________________________________

no-reply@classroom.google.com
027168.com

Not Possible Email Hijacking Attack

________________________________________________________________

no-reply@classroom.google.com
0815.ru

Not Possible Email Hijacking Attack

________________________________________________________________

no-reply@classroom.google.com
0815.ry
```

Figure 4.10 check sender's email address with blacklist email address

After this checking, extracted links within email check whether including parameter or not. If parameter including in links, this links included parameters of email address next check with legitimate government email data. If these links of email address are matched legitimate email data, this email is considered not possible Email Hijacking. If not, it is considered as possible Email Hijacking. This result can be seen as Figure 4.11.


Vulnerability Detection For HTTPS Spoofing and Email Hijacking of Web Application using Boyer Moore Algorithm
Welcome to Test yours Sites !!!
Start
Crawl Page
Extract Domain Names
Testing.....
Extract email info
Testing.....
Extract links
Tested_Accuracy
Exit

Enter URL or Email :: thazineaindrabo@gmail.com

Enter 1 for Attack or 0 for Not :: 0

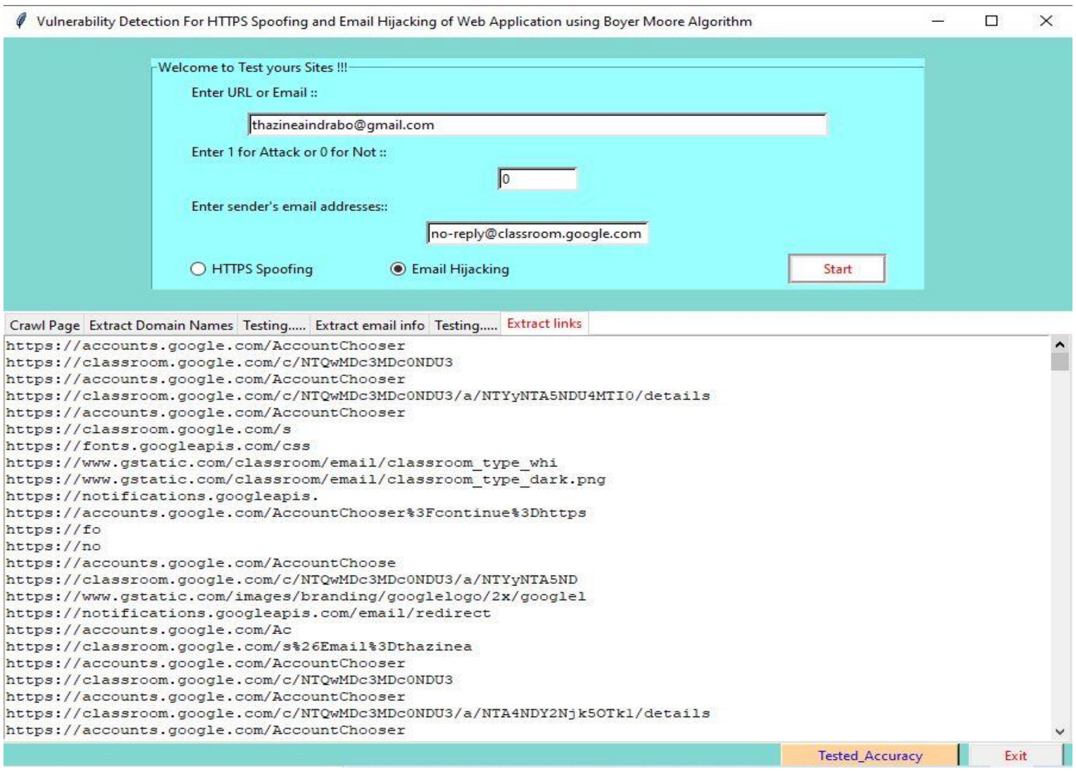
Enter sender's email addresses:: no-reply@classroom.google.com

HTTPS Spoofing [✓ Email Hijacking]

```
{Ministerial Office, Ministry of Cooperative and Rural Development} {contact@mcrd.gov.mm
}
no-reply@classroom.google.com
Possible Email Hijacking Attack

{Pyidaungsu Hluttaw Office} {unionassembly.ict@pds.hluttaw.mm
}
no-reply@classroom.google.com
Possible Email Hijacking Attack

{Pyithu Hluttaw Office} {pyithuhluttaw@pyithuhluttaw.gov.mm
}
no-reply@classroom.google.com
Possible Email Hijacking Attack

{Speaker's Office, Amyotha Hluttaw} {speakeroffice@amyotha.hluttaw.mm
}
```

Figure 4.11 Check email links included parameters with legitimate email

After that, the last panel can be seen extracted links in email body. This result can be shown in Figure 4.12.


Vulnerability Detection For HTTPS Spoofing and Email Hijacking of Web Application using Boyer Moore Algorithm
Welcome to Test yours Sites !!!

Enter URL or Email :: thazineaindrabo@gmail.com

Enter 1 for Attack or 0 for Not :: 0

Enter sender's email addresses:: no-reply@classroom.google.com

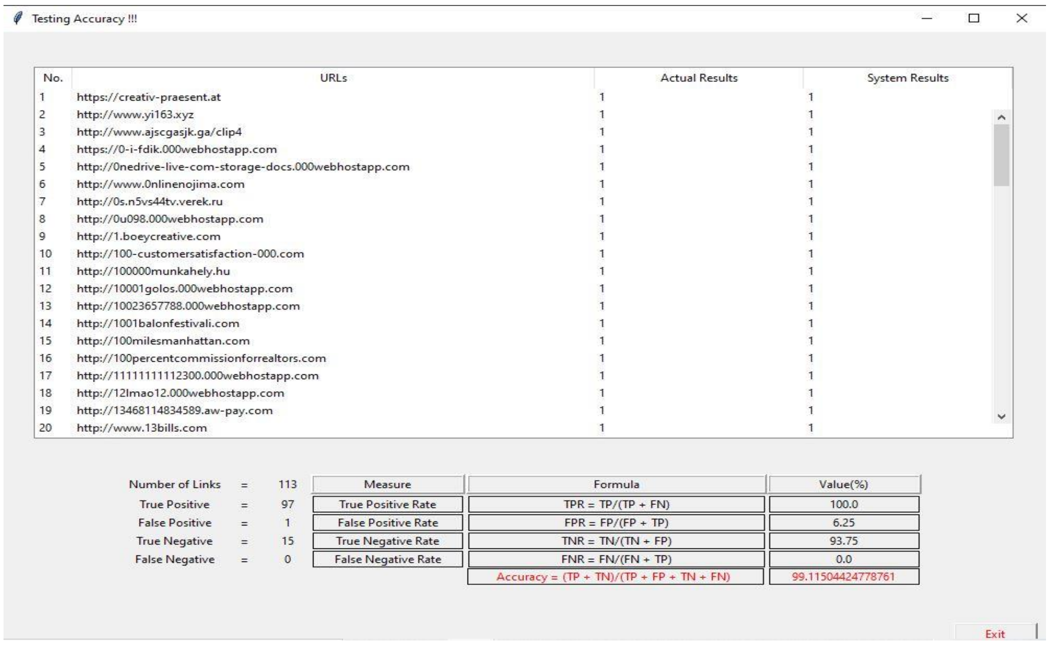
HTTPS Spoofing [✓ Email Hijacking]

Start
Crawl Page
Extract Domain Names
Testing.....
Extract email info
Testing.....
Extract links

```
https://accounts.google.com/AccountChooser
https://classroom.google.com/c/NTQwMDc3MDc0NDU3
https://accounts.google.com/AccountChooser
https://classroom.google.com/c/NTQwMDc3MDc0NDU3/a/NTYyNTA5NDU4MTI0/details
https://accounts.google.com/AccountChooser
https://classroom.google.com/s
https://fonts.googleapis.com/css
https://www.gstatic.com/classroom/email/classroom_type_whi
https://www.gstatic.com/classroom/email/classroom_type_dark.png
https://notifications.googleapis.
https://accounts.google.com/AccountChooser%3Fcontinue%3Dhttps
https://fo
https://no
https://accounts.google.com/AccountChoose
https://classroom.google.com/c/NTQwMDc3MDc0NDU3/a/NTYyNTA5ND
https://www.gstatic.com/images/branding/googlelogo/2x/googlel
https://notifications.googleapis.com/email/redirect
https://accounts.google.com/Ac
https://classroom.google.com/s%26Email%3Dthazinea
https://accounts.google.com/AccountChooser
https://classroom.google.com/c/NTQwMDc3MDc0NDU3
https://accounts.google.com/AccountChooser
https://classroom.google.com/c/NTQwMDc3MDc0NDU3/a/NTA4NDY2Njk5OTk1/details
https://accounts.google.com/AccountChooser
```

Figure 4.12 Extracting links in email

The comparison results of the proposed system and attacked datasets on detected URLs and Emails can be seen by clicking the "Tested Accuracy" button. Calculating of the accuracy shows the based on the detected result of proposed system. Figure 4.13 presents a tested screenshot showing the stored comparison result and accuracy calculation result.



No.	URLs	Actual Results	System Results
1	https://creativ-praesent.at	1	1
2	http://www.yi163.xyz	1	1
3	http://www.ajscgasjk.ga/clip4	1	1
4	https://0-i-fdik.000webhostapp.com	1	1
5	http://0nedrive-live-com-storage-docs.000webhostapp.com	1	1
6	http://www.0nlinenojima.com	1	1
7	http://0s.n5vs44tv.verek.ru	1	1
8	http://0u098.000webhostapp.com	1	1
9	http://1.boeycreative.com	1	1
10	http://100-customersatisfaction-000.com	1	1
11	http://100000munkahely.hu	1	1
12	http://10001golos.000webhostapp.com	1	1
13	http://10023657788.000webhostapp.com	1	1
14	http://1001balonfestivali.com	1	1
15	http://100milesmanhattan.com	1	1
16	http://100percentcommissionforrealtors.com	1	1
17	http://11111111112300.000webhostapp.com	1	1
18	http://12lmao12.000webhostapp.com	1	1
19	http://13468114834589.aw-pay.com	1	1
20	http://www.13bills.com	1	1

			Measure	Formula	Value(%)
Number of Links	=	113			
True Positive	=	97	True Positive Rate	TPR = TP/(TP + FN)	100.0
False Positive	=	1	False Positive Rate	FPR = FP/(FP + TP)	6.25
True Negative	=	15	True Negative Rate	TNR = TN/(TN + FP)	93.75
False Negative	=	0	False Negative Rate	FNR = FN/(FN + TP)	0.0
				Accuracy = (TP + TN)/(TP + FP + TN + FN)	99.11504424778761

Figure 4.13 Calculating the accuracy of the proposed system

4.3 Experiment Results

The analysis includes stored URLs and Email links, attacked datasets and the proposed system that was detected with the corresponding type of vulnerabilities. In examining, attacked datasets was taken to quantify the accuracy of the proposed system. These attacked datasets and the proposed system evaluated the vulnerabilities by using these stored URLs and Email links.

After the evaluation process is finished, both of these two methods had 97 URLs and Email links in true positive, 15 URLs and Email links in true negative and 0 URL and Email link in false negative. In false positive, attacked datasets has 0 URL or Email link and 1 URL or Email link in the proposed system. The proposed system and attacked datasets were nearly equal to the comparing results of true positive, false positive, true negative and false negative rates. Table 4.1 can be seen the percentage of the accuracy of the proposed system.

Table 4.1 Accuracy results for Proposed System

Measure	Value (%)
True Positive Rate	100%
False Positive Rate	6.25%
True Negative Rate	93.75%
False Negative Rate	0.0%
Accuracy	99.11%

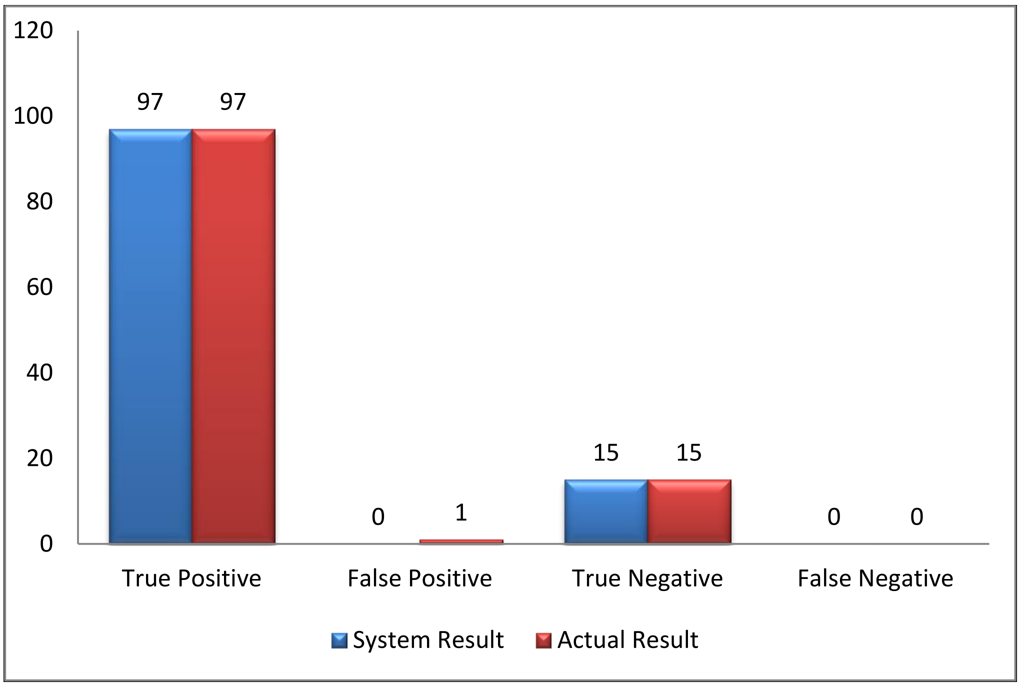


Figure 4.14 Comparison results of proposed system and attacked datasets

In Figure 4.14 shows the comparison results of the proposed system and attacked datasets.

The processing time was caused by the crawling URL result and numbers of sending email messages are different. So, the processing time is not similar over each input. Furthermore, after all testing completed, the result showed step by step in this proposed system and checked vulnerabilities for HTTPS Spoofing and Email Hijacking. So, users can choose the desired vulnerability type for testing.

CHAPTER 5
CONCLUSION

In this thesis, the researcher mainly focuses the detection of web applications vulnerability, particularly in two main types of Man-In-The-Middle (MITM) attack, namely HTTPS Spoofing and Email Hijacking. This thesis presents the proposed algorithm, Boyer Moore String Matching Algorithm use to detect URL and Email that have possible vulnerabilities for HTTPS Spoofing and Email Hijacking attacks. The experimental results of the system are compared with the proposed system and attacked datasets. The system is implemented by using python programming language. In this chapter, the summary of the main system conclusion and advantages of system, limitations and further extensions are described.

The proposed system uses URL (Uniform Resource Locator) or Gmail as input for detecting web application. It has the choice to either detect HTTPS Spoofing or Email Hijacking vulnerability. It provides the detailed checking, which includes the vulnerable URLs and Email links and step by step processing. It also applies Boyer Moore string matching algorithm to search strings in datasets, which makes to be more accurate for the proposed system.

The experimental result of the proposed system produces comparable results with attacked datasets. The users can decide the type of vulnerability, which is either HTTPS Spoofing or Email Hijacking. The proposed algorithm processes to get good result for accuracy by decreasing the rate of false positive and false negative. While testing the web application vulnerability, this algorithm makes effective and reliable processes.

In conclusion, the proposed system is customizable and reliable. And, it helps the web developers and web application users to protect their web applications from being attacked and who begin to learn the security by showing step by step of detection stages of web application.

5.1 Limitations and Further Extensions

In this study, the system does not consider other types of web application vulnerabilities of Man-In-The-Middle (MITM) in cyber security field such as IP Spoofing, DNS Spoofing, SSL Hijacking and so on. In future work, this will be processed to test with all vulnerabilities in websites, emails and retain the low false positive and false negative rate.

This system cannot check harmful text message in email body and can only check links in email body. Moreover, the system cannot examine blacklist emails addresses outside datasets. Legitimate email data are collected according system's needs but this data can be not enough. In URL, phishing links cannot work on HTTP request.

AUTHOR'S PUBLICATION

[1] Thazin Eaindra Bo, Zin Thu Thu Myint, "*Vulnerability Detection for HTTPS Spoofing and Email Hijacking Attacks of Web Application using Boyer Moore String Matching Algorithm*", National Journal of Parallel & Soft Computing, Yangon, Myanmar, 2022.

REFERENCES

[1] Ain Zubaidah Mohd Saleha, Nur Amizah Rozalia, Alya Geogiana Bujaa, Kamarularifin Abd. Jalila, Fakariah Hani Mohd Alia, Teh Fradilla Abdul Fahmana, "A Method for Web Application Vulnerabilities Detection by Using Boyer-Moore String Matching Algorithm", Information Systems International Conference (ISICO 2015), Universiti Teknologi MARA, Shah Alam 40000, Selangor, 2015.

[2] Argha Ghosh(&) and A. Senthilrajan, "An Approach for Detecting Man-In-The-Middle Attack Using DPI and DFI", A. P. Pandian et al. (Eds.): ICCBI 2019, LNDECT 49, Department of Computational Logistics, Alagappa University, Karaikudi, India, pp. 563–574, 2020.

[3] Email blacklist domain names data are collected https://github.com/disposable-email-domains/, visited on February 2022.

[4] Email legitimate data are collected from 'Government Directory Myanmar National Portal' website https://myanmar.gov.mm/governmentdirectory/, visited on March 2022.

[5] Kamran Mahmoudi, "Pattern mattching algorithms", Imam Khomeini International University, April 2017.

[6] OWASP Top Ten, [online] available: https://owasp.org/www-project-top-ten/, visit on February 2022.Simon Wahstrom, "Evaluation of string searching algorithms", Mälardalen University, Västerås, Sweden, 2012.

[7] Robert A. Sowah , Kwadwo B. Ofori-Amanfo, Godfrey A. Mills, and Koudjo M. Koumad, "Detection and Prevention of Man-in-the-Middle Spoofing Attacks in MANETs Using Predictive Techniques in Artificial Neural Networks (ANN)", Journal of Computer Networks and Communications Volume 2019, Article ID 4683982, Department of Computer Engineering, University of Ghana,2019

[8] Simon Wahstrom, "Evaluation of string searching algorithms", Mälardalen University, Västerås, Sweden, 2012.

[9] Tcamques10, "Different types of Vulnerability", https://www.ques10.com/p/3550/what-are-the-different-types-of-vulnerability-there/"?, visited on March 2022.

[10] Thomas Hamiton, Security Testing, https://owasp.org/www-project-top-ten/, visited on March 2022.

[11] U. Rahamathunnisa A.P (Sr), SITE, Vellore N. Manikandan A.P (SG), SITE, Vellore

U. Senthil Kumaran Associate Professor, Vellore C. Niveditha Student, MCA, "Preventing from phishing attack by implementing url pattern matching technique in web", Article in International Journal of Civil Engineering and Technology, VIT University, Vellore, September 2017.

[12] URL attack data are stored from https://github.com/mitchellkrogza/Phishing.Database/ , visited on February 2022.

[13] Wesley Chai, Confidentiality, Integrity and Availability, https://www.techtarget.com/whatis/definition/Confidentiality-integrity-and-availability -CIA#, visited on March 2022.

[14] What is Spoofing and How to Prevent a Spoofing Attack,[online]available: https://www.pandasecurity.com/en/media/center/ panda-security/what-ispoofing/, visit on March 2022.

[15] Xin Wanga,一, Runpu Wua, Jinxin Maa, Gang Longa, Jedeng Hana, "Research on Vulnerability Detection Technology for WEB MailSystem ",8th International Congress of Information and Communication Technology (ICICT-2018),China Information Technology Security Evaluation Center,China,2018.

[16] Zhiping Jiang, Kun Zhao, Rui Li, Jizhong Zhao & Junzhao Du , "identity spoofing attack detection and prevention for a wireless edge network" , Journal of Cloud Computing volume 9, Article number: 5 (2020).