Journal of Informatics and Web Engineering
Vol. 2 No. 2 (September 2023) eISSN: 2821-370X

Ensuring Privacy and Security on Banking Websites in Malaysia: A Cookies Scanner Solution

Yi Hong Tay [1], Shih Yin Ooi [1] *, Ying Han Pang [1], Ying Huey Gan [1], Sook Ling Lew [1]
[1]Faculty of Information Science and Technology, Multimedia University, Malaysia
**corresponding author: (syooi@mmu.edu.my, ORCiD: 0000-0002-3024-1011)*

Abstract - In this new era of science and technology, data can be said to be an extremely valuable asset for individuals, corporations, and even countries. Different parties attempt to obtain users' data occasionally, and the collection of web cookies is a prominent example. When users use a computer network, their data will be saved by the web server as cookies, including their private information. As people with bad intentions obtain this information, they can use it to commit cybercrimes and cause losses to the information owners. Thus, cookies management is vital for web users to protect their data. This paper proposes a cookies scanner for banking websites in Malaysia to help web users manage cookies. The scope is focused on banking websites as it is the most targeted website by cybercriminals. The proposed scanner will help users identify, understand, and manage cookies to keep their banking information safe. This paper explores existing cookie scanners to determine the proposed system's design and identify improvement areas. In this paper, we proposed a framework to develop the cookie scanner in the browser extension format. The system's access mode, workflow, functionalities, technical specifications, and requirements are discussed throughout the paper. To show our contribution, a copy of the proposed code implementation has been made available at https://github.com/gnohiy/cookies-scanner-for-banking-websites-in-malaysia.git.

Keywords— Cookies, Cookies Scanner, Cookies Detection, Cookies Management, Data Safety, Data Protection

Received: 12 June 2023; Accepted: 12 September 2023; Published: 16 September 2023

I. INTRODUCTION

With the evolution of science and technology, Electronic Banking is now being adopted and used by more and more people. Electronic banking, more commonly referred to as e-banking, is an electronic payment system enabling users to conduct various financial transactions through banking websites. For instance, e-banking allows users to make payments, transfers, and deposits at their fingertips without leaving the comfort of their homes. In this aspect, it is undeniable that e-banking has brought a lot of ease and convenience to the public.

However, despite all the advantages it has, e-banking also has its shortcomings. For example, e-banking is prone to hacking, illegal transactions, information exposure, fraud exposure, etc. These are all closely related to HTTP cookies. In this case, cookies in web technology are the data that a website sends to the web browser while users browse the website. In other words, cookies are the user's information that a web server collects and stores on the user's device. It is undeniable that cookies are very helpful as they enable a website to remember the user's information and preferences about their visit, and the website can then provide a better experience for the users

Journal of Informatics and Web Engineering
https://doi.org/10.33093/jiwe.2023.2.2.12

Published by MMU Press. URL: https://journals.mmupress.com/jiwe

during their future visits. But at the same time, cookies can also cause trouble to the users, mainly when the website uses their data to do something harmful, or in other words, their privacy is invaded. As mentioned above, users' information will be stored in cookies, including their private information, such as passwords. Thus, when individuals with bad intentions obtain this information from the cookies, they may use it to commit crimes.

Therefore, in this paper, a system that can be used to detect and classify cookies for banking websites in Malaysia is proposed. In this perspective, the system is expected to work as a browser extension and check the cookies collected when users browse the banking websites. When the checking is done, the system will generate a cookie scan report containing the collected cookies list, and the cookies will be classified as either protected or not protected. As it is hard to define whether a cookie is protected, the classification process will be done by examining the critical attributes of a cookie, such as the sensitivity of the user's information. With the proposed cookies scanner, it is hoped that users can be aware of the cookies collected from them and decide whether to continue allowing or blocking them.

A. Problem Statement

Whether it is realized or not, the number of criminal cases involving technology has increased rapidly in recent years. To prove this, the number of reported cybercrime cases in Malaysia increased by over 50% between 2019 and 2021, rising from 13000 to more than 20000, as reported by The Star [12].

Being one of the biggest platforms for online transactions, banking websites are always the target of cybercriminals. Concerning this, the dramatic growth of Internet banking fraud cases has caused many Malaysians to suffer losses. Thus, users should also play their roles instead of waiting for the banks to improve their security systems. In this case, what can users do to protect their private information to minimize the risk of fraud? Undeniably, cookies management is one of the best initiatives that users can take.

Through appropriate cookie management, users can prevent their personal information from being disclosed in the form of cookies to cybercriminals. However, this could hardly be achieved as most users have limited knowledge of web cookies, not to mention how to manage them. Various cookie scanners have been introduced to assist the public in managing cookies, but those scanners have limitations that need to be addressed for improved performance. For instance, the existing systems mostly involve tedious operations, provide helpless information, and do not explicitly show whether a cookie is harmful.

Therefore, in this paper, we introduce our own cookies scanner as a solution. The proposed system improves the existing systems, and the details are discussed in the following sections.

II. LITERATURE REVIEW

Often, when Internet users browse a new website, a notification will inform us that the page uses cookies to enhance our browsing experience and ask us to consent to use cookies [22]. Most of the time, users tediously click on the "Accept" button and continue browsing the website without further thought. The cookies alerts are supposed to make Internet users more concerned about their privacy. It is said so because some cookies may be vulnerable to the users' privacy [2]. In this case, a cookies scanner can play an important role, enabling users to realize what cookies are used on the websites through a cookie scan report.

A. What Is Cookie and Why Is It Important

Today, a cookie is no longer a new term to Internet users. However, most users have limited knowledge and understanding of cookie, and some have just heard of cookie but do not understand what it is and how it works [14]. So, what is a cookie?

In 1994, the cookie was first created by a web browser programmer, Lou Montulli, to optimize users' experience in E-commerce transactions [7]. In more detail, the initial idea of cookies was to allow the users at an e-commerce website to keep their desired items in the virtual shopping cart for later purchases. This could be done as cookies store the users' data and allow the website to retrieve those data later. This indicates that the website can easily

remember the users' choices and information when needed. Thus, cookies can be understood as some small text files used to store pieces of user data, and they are kept on the user's computer via web browsers [3].

As a cookie was introduced, there are a lot of benefits that come with it, which makes cookies an essential component in the Internet environment. From this perspective, one of the most significant benefits of cookies is that it improves users' browsing experience. To prove this, a website can recognize the user who has visited the site before by exchanging data with the cookies, making it easier and more helpful to visit the site again [23]. Taking e-commerce websites as an example, users are allowed to retain login credentials when revisiting the websites, and the websites will also provide them with a customized shopping experience based on their browsing history. In short, cookies can create a more convenient and personalized browsing environment for every user, showing why cookies are essential.

However, despite all the advantages it has, cookies are, at the same time, an online surveillance tool that poses a threat to users' privacy. In other words, cookies can also be a trove of private information for people with bad intentions to spy on, and it will be absolutely easy to track users' online activities through their respective cookies [2]. As a result, users will lose their online privacy. Things may also worsen when some unreliable websites resell users' data without consent or online criminals steal their cookies to commit a crime. Undeniably, a cookie itself is harmless, but it is also a fact that cookies can be misused to harm users. Thus, with so much usage of cookies today, it is vital to check and understand what cookies are used on the websites we browse, and a cookies scanner can help that a lot.

B. What Is Cookies Scanner and How Does It Work

It is not straightforward to determine whether a cookie is good or bad. Most of the time, a cookie will be classified as either harmful or harmless by checking if that particular cookie complies with relevant privacy laws, and here comes the need for a cookie scanner. In this case, cookies can be used by anyone, including the users and owners of a website [4].

A cookies scanner, sometimes known as a cookies checker, is an online tool for identifying the cookies used on a website. Not only that, but some cookie scanners will also provide information such as who sets the cookies, where the cookies are set, and what the cookies are used for [19]. Different types of cookie scanners are available on the market, and they can be varied in complexity and accuracy. A simple cookie scanner may provide fewer details, while a complex scanner will provide more specific details about individual cookies. As mentioned above, users and website owners can use a cookies scanner. For users, a cookies scanner is especially helpful when they are concerned about their online privacy or do not trust a website and want to ensure its cookie policy is accurate. As for website owners, cookies can be extremely useful when making their websites comply with online privacy laws [19].

To provide the functionalities mentioned above, there are several things that a cookies scanner has to perform. Firstly, before the checking can begin, the cookies scanner will need to obtain the URL of the website to be checked. It will then visit the website and crawl through its home page to activate all the related cookies [19]. For information, the cookies are those set by the website server. After identifying the cookies on the site, the cookies scanner will go through the cookies' properties and group them accordingly. It will also analyze the cookies by matching them against its cookie library [19]. When everything is done, the cookies scanner will produce a cookie scan report, which contains information such as the number of cookies, the type of cookies, the purpose of cookies, and so on. Finally, with all these details, users can determine whether a website is reliable and take some necessary actions to safeguard their privacy.

It is a truth that cookies checking can also be done manually. However, a cookies scanner will still be better for several reasons. Unlike cookies manually, a cookies scanner can make cookies-checking easier and faster. It is said so because checking cookies with a scanner is very straightforward, as the users will only need to provide the URL, and the whole checking process will be done in the background until the users are provided with the results. Also, compared to humans, a machine can perform faster, significantly reducing the time needed for cookie checking. Besides, the cookies scanner can also provide the users with a very detailed and concise report of the cookies. If this is to be done manually, then it will be a very tedious and complicated process, not to mention cookies checking is only useful when done regularly [4]. Thus, a cookies scanner is in need.

C. Comparison Between Existing Systems

Various companies have introduced their cookie scanners to the public due to demand. From this perspective, numerous cookie scanners are available on the market, each with its characteristics. Thus, a study is done on the existing systems to identify the areas for improvement and determine the design of the proposed system, as depicted in Table 1.

Table 1. Comparison Between Existing Systems

	System type	**Way to check cookies**	**Time required**	**Information provided**
CookieServe [8]	Web app	By providing a URL to the system	A few seconds	Total number of cookies, cookies classification, cookie name, domain, description, duration, and type
CookieYes Scanner [9]	Web app	By providing a URL to the system	A few seconds	Total number of cookies, cookies classification, cookie name, domain, description, duration, and type
Cookie Metrix [21]	Web app	By providing a URL to the system	From seconds to minutes	Total number of cookies, cookies classification, cookie name, domain, description, duration, and type
CookieChecker - by Cookiebot [25]	Web app	By providing a URL to the system	Ten to twenty minutes	Total number of cookies, compliance with EU Cookie Law, server location, cookies classification, cookie name, provider, type, expiry duration, and description.
Cookie-script [5]	Web app	By providing a URL to the system	Around fifteen seconds	Number of first-party cookies, number of third-party cookies, cookies classification, cookie key, domain, path, cookie type, expiration, and description.
Check My Cookies [17]	Browser extension	By activating the browser extension	A few seconds (fastest)	Domain, cookie name, cookie value, and cookie expiration.
CookieChecker - by CookieHub [6]	Web app	By providing a URL to the system	A few seconds	Cookies classification, cookie name, hostname, and expiry duration.

D. Why Browser Extension

After careful consideration, the proposed cookies scanner is decided to be done as a browser extension. As discussed in the previous section, most cookie scanners available in the current market exist and run as websites. For example, CookieServe, Cookie Metrix, Cookie-script, etc., all work as a website, and a cookies scanner in the form of a website are typically more potent. But still, the outcome of this research will be a browser extension, and why? A browser extension also has its strengths, and some of the strengths are as listed below [10]:

- Fast access. The browser extension can give users a shortcut to access the cookies scanning tool. Users can quickly scan the cookies on a website without opening a web application, which means no delay.
- Convenient and easy to use. Users can easily access and utilize the cookies scanner from the extension list of their browsers. Taking Check My Cookies as an example, the cookies scanning process can be executed with just one click.

- Cross-platform. A cookies scanner in the form of a browser extension can be used on any platform if there is a browser. For example, a browser extension can be accessed on laptops and desktops of any brand or model.
- Ability to integrate non-integrable. The functionality of browser extensions can be integrated into third-party websites without having to access the core or kernel.

E. How to Develop Browser Extension and The Platform Available

It is, in fact, possible for a web user to develop their browser extension. But of course, the user will at least need to be equipped with the knowledge of HTML, CSS, and JavaScript. In this perspective, browsers like Chrome and Firefox provide a platform for developers to implement and test their browser extensions. Thus, the development of browser extensions will be discussed in this section.

As Google Chrome is the most widely and commonly used browser today, the extension development will be viewed from the perspective of the Chrome browser. Four components must exist for creating a Chrome browser extension: the manifest.json file, Content script, Event page, and Pop-up page [1]. Each of these components will be further discussed to gain a better understanding. Firstly, the manifest.json file creation describes the metadata information of a browser extension [13]. For example, the extension's name, version, description, icon, and so on will all be declared in this file. This file is saved in JSON (JavaScript Object Notation) format because it allows the serialization and transmission of structured data over a network connection [24]. In other words, a JSON file will inform the browser about the extension's information and how it should behave.

Moving on, content scripts are created to read the web page details loaded by the user [1]. It is said so because a browser extension will need to access the content of a web page to pass information and make changes to it. The content script is essential in this case because it can change the JavaScript environment without causing any conflict with the page. As for the event page, it is an intermediary script between content and pop-up and is used to control specific tasks and states [1]. For instance, event pages will be loaded when the content script sends a message, when an extension is installed, and when an event is dispatched. Another alternative is the background page, but the event page is still being chosen as it consumes less memory.

Last but not least, the creation of a pop-up page. Unlike the previous elements, the pop-up page is the front-end element of a browser extension. As we know, a pop-up is something that users can see and interact with. Not only that but a pop-up can also be used to deliver specific information to users and even to get requests from users. Similarly, these are also the essential functions of a pop-up page. In a nutshell, the four elements that are mentioned above are equally essential for a browser extension. A developer must work hard on each component to produce a well-performing extension.

III. RESEARCH METHODOLOGY

This section discusses the proposed procedures and techniques used to create the cookies scanner, and the general concept is depicted in Figure 1.

A. Requirement Analysis

When there is a demand for a cookies scanner, the requirements for the system development are first identified. In this case, the requirements can be discussed from several different perspectives [26]. Firstly, the system's requirements. To ensure that the system produced can meet the needs of end users, we have carefully determined and decided the system's features, including the functionalities expected from the cookies scanner, the system's delivery method, and the system's workflow. We have reviewed users' needs carefully and studied numerous existing systems to identify the requirements. The platform and tools for the system development are also examined. Undeniably, a wide range of platforms and tools can be utilized to produce a browser extension; each varies from one another. Thus, it is essential to determine the best platforms and tools for system development. For our case, Google Chrome has been chosen as the platform to develop the cookies scanner, with the help of a software tool named Atom.

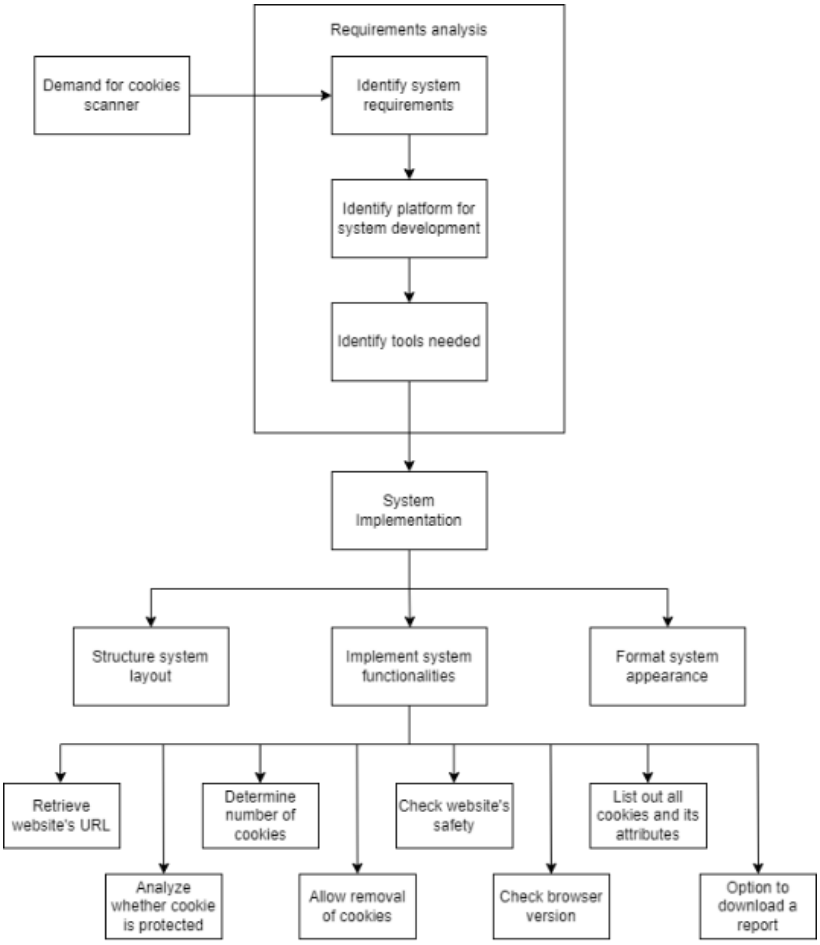

Figure 1. Block Diagram of Implementation Procedures

B. Proposed Framework of Cookies Scanner

As depicted in Figure 1, the development of our cookies scanner falls into three major categories: structuring the system layout, implementing the system functionalities, and formatting the system appearance. Each of these activities can be carried out either synchronously or sequentially.

i. Structuring System Layout

To structure the system layout, the first thing done is to create an HTML file, which can be better understood as a file with a .html extension. HTML is much needed in the cookies scanner development as the system is expected to work as a browser extension, and HTML is the core component that provides the basic structure of the extension. Inside the file, we have described the structure and presentation of our system by using the standard HTML syntax. In this case, a developer must be equipped with the knowledge of HTML programming. Our cookies scanner has five sections, each presenting a distinct set of functionalities or information to the users.

ii. Implementing System Functionality

A JavaScript file is created to implement the system's functionalities. JavaScript is used to control how the content of the cookies scanner will behave in response to the users' actions. Our system is expected to perform more than a typical cookies scanner; thus, numerous functions must be integrated.

a. URL Retrieval

The first function that is added to our system is the retrieval of the website's URL, which is also one of the most critical functions for the system. It is said so because the URL retrieved will be used to facilitate the operation of other functions, such as cookies scanning and website safety checking. For information, the URL retrieval function is developed to obtain the link from the user's active tab on the browser and then directly take it as the input for the system, thus eliminating the need for users to manually provide the website's URL. To realize this, Chrome.tab API is utilized, enabling our system to interact with the browser's tab system and retrieve the required URL. Once the desired URL is obtained, the information will be passed back to the cookies scanner and ready to be used.

b. Cookies Number Identification

Our cookies scanner also constructed a function identifying the total number of cookies. To be more precise, the total number of cookies refers to the sum of web cookies used on the website checked. In this case, Chrome.cookies API is used. This API allows our system to access, query, and read the browser's cookies. The URL from the previous function is supplied to the API to identify the cookies of interest. This allows our system to get the list of cookies detected from the corresponding websites. The total number of cookies is extracted from the list of cookies by performing a count on all entries.

c. Cookies Display

Similarly, the cookies listing function is implemented using Chrome.cookies API, just that different information is retrieved. The cookies information expected from this function includes the cookies' name, domain, category, type, and whether it contains sensitive information. Getting the name, domain, and category is relatively easy as the information can be directly obtained by accessing the name, domain, sameSite, and secure properties associated with the cookies. While for type, the cookie's type is categorized into either protected or not protected, and our system will determine this by examining the value set to the cookies' secure, sameSite, and httpOnly attributes. If and only if a cookie is set with "secure" for the secure attribute, "strict" for the sameSite attribute, and "true" for the httpOnly attribute, then it can be classified as protected. Lastly, to identify whether a cookie contains sensitive information, we have defined a list of keywords usually found in harmful cookies and then compared the value of the cookies detected with those keywords to see if there is any match.

d. Website Safety Check

To check the safety of a website, our cookies scanner is integrated with VirusTotal's API. As an introduction, VirusTotal is a virus, malware, and URL online scanning service. By utilizing its API, our scanner can access the service remotely. To implement the function, we first obtained an API key from the VirusTotal website to authenticate and authorize the API requests for our system. As the VirusTotal API provides different endpoints for different types of services, the specific endpoints and request parameters needed for website URL scanning are also identified. Then, all this information is defined in the function and used to construct an HTTP request that will be sent to the VirusTotal API server to request for checking. Finally, to extract the checking result, JSON.parse() method is utilized to process the response from VirusTotal, and the website safety status is displayed accordingly.

e. Browser Version Check

The browser version-checking function is implemented using the navigator.userAgent property of JavaScript. Using the property, the user agent string, which contains information about the browser and its version, is visible to the cookies scanner. However, since the string is too complex and contains too much information, a regex pattern is declared for the function to extract only the browser version from the user agent string. To identify whether the user's browser is up to date, a check is initiated on the browser version by comparing it against the latest release. From the ablation study, a conclusion is made to determine whether the browser's status is up-to-date or outdated.

f. Cookies Removal

Again, Chrome.cookies API is employed to construct the cookies removal function. In this case, parameters like the associated website URL and the name of the cookies to be deleted are passed to the API so that

deletion can be performed on the expected cookie. This function is made accessible to the users by creating a button, and each listed cookie is associated with a button to perform the deletion.

g. Report Download

Our cookies scanner also generates a cookies scan report for users to download. In this perspective, the report is generated by capturing the result screen that is displayed to the users. The html2canvas library and the jsPDF library are both in use to enable this. The html2canvas library captures a screenshot of the result screen, while the jsPDF library transforms the image captured into a PDF file, which can then be downloaded. We have also included a download button to trigger a download.

iii. Formatting System Appearance

We have styled and designed the system's appearance through a CSS file to make our cookies scanner more attractive. Usually, CSS is used along with HTML, as its primary function is to specify how the HTML elements should be displayed. For our case, we have defined the styles, format, and layout for all HTML elements we created according to a predefined prototype.

C. Proposed Cookies Scanner in Browser Extension

To deploy our cookies scanner as a browser extension, we have created a manifest.json file to define the extension's metadata, including the extension's name, description, version, permissions needed, etc. Then, all the related files are placed into a zipped folder and packaged into a distributable format. When these are done, the extension is published to the extension store and is reviewed by the store's team for approval. Finally, our cookies scanner has been published and made available for users to install and use. A copy of the proposed code implementation has been made available at https://github.com/gnohiy/cookies-scanner-for-banking-websites-in-malaysia.git.

IV. RESULTS AND DISCUSSIONS

In this section, we present the results of the cookies scanner implementation. A detailed description of the system's architecture and design is presented with different diagrams, and the system's interface is also included to show the actual outcome.

A. Outline of the Proposed Cookies Scanner

A simple pseudocode is shown below to provide a high-level overview of our cookies scanner. The pseudocode is intended to represent and describe the outline of the proposed system rather than delving into specific implementation details.

```
1.  Start
2.     If cookies checker is activated:
3.           URL = GetURLFromActiveTab()
4.           While IsValid(URL):
5.                 totalCookies = GetTotalCookies(URL)
6.                 websiteSafety = CheckWebsiteSafety(URL)
7.                 cookiesInformation = ScanCookies(URL)
8.                 browserVersion = CheckBrowserVersion()
9.           DisplayOverallResult()
10.          If UserTriggerCookieDeletion():
11.                RemoveCookie()
12.          If UserTriggerDownloadReport():
13.                DownloadReport()
14. End
```

B. System Architecture

The cookies scanner comprises several components, each providing a distinct functionality to support the system's successful operation. In this perspective, a system architecture diagram is constructed to better understand the system's components, as shown in Figure 2.

The core component responsible for handling and managing the system's internal operations for our cookies scanner is the so-called cookies scanner engine. In other words, it can be viewed as the control center of the cookies scanner, which interprets the instructions when users interact with the system. The scanner engine is responsible for invoking different APIs so that the system can scan cookies and other functionalities. Between the scanner engine and user interface, there is an intermediary component called the scan module, and its significant tasks are to receive requests from users, process the requests, and issue appropriate instructions to the scanner engine. A reporting and visualization component also coordinates the cookies' scan results and the results from the external APIs. The following presents a report containing all the relevant information to the users through the user interface.

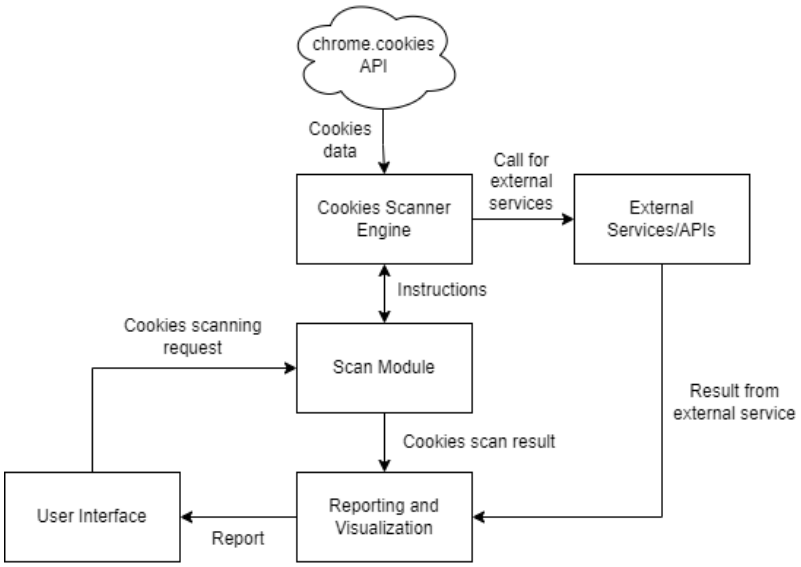

Figure 2. System Architecture Diagram of Cookies Scanner

C. Working Flow of the Proposed Cookies Scanner

The process flow of the cookies scanner is depicted in Figure 3, which mainly focuses on the cookies scanning process. Firstly, when a user activates the cookies scanner, it directly grabs the URL from the website that the user is visiting. This means that the users must ensure they are on the banking website to be checked before activating the cookies scanner. After the URL is obtained, the cookies scanner determines whether the URL is valid. If and only if the URL is valid, the cookies scanner scans the cookies on the banking website.

The cookies scanning process (Figure 3) begins with activating all cookies on the banking website. In this case, the cookies scanner crawls through the website's homepage based on the URL obtained to activate the cookies set by the site server. Once this is done, the cookies scanner identifies all the present cookies and categorizes them accordingly. Next, the cookies collected are passed into a predefined function to determine their properties, such as the cookies' name, domain, category, and so on. Finally, the cookies scanner presents a scan report to deliver all the cookies information to the user.

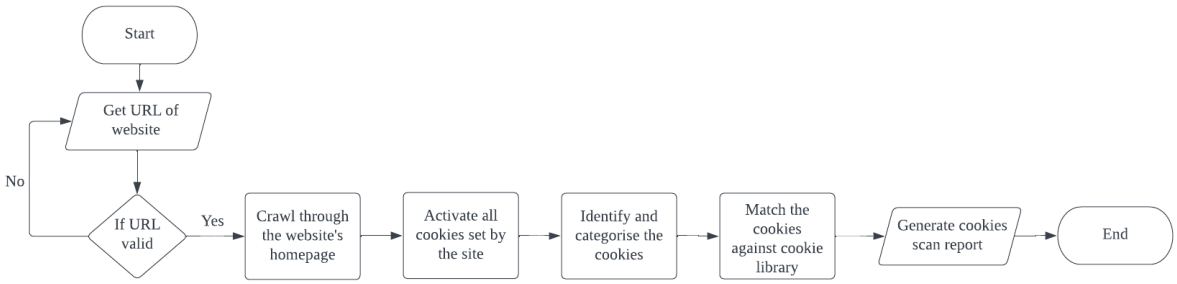

Figure 3. Working Flow of the Proposed Cookies Scanner

The context diagram below (Figure 4) shows the overview of the cookies scanner by illustrating the external entities that interact with it. Two entities interact with the cookies scanner: the user and the banking website. The system expects the website URL from the user indirectly and then provides a cookies scan report as the response. As for the banking website, the cookies scanner sends a cookies activation request to it and waits for a cookie activation reply to continue with its cookies scanning process.

Figure 4. Context Diagram

A straightforward use case diagram is displayed in Figure 5 to show the user-system interaction of the cookies scanner. Two significant functionalities of the cookies scanner are scan cookies and viewing cookies scan reports, which will require the system to generate a report. While viewing the report, the user can also download it.

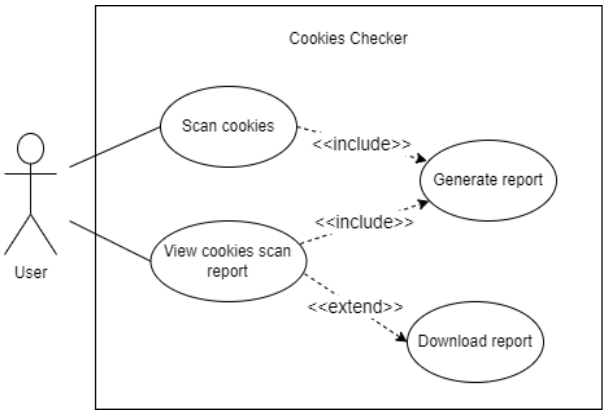

Figure 5. Use Case Diagram

D. System Interface

As mentioned in the previous section, our cookies scanner is a Chrome browser extension. When the users activate this extension, it carries out the cookies scanning procedure in the background and displays a cookies scan report to the users once the checking is done. The report contains helpful information for the users in protecting their information, which can be better understood from the Figure 6 below.

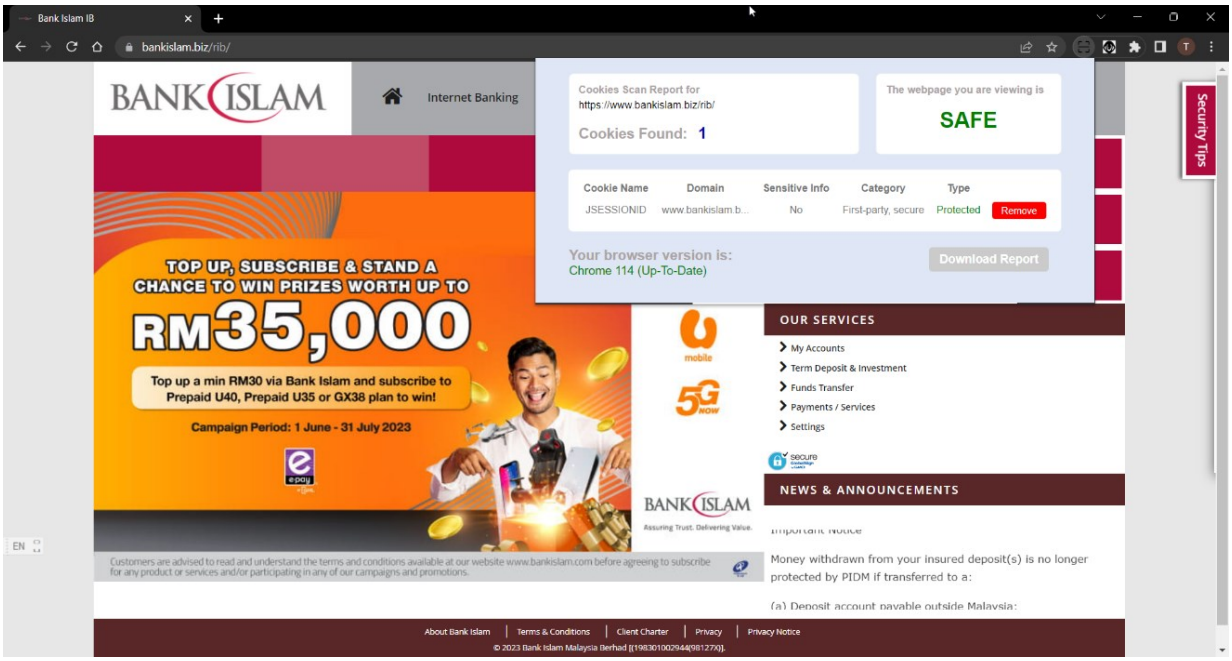

Figure 6. Use Interface

E. Experimental Results

Our cookies scanner has been deployed and tested on Malaysia's 10 most popular banking websites to try the system's functionality (refer to Table 2). The primary objective of the experiment is to determine the system performance and, simultaneously, examine the cookie landscape of those websites.

Table 2: Cookies Analysis Result

Bank	Number of Cookies Detected	True Positive (%)	Risk Assessment of Sensitive Information Leakage (%)
Maybank	14	100	7.14
CIMB Bank	27	100	11.11
Bank Islam Malaysia	1	100	0
Bank Simpanan Nasional (BSN)	14	100	0
Public Bank	13	100	0
Bank Rakyat	2	100	0
AmBank	14	100	0
Hong Leong Bank	21	100	9.52
OCBC Bank	7	100	0
RHB Bank	17	100	0

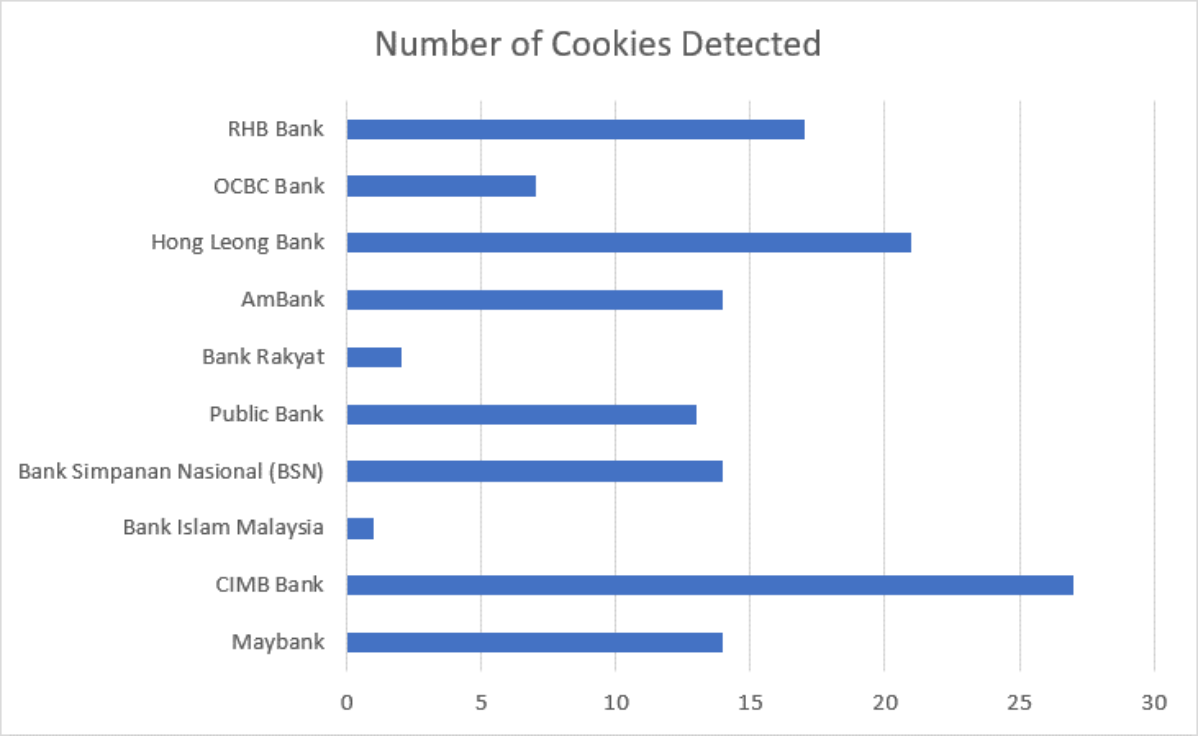

Figure 7. Cookies Detected on the 10 Most Popular Banking Websites in Malaysia

F. Discussion

The implementation results (Figure 7) show that our cookies scanner is quite promising and can serve as an alternative solution for assisting the public in protecting their privacy and security on banking websites. As users browse different banking websites, their information is stored by the web browser in the form of cookies. This may include their private information such as account number and password. If the cookies containing users' personal information are disclosed or exposed, the users' privacy is violated. Thus, proper cookies management is essential for the public to ensure their privacy and security on banking websites; a cookies scanner can greatly help that.

Even though various cookie scanners are available on the current market, most are non-user-friendly, and it can be said that none is helpful for banking website users. It is said so because most users have limited or no knowledge regarding cookies; thus, the technical information provided by those cookie scanners is meaningless. To prove this, users can hardly determine the reliability of a cookie solely based on the results produced by those systems, as it is not explicitly specified, and users have to spend a lot of time and effort to understand it clearly. Moreover, most of the existing cookie scanners are working as websites, meaning users need to visit the corresponding website and provide the URL of the website to be checked manually to initialize the scanning. Undeniably, this is somehow troublesome for the users.

Therefore, we have proposed our own cookies scanner as a solution. Our scanner is extremely easy and straightforward to use, as users are only required to add the browser extension to their browser and activate the extension when needed (a comparison is provided in Table 3). Once activated, our system can automatically grab the URL from the user's active tab on the browser and begin the scanning process in the background. The process takes seconds until users are presented with a cookies scan report. We have ensured that most users can understand the cookies scan report by including only helpful and easily understandable information such as cookies' names, domains, categories, and so on.

Most importantly, the report specifies whether the detected cookies are protected and whether the banking websites users browse are safe. Besides, our system has additional features such as the browser version check. Thus, with all

the information provided, it is believed that users can determine whether their personal information is prone to unauthorized disclosure in an easy way. The creation of this cookie scanner is meaningful and much needed.

Table 3: Differences Between Proposed System and Existing System

Proposed System	**Existing Systems**
Provides helpful and easily understandable information	Provide technical information which is meaningless for most users.
Categories cookies detected into either protected or not protected, which helps in cookies management	Do not explicitly show whether the cookies detected are harmful.
Equipped with additional functions like website safety and browser version checking	Focus on cookies scanning and do not provide additional functions.
It works as a browser extension.	Mostly work as a web application.
Directly and automatically grabs URL from website to be checked	Require users to supply the URL for cookies scanning purposes manually
Involves simple, fast, and automated scanning process	Involve tedious and time-consuming scanning process

G. Contributions

As the proposed system is successfully implemented, we have made several significant contributions in the field of cookie scanning. First, an automated cookies scanner has been produced, enabling simpler and faster cookie scanning. Unlike the common scanners, in which users have to perform various actions to initialize the scanning process, our cookies scanner only requires users to activate the corresponding browser extension, and the system will automatically do everything else until users are presented with a result. This has resulted in a better user experience and encourages user involvement in cookie management. Besides, the system produced is also equipped with fast response capability. To prove this, the cookies scanner can display cookie scan results to its users within seconds. Based on the literature study, most cookie scanners require seconds to minutes to produce a result. Thus, it is obvious that our system is far more effective when compared to the existing ones.

Last but not least, the cookies categorization feature of the system is also a significant contribution. While the existing systems only provide technical details of cookies to their users, the proposed cookies scanner can supply users with more meaningful information that can assist them in managing the cookies. For instance, users can easily determine whether a cookie is protected based on the category attribute specified in the system's cookies scan report. All in all, the implementation of the cookies scanner is proven to be meaningful with all the significant contributions it made.

V. CONCLUSION

This paper focuses on a cookie scanner solution for ensuring user privacy and security on banking websites in Malaysia. We have equipped the system with various features that help users keep their personal information safe and secure. To provide a smoother and more intuitive user experience, we have designed the cookies scanner's usage to be very straightforward, requiring minimal user intervention. Also, the system's user interface is well-organized and uncluttered, enabling users to access a concise report. Overall, the cookies scanner allows users to evaluate the websites they visit very quickly and effortlessly. However, there is also a limitation regarding the cookies scanner's

compatibility with other browsers, which hinders the system's effectiveness for some users. It is said so because the cookies scanner is designed specifically as a Chrome extension, and specific APIs utilized by the scanner are not accessible on other browsers. In other words, the system cannot operate on a platform other than Chrome and is, therefore, unavailable on other browsers. Thus, in the future, it is hoped that the system will be enhanced in terms of its compatibility with browsers like Mozilla Firefox and Microsoft Edge. The proposed enhancement is expected to ensure the seamless operation and usability of the cookies scanner across different browser platforms. By doing so, users can effectively use the system without being affected by their browser preferences. This improvement will expose the system to a broader range of potential users.

ACKNOWLEDGEMENT

The authors received no funding from any party for the research and publication of this article.

REFERENCES

[1] S. K. Arora (2018, January 31). How to make a Chrome browser extension from scratch | Understanding Chrome extension anatomy. https://medium.com/front-end-weekly/how-to-make-a-chrome-browser-extension-from-scratch-chrome-extension-development-basics-basic-ba1daee11123

[2] BigCommerce. (2023). What is a cookie and why is it important? (2023). Retrieved May 24, 2023, from https://www.bigcommerce.com/ecommerce-answers/what-cookie-and-why-it-important/

[3] Brave. (2023). What are browser extensions? (2023). Retrieved May 24, 2023, from https://brave.com/learn/what-are-web-browser-extensions/

[4] Cookie Checker. (2023). Retrieved May 24, 2023, from https://openli.com/dictionary/cookie-checker

[5] Cookie Script. (2023). Retrieved May 24, 2023, from https://cookie-script.com/

[6] CookieHub. (2023). Retrieved May 24, 2023, from https://www.cookiechecker.com/

[7] Cookies. (2023). Retrieved May 24, 2023, from https://www.trendmicro.com/vinfo/us/security/definition/cookies

[8] CookieYes. (2023). Retrieved May 24, 2023, from https://www.cookieserve.com/

[9] CookieYes. (2023). Retrieved May 24, 2023, from https://www.cookieyes.com/?ref=SB_24022022c

[10] Development of Extensions for Google Chrome, Firefox, Opera and other browsers. (2023). Retrieved May 24, 2023, from https://evergreen.team/development-services/extensions.html

[11] J. Fedewa (2021, March 27). What Is a Browser Extension? https://www.howtogeek.com/718676/what-is-a-browser-extension/

[12] K. Inus (2022, August 11). RM560mil lost due to cyber crimes last year, says Home Ministry. https://www.thestar.com.my/news/nation/2022/08/11/rm560mil-lost-due-to-cyber-crimes-last-year-says-home-ministry

[13] T. Kanjariya (2020, May 6). Getting Started with Developing Browser Extensions. https://medium.com/@TusharKanjariya/getting-started-with-developing-browser-extensions-eb4a7d8658b3

[14] Kaspersky. (2023). What are Cookies? (2023). Retrieved May 24, 2023, from https://www.kaspersky.com/resource-center/definitions/cookies

[15] D. Lancefield, M. Ambler, M. Rauber & R. Patel (2011). Research into consumer understanding and management of internet cookies and the potential impact of the EU Electronic Communications Framework.

[16] N. Mott (2023). Browser extension. Retrieved May 24, 2023, from https://www.pcmag.com/encyclopedia/term/browser-extension

[17] One IT. (2023). Retrieved May 24, 2023, from https://chrome.google.com/webstore/detail/check-my-cookies/idmefaajmbkeajdiafefcleiaihkahnm

[18] N. Routley (2020, January 20). Internet Browser Market Share (1996–2019). https://www.visualcapitalist.com/internet-browser-market-share/

[19] Safwana. (2022, July 22). Best Free Cookie Checker for Websites [2022]. https://www.cookielawinfo.com/free-cookie-checker-for-websites/

[20] T. Schiller (2021, March 4). A Brief History of Browser Extensibility. https://medium.com/brick-by-brick/a-brief-history-of-browser-extensibility-bcfeb4181c9a

[21] SqueezeMind. (2023). Retrieved May 24, 2023, from https://www.cookiemetrix.com/

[22] E. Stewart (2019, December 10). Why every website wants you to accept its cookies. https://www.vox.com/recode/2019/12/10/18656519/what-are-cookies-website-tracking-gdpr-privacy

[23] E. Stoycheff (2022, July 5). Browser cookies make people more cautious online, study finds. https://theconversation.com/browser-cookies-make-people-more-cautious-online-study-finds-184219

[24] TutorialsPoint. (2023). JSON - Quick Guide. Retrieved May 24, 2023, from https://www.tutorialspoint.com/json/json_quick_guide.htm

[25] Usercentrics. (2023). Retrieved May 24, 2023, from https://www.cookie-checker.com/

[26] C.Y. Seek, S.Y. Ooi, Y.H. Pang, S.L. Lew, and X.Y. Heng (2023), "Elderly and Smartphone Apps: Case Study with Lightweight MySejahtera", Journal of Informatics and Web Engineering, vol. 2., no. 1, pp. 13-24, 2023.