COLLEGE OF INFORMATION TECHNOLOGY
UNIVERSITI TENAGA NASIONAL

COOKIES TAMPERING VULNERABILITY SCANNER

MU'AZ BIN AHMAD

2013

COOKIES TAMPERING VULNERABILITY SCANNER

by

MU'AZ BIN AHMAD

Project Supervisor: Dr. Salman Yussof

A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE BACHELOR OF COMPUTER SCIENCE (SYSTEMS AND NETWORKING) (HONS), COLLEGE OF INFORMATION TECHNOLOGY UNIVERSITI TENAGA NASIONAL

2013

DEDICATION

I would like to dedicate this project report to my beloved supervisor which is Dr. Salman Bin Yussof who have give me the drive and discipline to tackle any task with enthusiasm and determination.

ACKNOWLEDGEMENT

First and foremost I am offer my sincerest gratitude to my supervisor Dr. Salman Bin Yussof who had supported me throughout my Final Year Project with patience and knowledge. Without help from him, my Final Year Project would not have been completed within the time frame. One simply could not wish for a better or friendlier supervisor.

I would also like to express my gratitude to my project examiner, Madam Azimah Binti Abdul Ghapar for giving me an idea to improve my project and research about cookies tampering vulnerability and attack demonstration.

I would like to express my thanks to all colleagues for their support and help during the project progress. Lastly, I appreciated and like to acknowledge my parents for supporting me throughout my studies at UNITEN and completing my Final Year Project successfully.

ABSTRACT

"Cookies Tampering Vulnerability Scanner" is an Android application that would be able to detect the cookies tampering vulnerability in the website by performing the scanning to website source code. During the scanning, this application will detect the vulnerable source code to the cookies tampering attack. Once, the vulnerable source code has been detected, the application will straighly notify the user that the website is vulnerable to the cookies tampering attack. Cookies tampering attack is one type of attack that modify the value of cookie during sending back to the server from the web browser. Commonly, cookies tampering attack will be done to the e-commerce website by tampering the price of the item into a lower price without noticed by the vendor of the website. So, it is important to secure the website from cookies tampering attack. This project is using qualitative research methodology through literature research and website review method in order to gather information in details about related topic to the cookies tampering attack. Furthermore, this project is using prototyping as the Software Development Methodology. The first prototype had been developed during the project 1 while the next three prototypes of the application has been developed during project 2. The final prototype indicate the fully operate android application that is used to scan and detect cookies tampering vulnerability in the website source code. Then, cookies tampering attack demonstration also had been perform during project 1 and project 2 presentation as the fufilment of the project objectives.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Project Background

Web application is widely used in order to do an online transaction between Internet users because it is much easier and faster compared to manual transaction. Most of the transaction usually will involve money transaction such as money transfer, online ticket, and bill payment. However, does all the online transaction that commonly used is safe?

Every online transaction needs a user to log in first in order to do a transaction. Cookies are used when the user log in into a website. When the user is authenticated, the cookie is saved that allows the website to know the users are already logged in as they navigate around the site. This permits them access to any functionality that may be available only to the logged in user. Then, any online transaction will need a form in order for it to gather the information from the user. However, the probability of the form to be tampered by an attacker is high when the form is not secured. For example, the most common method of storing state information is by using HTML hidden field which the browser

does not display that field. It is an easy choice for preserving state information in web application such as price of certain item that sell online. Once the hidden form is not properly secured in the website, the probability for it to be tampered is high because the attacker may tamper the price of the selling item. This will cause a company that provides online business lost a lot of money.

This project will develop an android application in order to detect the vulnerability from cookie tampering attack in the website by using mobile phones or tablet pc. The vulnerability check will be focused on the weakness of the form on the website through hidden form field variable. If the website is vulnerable to this type of attack, the application will notify that this website has a potential to be attacked.

In addition, the project also will perform a study on cookie tampering vulnerability and method that could be used to fix the vulnerability. The study is important in order to develop an application that will identify the vulnerability.

1.2 Problem Statement

Cookie is technically a small piece of text that is sent to the web browser by the server and is intended to be sent back to the server. It is unchanged each time it accessed the same server or another server in the same domain. Cookie are used for authentication, tracking, maintaining state over stateless HTTP, as well as maintaining specific information about the user such as their site preferences. However cookie is vulnerable to the type of attack such as cookie theft, cookie poisoning, and cross-site cooking. All this type of attack may give a huge impact to the money online transaction. For example, the attacker may tamper the amount that needs to be paid for some transaction into a lower value and proceed with the process until it is successful without being noticed by the company. This activity may cause the company that provides online transaction business loss a lot of money.

The security vulnerability for this type of attack is difficult to be detected since the weakness is residing in the website source code. In addition, most of the e-commerce website are using a form to get user input and the form is include with hidden form field which use to hold the specific value for user transaction purpose. In general, most of the hidden form field that resides in the website is not properly secured. The attacker may easily tamper the value when it is sent to the server in plaintext value.

Actually, the difficulty in this project is about to detect the code that is not properly secured from cookies tampering attack. It is considered as not secure when the hidden form field is not encrypted. The challenge in this project is to detect whether the hidden form field that reside in the website is encrypted or not.

1.3 Project Objectives

The project consists of three objectives which are expected to be met at the end of the project. The objectives of the project are:

1) To perform a study on cookie tampering vulnerability and methods that can be used to fix this vulnerability.
2) To show and demonstrate how the cookie tampering vulnerability can exploited.
3) To develop an Android application to automatically test websites for cookie tampering vulnerability.

1.4 Project Scopes

The project is about to develop an android application that will be able to scan and detect the vulnerability in a website from cookie tampering attack using mobile phones or tablet pc. However, the project has its own boundary that limiting the project. The statement below state the scope of the project:

- The application will only detect the vulnerability on weakness of hidden form field variable in the website.
- Demonstrate cookies tampering exploit on the localhost machine using tamper data add-on.
- The application will only automatically scan the vulnerability on the entire page that the user access.
- The application will only detect the encryption of hidden form field value in PHP website.

1.5 Expected Benefits

There are some advantages that will be obtained after performing this project. The expected benefits from this project are:

- Produce mobile tool which is easier for identifying the cookie tampering vulnerability in the website.
- Produce a mobile tool which will be helpful in order to do a penetration testing.

- Expand knowledge in the network security field which will be helpful for working in the real industry after graduation.
- Gain new experience in security application development.

1.6 Project Requirement

There are several requirements that need to be setup in order to start developing an android application. The requirements that are needed in the project include:

- Java Development Kit (JDK)
- Android Software Development Kit (SDK)
- Integrated Development Environment (IDE)
- Android emulator

There are several requirements that need to be installed in order to do attack demonstration. The requirements that are needed to do attack demonstration include:

- Mozilla Firefox
- Tamper Data add-on
- Wamp Server

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

A literature review is a critical and in depth evaluation of previous research. It is a summary and synopsis of a particular area research that allowing anybody to read the paper to get a reason to pursuing the particular research program. The research of this project will be focused on the topic related to this project, research methodology, software development methodology, possible development tools and software to be used, review of the current system, and review of the similar system.

Most web applications work by passing information from a web browser on the user machine to the web server and vice versa. Cookies are small text files that are saved on a user machine to store information from a website. Many web applications used cookies to save information such as the user id, password, and account number. The cookies stored in the user machine to maintain information that allow the application to authenticate the

user identity, personalizes user content based on user identity, and speed up the transaction. The topic below will be describes in details about cookies.

2.2 Research Methodology

Research commonly refers to search for knowledge. Research is also known as an art of scientific investigation. There are many ways to gather the information and commonly the method used to get the information are literature searches, personal interview, reviewing the documentation, telephone survey, email survey, internet survey, and questionnaire. It is very important to gather the data in order to make sure that the application that will be developed is successful and accepted by the user [4].

Qualitative research is used as the research methodology in this project since it is commonly used to understand the meaning by describe and understand the experience, idea, belief, and value [6]. There are two methods that perform in qualitative research for this project which are literature search and website review.

Literature search much involve a material that is readily available such as article, documentation, report, tutorial, and book. It can be done either through Internet search or reading a book. In this project, the literature search is used to gather information about the details of cookies tampering attack. The details of the attack include the vulnerability that brings to the attack, ways to prevent the attack, impact of the attack, case study of the

attack and ways to do the attack. All material for this project is commonly from article, tutorial, and book. The material for this project is obtained from the Internet. Furthermore, most findings for this project research are come from the Internet search rather than reading a book.

The other research method that used is this project is website review. Website review is one method that can be used to gather the information in order to define the project requirement. Actually, website review is done through the review of the weakness in the website source code that potential for cookies tampering attack. This attack can easily be done if the website has a hidden form field that is not encrypted.

The hidden form field like figure 2.1 can easily be tampered because it is not encrypted. It is the weakness that is always found in the website that provides an online transaction. If the hidden form contains the price of the items, it may be easily tampered by an attacker. The problem can be solved by using an encryption mechanism by encrypting all the hidden form fields and compare the signature whether it is equal or not with the original signature when it is submitted. If the signature is not equal, the form has been tampered. Figure 2.2 shows codes for encrypted hidden form field.

```
<input type="hidden" name="userid" value="ktrout">
<input type="hidden" name="credit_ok" value="1">
<input type="hidden" name="form_expires" value="20001001:12:45:20">
```

Figure 2.1: The code of hidden form field that not encrypted

```
<input type="hidden" name="userid" value="ktrout">
<input type="hidden" name="credit_ok" value="1">
<input type="hidden" name="form_expires" value="20051001:12:45:20">
<input type="hidden" name="signature"
value="YJSG2/fXQRSsvLdDXJpjF/xLLYo">
```

Figure 2.2: The code of hidden form field that is encrypted

Based on the website review that had been done, the application that will be developed should be able to detect the hidden form field that is not encrypted in order to detect the vulnerability of cookies tampering attack. So, the code detection will use java language since the development of Android application uses java language.

2.3 Background of Cookies

Technically cookies are small pieces of text which is sent to the web browser by a server and is intended to be sent back to the server. It is unchangeable each time it accesses the same server or another server in the same domain. It is commonly used for authentication, tracking, maintaining state over stateless HTTP, as well as maintaining specific information about the user such as their site preferences. In addition, cookies are originally developed to be used in the online web applications such as e-commerce and online transaction. It also allows the content to be changed based on the user actions between browser sessions [1].

Cookies are also used to manage the user log in into website. When the user enter username and password at the login page, the authentication will be performed and after the user is authenticated, a cookies is saved and this allows the web site to know the user that has already logged in as the user navigate around the site. This permits the user to access any functionality that may be available only to the logged in user. Other than that, the cookie is used to save user preferences for a site. So, the site presentation and functionality can be personalized based the preferences of the user [1].

Then, cookies are used to track the user action across the site or domain. There are some third party cookies that allow for tracking across multiple sites. Most tracking is done within the site domain to gather usage data for that site. This tracking is often done by advertising company in order to build usage profile to allow more targeted marketing [1].

The cookies will only be destroyed when the user closed the browser unless a deletion date has been set. Once the deletion date has been set, the cookies will be destroyed on the particular date and these cookies are called persistent cookies [1].

2.4 Cookies Risk

Commonly, cookies is use to identify user in the web site. The cookie value is initially set by the site and stored on the user machine. Then, every time the user goes to the website, the browser will check the machine whether the cookie has been set for that domain or not. If the cookie has been set for that domain, the web browser will send the cookies data to the web server through the HTTP header. Several vulnerabilities can result from here.

First and most obvious is the storing of sensitive user information within the cookies itself. If another user has access to the client machine, the cookies data can be stolen. A simple scenario to consider is a web application that automatically logs user in based on user id stored in a cookie. If the ID information is compromised, an attacker may potentially impersonate that user and have access to this account.

In addition, the related issue is the tampering of the data stored in cookies. Consider a website logs the user in based on ID stored in the cookie. An attacker could easily iterate through several combination of possible user ID and potentially gain unauthorized access to other user accounts.

Another issue involved storing the price of items for an e-commerce site in cookies. The particular vendor would allow users to browse the site for items and then add selected items to virtual shopping cart by storing the item id and corresponding price on the

shopper's machine as cookies. The attack vector here is simple where the attacker may set an arbitrary price for any item that they want.

2.5 Types of Cookies Attack

2.5.1 Cookie Theft

Cookies are supposed to be sent only between web browser and web server in the same domain that the cookie is set. If the cookie is being sent over an ordinary HTTP connection, it is visible to anyone across the network to sniff the packet. It means, the cookies cannot contain any sensitive information. Sometimes, HTTPS is used to overcome this problem but that is not a solution because it only solves the problem associated with having sensitive data stored in cookies [1].

Cross-site scripting can be used in order to send cookies to the server that should not be receiving that data. Encryption does not help to stop this cookies theft which is often done using simple snippet of HTML posted to a site that the users can be tricked into clicking on the link which will sends the user cookies for that site to the location specified by an attacker. Because the request is coming from the same domain intended for the cookie, there are no problems. Then, the cookies can be exploited by connecting to the same site using the stolen cookies [1].

2.5.2 Cookie Poisoning

The cookie supposed to be sent back to the server without modification but attacker can modify the value of the cookie before sending it back to the server. This is typically done to carry out some sort of attack against the server that relates to some sort of data contained in the cookie.

If the cookie contains the price per item for something in the shopping basket, a change to this value in the cookie may cause the server to change the value into a lower price for that item. The process of modifying a cookie before it is sent back to the server is called cookie poisoning. Sometimes, cookie poisoning is used after cookie theft [1].

2.5.3 Cross-Site Cooking

In fact, each site supposed to have its own set of cookies and it cannot be altered or set for any other site. There are flaws that cause cross-site cooking vulnerabilities and allow malicious site to break this rule. This is similar to cookie poisoning, but instead attacking the site itself, the attacker will attack non-malicious user with vulnerable browsers [1].

Many browsers have a flaw in dealing with relaxed cookie domains. The browser is supposed to require a two-dot specification for all domains under top-level domain. This is supposed to prevent the setting of cookie for a subdomain like ".com". So, the actual

intent is by using two-dot name such as "mydomain.com". This break when it gets to the international naming system for some domain like ".com.au". In some browser, it is possible for a cookie to be set for the entire ".com.au" domain. Another problem is about how the browser deals with periods. There is typically no check to see whether there is anything between the periods or if there is trailing period being used to override the local domain search path. This means that a cookie can be set for ".com" to redirect to "http://www.mydomain.com". The address is not the real one. Probably many users do not care about trailing period and some seasoned user may not be adequately suspicious [1].

2.6 Anatomy of an Exploit

2.6.1 Cookie Theft

The attacker posts an auction that includes a link to what is advertised to be additional pictures or information about the object in the auction. When the user clicks to the link, the cookie for the auction website will be sent to the attacker server where the CGI script logs the information. Now, the attacker can look through the list of cookies and pick some of the most recent cookies to be used in order to login into the auction site and spoof the user [1].

2.6.2 Cookie Poisoning

The attacker visits an e-commerce site and adds an expensive item to his shopping cart. Then, the user examines the cookie stored on his system from that site to see whether the cookie includes the total cost of the items in the attacker's cart. The attacker will modify the cookie on his system to change the total to $5.00 and resave the cookie. Then, the user return to e-commerce site and check his cart to see the total is now $5.00 and proceed to order the item for the false price [1].

2.6.3 Cross-Site Cooking

The attacker crafts a cookie for domain ".com.uk" and set up a website to distribute the cookie. Then, the attacker will post the link to his web site on various bulletin boards or through emails. Once the user clicked on that link, they are given the attacker's crafted cookie that can overwrite or disrupt the real business they do with web sites in the international domain [1].

2.7 Real World Example

"Cookie theft vulnerability was reported in January 2005 in Froogle which is shopping search tool that launch by Google. Although the details reported are sketchy, it appears that malicious java-script in URL points to Froogle. Once the user clicks that link, a java-script executes a redirect to a malicious website which then steals the user's Google cookie. This stolen cookie apparently contains the username and password for the Google

Accounts centralized log in service information that is used by multiple Google service" [1].

2.8 Protecting Customer Information: Case Study

"In 2003, California passes an identity theft prevention law that, among other things, requires companies that do business with California citizens to disclose security breaches that might have resulted in loss of customer's confidential information. Vendors that fail take reasonable precautions to protect confidential information, including credit card numbers and other personally identifiable information, from access by unauthorized persons can be subject to fines, lawsuit, and loss of business" [2].

"In April of 2004, the State of New York fined New York City-based *BarnesandNoble.com* $60,000 for exposing customer data through a flaw in their web site. Personal information (but not credit card numbers) could be accessed, and purchased could be made against another person's account" [2].

Based on the above cases, the vendors should take a good responsibility in order to protect their customer information from being stolen or manipulate by third party. So, the vendor should take a responsibility to increase the security mechanism in order for them to provide an online transaction especially e-commerce transaction.

2.9 Cookies Tampering Attack

Cookies is small piece of text that is send to the web browser by the server and is intended to be send back to the server. Cookies tampering attack occur during the intent of web browser to send back the cookies to the server. The attack is possible by using the tools like "Tamper Data" to modify the content in the cookies. Tamper Data is actually a Mozilla Firefox Extension that gives user the power to views, record and modify ongoing HTTP request. In addition, Tamper Data is commonly uses to view and modify HTTP or HTTPS header and POST parameter [5].

Mozilla Firefox and Tamper Data are needed in doing cookies tampering attack. The Tamper Data will be enabled during the web browser intense to POST the data to the server. During this stage, cookies will be tampered. The steps below show how the cookies is being tampered using Tamper Data. Figure 2.3 show the data in cookies before it is being tampered. Figure 2.4 show the data in cookies after it is being tampered. The data that is being tampered in Figure 2.4 is "userid". It had been tampered from "ktrout" to "anonymous". Actually, the "userid" that is tampered is hidden form field that hold the value "ktrout". Figure 2.5 shows the result when the data is submitted to the server. The result has printing the "userid". Now the "userid" that printed is change from "ktrout" to "anonymous" after it is being tampered. That is how cookies tampering attack is done using Tamper Data.

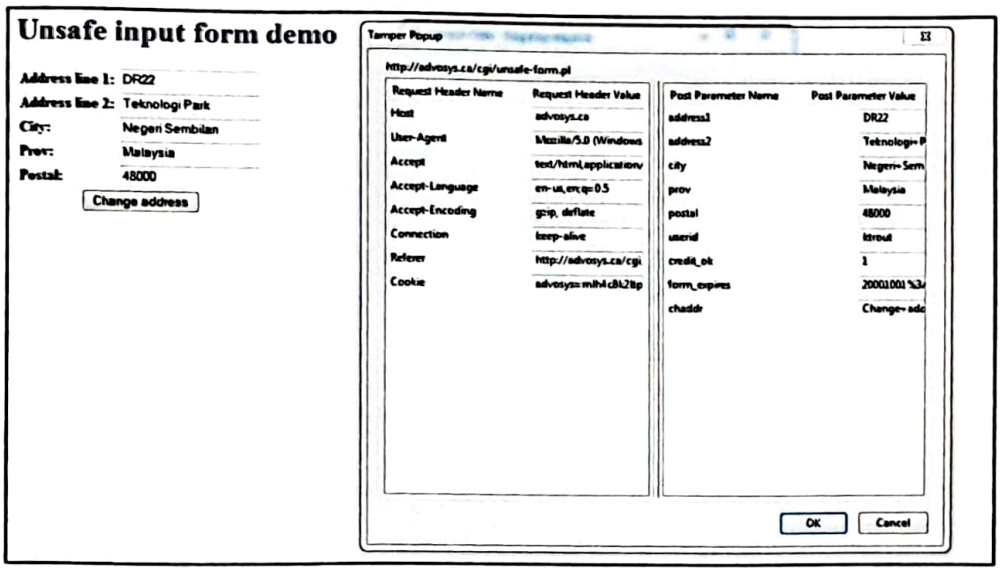

Figure 2.3: The POST data value before modified

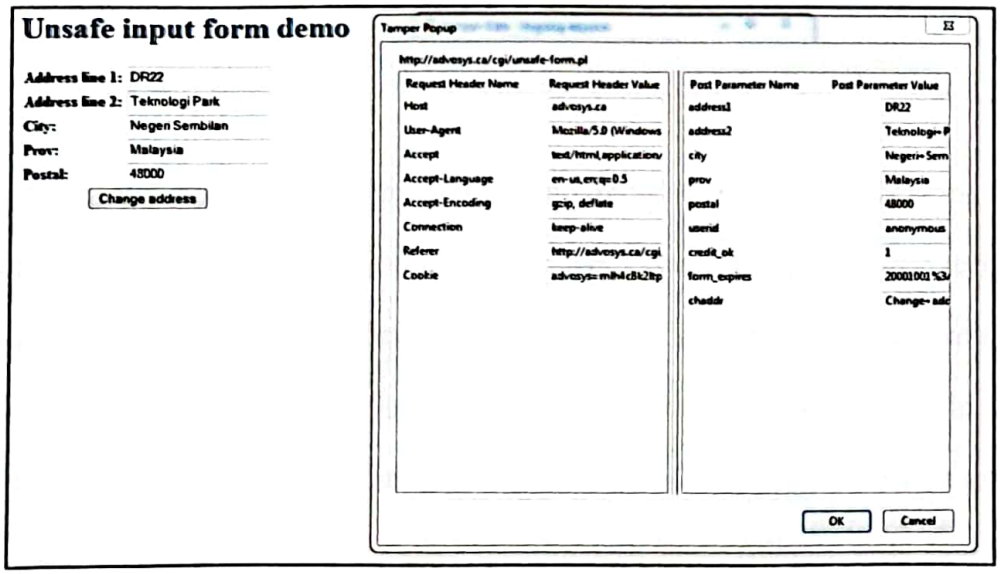

Figure 2.4: The POST data value after being modified

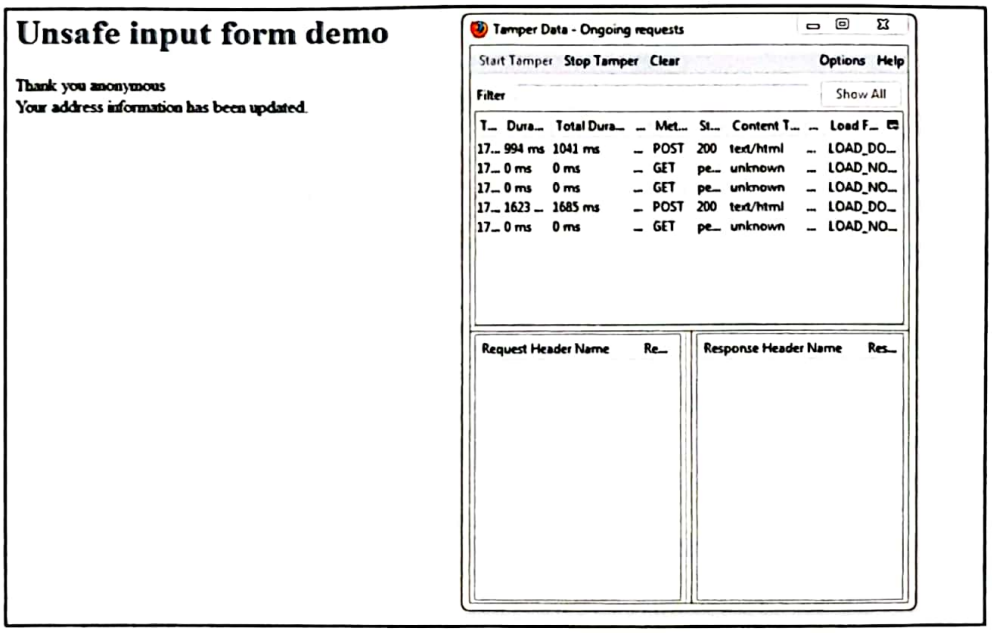

Figure 2.5: The outcome of the attack

2.10 Android Application

Android is one of a mobile platform. It is a Linux based mobile phone operating system developed by Google. Android is unique because Google is actively developing the platform but giving it away for free to hardware manufacturer and phone carriers who want to use Android on their devices. An Android modified version is used in Google TV, Samsung Galaxy Tab and countless other devices. Google had formed a group of technology and mobile company called the Open Handset Alliance with the goal of contributing to Android development. Most of the members have the goal of making money from Android, either by selling phones, phone service, or mobile applications. Anyone can download Software Development Kit (SDK) in order to write an application for Android phone. These applications can be downloaded from the Android Market [3].

2.11 Software Development Methodology

Software development methodology can be considered as a process of creating software much on organizational aspects. It is referring to the documented collection of guidelines, procedure and standard to ensure the quality of application development in order to meet user requirement and expectation in an efficient manner. The development methodology has listed a series of operations and procedures that need to be followed during the application development. M.Marakas explained that methodologies are approaches that are followed and it contains several steps that guide and organize the application development [22]. A good methodology needed to ensure the project can be developed smoothly until the end of the project with less risk and mistakes.

There are several types of software development methodologies that are suitable to the Information Technology (IT) project. The methodologies include Waterfall Model, Prototyping, Rapid Application Development (RAD), Iterative and Incremental Development Model, and Spiral Model.

2.11.1 Waterfall Model

The waterfall model is one of the popular methodologies that are used in software development. This methodology consists of several phases and each stage needs to be completed before proceeding to the next stage. This model is a type of linear software development methodology. Basically waterfall model will divide the phases into analysis,

design, implementation, testing, integration, and management and maintenance. During analysis phase, the research is being conducted and it is include with brainstorming about the project and the purpose of the project to be fulfilled. The next phase is the design phase. The design phase includes the basic of software design on the paper. When the basic design is approved, more elaborated technical design can be planned. The next phase is implementation phase where the source code of the program is written. Then it moves to the testing phase. During testing, the whole design and construction is put under a test to check the functionality. If there are any errors, then they will surface at this point of process. After the system has successfully tested, it will move to the next stage which is integration phase. During this phase, the company will put the system in use. The last stage in waterfall is management and maintenance phase. This phase is needed in order to ensure that the system will continue to perform as desired [7].

The sequence of development will ensure the adequacy of documentation and design in order to increase the quality, maintainability, and reliability when developing an application or software (Royce 1970) [8].

2.11.2 Prototyping

Prototyping model is one of the popular software development methodologies after waterfall model. This model is made first based on the final product. It is a model which is not based on strict planning but it is an early approximation of the final product in

developing an application or software. Actually, a prototype is a sample to test the process and the better final product will be built based on the sample. This type of development is employed when it is very difficult to obtain the exact requirement for the user. From time to time, the user will keep giving feedback based on the prototype that had been made. The complete application or software will be developed based on the given feedback from the user. After that, the System Requirement Specification (SRS) should be prepared and after the completion of this, the more accurate SRS should be provided and the development work can be started [10].

2.11.3 Spiral Model

The spiral model is one of the system development methodologies. In order to combine the advantage of top-down and bottom-up concept, the spiral model has used the combination of prototype and design elements. Moreover, the spiral model is a system design method that is widely used in Information Technology (IT). It combines the features of two methodologies which are waterfall model and prototyping model. This methodology is suitable for large, expensive and complicated project [11].

2.11.4 Rapid Application Development (RAD)

Rapid Application Development (RAD) is an application development technique that uses prototypes, iterative customization, and Computer Aided Software Engineering (CASE) tools. It is a software development methodology which gives a focus on building an

application in a short period of time by compromising on the usability, features, and execution speed. Generally, by using this methodology, the application development and design would be done within 60 to 90 days [12].

RAD is used to deliver a qualified software development within relatively low investment cost and usually provides ability to be changed according to the user demand. This approach may involve a compromise in functionality and performance in order to enable a faster application development and facilitate application maintenance (*James Martin, 1991*) [13].

2.11.5 Iterative and Incremental Development Model

Iterative and Incremental Development Model is also known as Phased Development Model. It is an Agile Software Development method which combines iterative and incremental technique to deliver functionality early. Iterative development refactors code repeatedly and making a progress through successive refinement. A developer may repeatedly perform a little modeling, coding, testing, and integration by using test-driven development. Incremental development builds and delivers software to a production environment within a series of a small and regular release with expanding functionality. Through this development, the application will be delivered in incremental release over the time. Each of the release will have an added new functionality from the previous release. In addition, Iterative and Incremental Development was developed in response to

the weakness of the waterfall model when it starts with initial planning and ends with deployment with cyclic iteration in between [14].

2.12 Development Tools and Software

Android is an open source mobile platform and it is linux-based, multiprocess, and multithread operating system. Nowadays, android operating system is widely used in mobile phones, tablet PCs, and other mobile devices. Android application becomes one of the popular and demanding applications to the users due to rapidly growing of the technology. That is why android application development becomes one of the most popular project in expanding the use of android operating system. There are several development tools and software required in order to build an Android application such as Java Development Kit (JDK), Android Software Development Kit (SDK), Integrated Development Environment (IDE), and Android Emulator.

2.12.1 Java Development Kit (JDK)

Java Development Kit (JDK) is the program development environment that is used for writing java applets and application. The JDK is developed by Sun Microsystem's JavaSoft division. The JDK is required since an Android application development uses java as the programming language [15].

2.12.2 Android Software Development Kit (SDK)

Android is a software stack for mobile device which includes an operating system, middleware and key applications. Then, the android SDK provides the tools and library that necessary to the beginning for developing an application that runs on the android devices [16]. The android SDK is composed of modular package that can be downloaded separately using Android SDK Manager [17].

2.12.3 Integrated Development Environment (IDE)

Integrated Development Environment (IDE) is a software application that consists of a code editor, a compiler, a debugger and graphical user interface (GUI) builder. The IDE provides comprehensive facilities to the computer programmer for software development such as developing an Android application. The examples of IDE are Eclipse, IntelliJ IDEA, NetBeans, and JBuilder [9].

2.12.3.1 Eclipse

Eclipse is multi-language software development environment which compromising an Integrated Development Environment (IDE) and extensible plug-in system. It can be used to create diverse end to end computing solution for multiple execution environments. Mostly, Eclipse is written in java and it also can be used to develop application in java and other programming language including Ada, C, C++, COBOL, Perl, PHP, Phyton, R, Ruby, Haskell, Groovy, Scala, Clojure, and Scheme. The development environments

include the Eclipse Java Development Tools (JDT) for Java, Eclipse CDT for C or C++, and Eclipse PDT for PHP and among others [18].

2.12.3.2 IntelliJ IDEA

IntelliJ IDEA is a code-centric Integrated Development Environment (IDE) which focus on developer productivity. It is one of the commercial Java IDE which produced by JetBrains. The example of language that supported by IntelliJ IDEA includes Java, JavaScript, XML, HTML, and CoffeeScript. Furthermore, Intellij IDEA is intelligent code assistance which include with multiple productivity-boosting features like smart code completion and on the fly code analysis [23].

2.12.3.3 NetBeans

NetBeans is a free, open-source of Integrated Development Environment (IDE) for software developers. It is the tools that can be used to develop mobile application, enterprise web, and create professional desktop either by using java platform, C, C++, PHP, JavaScript, and Groovy. The newest version of NetBeans provides significant improvement with new static code analysis capabilities in Java Editor and smarter project scanning. NetBeans can be run on Windows, Linux, Mac OS X and Solaris [24].

2.12.3.4 JBuilder

JBuilder is one of the Java Integrated Development Environment (IDE) which available with support for the leading commercial and open source Java application server. It is include with the ability to profile a web application, enable Java developers to deliver high performance and scalable application. Furthermore, it is part of Embarcadero family of software development tools for Java, NET, Windows, Mac, web, and mobile including RAD Studio, C++ Builder, and Rad PHP [25].

2.12.4 Android Emulator

Android emulator is a virtual mobile device that runs on the computer. It helps an android application developer to test the Android application without using a physical device. Once the Android emulator has been installed in the PC, all Android application could be run on the PC like in the Android device. Generally, an Android emulator is included within the Android SDK [19].

2.13 Similar Applications

Since the web application is becoming more crucial to the world, the growth of web security issues is widely increasing. Actually, web security is very important because the website contains the relevant information about the company or organization. Sometimes, the information is a secret and is only published to the respective staffs. Nowadays, website defacement is very common. The most common website vulnerabilities are SQL-

injection, cross-site scripting, and cookies tampering which lead towards the defacement. So, it is important to do a vulnerability scanning on the website in order to find the vulnerability and fix it before it is too late. Actually, there are many tools available for scanning the vulnerability in the website such as Open Web Application Security Project (OWASP) – Zed Attack Proxy (ZAP), Nikto vulnerability scanner, Netsparker web application security scanner, and Skipfish web vulnerability scanner [20].

2.13.1 Open Web Application Security Project (OWASP) – ZAP

OWASP is a non-profit organization which is focusing on improving the security of web application. Zed Attack Proxy (ZAP) is an easy to use integrated penetration testing tool for finding the vulnerabilities in web applications. This tool has an automatic scanning functionality and it has a set of tools that allow user to find vulnerability manually. ZAP also provides a basic port scanner to shows which ports are open on the target sites. It also allows user to see all of the requests and responses that are made to a web application [20]. The screenshot in Figure 2.6 shows ZAP interface.

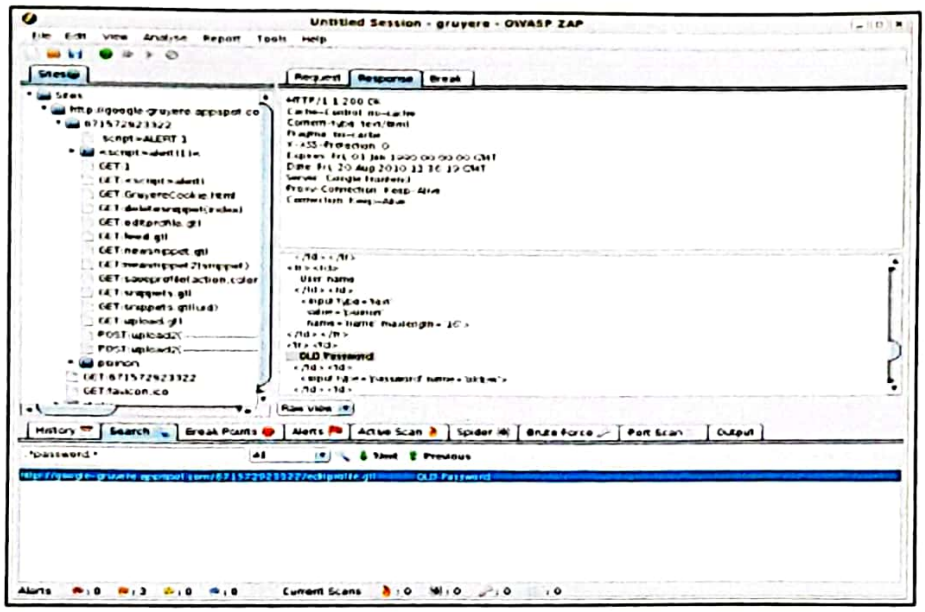

Figure 2.6: Zed Attack Proxy Interface

2.13.2 Nikto Vulnerability Scanner

Nikto is open source software that has been used by a large community to find the vulnerability on a web application. It is used to check a potential dangerous CGI files in the web server, check outdated version of the server and specific problem on the server version, check the plug in and misconfiguration files, and find out the insecure files and program [20].

2.13.3 Netsparker Web Application Security Scanner

Netsparker is a web application security scanner on windows platform. It will firstly crawl the website and then attack on each and every link to find out the vulnerabilities

regardless of the platform of the website. This tool is able to find different vulnerabilities including SQL-injection, cross-site scripting, local file inclusion and remote code execution [20]. The screenshot in Figure 2.7 shows Netsparker interface.

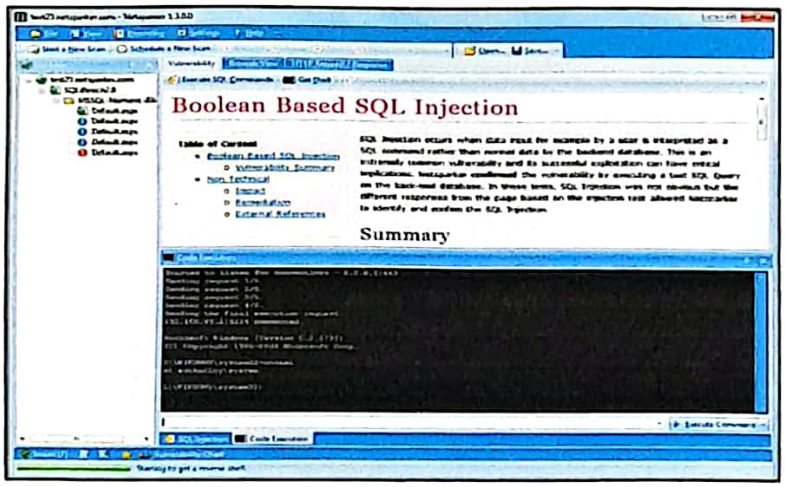

Figure 2.7: Netsparker

2.13.4 Skipfish Web Vulnerability Scanner

Skipfish is an automatic web application security tool that has been designed to find the vulnerabilities on a web application. This tool can be used to find the vulnerability on the critical website before the website is exploited by the hacker. Skipfish is applicable on multiple platform including Linux, BSD, MAC and Windows. It is a powerful scanner tool that crawls targeted website and fully scans all the pages. Skipfish is available on backtrack 5 [20]. The screenshot in Figure 2.8 shows Skipfish interface on backtrack 5.

Figure 2.8: Skipfish on the backtrack 5

2.13.5 Veracode

Veracode is an accurate vulnerability scanning tool for industry use. It combines three different testing methodologies which are static analysis, dynamic analysis, and manual penetration testing for comprehensive software application and Web vulnerability scanning test. Veracode static analysis provides an innovative and highly accurate testing technique called binary analysis where most vulnerability scanning tools look at application source code. Veracode actually scans the binary code rather scanning the source code which is often ineffective since the source code may be unavailable for practical or proprietary reasons. Scanning binary code allows the enterprise to review an entire application, delivering a far more accurate and comprehensive analysis [21]. The screenshot in Figure 2.9 shows Veracode interface.

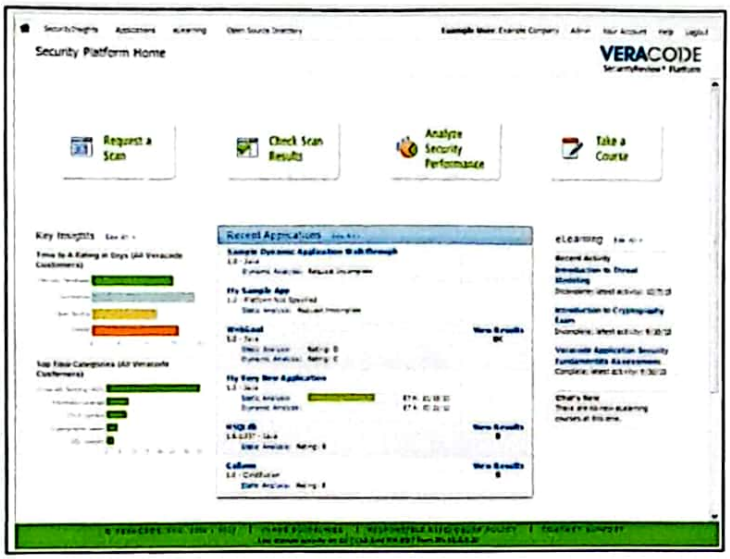

Figure 2.9: Veracode interface

CHAPTER 3

ANALYSIS

3.1 Introduction

This chapter will discuss in detail about the methodology that had been chosen, the observation about the importance of this project, and the impact of this project to solve the problem mentioned in the problem statement in chapter one. In addition, this chapter will mention in detail about the project functionality based on the research that had been done and the tool and software that will be used in order to develop this project.

3.2 System Development Methodology

System development methodology is the most important part in order to develop a project or new application. It refers to a guideline of developing a project from the beginning until the end. However, there are different types of methodology that could be chosen and used. The choice of methodology to be used depends on the software to be developed.

The finding of the research shows that the most suitable methodology for this project is prototyping because this methodology is based on an early approximation of a final product. Moreover, this project is about developing a new application which requires a guideline that is not based on strict planning such as in the waterfall model. In addition, the application to be developed lacks a detail requirement and the prototyping methodology works well in this situation.

Prototyping is one of the software development methodologies that reduce the time and the cost of the project. In addition, prototyping methodology is very suitable for small project that has limited time to be completed because this methodology can reduce the time needed to complete the project since there is more of a focus on building a project rather than defining the requirement and scope at the beginning of the project. That is the main reason of choosing prototyping methodology in this project because this project is about developing a small application within limited period of time.

There are several steps in applying prototyping methodology in this project. The first step in prototyping methodology is defining the application requirement. The application requirement in this project is determined through observation and case study that had been made in the current situation where there are many cases involving cookies tampering attack which cause the company offering an online transaction to lose a lot of money in their business. So, the weakness that may cause cookies tampering attack is

identified in this stage in order to build an application that might be able to detect the vulnerability in the website.

The next stage is to build preliminary design of an application. After defining the application requirement, the preliminary design of the application is created. The application design is created based on the requirement that has been identified. In this project, the preliminary design will create an application that would be able to scan the hidden form field in the website that is not encrypted in order to determine the vulnerability for cookies tampering attack.

After that, the first prototype of the application is constructed based on the preliminary design created in the previous stage. This prototype usually is a scaled-down application which represents an approximation of the characteristics of the final product. Then, the project supervisor will evaluate the first prototype after it is completed. Based on the comment given by the project supervisor from the first evaluation, the second prototype is constructed to improve the application that created. After that, the second prototype will be evaluated by the project supervisor in the same manner as the first prototype. The preceding steps will be iterated as many times as necessary until the project supervisor is satisfied with the prototype that created. Then, the application is constructed based on the final prototype. Then, the application will be thoroughly evaluated and tested.

There are four prototypes that will be created in this project in order to meet the final product. The first prototype will create an android application project that would able to connect to the web server when URL is entered. The second prototype would be an improved version of the first prototype by creating an application that would be able to connect to the web server and read the website source code. The next prototype will create an application that will be able to perform vulnerability scanning on the website source code that read and detect cookies tampering vulnerability in the website source code. The last prototype will create an application that has user friendly interface within the functionality to detect vulnerability in the website for cookies tampering attack.

3.3 Application Functionality

Application functionality determines the functionality of the application that will be developed in this project. The application that will be developed has functionality to connect to the web server, read the website source code from the web server, scan and detect the hidden form field in the website that is not encrypted and lastly, the application will prompt the warning message when the system found the vulnerability in the website.

3.3.1 Connect to the Web Server

The application interface will require the user to input the URL of the website that will be scanned. After the URL is entered and submit, the application will be connect to the web server in order for the application to perform the next functionality.

3.3.2 Read the Website Source Code from the Web Server

The next functionality is read the website source code and put the source code into the buffer reader that created. The source code of the website will be read and tokenize line by line. The source code will be scanned in the next function.

3.3.3 Scan and Detect the Hidden Form Field

The next functionality is scanning and detecting the hidden form field that is not encrypted. The scanning will be performed automatically when the user input and submit the URL of the website. The result of the scanning will determine whether the website is vulnerable or not for cookies tampering attack.

3.3.4 Warning Message

The last functionality is to prompt the warning message when the result of the scanning found that the hidden form field is not encrypted. The application will prompt the warning message "This website is vulnerable to the cookies tampering attack!!!" when the result shows the hidden form field is not encrypted. When the result shows the hidden form field is encrypted, the application will prompt the message "This website is contains encryption mechanism that made it safe from cookies tampering attack". The last one is when the result shows the hidden form field does not exist in the website, and then the application will prompt the message "The website is safe from cookies tampering attack".

3.4 Development Tools and Software

There are several tools that had been decided to be used in order to develop the project. The development tools and software that had been decided to be used in this project include Java Development Kit (JDK), Android Software Development Kit (SDK), Eclipse, and build in Android emulator in Android SDK. The combination of this tools and software is compatible and is always used in order to develop an Android application. Furthermore, there are many support of using these tools and software in order to develop an Android application. It makes things easier in order to develop an Android application since there are many supports in the Internet. In addition, all of the tools and software is open source which is free to be used. That is the main reason of choosing JDK, Android SDK, Eclipse, and build in android emulator in Android SDK as the development tools and software for this project.

The first software that should be installed before developing an Android application is Java Development Kit (JDK). JDK is developed by Sun Microsystem and now owned by Oracle Corporation. It used for writing java applets and application. Commonly, JDK needs to be installed in writing the java programming.

The next software that needs to be installed is Android Software Development Kit (SDK). Android SDK is used to provide the tools and library that is necessary in writing the

program for Android application that runs on Android devices. This is the most importance software in writing Android applications.

Then, Integrated Development Environment (IDE) which is Eclipse is the next important software that needs to be installed in order to write an Android program. Eclipse is a multi-language Software Development Environment. Most of the time Android application development will use Eclipse as the software development environment. Moreover, Eclipse interface is easy to use and understand while in term of support Eclipse is much better than other Software Development Environment like Netbeans, JBuilder and Intellij IDEA.

Lastly, the tool like Android emulator is very important in order to work as virtual mobile device that runs on a computer. Actually, there is built-in Android emulator in the Android SDK which needs to be configured before it is able to be used. In this project, the built-in Android emulator will be configured and used in order to test the application that will be developed. By using Android emulator, all the testing procedures will be less complicated. After the application runs smoothly in the Android emulator, it will be tested in the real Android devices. So, the testing phase will be much faster and efficient.

CHAPTER 4

DESIGN

4.1 Introduction

This chapter will briefly explain about the application that will be developed, flow chart which determines how the application operated, application structure which determines the whole application, and interface design of the application. This chapter will be more specific towards the application that will be developed in order to make the project clearly defined.

4.2 Explanation of the Proposed Application

"Cookies Tampering Vulnerability Scanner" is an Android application that would be able to scan and detect cookies tampering vulnerability in the website. This application will perform a scanning on the website source code in order to find the hidden form field that is not encrypted that may bring the website towards cookies tampering attack. The

application will prompt the warning message once it has detected the hidden form field that is not encrypted. If the application detects encrypted code for the hidden form field, it will prompt a message that will tell user, the website is safe from cookies tampering attack.

4.3 Application Flowchart

The flowchart is the best way to indicate how the application operates. By using the flowchart, application process will be clearly defined. Figure 4.1 shows the flowchart for "Cookies Tampering Vulnerability Scanner".

The flowchart of the application shows the operations of the application. The operation starts when the user input the URL of the website to be scanned. When the user click on submit button the application will start perform the operation. Firstly, the application will connect to the web server. If the connection is failure, the application will prompt the error message that indicates the connection is fail. When the connection is successful, the application will read the website source code and place it into the buffer reader. Then, the code will be tokenize by space before perform the scanning to detect the vulnerable code. When the source code is vulnerable to the cookies tampering attack, the application will prompt the warning message that indicates the website is vulnerable to the cookies tampering attack. If the source code is not vulnerable, the application will prompt the positive message. The user will end the operation by close the application.

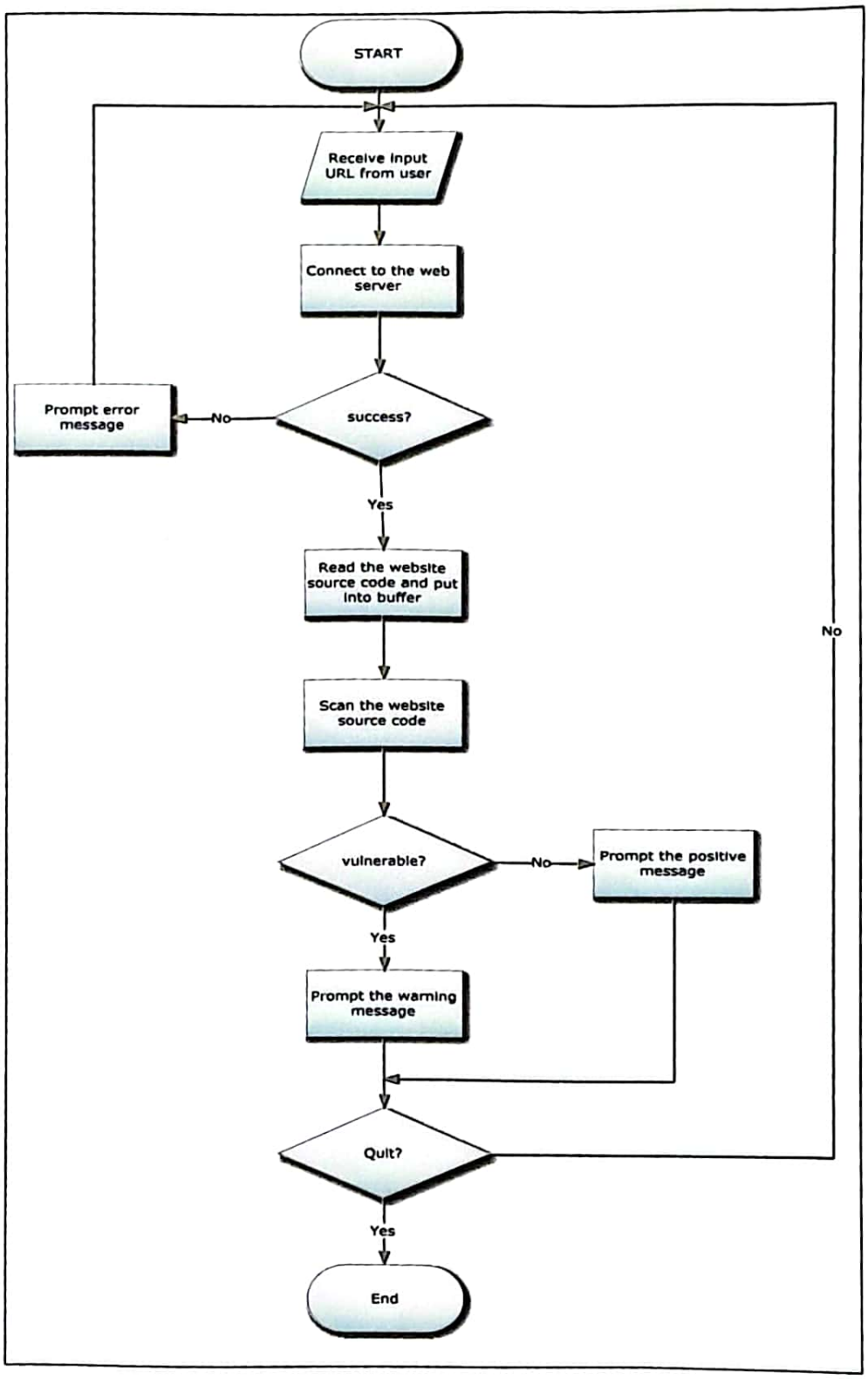

Figure 4.1: Flowchart for cookies tampering vulnerability scanner

There are three major operations in cookies tampering vulnerability scanner which are connect operation, scanning operation, and detect operation. The scanning operation will be combines together with the read operation. Figure 4.2 shows the flowchart for connect operation in cookies tampering vulnerability scanner while Figure 4.3 shows the flowchart for scanning operation in cookies tampering vulnerability scanner. Then, Figure 4.4 shows the flowchart for detecting vulnerable code and prompting an alert to the user. All three major operations that show in the entire flowchart measure the complete operation in order to make sure full functionality of the application to detect cookies tampering vulnerability in the website.

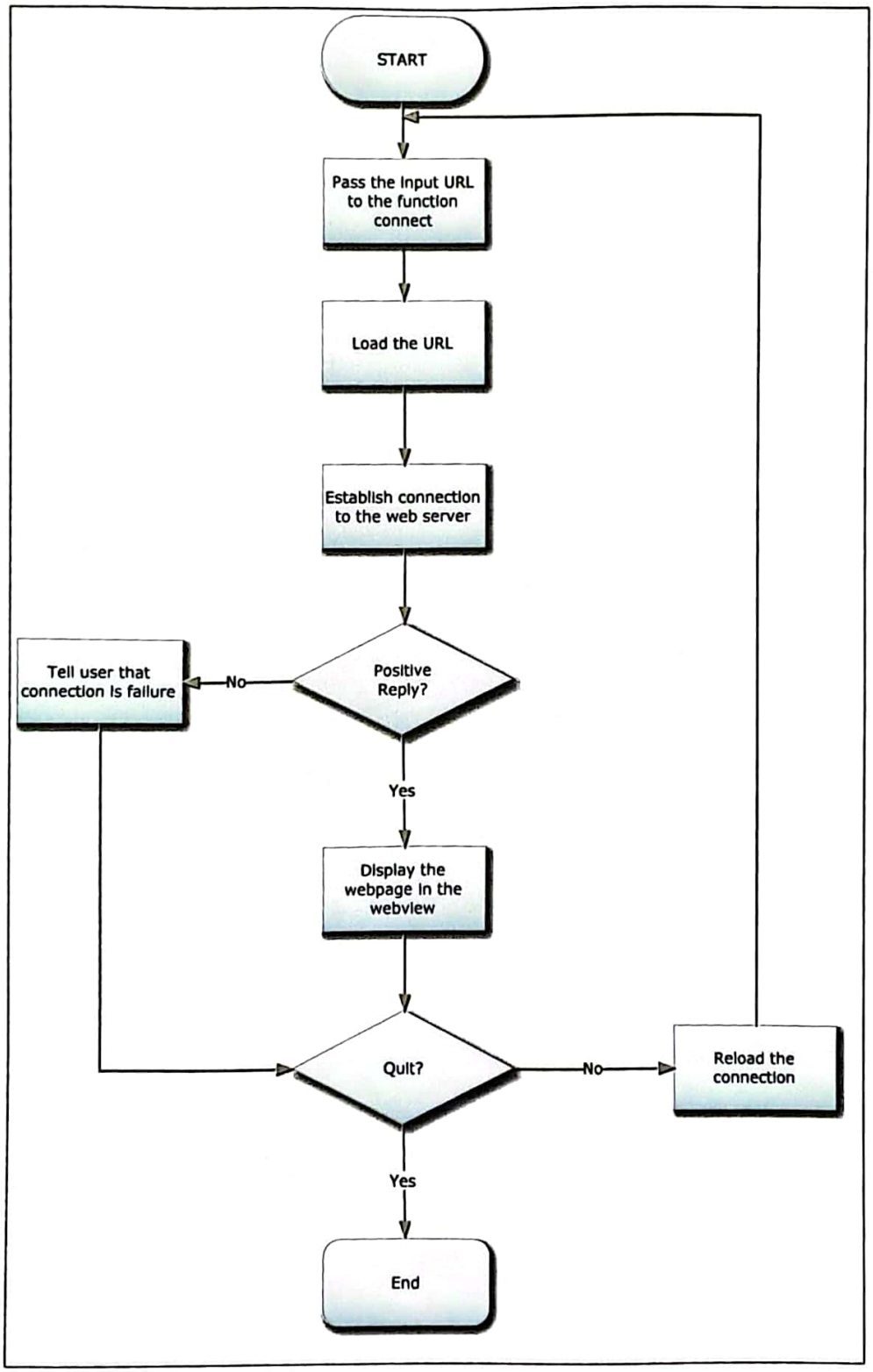

Figure 4.2: Flowchart for connect operation

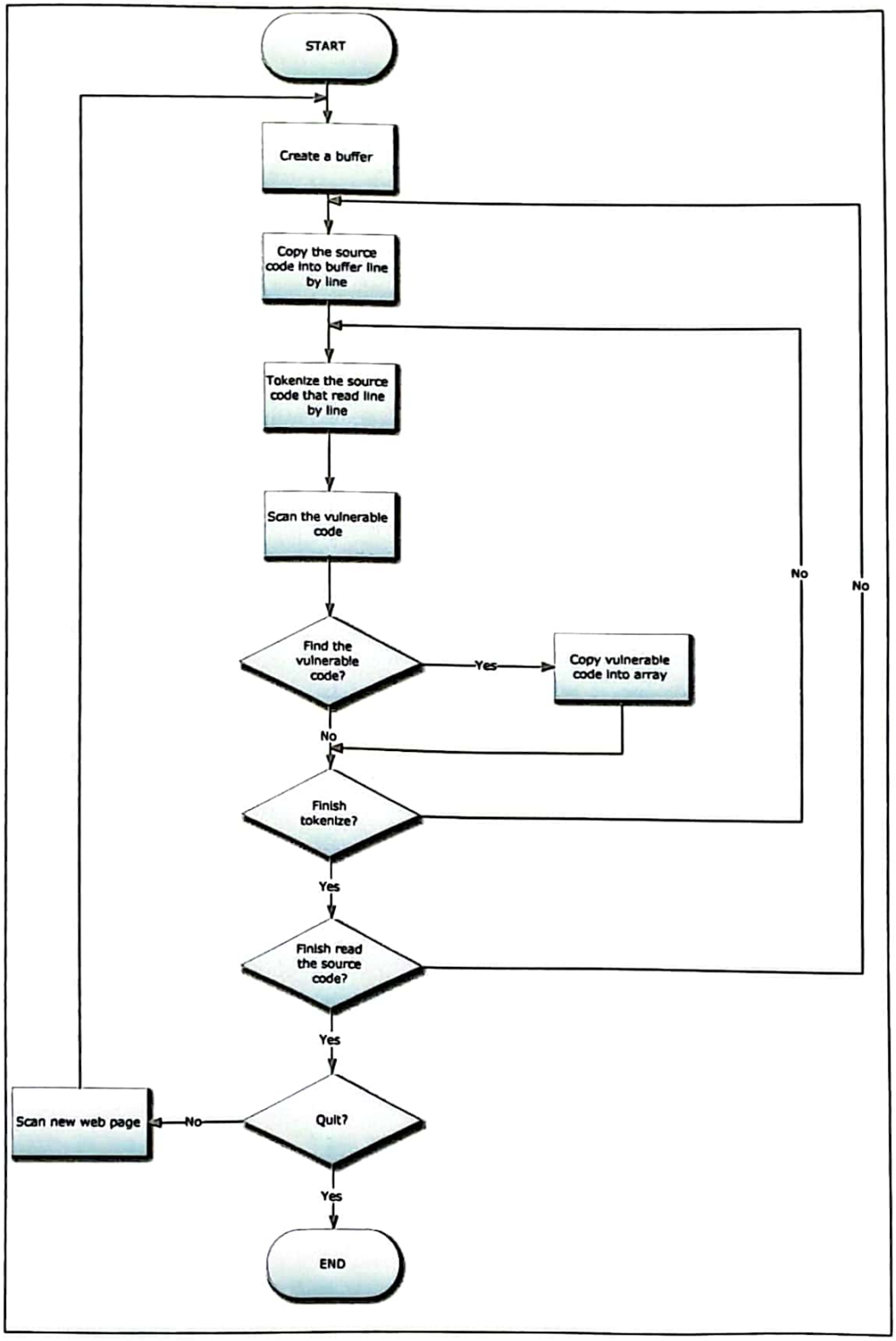

Figure 4.3: Flowchart for scanning and read operation

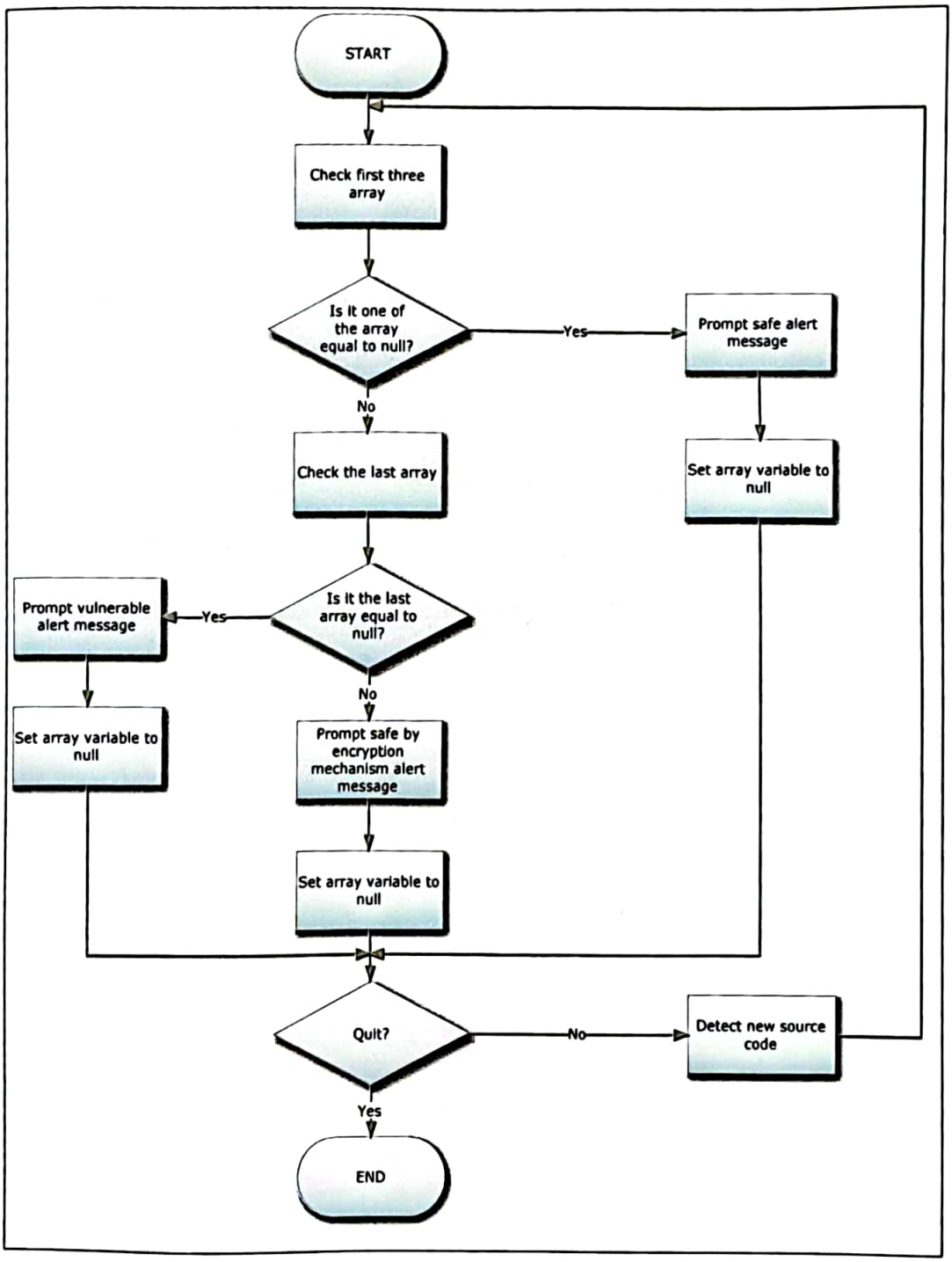

Figure 4.4: Flowchart for detect operation

4.4 Application Structure Chart

This section will show the structure of the application that will be developed. The structure will give a clear view of the whole application in order for it to perform its function. "Cookies Tampering Vulnerability Scanner" has three major separate functions in its structure. The functions include the main function, main XML function and control permission function. Main function is divided into four separate functions which are function to connect to the web server, function to read the source code from the web server, function to scan the source code and function to detect the source code. User message will be handled by function to detect the source code while error message will be handled by function to connect to the web server. Figure 4.5 shows the structure chart for "Cookies Tampering Vulnerability Scanner".

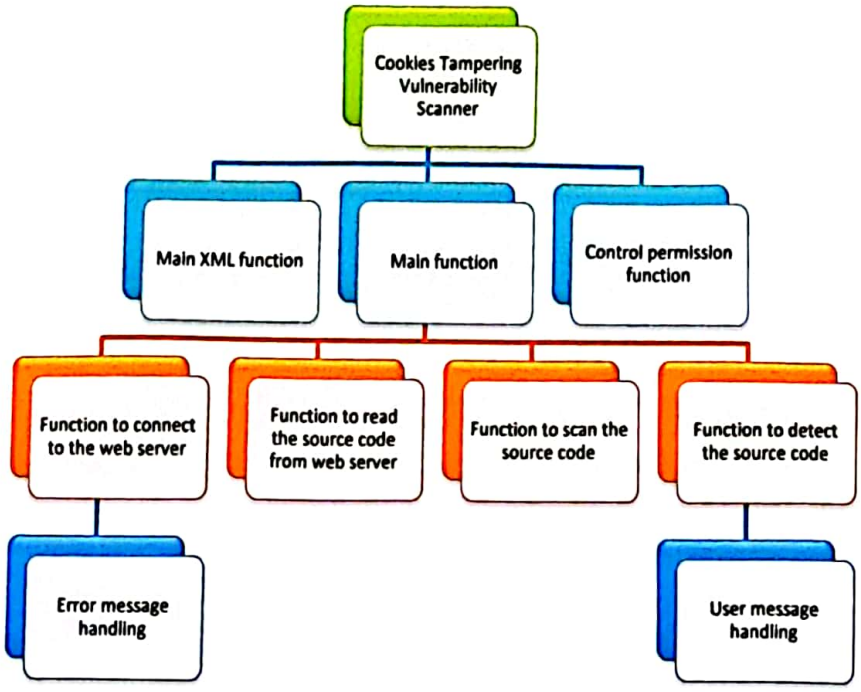

Figure 4.5: Structure chart for cookies tampering vulnerability scanner

4.5 Interface Design

The interface design will show the design of the application interface. The interface design is done based on the determination of the application functionality. This section presents the interface design for the main page of the application, the error message that will appear during the failure of connection to the website, alert message that will appear after detecting the vulnerability, alert message that will appear after detecting the encryption mechanism in the hidden form field and the alert message that will appear when the vulnerability code is not found in the website.

Figure 4.6 shows the design of the main page for the application while the screenshot for the next figures which are Figure 4.7, Figure 4.8, Figure 4.9 and Figure 4.10 indicate the design of the error message when the connection failure, the alert message when vulnerability is detected, the alert message when encryption mechanism is detected, and the alert message when the vulnerability is not found.

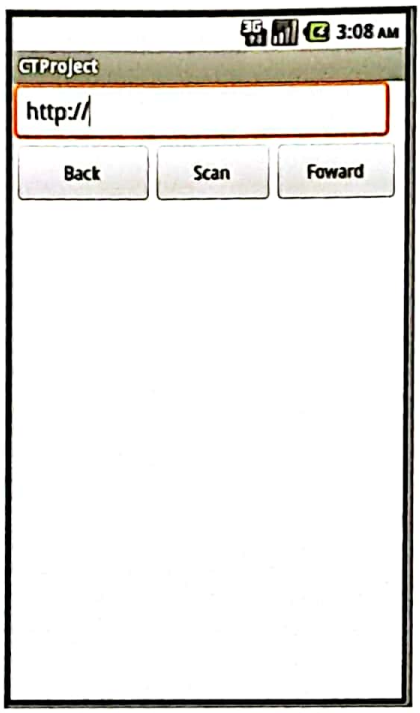

Figure 4.6: Interface design for main page

Figure 4.7: Error message for connection failure

Figure 4.8: Alert message after detecting the vulnerability

Figure 4.9: Alert message after detecting encryption mechanism in the hidden form field

Figure 4.10: Alert message after vulnerable code is not found in the website

CHAPTER 5

PROTOTYPE

5.1 Introduction

Prototyping is one of the Software Development Methodology based on early approximation of the final product in developing an application or software. First prototype need to be developed during project 1 since this project is using prototyping as Software Development Methodology. The next prototype will be developed during project 2 until meet the final product.

There are four prototypes that had been developed in this project in order to meet the final product. Every prototype will be presented to the project supervisor. Then, the project supervisor will give a comment and idea for developing the next prototype. This procedure will be continues on every prototype that developed.

In this project, there are two different prototypes that had been developed which are application prototype and website prototype that had been used for cookies tampering attack demonstration. All the prototypes that had been developed will be explained in deep on the topic below.

5.2 First Prototype of an Android Application

The first prototype is the basic Android application that is able to receive the user input and perform an operation to connect to a web server. The first prototype is like a web browser for Android application. Once the user opens the application, it will automatically load to the UNITEN website. After that, the user can enter the preferred website URL to be visited. Then, the application will load the URL and display the webpage of the website.

The second prototype will be developed based on this first prototype. The screenshot below indicate the first prototype application that had been developed during the project 1. The screenshot is capture from Android emulator which used to test the application that develops.

Figure 5.1: First Prototype CookiesProject

Figure 5.1 shows the CookiesProject had been successfully installed in Android emulator. The application is ready to be open by the user.

Figure 5.2: Automatically load to UNITEN website

Figure 5.2 shows that the application will automatically load into UNITEN website when the user opens the application.

Figure 5.3: Load into Yahoo website

Figure 5.3 shows that the application will be loaded into Yahoo website when the user enters URL of the Yahoo website. So, the first prototype is now able to connect to the web server.

5.3 Second Prototype of an Android Application

This prototype is an extended from the first prototype which developed during project 1. The second prototype is an application that able to read the website source code and place

it into the buffer reader. After that it will display the website in the WebView and print the website source code in the TextView. This prototype only perform basic operation like scan, read and display in order to detect the vulnerable code for cookies tampering attack. The detection of vulnerable code will be included in the next prototype.

The screenshot below indicate the second prototype that had been developed during project 2. Figure 5.4 show that the application is load into Myeg website and display the website source code while Figure 5.5 show that the application is load into GSC website and display the website source code.

Figure 5.4: Load into Myeg website

Figure 5.5: Load into GSC website

5.4 Third Prototype of an Android Application

The third prototype is an extended prototype from the second prototype. In the third prototype, the detection mechanism has been included in order to detect the vulnerable source code in the website. However, this prototype only can detect the vulnerable source code in the website on the first page that user visit. When the user clicks to the link in the website, this application could not be able to scan and detect the vulnerable source code automatically.

Figure 5.6 below shows the screenshot when the application from the third prototype scan GSC website and prompt the vulnerability message while Figure 5.7 shows the screenshot of the application which scan the Google website and prompt positive (safe) alert message. Lastly, Figure 5.8 shows the screenshot of the application which prompt an error message when the user enters wrong URL.

Figure 5.6: Scan GSC website and prompt an alert message

Figure 5.7: Scan Google website and prompt an alert message

Figure 5.8: The application prompts an error message

5.5 Fourth Prototype of an Android Application

The fourth prototype is the last prototype of the application in this project. In this prototype, the application is enhanced from the third prototype to scan and detect the vulnerable source code when the user visit to the website and click the link in the website. When the user click on the link in the website, the application will automatically perform a scanning and detect the vulnerable source code in the website. Then, the interface has been changed to the simple interface and the application will not automatically load into UNITEN website when the user opens to the application.

Figure 5.9 shows the screenshot of the application which scan the first page of Myeg website while Figure 5.10 shows the application which scan another page in Myeg website when the user click the link of "Contact Us" in the website.

Figure 5.9: Load into first page of Myeg website

Figure 5.10: Load into "Contact Us" webpage in Myeg website

5.6 Website Prototype

The website prototype in this project is used for attack demonstration purpose. It had been created using PHP and MYSQL database. Furthermore, it is e-commerce website for mobile shopping. The website created actually is unsecure e-commerce website and it can be easily tampered using tamper data add-on. Below is the screenshot of the website prototype that had been created and how the cookies are being tampered in this prototype.

Figure 5.11 shows the main page of the website prototype which sell the mobile phone while Figure 5.12 shows the login page of the registered user. When the user has login into the website, the page will be redirected to mobile shopping page. Figure 5.13 shows the page of the mobile shopping. The user needs to fill up the order form such in Figure 5.14 since they want to buy the mobile phone. Figure 5.15 shows the confirmation page when the user has submitted the order form.

Figure 5.11: The main page of the website

Figure 5.12: Login page of the website

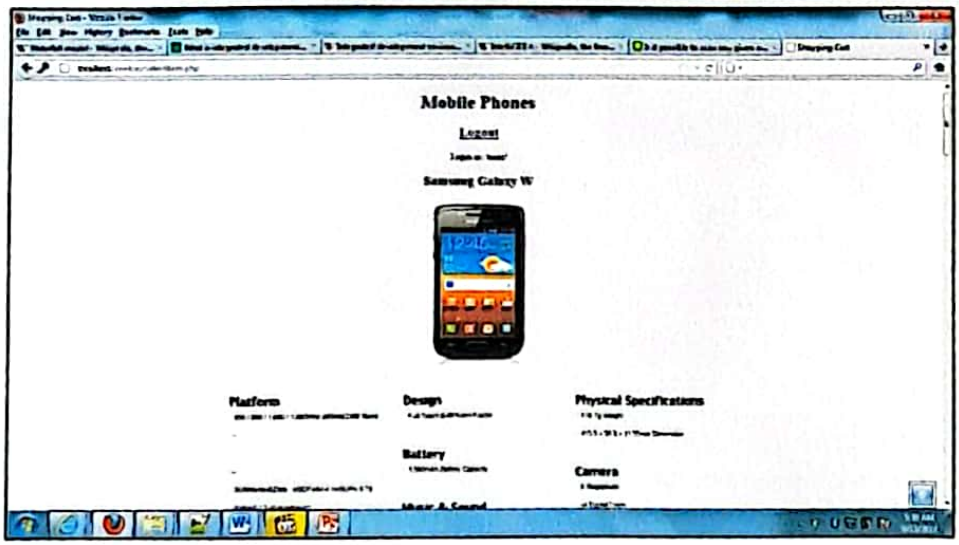

Figure 5.13: Mobile shopping page

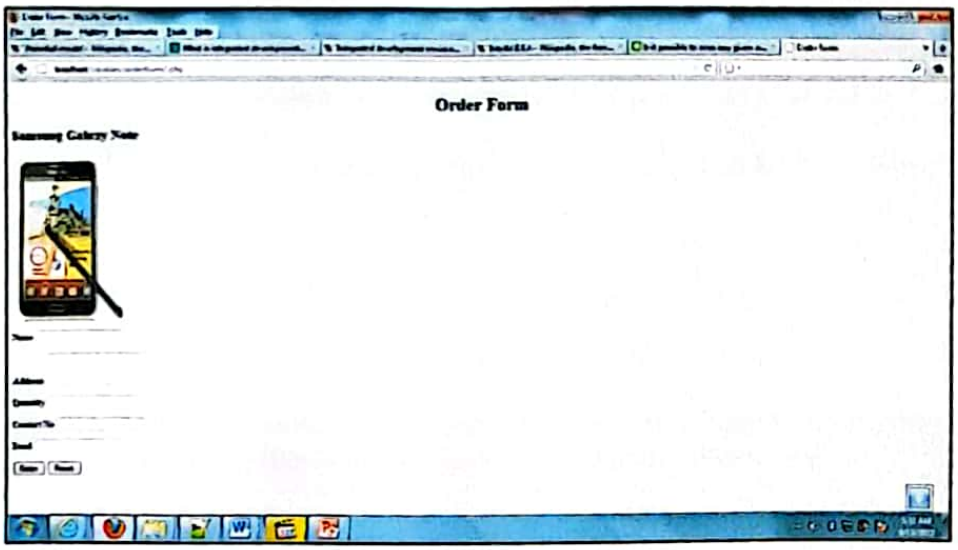

Figure 5.14: Order item page

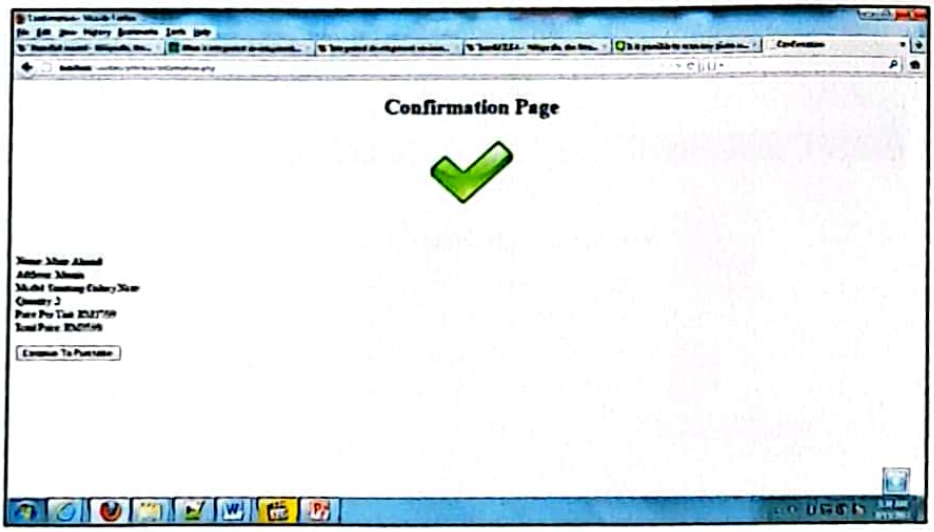

Figure 5.15: Purchasing confirmation page

Figure 5.16 shows the initial amount of price that need to be paid while Figure 5.17 shows the amount of price that being tampered into the lower price. Then, Figure 5.18

shows the price that need to be paid by the user and Figure 5.19 shows the receipt that being generated when the transaction is successful. The price that generate by the receipt is the actual price of the items since the price that being paid had been tampered into the lower price by the user.

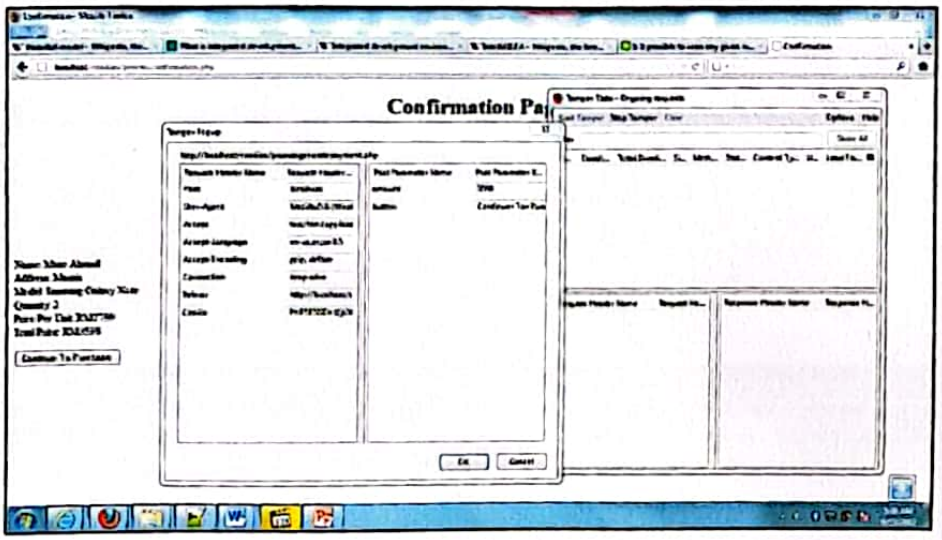

Figure 5.16: The initial amount of price

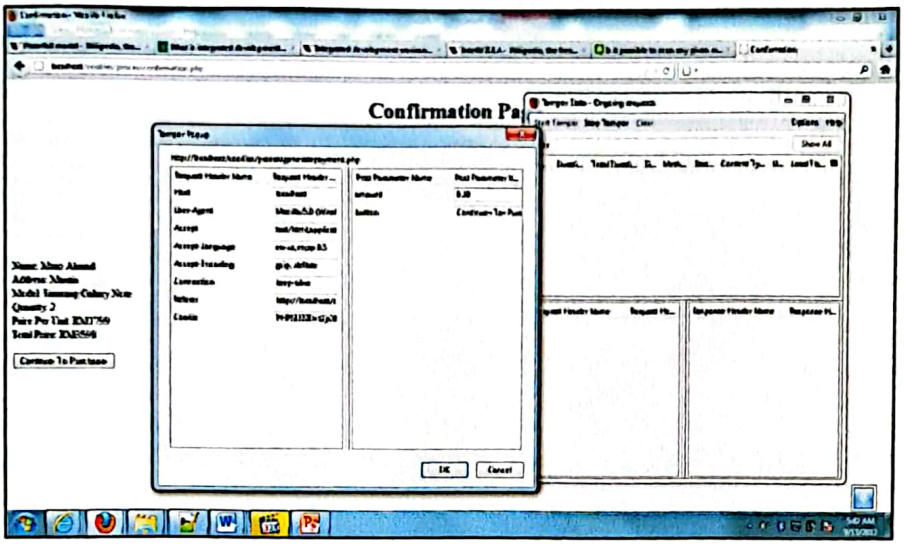

Figure 5.17: The amount of price being tampered

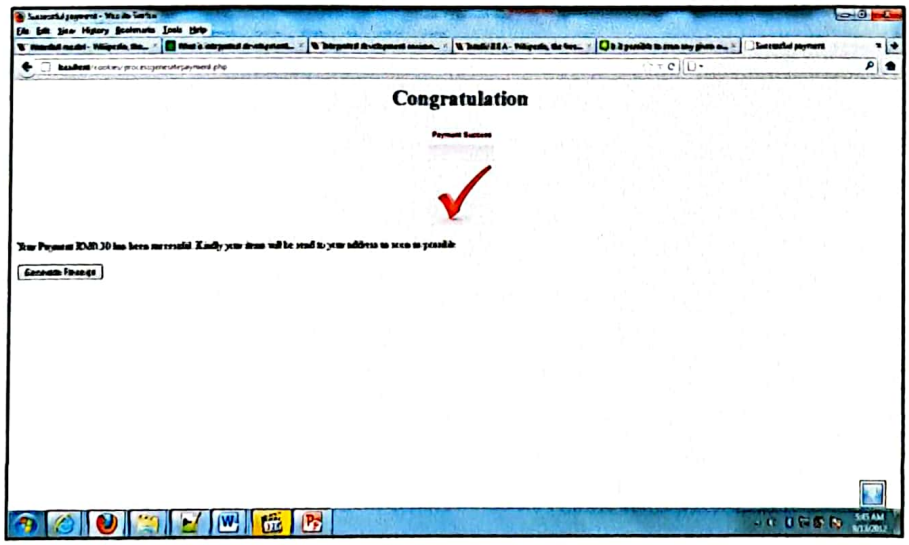

Figure 5.18: The price had been paid for the transaction

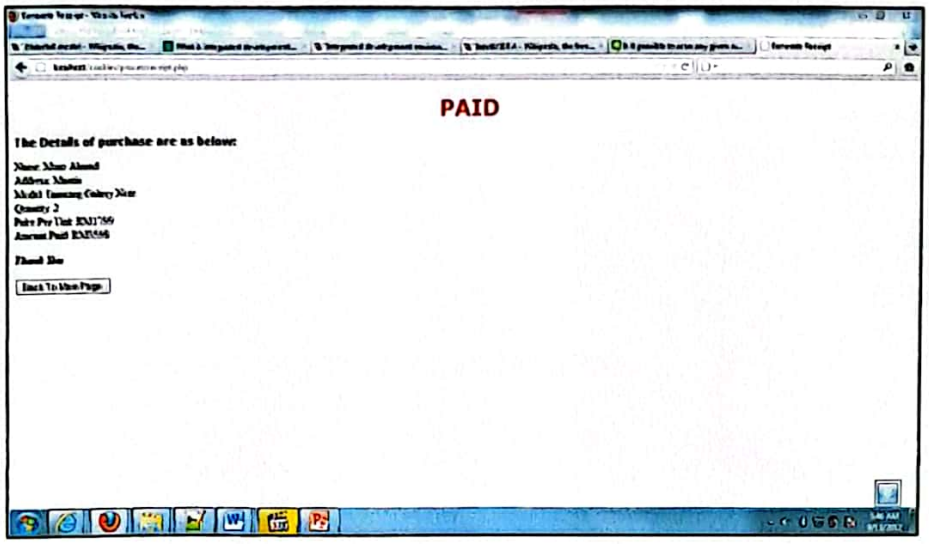

Figure 5.19: Generate receipt for the successful transaction

CHAPTER 6

IMPLEMENTATION

6.1 Introduction

Implementation is one of important part to describe how the project is developed and implemented. This chapter will describe in deep about developed application, technical details of implementation which indicate about the code had been implemented to perform a function of the application and the last part will indicate the finalize screenshot of developed application.

6.2 Description of Developed Application

"Cookies Tampering Vulnerability Scanner" is an application that able to scan and detect the vulnerability in the website source code from cookies tampering attack. This application is just an android scanning tool. The application also can detect only PHP encrypted value in the hidden form field. Once the application detect the encrypted value in the hidden form field, it will prompt an alert message that indicate the website contain encrypted mechanism that made it safe from cookies tampering attack. Once the

application detect hidden form field without encryption, it will prompt an alert message which indicates the website is vulnerable to the cookies tampering attack. The application will prompt an alert message that indicates the website is safe from cookies tampering attack when there is no vulnerable website source code. Then, the application will prompt an error alert message when there is no internet connection or the user has entered wrong website URL to be scanned.

The interface of the developed application is very simple. The application interface have a text field which require user to enter the URL of the website to be scanned, the scan button which used to perform scanning on the specified website after entering the website URL in the text field, back button which used to back on the previous page, and forward button which used to load into the forward page in the website. The scanning will not be performed when the user pressing the back and forward button.

When the user opens the application, they will be redirected to the main page of the application. On the main page of the application, the user needs to enter the URL of the website to be scanned. After that, the user needs to press the scan button in the main page and the application will perform scanning to the website webpage and prompt the alert message. The scanning will automatically performed when the user clicked on the link in the visited webpage. The URL of the link will automatically be updated in specified function in the application.

6.3 Technical Details of Implementation

Technical details of implementation will further describe about how the application is developed using selected development tools which had been mentioned in chapter three of the report. Moreover, since the application is developed using prototyping methodology, then the coding for each prototype of the developed application will be mentioned in this chapter in order to make a clear review in technical on how the application is function and perform the task. The coding of the application had been proper commented in order to make sure the code is understandable for the future used whether to enhanced the application or edit the coding of the application.

6.3.1 How Development Tools and Software Were Used

There are few development tools and software that is used in order to develop this application. The software and tools that used to develop this application include Eclipse which is Integrated Development Environment (IDE) that support android application development, Android Software Development Kit (SDK), Java Development Kit (JDK) and build in Android emulator. JDK is used since the java language is need to be used in development of android application when Android SDK is used to support android development such as the library of android build in function. Then, Eclipse Indigo is used to create and develop an android project. Eclipse is one of the IDE that commonly be used in order to develop an android application due to ease of use and many support. Android emulator is used in android development in order to test the application. It makes

a testing of the project become faster and easier. It works likes a real android devices in the development environment.

6.3.2 How The Application Is Developed

The application is developed using a prototyping methodology. There are four prototypes in order to complete the application development. The first prototype is work like a web browser in order to establish a connection to the web server. The second prototype is an application that will be able to scan and display a website source code while the third prototype is an application that will able to scan, detect the vulnerable source code in the website and alert the user using alert message. The last prototype is the fourth prototype which is the complete application that will be able to scan, detect the vulnerable source code in the website and alert the user automatically when the user click on the link in the website.

6.3.2.1 Development of the First Prototype

The first prototype is the initial stage of developing the application. In the first prototype, the application is able to establish the connection to the web server and display the error message once the connection is failure. The permission to have an internet connection should be given at "AndroidManifest.xml" in order to allow the application to have an internet connection. The internet connection is needed by the application to connect to the

web server. The code below is the code that is used to allow an internet connection for the application.

```
<uses-permission android:name="android.permission.INTERNET"/>
```

The application connects to the web server by using try and catch method. The website will be displayed once connection to the web server is successful and the error message will be displayed once the connection to the web server is failed. The coding below indicate how the application is established the connection to the web server and handle the error message.

```
try{
                String scanweb=url.getText().toString();    //Declare the url called scanweb
                URL url=new URL(scanweb);                   //Create URL object
                mWebView.loadUrl(scanweb);                  //Load into the website that specified by user

            } //End of try

            //Declare the catch to handle the error
            catch(Exception e){
                showAlertDialog2();     //Prompt error alert message
                break;                  //Break the switch statement

            } //end of catch
```

In the coding above, the application will enter into try to load the website that specified by the user based on the URL that entered. The application will display the website once

the connection is successful. When the connection is failed then, the application will load into catch and display the error message. The first prototype in this project is the simple application that working like a web browser which is able to handle the connection between the application and the web server. The enhancement of the first prototype to add further function for the application will create the next prototype.

6.3.2.2 Development of the Second Prototype

The second prototype is the enhancement of the first prototype. The application created in the second prototype is able to scan and display the website source code. Once the application is successful establish the connection to the web server then, the application will create a buffer reader to hold the website source code before read the source code of the visited website. After that, the application will use while loop in order to read the website source code line by line and display the source code in the text view. The output of the second prototype will display the website in the webview and website source code in the textview. The coding below will indicate the module of the application that will read and display the website source code in practically.

```
String temp=null;                          //Declare temporary String variable called temp to null

//Create a new BufferedReader object, which reads from url that specified
//Create an InputStreamReader which used to convert an InputStream from specified url to a reader object which can be used by BufferedReader
BufferedReader in= new BufferedReader(new InputStreamReader(url.openStream()));

//Create while loop to read the webpage source code line by line until finish
while((temp=in.readLine())!=null){

    mycode.append(temp.toString());                          //Append the source code to string
    mycode.append("\n");                                     //Enter to the new line
    mycode.setMovementMethod(new ScrollingMovementMethod()); //Scroll the to view the display source code
}
```

In this stage, the application prototype will only would be able to read and display the website source code. It will be very useful in order to create an application that would be able to detect the vulnerable code of the website in the next prototype.

6.3.2.3 Development of the Third Prototype

The third prototype is the further enhancement from the second prototype. The application created in the third prototype is able to detect the vulnerable code and prompt an alert message whether the website is vulnerable to the cookies tampering attack or not. The implementation of the coding in the third prototype is much complicated compared to the first and the second prototype.

Firstly, four variables are declared to hold the vulnerable code that is used in the detection stage. The coding below indicate the variable that created to hold the vulnerable code.

```
//Declare the code that will be detected in the webpage souce code which is vulnerable code for cookies tampering attack
String vul1="method=\"post\"";
String vul2="<input";
String vul3="type=\"hidden\"";
String vul3_1="type=\"hidden\">";
String vul4="value=\"<br";
```

After declaring the vulnerable code, then the application will create an array that will be used to hold the vulnerable code once it is detected. Then, the application will tokenize the code based on the space and check the vulnerable code in for loop. The vulnerable code will be copied into an array once it is detected in for loop. The coding below indicates the module of the application in order to tokenize the website source code, detect the vulnerable code and copy the code into an array that created.

```
while((temp=in.readLine())!=null){
    //Create a String that will hold in line source code that is read
    String code=temp.toString();
    //Create a string called value that will tokenize the String by space and hold the value
    String[] value=code.split(" ");

    //Check the tokenize value to find vulnerable code using for loop
    for(int i=0;i<value.length;i++){
        //if the value is equal to vul1 string, copy the value into first array of vultemp
        if(value[i].equals(vul1)){
            System.arraycopy(value, i, vultemp, 0, 1);
        }

        //if the value is equal to vul2 string, copy the value into second array of vultemp
        else if(value[i].equals(vul2)){
            System.arraycopy(value, i, vultemp, 1, 1);
        }

        //if the value is equal to vul3 or vul3_1 string, copy the value into third array of vultemp
        else if((value[i].equals(vul3))||(value[i].equals(vul3_1))){
            System.arraycopy(value, i, vultemp, 2, 1);
        }

        //if the value is equal to vul4 string, copy the value into fourth array of vultemp
        else if(value[i].equals(vul4)){
            System.arraycopy(value, i, vultemp, 3, 1);
        }

    } //End of for loop

} //End of while loop
```

Then, the application will check the content of an array in order to prompt an alert message. The content of an array will indicate whether the website is vulnerable or not. The array that created has four elements. The first three elements contain the vulnerable code which is the code of hidden form field in the website while the fourth element contains the encrypted code for hidden form field in PHP language. The application will check the first three elements in the array in order to detect whether the website has hidden form field or not. If one of the first three arrays is empty, the application will tell the user that the website is safe from cookies tampering attack. When the first three arrays are full then, the application will check whether the hidden form field is encrypted or not. The encryption will be detected by checking the fourth element of an array whether it is empty or not. The hidden form field is not encrypted when the fourth element in array is empty. The application will prompt an alert message that indicates the website is vulnerable to the cookies tampering attack when the fourth element in array is empty. When the fourth element in array contain a value, the application will prompt an alert message that indicates the website contain an encryption mechanism that made it safe from cookies tampering attack. The coding below is the main part in the application that indicates how the application detect whether the website is vulnerable to the cookies tampering attack or not.

The coding below indicate the code used to check the first three elements in the array in order to detect whether the website contain the hidden form field or not.

```
//Check first 3 array in vultemp using for loop
for(int j=0;j<3;j++){
    //If one of the three array is null prompt an alert box that tell the webpage is safe from cookies tampering attack
    if(vultemp[j]==null){
        showAlertDialog3();     //Prompt safe alert message
        vultemp=null;           //Set the vultemp array to null
        break search;           //Break the switch statement
    }

} //End of for loop
```

The coding below is executed when the website contains the hidden form field. The code below is used to check whether the hidden form field in the website is encrypted using PHP language or not.

```
//If all first three array is full AND the fourth array is null prompt an alert box that tell the webpage is vulnerable to the cookies tampering attack
if(vultemp[3]==null){
    showAlertDialog1();     //Prompt vulnerable alert message
    vultemp=null;           //Set the vultemp array to null
    break;                  //Break the switch statement
} //End of if

//If all four array is full prompt an alert box that tell the webpage contain encryption mechanism and it is safe from cookies tampering attack
showAlertDialog4();     //Prompt safe alert message
vultemp=null;           //Set the vultemp array to null
break;                  //Break the switch statement
```

In this stage, the application will be able to perform full function in order to detect the vulnerable code for cookies tampering attack in the website. However, the application is unable to scan the vulnerability in the website automatically when the user click on the link in the website that visited. So, the fourth prototype will enhance the application to

detect the vulnerability in the website automatically when the user clicks on the link in the visited website.

6.3.2.4 Development of the Fourth Prototype

The fourth prototype is the enhancement of the third prototype and it is the last prototype in this project. This prototype will create the final product of the development application. In this prototype, the application is able to scan and detect the vulnerable website source code automatically when the user visit the website and click the link on the page of the website.

There are two classes that are created in order for the application to perform the scanning automatically. One of the classes operates to receive the updated URL when the user click on the link in the website and the other one class is used to handle an alert message. The coding below is the code that is used to declare the two classes in the main function. The class that created to receive the updated URL is called "ourViewClient" while the class created to handle an alert message is called "MyWebChromeClient".

```
mWebView.setWebViewClient(new ourViewClient());            //Declare a class that will handle when user click a link in the webpage
mWebView.setWebChromeClient(new MyWebChromeClient());      //Declare a class that will handle alert message when user click a link in the webpage
```

The class created to receive an updated URL and perform scanning automatically is called "ourViewClient". The function in this class will perform the scanning and detect the vulnerable website automatically when the user clicks on the link in the visited website. The coding below show the function that created to receive the update URL when the user clicks the link in the website.

```
//Create a class that will handle scanning when user click on the link in the webpage
public class ourViewClient extends WebViewClient{

//Create a function that will override url and webview when user click on the link in the webpage
@Override
public boolean shouldOverrideUrlLoading(WebView v, String updateurl){
```

Based on the coding above, the updated webview is received by the function as "v" while the string of the updated URL is received by the function as "updateurl". All the parameter will be used to scan and detect the vulnerable website automatically.

The other class will be used to handle an alert message when the user clicks on the link in the website. The alert message will be generated by the function in "ourViewClient" class and it will be handled by a function in the class called "MyWebChromeClient". There is specific function in this class that will be handled and displayed the generated alert message. The coding below show the code of the function that will handle and display generated alert message in "MyWebChromeClient" class.

```
//Create a class called MyWebChromeClient that will handle alert message when user click on the link in the webpage
public class MyWebChromeClient extends WebChromeClient {

//Create a function that will handle javascript alerts when user click on the link in the webpage
@Override
public boolean onJsAlert(WebView view, String lateste url, String message, final android.webkit.JsResult result){
```

Based on the code above, the function in "MyWebChromeClient" class received an alert message as the string called "message". This function will handle the message to be displayed as java script alert message when it is needed by a function in "ourViewClient" class.

The complete developed application will be the final prototype. In this project, this is the final prototype which created an application that would be able to scan a website source code, detect the vulnerable code in the website source code, and prompt an alert message whether the website is vulnerable to the cookies tampering attack or not. The application also will be able to detect the vulnerable source code in the website automatically when the user clicks on the link in the website.

6.4 Screenshots of Developed Application

The screenshot below shows the final prototype of developed application which can be used to scan cookies tampering vulnerability source code in the website. The screenshot in Figure 6.1 will show the main page of the application. Figure 6.2 and Figure 6.3 will

show an alert message that will be prompted for vulnerable and the safe website. In Figure 6.4, the screenshot show an error message while the application is failed to connect to the web server.

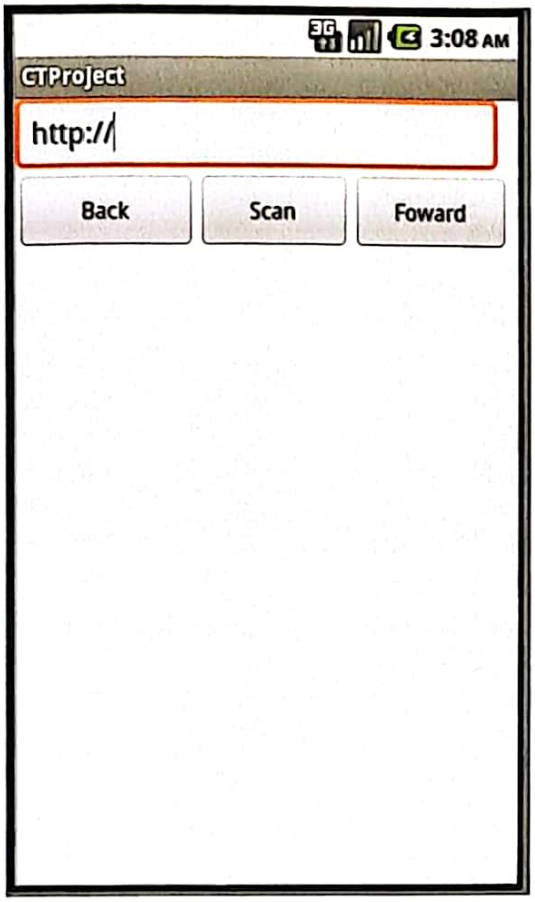

Figure 6.1: Final application prototype main page

Figure 6.2: Final application prototype detects vulnerable website source code

Figure 6.3: Final application prototype detects the website is safe

Figure 6.4: Final application prototype is unable to connect to the web server

CHAPTER 7

TESTING AND VERIFICATION

7.1 Introduction

Testing and verification is an important part in application development such as this project. During this part, the application functionality will be tested and verify based on the final prototype of the application in order to make sure the application that had been develop has function in the right way. If the application not working in the right way during this testing, it should be rectify before testing again. The application will be released to the user once it has passed the testing. That is the reason why testing and verification is an important part in application development project.

7.2 Verify The Code Scanned By The Application

The testing will be based on the final prototype of the application. During the testing, the code that is scanned and detected by the final prototype of the application will be verified in order to make sure the detection is valid or not. The testing had been done to a few website in order to get an accurate result. When the application detects the website is

vulnerable to the cookies tampering attack, the vulnerable code in the real website will be verify either it is true or not. During this testing, GSC website is chosen as the vulnerable website and after scanning, the source code of the page in the GSC website will be review to find out either the application detect valid vulnerable code or not. The screenshot below in Figure 7.1 shows the final prototype of the application detects the main page of GSC website has vulnerable code of cookies tampering attack while Figure 7.2 show the actual code of the main page of GSC website. Based on the testing below, this final application is successful in detecting the vulnerable code in GSC website.

Figure 7.1: Final application detects vulnerable code in main page of GSC website

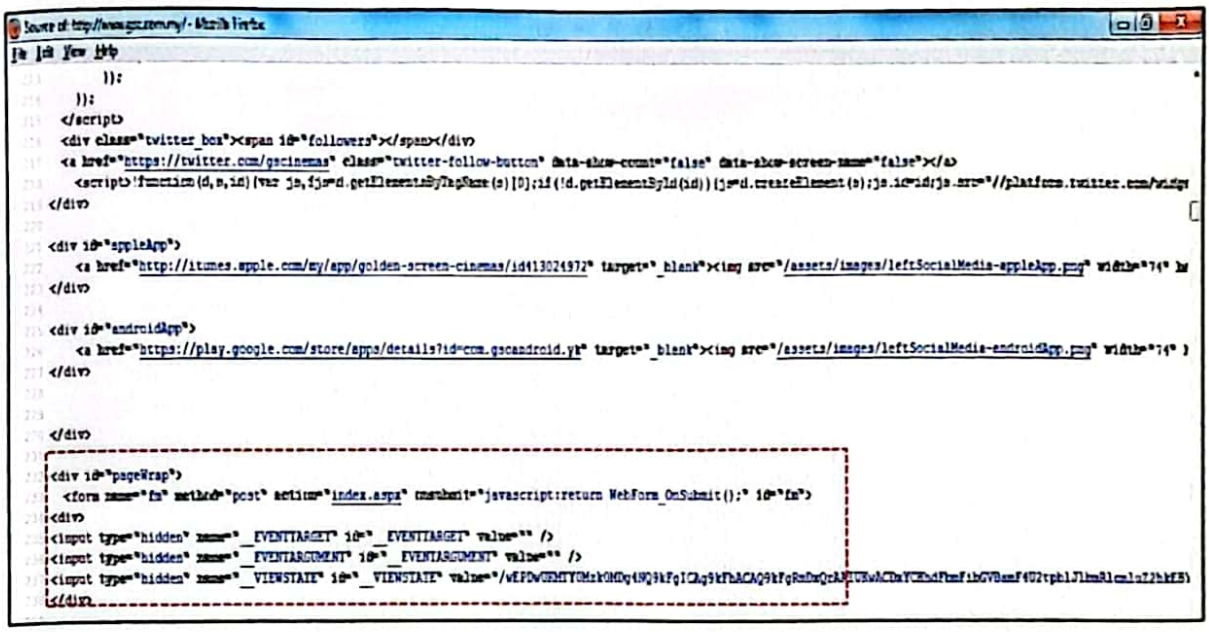

Figure 7.2: The vulnerable code in the GSC main page

The testing for safe website will be done to the TGV website and the source code of TGV website will be review. Figure 7.3 show the final prototype of application detects the main page of TGV website is safe from cookies tampering attack while Figure 7.4 shows the actual source code of the main page in TGV website. Based on the testing below this final application is able to detect the website that is safe from cookies tampering attack.

Figure 7.3: Final application detects the main page of TGV website is safe

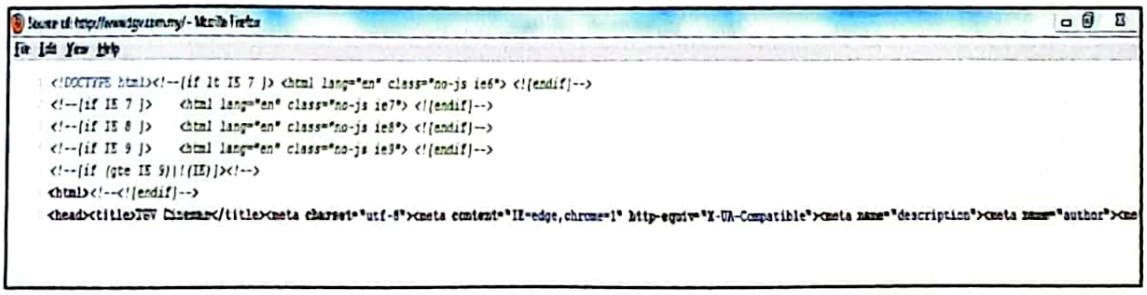

Figure 7.4: The safe code in the TGV main page

7.3 Exploit

Since the website is vulnerable to the cookies tampering attack, the value of the cookies that is sent to the server can easily be tampered by a hacker using a tool that exists in the market. The application will be used to find the targeted website for testing purposes. After some website is found vulnerable to the cookies tampering attack, then it will be set as the targeted website for cookies tampering attack. As a result of this testing will be covered on how vulnerable the website it is and what is the impact of this vulnerability to the company. This testing will exploit the targeted website by tampering the price of the item that using online payment. This attack will cause the company lost a lot of money if the vulnerability is not fixed. The screenshot below shows on how the website is exploited by cookies tampering attack.

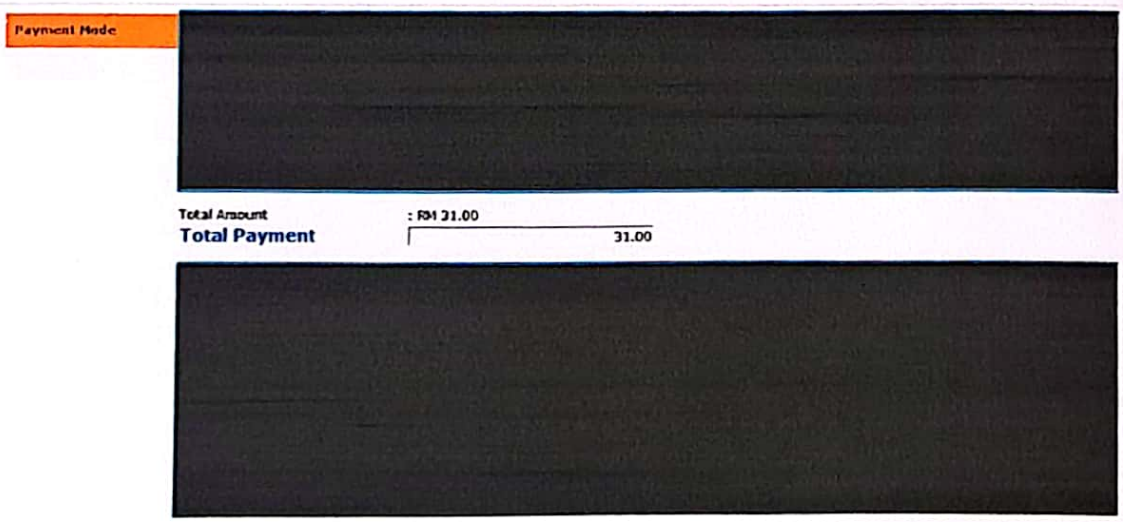

Figure 7.5: Confirmation to pay the item price

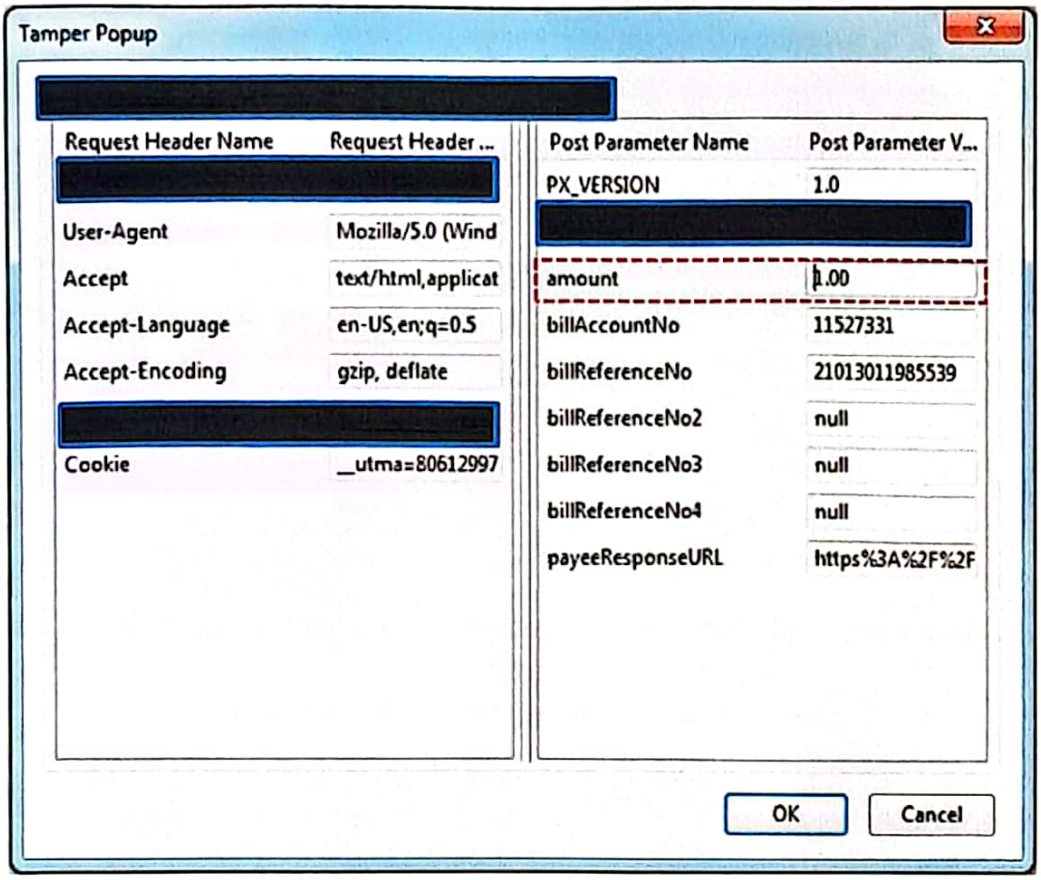

Figure 7.6: Tampering the amount of the actual price

Figure 7.5 shows the confirmation to pay the item price using online payment and the amount of the item is RM 31.00 while Figure 7.6 shows the amount that need to be paid has been tampered to RM 1.00 before sending to the server. The outcome of the attack has been shown in Figure 7.7 when the amount that needs to be paid by the attacker is only RM 1.00 instead of RM 31.00.

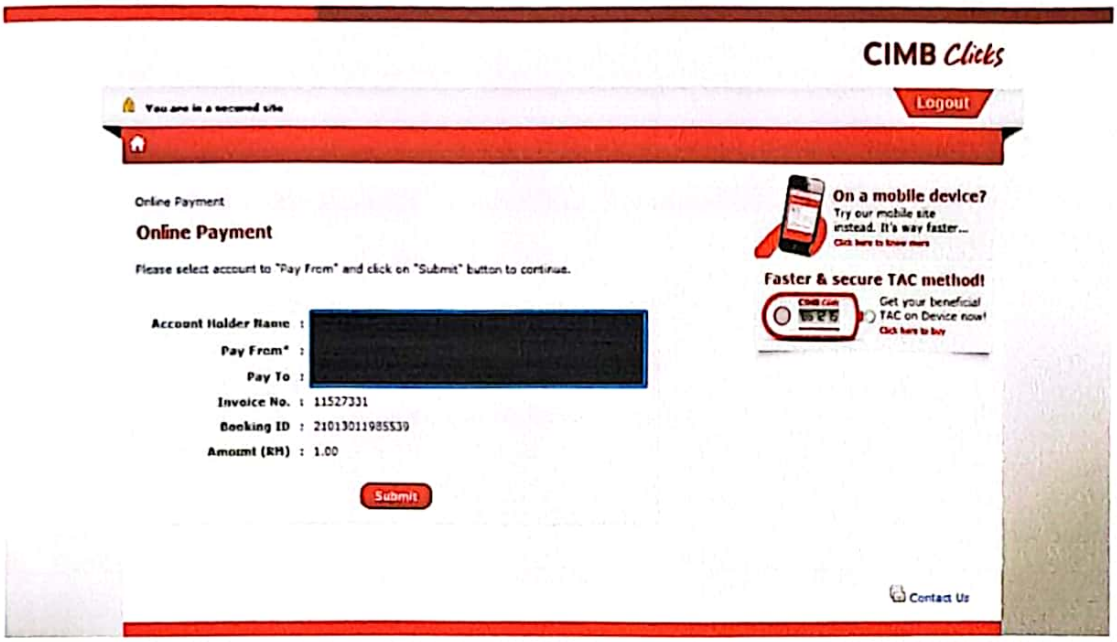

Figure 7.7: Amount that needs to be paid for the item after tampering

The company will lose a lot of money if the website is vulnerable to the cookies tampering attack. Although this attack looks not very serious but the impact is very huge if the company does not take any action to fix this vulnerability. So, this application is useful to test the website either it is vulnerable to the cookies tampering attack or not.

CHAPTER 8

CONCLUSION

8.1 Result

The researches about cookies tampering attack and methodology that will be used during the project are done during the project 1. The research is focused on how the attack is done, the vulnerability that caused the attack, and how to solve the problem. All the research takes about two months in order to get better understanding about cookies tampering attack. After getting a better understanding about cookies tampering attack, the application design such as flowchart and structure chart is constructed for documentation and development purposes.

After that, the environment for developing the project is constructed. All the software and tools needed are well installed and configured in order to start developing the first prototype which is an Android application that would be able to connect to the web server. The next until final prototype has been developed during project 2. There are three prototypes that had been developed during the project 2. All the prototypes have been

developed stage by stage based on the planning during project 1. The final product would be the last prototype of this project.

"Cookies Tampering Vulnerability Scanner" is an android application that would be able to scan and detect the cookies tampering vulnerability in the website source code. The detection will be based on the vulnerable code that found in the website. After the application detects the vulnerable code in the website, it will prompt an alert message that indicates the website is vulnerable to the cookies tampering attack. This application is useful to test the e-commerce website which provides an online shopping in order to prevent the website from cookies tampering attack. While the website is vulnerable, the owner of the website should take action to ensure the website is safe from this kind of attack.

As the result, this application is successful in order to detect the vulnerability in the website source code from cookies tampering attack. It is suitable to be used as the tool for penetration testing to the e-commerce website such as Zalora, GSC and Myeg.

8.2 Problems Encountered

There are several problems that have been encountered during develop this project. The first problem encountered is about the difficulties to get the resources about cookies tampering attack. Ironically, this attack is not popular like SQL-Injection attack, DOS attack, and network mapping attack but the impact of this attack is huge. This difficulty caused the research to take about two months in order to get better understanding about this type of attack.

The second problem encountered during this project is about the difficulties to make a decision about the methodology that is suitable for this project. There are two methodology that is under consideration which are prototyping methodology and waterfall methodology. Each methodology has its own advantages and disadvantages. Finally, prototyping is found to be the best methodology for this project after considering the details about the project.

The next problem encountered is the difficulties to setup the environment for developing an Android application. This environment needs to be setup in order to start developing the first prototype. The first prototype needs to be completed during project 1. Finally, after referring to several tutorials, the environment for developing an Android application is successfully setup in the computer.

The next problem will be the problem encountered during development phases. The problem include the difficulties to program the application in order to read and scan the website source code. It takes around three weeks to solve this problem and after the problem is solve, there are some problem to program the code in order to detect the vulnerability in the website. After one month research about this problem, the solution that takes is by using an array to detect the vulnerable code. Once the vulnerable code is found, it will be copied into an array. The vulnerability is detected by checking the array and prompted the alert message.

The major problem that faces during the development phases is to scan the website source code when the user clicks on the link in the main page of the website. It takes around two month in order to solve this problem. It is the most complicated part in this project. After trying several solutions for this problem, finally this application had been able to scan the user on click page.

8.3 Limitations

Although this application works smoothly to detect the vulnerable code of cookies tampering attack in the website, but it's still have a limitation where this application is only applicable to detect the encrypted value of hidden form field in PHP language while there are many other language that is used to develop a website such as ASP.Net and C#.

8.4 Future Work

The developed application still has a room to be expanded. The application can be enhanced to detect the encrypted value of hidden form field for other language such as ASP.Net, Pearl, and C#. The next enhancement could be added the attack function into this application. For example, the enhanced application would be able to scan cookies tampering vulnerability in the website source code and after the vulnerability in the website source code is found, the application can be used to tamper the cookies value in the website.

REFERENCES

[1] Maura A. van der Linden. 2009. Vulnerability case study: Cookie Tampering.

URL: http://www.infosectoday.com/Articles/Cookie_Tampering.htm

Last date of extraction: 20/7/2012

[2] Herbert H. Thompson, Scott G. Chase, "The software vulnerability guide", Charles River Media, Inc., pp.297-310, 2005.

[3] Marziah Karch. 2012. What is Google Android?

URL: http://google.about.com/od/socialtoolsfromgoogle/p/android_what_is.htm

Last date of extraction: 24/7/2012

[4] Author. 2012. Ways to get information.

URL: http://www.statpac.com/surveys/research-methods.htm

Last date of extraction: 24/7/2012

[5] Author. 2011. Tamper Data: A Firefox Extension.

URL: http://www.ehacking.net/2011/05/tamper-data-firefox-extension.html

Last date of extraction: 10/8/2012

[6] Gina Wisker. 2012. Choosing appropriate research methodologies and methods.

URL: www.palgrave.com/skills4study/studentlife/postgraduate/choosing.asp

Last date of extraction: 25/7/2012

[7] Author. 2012. Waterfall Model.

URL: http://www.waterfall-model.com

Last date of extraction: 11/8/2012

[8] Author. 2012. Chapter 3: Research methodology.

URL:http://dspace.fsktm.um.edu.my/xmlui/bitstream/handle/1812/213/Chapter%220.pdf?sequence=8

Last date of extraction: 25/7/2012

[9] Author. 2012. Integrated Development Environment.

URL: http://en.wikipedia.org/wiki/Integrated_development_environment

Last date of extraction: 25/8/2012

[10] Author. 2012. Prototype model: advantages and disadvantages.

URL:http://www.ianswer4u.com/2011/11/prototype-model-advantages-and.html#axzz21kFJIjsC

Last date of extraction: 26/7/2012

[11] Author. 2012. Spiral model.

URL: www.sccs.swarthmore.edu/users/08/ajb/tmve/wiki100k/docs/Spiral_model.html

Last date of extraction: 27/7/2012

[12] Author. 2005. Rapid Application Development.

URL: http://www.blueink.biz/RapidApplicationDevelopment.aspx

Last date of extraction: 27/7/2012

[13] Author. 2012. Rapid Application Development.

URL: http://en.wikipedia.org/wiki/Rapid_application_development

Last date of extraction: 28/7/2012

[14] Simon Baker. 2005. Iterative and Incremental Development.

URL: www.energizedwork.com/weblog/2005/07/iterative-and-incremental-development

Last date of extraction: 28/7/2012

[15] Margaret Rouse. 2005. Java Development Kit.

URL: http://searchsoa.techtarget.com/definition/Java-Development-Kit

Last date of extraction: 29/7/2012

[16] Author. 2012. Android.

URL: https://developers.google.com/android/

Last date of extraction: 27/7/2012

[17] Author. 2012. Exploring the SDK.

URL: http://developer.android.com/sdk/exploring.html

Last date of extraction: 28/7/2012

[18] Author. 2012. Eclipse software.

URL: http://en.wikipedia.org/wiki/Eclipse_(software)

Last date of extraction: 29/7/2012

[19] Author. 2012. Android emulator.

URL: http://developer.android.com/tools/help/emulator.html

Last date of extraction: 30/7/2012

[20] Author. 2012. Top 6 vulnerability scanner tool.

URL: http://www.ehacking.net/2011/08/top-6-web-vulnerability-scanner-tool.html

Last date of extraction: 31/7/2012

[21] Fergal Glynn. 2012. Vulnerability scanner tool.

URL: http://www.veracode.com/security/vulnerability-scanning

Last date of extraction: 31/7/2012

[22] Marakas, George M. 2006. System Analysis & Design: An Active Approach (Second Edition). New York: McGraw-Hill/Irwin.

[23] Author. 2012. Intellij IDEA.

URL: http://en.wikipedia.org/wiki/IntelliJ_IDEA

Last date of extraction: 25/8/2012

[24] Author. 2012. NetBeans IDE 7.2 release information.

URL: http://netbeans.org/community/releases/72/

Last date of extraction: 26/8/2012

[25] Author. 2012. JBuilder.

URL: http://www.embarcadero.com/products/jbuilder

Last date of extraction: 26/8/2012