A Proposal of a Cross-Browser User Tracking Method with Browser Fingerprint

Vu Xuan Duong

Faculty of Environment and Information Studies

Keio University

5322 Endo Fujisawa Kanagawa 252-0882 JAPAN

Submitted in partial fulfillment of the requirements

for the degree of Bachelor

Advisors:

Professor Hideyuki Tokuda

Professor Jun Murai

Associate Professor Hiroyuki Kusumoto

Professor Osamu Nakamura

Associate Professor Kazunori Takashio

Assistant Professor Rodney D. Van Meter III

Associate Professor Keisuke Uehara

Associate Professor Jin Mitsugi

Lecturer Jin Nakazawa

Professor Keiji Takeda

Abstract of Bachelor Thesis

A Proposal of a Cross-Browser User Tracking Method with Browser Fingerprint

Today, modern web browsers and browser plug-ins provide a rich set of interfaces for many Rich Internet Applications. But on the other hand, when web users access to the Internet, there are many browser's information sending upon each request, for example, their installed plug-in set and even installed font set on their system. Especially this information is sending without users' awareness. But the problem is when we combine all information together, they could become a personally identifiable information.

This thesis introduces a method to track users online behavior using information that certainly sending upon each request connection which we call Browser Fingerprint. We will show that without Cookies and user awareness, we can track users browsing history each time they access to web server. Besides showing that users could be profiled like using third party cookie, we also show that browser fingerprint can be linkable even user using many browsers on the same computer or even browser from other devices.

To make a clear picture of current privacy and security problem, by collecting only information transmitted upon each request connection to the Web server, we discuss how to defend against Browser Fingerprint. However we reach the conclusion where currently there is still a trade-off because a method to defend against Browser Fingerprint could itself become fingerprinting information. We will also discuss about privacy aspect by building a scenario to show how users privacy are threatened each time information is collected enough.

Keywords:

Browser, User Tracking, Privacy

Vu Xuan Duong

Faculty of Environment and Information Studies

Keio University

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we will provide an overview about current privacy problems and the raising of privacy concerns with the growth of social network as the background of this thesis. Then, we talk about the research goals and the structure of this thesis.

1.1 Background

Along side with the rapid development of the Internet, Social Network Services (SNS) are playing an important rolls in many parts of life. According to statistic data from Facebook[4], there are more than 800 million active users every seconds, more than 50% of active users log on to Facebook in any given day. Facebook argues that there are also more than 900 million objects that people interactive with pages, groups, events and community pages. SNS-es are changing the way people communicate, work, and play. But when personal and identifiable information is published on the SNS with unclear privacy policy from SNS provider, it starts raising a security and privacy concern in group researchers all over the world. The most recently research about Facebook showed that Facebook is tracking which websites users visit even after they have logged out of the service[5]. According to hacker and blogger Nik Cubrilovic, "even if we are logged out, Facebook still knows and can track every page you visit, pointing to Facebook cookie that remain active even after the user signs out". Moreover, questions about what social networks mean for personal privacy and security have been brought to a head by research at Carnegie Mellon University that shows that Facebook has essentially become a world-wide photo identification database. Paired with related research, the researchers claim they are looking at the prospect where good, bad and ugly actors will be identify a face in a crowd and know sensitive personal information about that person[6].

Piece of Personal Identifiable Information	Level of Availability			
	Always available	Available by default	Unavailable by default	Always Unavailable
Personal Photo	9	2	1	0
Location	5	7	0	0
Gender	4	6	0	2
Name	5	6	1	0
Friends	1	10	1	0
Activities	2	8	0	2
Age/Birth Year	2	5	4	1
Email Address	0	0	12	0
Phone Number	0	0	6	6

Table 1.1: Personal Identifiable Information Availability Counts in 12 Online SNS (cited from [2])

Besides, without users' awareness, many advertisement companies have been profiling user's web browsing history. This activities often refer to terminology "Targeted Advertisement" whereby advertisements are placed so as to reach consumers based on various traits such as demographics, purchase history, or observed behavior[7][8]. From marketing and business point of view, these techniques allow them to increase the effectiveness of their campaigns and pull the attention from Web users. When a customer visits a web site, the pages they visited, the amount of time they view each page, the links they click on, the searches they made, etc, are collected to create a profile that links to that customer's web browser. The situation became extremely complicated with appearance of Advertising Networks such as Google Ads. Customer's activities is not only tracked onsite but also in different way to individual sites. This cause many online users and groups give their concerns about privacy issues around activity of Advertising Networks. Though, this is a controversy that the behavioral targeting industry is trying to contain through education, production constraints to keep all information non-personally identifiable or to obtain permission from end-users.

There is growing awareness among web users that many people now block, limit or periodically delete third party cookies set by Advertisement Networks. Along side, there are many researches carrying on cookie policy to protect personal identifiable information leakage from cookie on upon each access to the Web server. For example, Mozilla and Google are both working on Do Not Track

header that can control the way the users' information are collected and used online[9]. Do Not Track header allows users to let a website know they would like to opt-out of third-party tracking for purposes including behavioral advertising. It does this by transmitting a Do Not Track header every time data is requested from the Web.

However, that does not seem enough when many more new techniques have been invented to track people. Information that people thinks certainly there is no connection with their personal information also could become identifiable information and cause serious problem. When the information linked with together, it could be used to easily track out user's web browsing history without any help from Cookies.

1.2 Challenge and Research Goal

This thesis introduces a method to track user's online behavior using information that certainly sending upon each request connection which we will call Browser Fingerprint. We will show that without Cookies and user awareness, we can track users' browsing history each time they access to web server. Besides showing that users could be profiled like using third party cookie, we also show that browser fingerprint can be "linkable" even user using many browsers on the same computer or even browser from other devices.

To make a clear picture of current privacy and security problem, by collecting only information that transmitted upon each request connection to the Web server, we will also show that currently there is no way to defend against Browser Fingerprint. We will also discuss about privacy aspect by building a scenario to show how dangerous and how users' privacy are threatened each time information is collected enough.

1.3 Structure of Thesis

The rest of this thesis is organized as follows. Chapter 2 describes some user tracking methodologies using Cookies, the evolution of Cookies that make users harder to remove or discover. After discussing some of common problems with Cookies, we will explain the way Cookies can be used to track user. Chapter 3 describes related work and researches. In Chapter 4, we describe approach of fingerprinting browser and the usage of browser fingerprint to track user's online behavior. Chapter 5 is the implementation and design of the system. Chapter 6 discusses results and evaluation from experimental. Finally in Chapter 7, we present conclusions and future work.

Chapter 2

User Tracking Methodologies

Privacy concerns that raised by Cookies are still one of the big privacy problems to current Internet users since Cookies are used as the main method of advertisement companies to track and profile users. With the appearance of many new Social Networks, advertising companies like Google also have much more rich places for advertisement activities. But it turns out the problem is becoming more complicated today since sites want to track users have new technical options that are hard for user to respond to.

In this chapter, we introduce about Original Cookies (Cookies) and recently appeared Super Cookies. After explaining about many privacy concerns raised by Cookies and Super Cookies, we will explain how Cookies can be used to track the Internet users' behavior.

2.1 Original Cookies

According to Wikipedia[10], cookies are also known as HTTP cookies, web Cookies, or browser cookies, used for an origin website to send state information to a user's browser and for the browser to return the state of information to the origin website. The state information can be used for authentication, identification of a user session, user's preferences, shopping cart contents or anything else that can be accomplished through storing text data on the user's computer.

In this thesis, Cookies mean Original Cookies if we do not specify anything.

2.1.1 A historical overviewer of Cookies

HTTP[1] was designed to share resources on the current computer to the other by using an unique URL[2]. When this protocol was designed, it was only allow to display the same document when accessing on the same URL. But as the changes of the Internet and World Wide Web, it needed to share a dynamic pages that can slightly change depending on the particular situation. Obviously, HTTP could not meet all the requirements.

Therefore, on 1994, the Developers from Netscape Communications Corporation have created an extension to the HTTP specification that allow to add new type of data which would be set by the Server and sent by browser to that server in every request. This piece of data was called "Cookies" and first appeared in Mosaic Netscape 0.9 beta.

Cookies provide four main advantages to a software developer: holding the session data, as in a shopping cart, storing login credentials, providing customization or personalization features, or they can be used to track user's activities. These uses are not exclusive because a single site may use one or more of these techniques.

There are two types of Cookies : First Cookies and Third Party Cookies. However, there was no obvious definition to distinguish between First Cookies and Third party Cookies. Since the raised of privacy concern when a website try to write a piece of data to user's browser to analysis access pattern to track user, browser vendors had to make a clear definition and policy to control the cookies that used to hold session data between domain/subdomain and the cookie that used javascript or iframe called third party cookie. Third party Cookies here means that their values can be read by external website (Third Party Websites) resulting in tracking user through different websites which have the same javascript code or iframe. For more details, image you are using Google service, and access to google.com, the cookie will be set is first cookie. Without first cookie, you must login to google service such a Gmail again each time a page refreshed. In contrast, if you access to page that content google advertisement, cookie from *.google.com which is set by javascript would be third party cookie.

But the situation here is becoming a bit complicating because we also need to distinguish first cookie and third cookie upon send and receive data. Since there are many warnings from experts about privacy concern, there are highly controversial debates about using of cookie and web users should/should NOT using cookie as browsing the Internet if they want to protect their privacy. Of course, it is evident that no friendly user interface would be provided without using of

[1]Hyper Text Transfer Protocol

[2]Uniform Resource locator

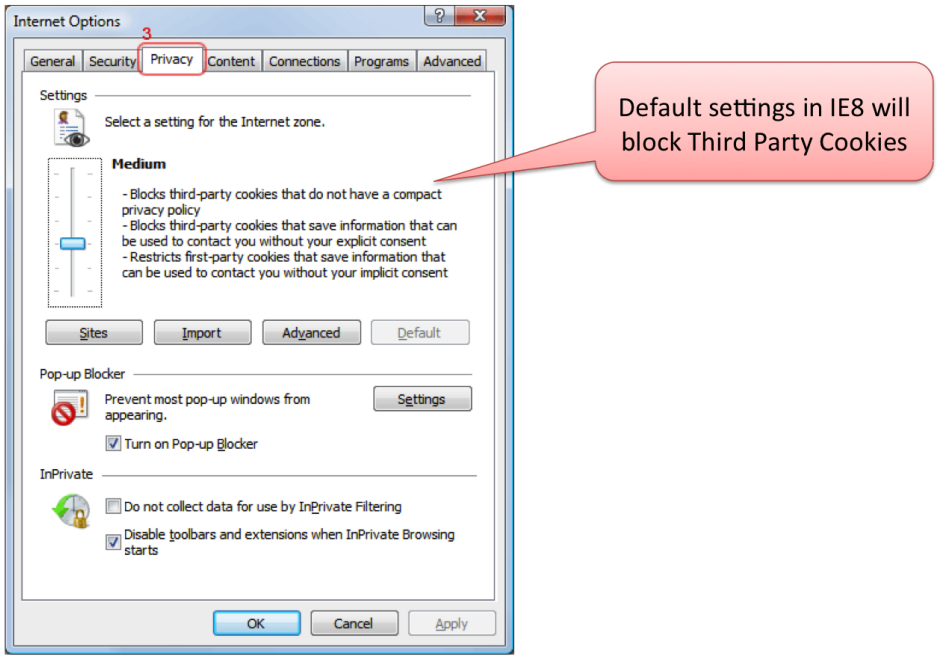

Figure 2.1: Cookies and Rasing in Privacy Concerns

cookie and it is terrible to image a user had to login and login, again and again hundred times if he/she want to check his/her new coming emails. As a result, cookie is still used, but with some maintain from browser vendor to make it more "secure" by force cookie only available on the same domain/subdomain but still cause the side effect. That is when the third party cookie is saving as the first cookie. In other words, the external javascript or iframe code set a part of data on user's computer as a first cookie but the cookie was sent as the third party upon each request. Therefore, the variation becomes very complicating since there is not clearly to determine if a part of data was saved as first or third party cookie, causes the difficulty to say if a cookie is used to store credentials or to track users' online behavior.

2.1.2 Security problems caused by existence of cookie

There are many security problems which can not be maintained or solved by the existence of cookie. For example, the leakage of information to other domain without user's awareness when the currently login state was kept retaining. These problems cause a critical risk on the client side, and even more seriously, bring zero-attack to browser side. This may result the hackers taking control over the system. The reason why the risk still remaining for such a long time is that there seems to be no way to clarify the problems because server side should fix these problems or client side should fix these problems is becoming a very unclear question all the time.

Here some highlight security problems remaining as the existence of cookie

- CSRF Attack [3] and XSS Attack [4] : When a website could be attacked by using CSRF, an iframe or image or javascript code could be injected into website and executed without user's awareness
- Phishing : Since the login state is kept retaining, user could be easily redirected to a phishing website and lose login credentials or even more important information such as credit card number
- JSON [5] hijacking : JSON response could be read through cross-domain and cause leakage of personal information when login state keeps retaining

It is obvious that there are many cases in which cookie is not used to stored credentials, and the security problems should be solved by client side (browser side) but :

- There is a need to control access to URL that content response information related directly to user
- If there is a loaded external source, credential information must not be sent upon each request. In other words, third party should not be sent over each request to external website.

If we could archive these rules, we could solve almost problems with cookies. Unfortunately, until now there still be no way to archive all these goals at the same time.

[3] Cross-site Request Forgery

[4] Cross Site Scriting

[5] JavaScript Object Notation

2.2 Super Cookie - Harder to see and remove

As the increasing of data sent upon each request, cookies' limitation has appeared since they could only available to hold a small size of data as well as the need to send cookies with every request. From Internet Explorer 5, Microsoft introduced a new mechanism for storing data on a user's system called the userData store. This method is well-known to web developers as a new way to increase storage and site data management capabilities but also keep maintaing the same-origin security policy. And this mechanism has been called "Super Cookies" (also called "Flash cookies" or "Zombie cookies") as a huge amount of data it could hold in comparing with "Original Cookies"

Super Cookies serve the same purpose as regular cookies. But in term of Privacy, when Super Cookies are using to track user preferences and browsing histories, they are definitely difficult to detect and subsequently remove. The Super Cookies secretly collect user data beyond the limitations of common industry practice.

Unlike Original Cookies and their standard policy, most of Super Cookies are stored on the different locations on the user's computer, for example in a file used by a plugin. The most common case we can give here is Adobe Flash plugin that currently installed on almost Internet user's browsers. To compare with "Original Cookies", Super Cookies are harder to find and remove, because they are standing outside browser's control. As a result currently "Delete Cookies" built-in function of common browsers can not remove Super Cookies. In typical cases, browsers can limits the Cookies to be written, read and removed by the site that created it. However, on the other hand, Super Cookies allow multi-sites track and record users online behavior. Furthermore, according to researchers at Stanford University and University of California at Berkley in 2008, at that time, websites such as MSN.com and Hulu.com have been installing files known as super cookies which are capable of recreating users profiles after people deleted regular cookies and without the knowledge of the user[1]. This does make sense since these type of Super Cookies can regenerate themselves to prevent from being removed by the users.

To tell more details, in the recent research results about "Cleaning Up After Cookies" carried by Katherine McKinley at iSEC Partners, he figured out that, the most popular Adobe Flash plugin does not provide any data protection schema to all current common browsers[11]. That means Flash is completely independent from browser's control. Figure 2.2 shows the preferences panel on Mactonish computers. Flash does not have a browser interface that allow user can modify privacy settings or local store with Flash plugin.

However, the situation is worst because by using their own local store mechanism, Flash allows data can be access from cross browsers, even in private browsing modes. In his research, Kather-

ine McKinley also shows that this causes potentially leakage of identifiable information such as user aliases or identification numbers and even bank multi-factor authentication images or codes. This once again brings Cookies and privacy problems to the top concern from the first time the researchers found of their usage as new tracking mechanism. More dangerously, many of the companies had found to be using the this new techniques to track the Internet users without any awareness, and faced very strong criticism for collecting personal data. And they only stopped it after getting contact with researchers.

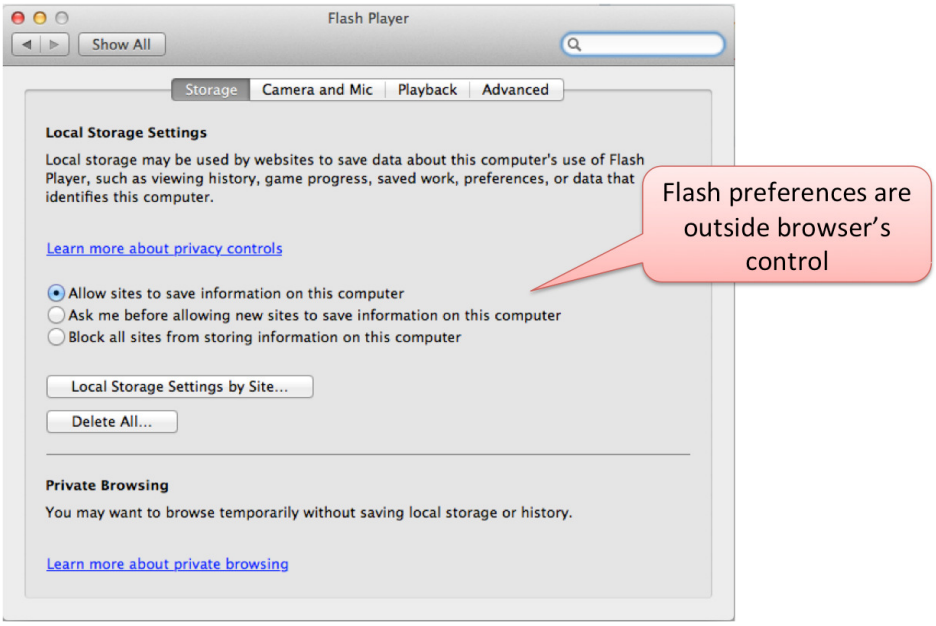

Figure 2.2: Flash Cookies Preferences Panel on MAC Computer

Mostly like "Original Cookies", Super Cookies also cause many privacy problems as we introduced in previous sections. We are not tend to explain them again in this sections. But in conclusion, we want to emphasize that Super Cookies has appeared as a new generation of Cookies with extremely annoying characteristic that threatening the Internet users' privacy. In his articles about Super cookies, Seth Schoen (Electronic Frontier Foundation) said that, the Internet users should have the right to know how they are being tracked. But in contrast, Super cookies have "made that extremely difficult with regard to Flash cookies, since they're stored outside of the browser's control, and since the official Flash plug-in isn't open source, users can't easily fix this for themselves" [12]. And clearly, there's a lot of work need to be done to bring these problems to the light.

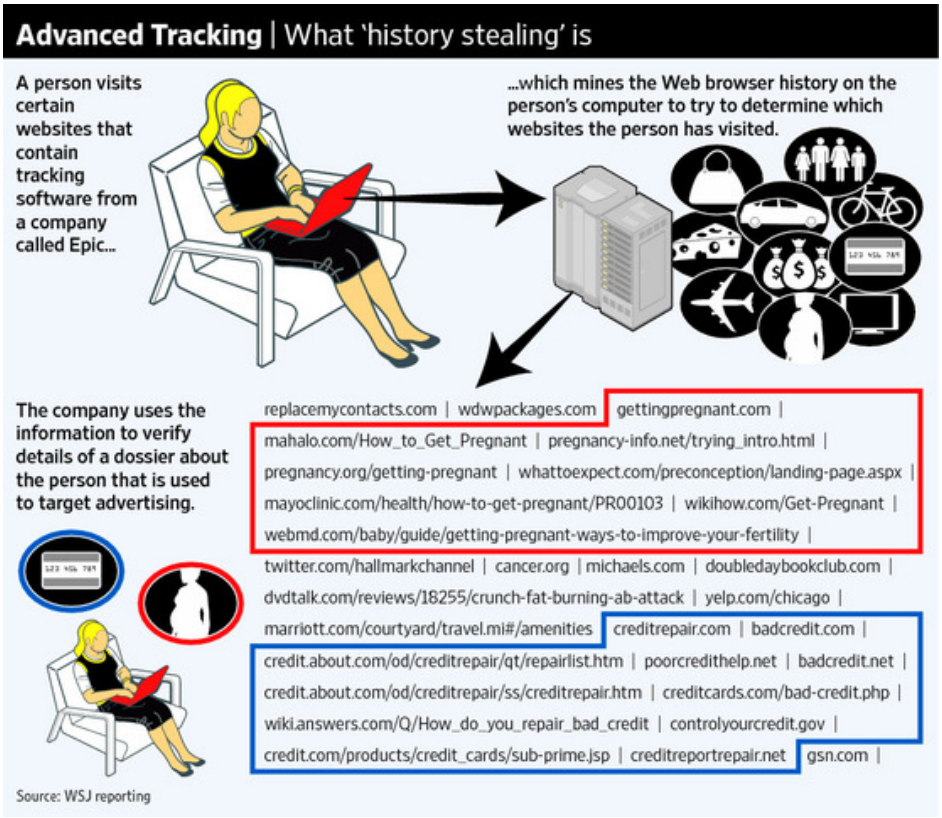

Figure 2.3: Advanced Web Tracking methodologies using Super Cookies(Wall Street Journal, Stealthy SuperCookies[1])

2.3 How to track user's behavior

In the previous sections, we briefly introduced two types of Cookies : Original Cookies and Super Cookies as a powerful mechanism to track and profile the Internet users. In this section, we will explain how Cookies can be used to profile and track users by giving a simple example.

Firstly we must understand how Cookies work. Figure 2.4 shows how Cookies are set by the server (Cookies are always set by server side). Typically, when an user accesses to the Web Server with the browser, before displaying a web pages, the browser must send an HTTP request to the Server. In response message, the Server will check if currently Cookies are set or not. If Cookies are not set yet, the Server will set Cookies value and send back to the Client (the Browser) with HTTP request response. From that, in every HTTP request to the server, the browser will sent the request with set Cookies.

Figure 2.4: Cookies will be sent in all HTTP Request

To explain how Cookies can be used to track users, let us start with an example of targeting advertisement. Figure 2.5 shows the tracking and profiling processes. Suppose you are accessing to web pages with many advertisement from third party for the first time. The third party will try to set Cookies and an unique ID on their computer. At the same time, they also build a profile of you based on the unique ID with specific pages that you had visited. The next time you visit the same page of a website, because Cookies will be sent in all HTTP request, previously set Cookies will also be sent to Server and profile that built up before will be updated. It repeats exactly the same when user access to another site but with the same advertisement from the same company. As the time passes away, advertising companies will build up a single giant profile of your Internet browsing history. In case of targeting advertisement, suppose you have visited two sites with the same content, for example, sporting selling sites. By analysing browsing history, advertising companies can know about what you are interested in. Next time when you visit another site, based on your profile, an associated advertisement will be also displayed. That is the simple way that

many companies using to track and profile the Internet users. Even if users deleted the cookies, they will build a new profile based on Internet browsing history of those users.

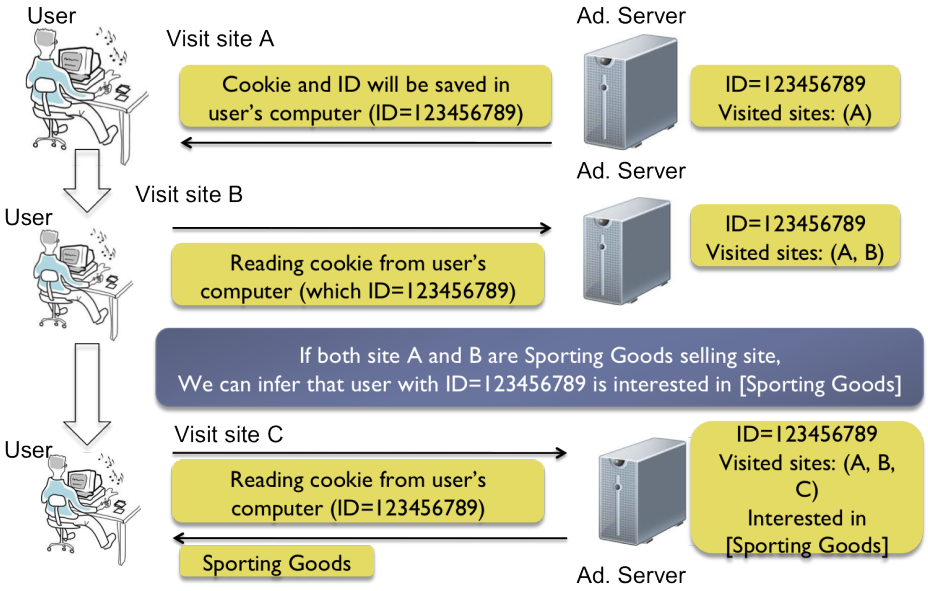

Figure 2.5: How cookie could be used to track user's behavior

As we can see above, advertising companies can track you even over multiple different websites effectively. They use either Cookies, Super Cookies. By linking the Internet users profile records of each new page they had visited, these companies can build up a long term profile of most of the things users do on the Internet with browsers. According to an investigation from Peter Eckersley, one of the largest online jobs site in the United State included JavaScript code from 10 different tracking domains[13]. Peter Eckersley argues the fact that when you search for a job online, your profile including personal information also be sent to dozens companies that you are event not awareness. Further more, a recent research paper by Balachander Krishnamurthy and Craig Wills shows that social networking sites like Facebook, LinkedIn and MySpace are giving tracking companies an easier way to add your name, lists of friends, and other profile information to the records they already keep on you[2]. The main theme of the paper is that when you log in to a social networking site, the social network includes advertising and tracking code in such a way that the 3rd party can see which account on the social network is yours. They can then just go to your profile page, record its contents, and add them to their file. Of the 12 social networks surveyed in the paper, only one (Orkut) didn't leak any personally identifying information to 3rd parties.

2.4 Summary

This chapter covered the big picture of current cookies from the point of view of privacy concern caused by existence of them. Beside the original cookies, we introduced about the super cookies the ones that serve the same as original cookie but have ability to hold a much more larger amount of data and even difficult to detect and delete. However, since the claim on privacy raising inside the big group of users, super cookies's privacy is maintaining and could be control or delete by user even they are still standing beside browser's control. We also explain in details how cookies can be used to track user's behavior through sites or big advertisement network, etc. Considering these problem, in this thesis, we proposed an even more difficult method also privacy concern that users can not avoid of being tracked using browser fingerprint.

Chapter 3

Related work and Research

3.1 Browser Fingerprinting Techniques

3.1.1 How unique your browser is

On September, 2010, research carried on by experts at EFF [1] shows that in the best case it is possible to expect that only one in 286,777 other browsers will share the same configuration[3]. Meanwhile, whenever the Internet users connect to server by modern web browsers, there are much sensitive information transmitted upon each request. That makes modern web browsers become "device fingerprinting" via that available information. They collected a large samples of browsers that visited the test site, and observed that the distribution of fingerprint contains at least 18.1 bits of entropy to distinguish browser with the others. Even more, among browsers that support Flash or Java, they show the situation is worse, with the average browsers carrying at least 18.8 bits of identifying information. As a results, 94.2% of browsers with Flash or Java were unique in the collected samples of EFF

In this research, they also discuss what privacy threat browser fingerprinting in reality life and how to protected users from that threats. Unfortunately, there is paradox since each protection method could become fingerprinting information which carry a tradeoff between protection against fingerprintability and certain kinds of debuggability, which in current browsers is weighted heavily against privacy.

[1] Electronic Frontier Foundation

3.1.2 Fingerprinting Information in JavaScript Implimentation

Besides many techniques to try to make fingerprint devices or users on the Internet, the authors at University of California and University of St. Thomas (USA) introduced a very new method to make browser fingerprint by using browser's JavaScript execution time that can not mitigate in practice[14]. The use of JavaScript execution time fingerprinting technique can be use to detect not only browser's version but also operating system and micro-achitecture due to the dependence of JavaScript Engine on clock speed of CPU.

In addition, toward the Firefox user, more specific user who installed NoScript extension, could be fingerprinted by determine the domains name added into the whitelist.

Even their experiments represent a lower bound of the effectiveness, they also emphasize the conclusion that "it is possible to distinguish between browsers, along with the underlying system hardware and software, based solely on scripting benchmarks". And in deeper investigation, it is believed to give more informative details such as hardware revisions with a processor family or even clock speed of CPU, cache size and amount of RAM [2] on the target system. Presented techniques also are able to work properly on mobile devices in expectation.

3.2 Other user tracking techniques and Privacy related

3.2.1 User Tracking by Digital Information

On behalf of researcher on privacy field, the authors make sensitive digital information that can be used to profile users come to the light[15]. They proposed a method to capture information on the computer network and classify informative one in three categories. According to this thesis, each time a user profiled, his/her activities such as when he/she is active on the network, or location of the computer, etc, would become visible. This makes a big threats against users privacy since all their activity become visible to the others who are using give tracking system.

The result also showed that both specifying an individual user and profiling users' activities was possible based on the method presented in the thesis. Besides, the controversial discussion on privacy problems and the guidelines for users to ensure their privacy and personal information were also given in this thesis.

```
GET /clk; 203330889; 26770264;z;u=ds&sv1=170988623...
Host : ad.doubleclikc.net
Referer : http://www.facebook.com/profile.php?
          id=123456789&ref=name
Cookie : id = 2015bdgb9ec || t = 1234359834 | et = 730 | cs = 7aepmsks
```

(a) Via Referer Header

```
GET /__utm.gif?..utmhn=twitter.com&utmp=/profile/jdoe
Host: www.google-analytics.com
Referer: http://twitter.com/jdoe
```

(b) Via Request-URI

```
GET ...&g=http%3A//digg.com/users/jdoe&
Host : z.digg.com
Referer : http://digg.com/users/jdoe
Cookie : s_sq = ... http%25353A//digg.com/users/jdoe...
```

(c) Via Cookie

Figure 3.1: Leakage of Online Social Network ID to Third-parties[2]

3.2.2 Leakage of personally identifiable information

A recent research paper by Balachander Krishnamurthy and Craig Wills shows that social networking sites like Facebook, LinkedIn and MySpace are giving the hungry cloud of tracking companies an easy way to add your name, lists of friends, and other profile information to the records they already keep on you[2].

The main theme of the paper is that when you log in to a social networking site, the social network includes advertising and tracking code in such a way that the 3rd party can see which account on the social network is yours. They can then just go to your profile page, record its contents, and add them to their file. Of the 12 social networks surveyed in the paper, only one (Orkut) didn't leak any personally identifying information to 3rd parties.

There are some interesting technical details in how the social networking sites leak this data. In some cases, the leakage may be unintentional, but in others, there is clever and surreptitious anti-privacy engineering at work.

[2]Random Access Memory

3.3 Summary

This chapter described two newest related Browser Fingerprinting techniques : Browser Fingerprint with Information that sending to server upon each HTTP request[3] and Browser Fingerprint information in JavaScript Implementation[14]. In this thesis, information that we used to create Browser Fingerprint about the same as [3] but we are going to evaluate a stronger statement wherever Browser Fingerprint can be linked together or NOT. This chapter also covered two privacy related research about the leakage of digital information on wifi Network and leakage of identifiable information via Social Networks. These described problems give more details about current privacy research status.

Chapter 4

Browser Fingerprint Methodology

As we introduce in previous chapters, the most common way to track users is using HTTP Cookies, often set by third party analytics and advertisement domains.

However, there is growing awareness among web users that HTTP cookies are serious threat to privacy and many people now block, limit or periodically delete them. Awareness of Super Cookies is lower but vendors like Adobe is making their Super Cookies comply with the browser's normal HTTP cookies privacy settings because of claims from many researchers and normal users.

In their research paper, researchers from Electronic Frontier Foundation said that "An user seeking to avoid being follow must pass three tests: The first is tricky: find appropriate settings that allow sites to use cookies for necessary user interface features, but prevent other less welcome kinds of tracking. The second is harder: learn about all the kinds of Super Cookies, perhaps including some quite obscure types, and find ways to disable them. Only a tiny minority of people will pass the first two tests, but those who do will be confronted by a third challenge: fingerprinting"[3].

As a tracking mechanism for use against people who limit cookies, fingerprint-ing may be much harder for investigators to detect than Super Cookie methods, since it collects only evident information sent over each HTTP request. In this chapter, we give a definition of Browser Fingerprint in this thesis and also a method to link Browser Fingerprints on the same devices or from many different devices together.

4.1 Definition of Browser Fingerprint in this thesis

According to wikipedia, a fingerprint in its narrow sense is an impression left by the friction ridges of a human finger[16]. In a wider use of the term, fingerprints are the traces of an impression from the friction ridges of any part of a human hand. However, the terminology "Browser Fingerprint" is

closest to "Device Fingerprint". Device Fingerprint often refers to a compact summary of software and hardware settings collected from remote computing device[17]. In this case, Browser Fingerprint is also the same. It is a combination of all information that certainly sent in every HTTP request to the Server. But it is interesting that most of users do not aware that these information could be collected to against their privacy issues.

We implemented a browser fingerprinting algorithm by collecting a number of commonly and less-commonly known characteristics that browsers make available to websites. Some of these can be inferred from the content of simple, static HTTP requests; others were collected by AJAX . We grouped the measurements into separate strings, though some of these strings comprise multiple, related details. The fingerprint is essentially the concatenation of these strings.The source of each measurement and is indicated in Table 4.1.

Variable	Value
User Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_1)
Plugin	QuickTime Plug-in 7.7.1
HTTP ACCEPT Header	text/html, application/xml; q=0.9
System font	Abadi MT Condensed Light
Screen Size	1440x900

Table 4.1: Browser Fingerprint sample

To create each fingerprint like above, we collected the following information. Table 4.1 shows an over-viewer of each individual Browser Fingerprint fields.

- HTTP request header : According to Wikipedia[18], HTPP headers fields are components of the message header of requests and responses in the HyperText Transfer Protocol (HTTP). They define the operating parameters of and HTTP transaction. There are many fields in a HTTP request header but in this thesis, we only extract and use some of very common fields that help to identify browser, OS (Operating System) and used in context to distinguish client side settings (Is Cookies, Local Storage, etc, turning on)

- Installed Plugin (in case Firefox, Chrome): Modern browser provide a rich set of interfaces to extent browser's function called "plugin". In many case, Web users install plugin to help themselves having better experience when interacting with Web Page . How ever, it turns out that, these set of plugin could be very different from other users due to individual characterize

and individual needs. In browser finger print, it is one of the very strong fields that help us distinguish users's browser. In a recent research[3], the experts from Electronic Frontier Foundation shows that information from plugin list can hold average 15.4 bits information.

- Installed Font : We collect set of installed font on user's computer through java and flash object. Like plugin set, collected font set is also very strong to identify and link browser on the same computer. On the next section, we will specify m ore details about the idea to use font as an element to link browsers that are used on the same computer together.

- Request-URI and HTTP_Referer: These fields are optional fields of HTTP request header that allows client to specify the address of the document (or element within the document) from which the URI in the request was obtained and request. The reason why we put these two fields into one field of Browser Fingerprint is these two fields play the most important roll in analysis access from user to squeeze into some big categories from which we could hope to link Browser Fingerprints from many devices. This filed combine with access time may become very powerful identifiable information in tracking users.

- Screen resolution and Color Depth

- Access time and client time zone

<table>
<tr><td colspan="4">Field name</td><td>Description</td></tr>
<tr><td colspan="4">HTTP Request Header</td><td rowspan="2">Message header of request and response in open HTTP transaction</td></tr>
<tr><td>User Agent</td><td>HTTP Accept</td><td>Accept encoding</td><td>Accept Language</td></tr>
<tr><td colspan="4">Install Font List</td><td>List of system font that installed on the client side (using Flash object)</td></tr>
<tr><td colspan="4">Install Plugin List</td><td>List of extension supported by major modern browsers</td></tr>
<tr><td colspan="4">Request-URI and HTTP_Referer</td><td>Get currently accessing pages and reference pages</td></tr>
<tr><td colspan="4">Screen resolution and Color Depth</td><td>Screen resolution of devices and color depth</td></tr>
<tr><td colspan="4">Access time and time zone</td><td></td></tr>
</table>

Table 4.2: Browser Fingerprint Fields

As individually, it is thought that each field has individual roll in identifying users. Which fields carry the most or least information in tracking users will be discussed more on Chapter 6 when we evaluate our method.

4.2 Link Browser Fingerprints on the same computer

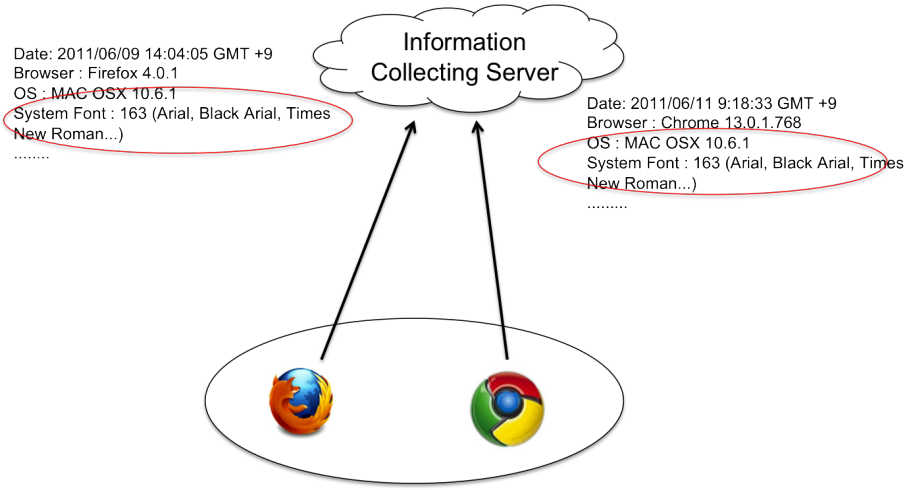

Figure 4.1: Method to link browser on the same computer together

In this section, we are going to write about a method to link different browsers on the same together when an user are using many different browsers on the same computer.

The basic idea is using information that will not change through different browsers. In this case, system font information is the strongest information. According to [3] fonts carry about 13.9 bits of information in order to identify different browser. Even in the same research, the researchers at EFF claimed that Plugins contain average 15.4 bits of information, but since they do NOT calculate the ability of fonts to link browsers on the same computer, we expect that Fonts contain more information than any other fields in Browser Fingerprint. Figure 4.1 above shows that by comparing obtained font set, we will obtain the results if two browsers on the same computer have the same font set in fingerprinting information. But experimental results also showed that obtained fonts in Browser Fingerprints of the same computer do NOT need always the same. They can be different from each other, however, we found an important relation between two font sets from Browser Fingerprints of the same computer. Typically, Font set of Browser Fingerprints from browsers on the same computer are expected to be the same, but in some cases they are set and subset of each other. In practical, this method is expected to detect fingerprinting information of browsers on the same devices very effective.

4.3 Link Browser Fingerprints of different devices

In general, different devices' browsers will have a completely different fingerprint. Therefore, it must be very difficult to link fingerprint of different devices together. However, here, we present several methods in practical. Even the effective still need debating, we present our method as a case study. The limited results obtained from this method would become motivation for us in future work in order to make this research more valuable.

Case 1

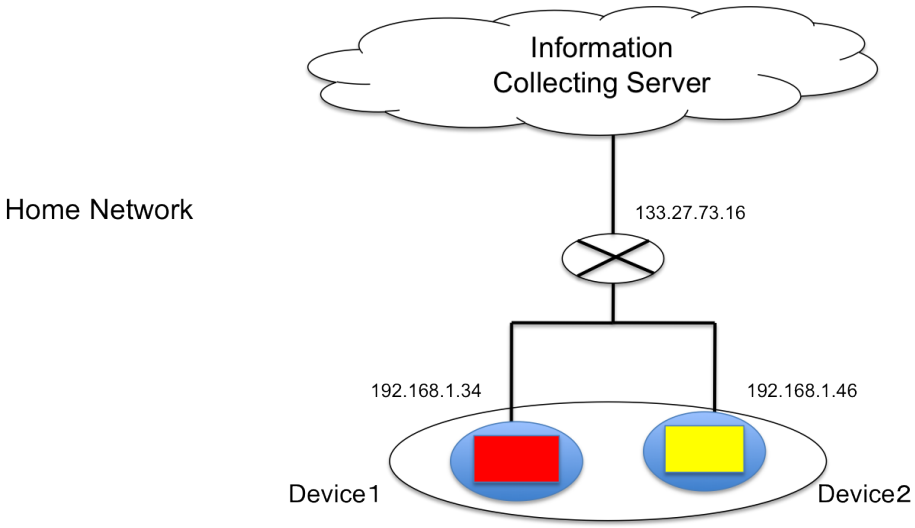

Figure 4.2: Home Network and the availability to link Browser Fingerprint from different devices

It turns out that at Home Network, in spite of multi devices are using to access to web server, there is the same IP Address remaining on the server side. In some cases, even it is possible to guess whenever Browser Fingerprints from browsers on the same network, it is difficult to make conclusion whenever these browsers belong to one user or NOT. Especially, at company network or at home services where multi-user taking activities there. In addition, IP Address is changing from time to time and the new IPv6 is replacing IPv4, it is overwhelmingly trackable with IP Address only.

Case 2

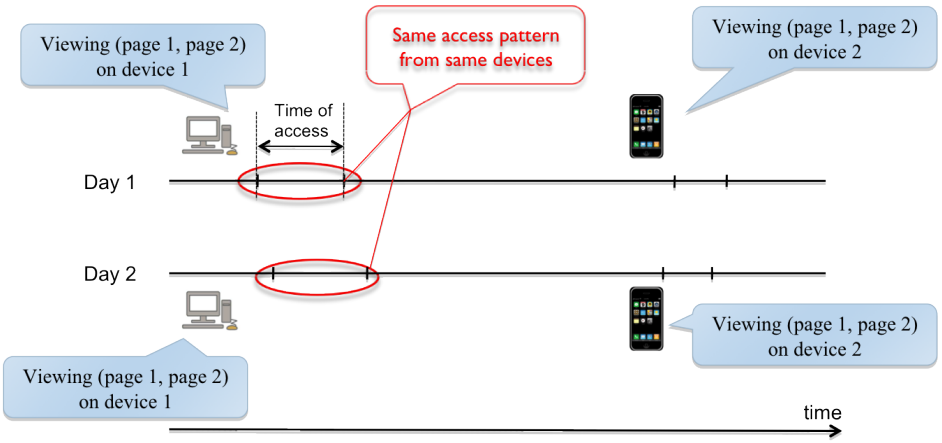

Figure 4.3: Access Pattern and Availability to link Browser Fingerprint from different devices

Mining traversal patterns on the Internet is one of critical issues for exploring the user access behaviors. And it is thought that if we based on some factors such as, the pages user had visited or based on time user had visited that web page, it could possibly link Browser Fingerprint on different devices together. Figure 4.3 shows a typical case where we could see clearly the correlation relation between two Browser Fingerprints. Based on the fact that an user may have a habit to visit only his interested pages again and again over time, we could link two completely different Browser Fingerprints in the following scenario. For example, in the first day, there are access from device 1 to page 1 and page 2 for about t1 time, device 2 to page 1 and page 2 for about t2 time. About at the same time on the second day, the same device (suppose each device's fingerprint is unique) accessed to the same pages with the same time of view. In this case, there is highly possibility that these two device fingerprint from the same user.

In fact, there is a need to collect a large set of Browser Fingerprint from a normal site and analyse user access pattern to prove these above scenario. But since it is limited in experience, we only could investigate how efficiently a small group of Users could be squeeze with access pattern field from Browser Fingerprint.

4.4 Summary

In this chapters, We gave a clear definition of Browser Fingerprint and introduced ideas to link multi-browsers on the same computer and from different devices together based on Browser Fingerprint. Browser Fingerprints from the same computer can be linked by the shared Operating System Characteristic where fonts are the strongest factor. And beside the difficult to link Browser Fingerprint from different devices, we present our method as a case study in this chapter. The ideas are quite simple and seem easily to implement sometimes could be a very effective method which we will prove in evaluation in the next chapter.

Chapter 5

System Design and Implementation

This chapter describes the implementation of Online User Tracking System that used in this thesis. In order to satisfy the system requirements, which will be introduced later in the same chapter, we divided the system in smaller units, and we will mainly explain the Information Collecting modules which is the most important part of this system.

5.1 System abstract

This section gives the first glance at Online User Tracking System that developed along side with this thesis. Beside describing about system requirements, we also explain how this system work and what kind of data we are collecting from users. Experiment set up that is slightly different from these specifications will be explained more details in a different chapter

5.1.1 An Overview and System requirements

The most important purpose of developed this system is tracking users' online behavior. However, "Tracking" here does not only mean that the system has the ability to follow which pages user has accessed or what kind of data user has downloaded. In this thesis, we define some more stronger requirements that require this system must follow an user and must know if that user access to the server again. Besides, based on the fact that, the Internet users currently does not use only one browser but maybe many different browser at the same time, we make a further more condition that system must satisfy : Detect the same user if he/she is using many browser from the same computer or even from different devices each time he/she accesses to the Web server.

In fact, modern web browsers and browser plug-ins provide a rich set of interfaces for many

Rich Inter- net Applications. By customizing their browsers using plug-ins or changing the default settings, users should aware that these settings could be used to profile them because among many others Internet users, these settings might be very different. By analysing carefully which information can collect in order to track and profile user, we decided to collect the follow information that certainly sending upon each request connection and call them browser fingerprint as we defined in Chapter 4

- HTTP request header
- Installed Plugin (in case Firefox, Chrome)
- Installed Font
- Remote host IP address
- Terminal screen size

We divided the hold system in three big modules : Plugin collecting module, Client font collecting module, and HTTP Header collecting module. Each module's implementation will be discuss in detail in next section.

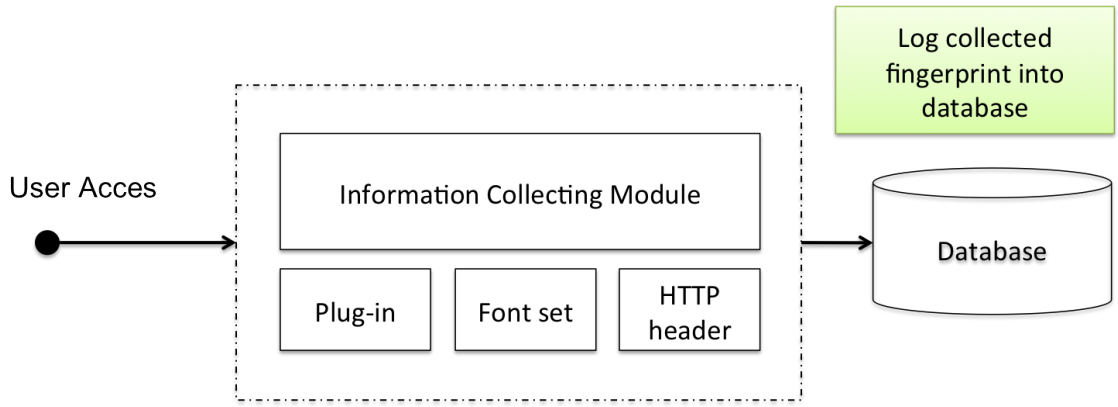

Figure 5.1: System Overview and some most important modules

Figure 5.1 gives us an overview of the system.

To more details, this thesis's scope is not tend to implement a completely Online User Tracking Platform but we only implement the most important part Information Collecting modules and log the collected information into database. User Tracking step will be done by analysis the correlation between logs in order to archive the goals of the thesis.

In additional, current specification is slightly different from experiment set up. In experiment, in order to analysis the accuracy when link browser together, we set cookies on every visited

browsers that their settings allow cookies. This definitely have no other usages except supporting to analysis steps. More details will be discussed in Chapter 6 where we will explain more details about experiments.

5.1.2 How it works

The following figure shows an image of how this system work.

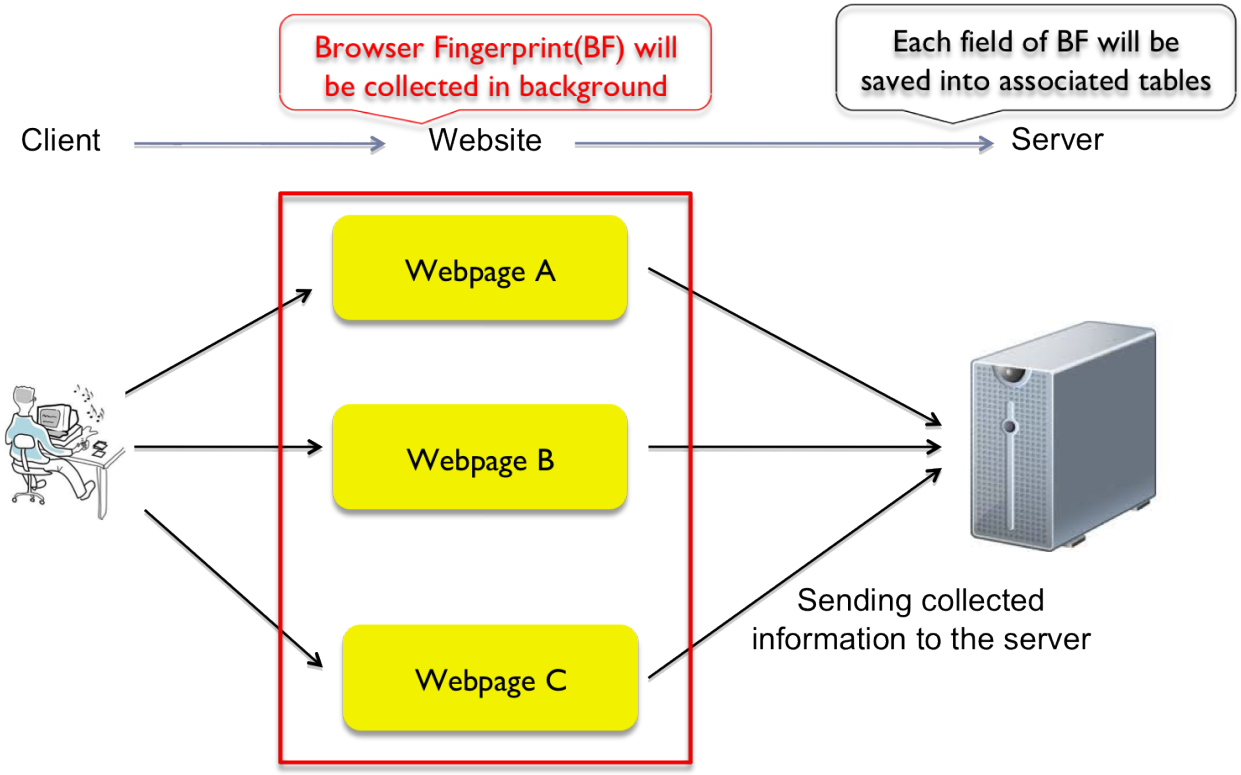

Figure 5.2: System Image

Figure 5.2 shows how system work. In general, system will be implemented on the server side and run as a background program. In other words, Information collecting modules will work behind the web server and each time an user access to web pages, every needed information will be collected and sent back to web server. Here data will be divided into some fields and inserted into associated tables. In this system, information will be collected each time users access to a new page (on the same domain, or refresh the current page).

5.2 Database Schema

In order to easy to manage data, we divided browser fingerprint into some fields like we have defined in previous chapter. Data will be save in normal way into Database and divided into some tables. Here is the main tables of database schema.

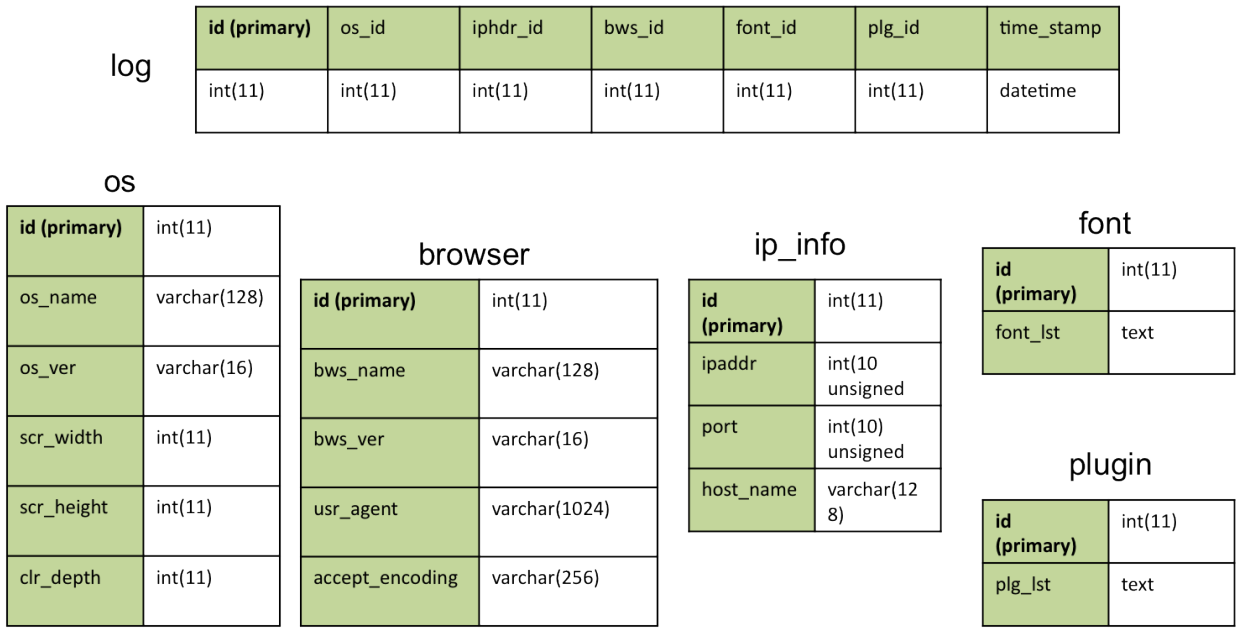

Figure 5.3: Database Schema

As we can see in Figure 5.3, each table will hold a field data of collected fingerprint and joined together in "log" table in order to easy to manage. In practical, we implement data base using MySQL (version 5.1.49) as a Relational Database Management System.

5.3 Information Collecting modules and Implementation

System that develop along side with this thesis will be implemented in the following environment (see table 5.3)

Development Language	Development OS	Database Management System
PHP + JavaScript	Ubuntu Server 10.10	MySQL version 5.1.49

Table 5.1: System Enviroment

In this section we will discuss in details about how each modules implemented. Information Collecting modules is the main and also the most important modules in this thesis. As we introduced in section 5.1 we divided this big modules into three smaller ones : HTTP Header Collecting Module, Font Collecting Module and Plugin Collecting Module.

5.3.1 Implementation of HTTP Header Collecting Module

HTTP Header is composed from many components of the message header of requests and responses in an open HTTP transaction. It is already known that there is many fields in header but here we only collect some popular fields such as User-Agent, Accept-Encoding and Accept-Language fields. The default JavaScript Engine or server side also support function to collect these information very easy and effective.

5.3.2 Implementation of Plugin Collecting Module

Plugin is a terminology refers as a small program or set of software components that adds new functions or abilities to larger program. Browser plug-ins often offer a rich set of interface and functionality to extent browser's ability from customize the way browser look to add new function that does not default implemented in browser. Currently, Chrome and Firefox are two major ones fully support plug-in with their very rich set of plug-in from their stores. Besides, Opera also support and these three major browsers's plugin can be easily get by using JavaScript

Here is sample source code :

Listing 5.1: JavaScript Implementation of Plugin Collecting Module (Sample pseudo Code)

```
Bros.getPluginList = function(){
  // This code should work properly in general Chrome, Firefox and Opera browsers
        var list = "";
        for(var i = 0; i < navigator.plugins.length; i++){
                list += navigator.plugins[i].name + "/" + navigator.plugins[i].
                    description + "/" + navigator.plugins[i].filename + ",";
        }
        return list;
};
```

In Internet Explorer, even plugin is also supported but things become much harder since Microsoft has created a different standard from the other browser vendors for plug-ins. In this case we used library to get less information, only name or version of some popular plugin such as Java, Flash and Adobe Acrobat PDF preview if they are installed

Listing 5.2: Internet Explorer Plugin name and version guess (Sample pseudo Code)

```
// In case browser is Internet Explorer
if (list == "") {
    var pp = new Array();
    pp[0] = "Java"; pp[1] = "QuickTime"; pp[2] = "DevalVR"; pp[3] = "Shockwave";
    pp[4] = "Flash"; pp[5] = "WindowsMediaplayer"; pp[6] = "Silverlight";
    pp[7] = "VLC";
    var version;
    for ( p in pp ) {
      version = PluginDetect.getVersion(pp[p]);
      if (version)
        list += pp[p] + " " + version + "; "
    }
    list += ieAcrobatVersion();
  }
  return list;
};
```

5.3.3 Implementation of Font Collecting Module

Basically, system font from user is not revealed through JavaScript. However, here is a tricky techniques where we try to embed an Flash object into webpage and manipulate this object with some help from other JavaScript Library.

Listing 5.3: Client Font Detection (Sample pseudo Code)

```
// Client Font detection
var fontDetect = new FontDetect("font-detect-swf", "flash/FontList.swf", function(
    fd){
                        var fonts = fd.fonts();

                        for (var i = 0; i < fonts.length; i++) {
                                Bros.font_lst += fonts[i].fontName + ", ";
                        }
}
```

5.4 Summary

This chapter has detailed the implementation of an Online User Tracking System that developed along side with this thesis. Not like other web application, this system run in background and return collected information back to the Server side and insert into database. Beside specify database schema, we also explain in details how to implement each module and also provide some sample source code. In the next chapter we will set up the testing environment and evaluate experiment results of this system.

Chapter 6

Experiment and Evaluation

In this chapter, we will present method and set up of experiments, which was used to evaluate the performance and accuracy of proposed method. Through this chapter, we will explain evaluation fields and also how experiment sets up. In the last sections, beside showing the result, we also have some discussion and counter measurement to defend against fingerprint and how to improve user privacy.

6.1 Evaluation Overview

As we mentioned in Chapter 1, in this thesis we produce a method to track user's online behavior with Browser Fingerprint. In order to prove that without cookie we can still track users' browsing history each time they access to web server, we must evaluate the unique of Browser Fingerprint to ensure that fingerprint set is unique in all accesses. In addition, we also argued an even stronger statement that Browser Fingerprinting can be linked together in case users use many browsers from the same computer or other devices.

Therefore, in evaluation there is need to evaluate the three following items :

- How unique the browser fingerprint is : Each access to Web Server will be fingerprinted and we must evaluate that if Browser Fingerprint could be used to identify and distinguish each access with the other. That means Browser Fingerprint of the same browser must be the same and different from the other one.

- How effective that browser fingerprint on the same computer can be linked together : Since the Internet users may use many browsers at the same time, we still expect that there is a correlative relation among Browser Fingerprint of the same computer. This experiment is to

test if we could efficiently link them together.

- How effective that browser fingerprint from many devices of an user can be linked together : Furthermore, not only linking browsers from the same computers, since the Internet users may schedule for viewing a number of the same pages at some specific time from their browsers, there is a hope to link Browser Fingerprints of their devices together in a limit number of cases. By building a scenario and even also setting up experimental scenario, we evaluate that if we could possibly find out the relation of two completely different Browser Fingerprints.

Because these experiments have the different types of setting up and evaluative methods, in each individual experiment, we will explain in details how to set up and also type of evaluation with results and discussions.

6.2 Experimental set up

Experiments use Linux Ubuntu Server Kernel 2.6.35-31 from laboratory. On this, we are launching a web application to share information among each laboratory's members which we call "Wiki". Our developed program is setting up in the background so that, each time, an ISC member accessese to Wiki, Browser Fingerprint will be collected, and the results will be saved into database. Figure 6.1 shows the experimental environment.

There are three types of experiment carried on in this thesis as we defined in previous section, where we must evaluate the unique of browser fingerprint, the effectiveness method that allows us to link browser fingerprint on the same computer and the last one, the effective of method that links Browser Fingerprint on the other devices together.

In the first type of experiment, we just set up and asked experimenters to access to server, logged information will be analysed carefully fields by fields. To more details, currently in our definition of fingerprint, there are many fields chosen. However, to evaluate which fields is the most or least important, and also, which fields contain the most numbers of information bits in order to distinguish Browser Fingerprints of different browsers, there is a need to analyse the roll of each field individually and also in combination together.

In the second and the third experiments, we asked about 20 experimenters using their browsers devices to access to set up experimental pages. We questioned experimenters to accept Cookies from the Server, and after that, we take note of Cookies in order to check if we could link Browser Fingerprints correctly. Again, this setting up is a little bit different from developed program. Here cookie is set to check the accuracy of our proposed methods only, and it does not have any other

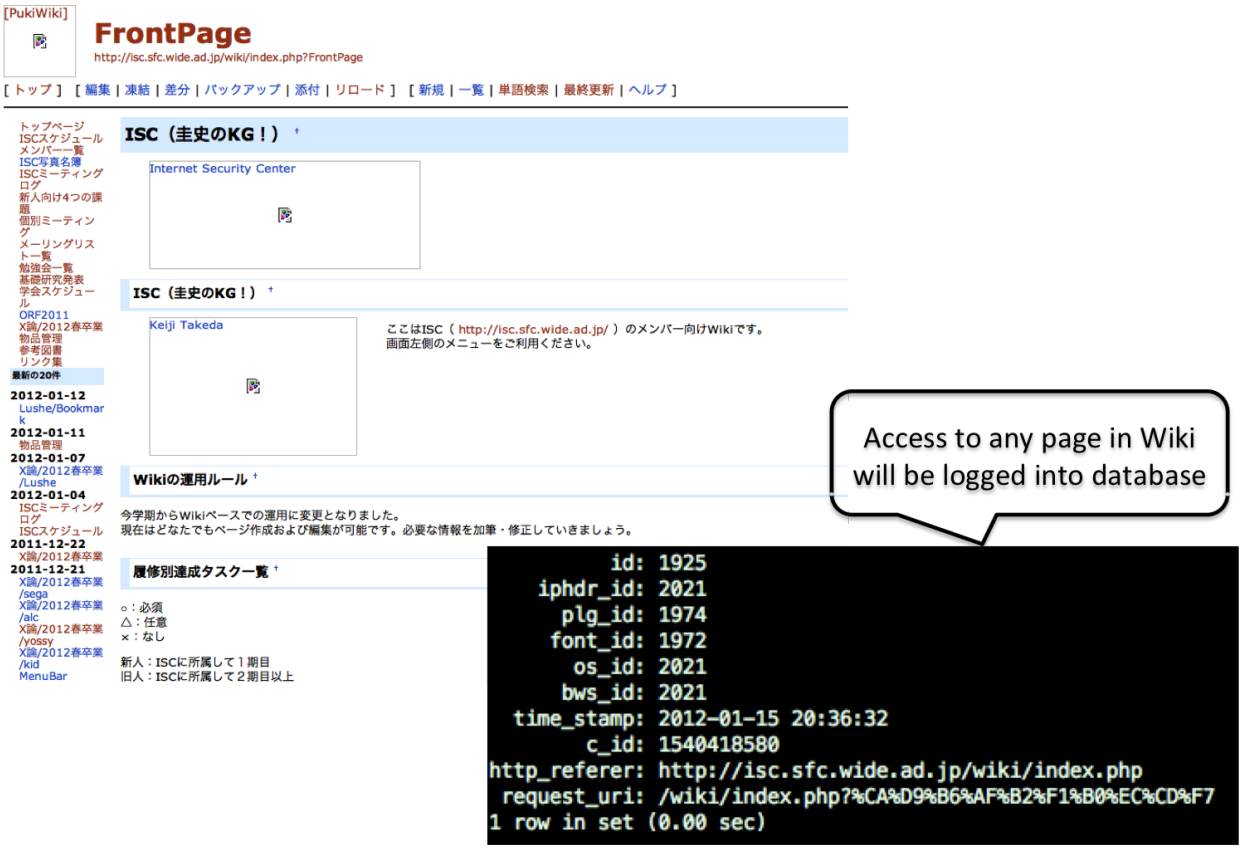

Figure 6.1: Experimental Setup

purposes.

Evaluation results and discussion about the results will be shown in the next section and its subsections.

6.3 Evaluation results and discussion

In this section, besides showing the results from experiment, we also discuss about availability of using fingerprint to track users. Some privacy problems, security concerns, and also counter measurement of how to defend against to be fingerprinted also mentioned in this section, too.

6.3.1 The unique of Browser Fingerprint

Evaluation Method

Among users who accepted cookies from our system, we analyse their Browsers Fingerprint and check the result again with Cookies. The same browser will content the same cookie value. Therefore, we only need to calculate and analyse how different of fingerprint sets that have different

cookie value in log table.

Result

Fingerprint is constructed from the following set :

- User Agent
- HTTP Accept Header
- Browser Plugins
- System Fonts
- Screen size and color depth
- IP Address

In order to evaluate the roll of each field in distinguishing each fingerprint, we carried on a test that showed how unique each field in fingerprinting information. Testing with User Agent, HTTP Accept Header, Browser Plugins Set and System Fonts Set, we obtained the result as the following

Total different browsers	177
Different User Agent	112 (63.28%)
Different HTTP Accept	10 (5.65%)
Different Plugin sets	128 (73.32%)
Different Font sets	106 (59.88%)

Table 6.1: The unique of Browser Fingerprint

During the experiment, among 2131 number of access we obtain only 177 different cookie values, therefore, in this case, we can say that, there are about 177 different browsers in this experiment. By analysing obtained log from user, we got the results that shown in Table 6.3.1. The results show that if we analyse each field of fingerprint individually, there are 112 different User Agent among 177 different browsers (63.28%), only 5.65% access browsers have different HTTP Accept field. Note that HTTP Accept is a field inside HTTP Accept Header. It is completely different from HTTP Accept Header. In addition, 73.32% browsers have different Plugin sets and the different Font sets are figured out around 59.88% among all browsers.

Table 6.3.1 was cited from research about the uniqueness of browser fingerprint and the number of information each of fields is carrying at Electronic Frontier Foundation [3]. According to this

research, User Agent contents 10.0 bits of information, Plugin contents 15.4 bits of information, Font contents 13.9 bits of information and HTTP Accept contents only 6.09 bits of information. Comparing this experiment, it is appeared about the same as our obtained results because "bits of information" here means how much information each filed carrying in order to distinguish from other browser fingerprints.

Varibale	Entropy (bits)
User Agent	10.0
Plugins	15.4
Fonts	13.9
video	4.83
supercookies	2.12
http_accpet	6.09
timezone	3.04
cookies_enabled	0.353

Table 6.2: PanoptiClick Fingerprint and number of bits of sensitive information ([3], Appendix A)

In this experiment, we expected System Fonts set content the highest information to identify the browsers. However, looking deeper inside the experimental logs, since currently there is no way detect installed font sets on mobile system, we carried another analyzing where we divide access from computer and mobile system into two different sets, and analyzing again. Unfortunately, results is almost the same because there is only 60.28% different font sets among all was detected. The reason is maybe in some cases, there are accesses from browsers on the same computer resulting the same font set in the experimental logs.

At last, when combining all fields into fingerprint together, since the total experimental Browser Fingerprint sets are very small, we obtained that there is obviously no same fingerprint in all experiment set.

The same configurations, The same Browser Fingerprint

Despite the ideal result we obtained from small experimental set, we carried on other experiments where we asked an experimenter to upgrade all their software and mobile OS system to the latest version and install the latest Opera mini version. After analysing browser fingerprint we obtained the same fingerprint from two mobile devices as showing in table 6.3.1. It is thought that completely

different devices can have exactly the same Browser Fingerprint. Especially in the same computer system of the same companies where all computers are set up exactly the same way or in mobile systems, it does not seem that Browser Fingerprint is an effective way to track users.

Variable	Value
User Agent	Opera/9.80 (iPhone; Opera Mini/6.5.2.26801/26.1395; U; en) Presto/2.8.119 Version/10.544
Accept Encoding	gzip, deflate
Accept Language	EN
HTTP ACCEPT header	text/html, application/xml;q=0.9, application/xhtml+xml, image/png, image/webp, image/jpeg, image/gif, image/x-xbitmap, */*;q=0.1
Plugin	Does not supported on Opera Mobile for iPhone
System font	Can not get system font sets from Mobile System
Screen Size and Color Depth	416x320x4

Table 6.3: Same Browser Fingerprint from two mobile devices

6.3.2 The effectiveness of method that link browser fingerprint on the same computer

Evaluation Method

In this experiment, we asked twenty experimenters to accept our cookie and access to our experiment pages. They are asked to install at least 2 browsers on their computer, and asked to access to our experiment pages using all installed browsers. After information collected, we took note of all cookies and created a cookie set for each of them. In practical analysis, we will compare Browser Fingerprints from the same set (each browser in the set is specified by unique Cookie ID), if they could be linked efficiently by our method. Font set differences also mention in this experiment.

Result

Following the Cookies provided by twenty experimenters, obtained results show that there are only four (20%) experimenters that have exactly the same fonts set from different browsers. Analysing log results showed that in the remaining of 80% experimenters leave exactly two different font sets even they accessed to the Web server using at browsers on the same computer.

We decided to analyse these two font sets on each case of each experimenter again. Firstly, in each font set we sort fonts in lexicographical order and try to pick up all the common fonts share by that two ones. We figured out that, even these two ones do not have to be the same, but they do have a relationship : **set and subset of each other** because the number of common fonts in two sets equals the number of fonts in less element one. Analysing results on all 80% experimenters, we have also obtained the same results. Furthermore, looking into browser name, we know that, on the same computer of experiments, if they use (Firefox and Opera) or (Chrome and Safari and Internet Explorer), two fingerprints from browsers in the same group will have exactly the same font set as the other. In other word, if we divide the most five common browsers (Firefox, Chrome, Internet Explorer, Safari and Opera) into two groups : Group one (Firefox, Opera) and Group two (Chrome, Safari, Internet Explorer), then when we access to experimental pages using two browsers of the same groups on the same computer, obtained browser fingerprints will share the exactly the same font set. And vice versa, if these two browsers do NOT belong to the same group, the will have set - subset relation. More concrete, Font set in fingerprinting information of browsers from group two will be the subset of the ones in group one.

6.3.3 The effectiveness of method that link browser fingerprint from different devices

The same as the second experimentation, besides asking experimenters to accept our cookie and access to our experimental pages, we also asked experimenters to choose two random pages and access every day from their smart phone and their personal computer. The main purposes of this experiment is to prove that : Traversal Pattern or Access Pattern some time could become identifier in linking web browsing history. Figure 6.2 shows a typical case where browser fingerprint from others devices can be linked together.

Since this experiment does not sound natural because we asked experimenters to access to our pages on a specific time and they are also asked to access the same pages from the their devices. In practice, we must set up and follow Traversal Pattern and access time to limit or divide users in some groups to find an efficient way to link their Browser Fingerprints together. However, we leave this as one of the future work since we failed to extend our experimental scope.

Back to current experiment results, among twenty experimenters there are only three experimenters have accessed the same two pages on the same devices as asked. Fingerprint information from the others therefore looks like they have accessed random pages from the home page of experiment wiki pages. This makes information overwhelmingly trackable. And we fail to link fingerprint from their two devices. Because we cannot ignore the experimenters who failed to follow the experiment set up, we only can say linking two completely different Browser Fingerprints from two completely different devices still is the unsolved problems in this thesis.

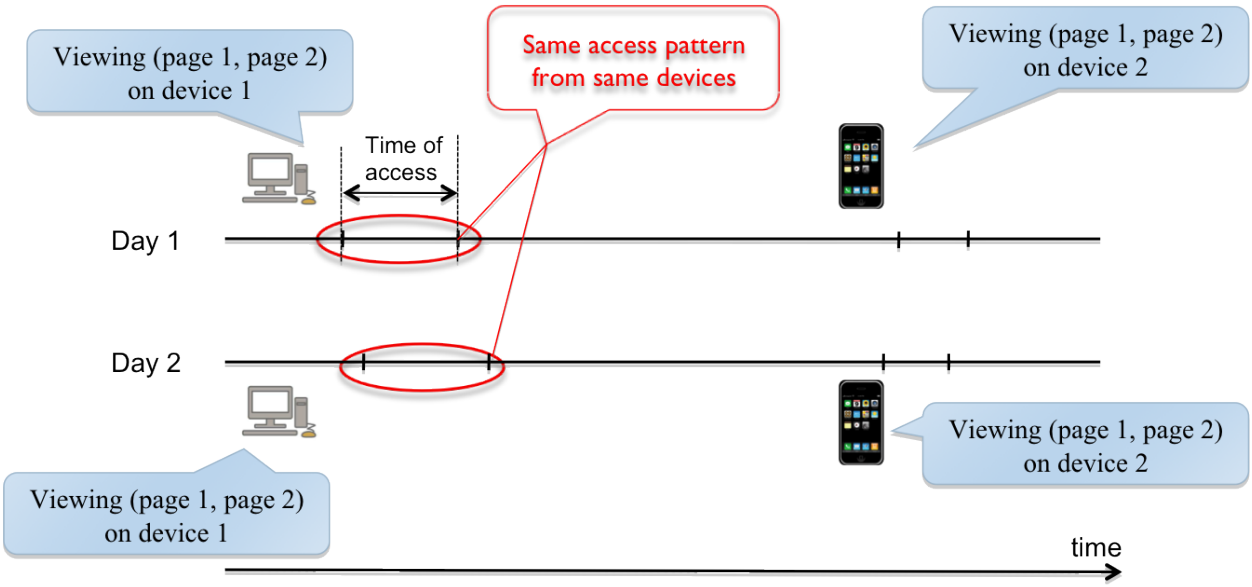

Figure 6.2: A typical case where browser fingerprints from different devices can be linked together

6.3.4 Discussion

Let us start with an example. If we ask whether a fact about a person identifies that person, it is not easy to simply answer yes or no. If all you know about a person is their ZIP code, you don't know who they are. If all you know is their date of birth, you don't know who they are. If all you know is their gender, you don't know who they are. But it turns out that if you know these three things about a person, I could probably possible to know who that person is! Therefore, each of the facts is partially identifying.

In case of browser fingerprint, things seem to appear the same like above example in many aspects. Obviously, in experiment we obtained that there are 112 different User Agent among 177 different browsers (63.28%), only 5.65% access browsers have different HTTP Accept field, 73.32% browsers have different Plugin sets and the different Font sets is figured out around 59.88% among all browsers. But when testing set is very small (around 200 like in this experiment), all browser fingerprint is different from each other when we combine all these fields together. However, according to the same related research about Browser Fingerprint, in the particular sample of browsers observed by Panopticlick, 83.6% had unique fingerprints[3].

From this obviously fact, it is thought that Browser Fingerprint could be used in a wide range of applications. Besides availability to track user even they are using many browser at the same time, by clearly appear unique in almost large amount of fingerprint sets, the first application we can give here is, Browser Fingerprint could be used in marketing where we need to count the number of accesses from users. In different with original Cookies or recently used super Cookies, even Cookies setting is turned on or off, Web server Administrator still can track users' behavior. In spite of testing sample and experiments still need more time and need testing in wider range of scope, we are expecting results in almost stay the same. Another application is, in the ideal cases, from security point of view, this system can be used to find criminal who attempted to attack the web server because their fingerprinting information will always be logged back into server and there is no way to avoid being tracked. This does make sense since recently there are many security holes in browsers and web applications exploit by using javascript like XSS. Or even attacking by using SQL injection could be detected since we log access page parameters in to the databases. In other words, our developed system is expected to use in a wide range from small scope to a very big scope to deal with not only security problem but also normal application in business and life.

However, beside some convenience like we expressed above, there still exists a privacy concerns since each access is identifiable with Browser Fingerprint. In conclusion of their research paper, Balachander Krishnamurthy and Craig E. Wills said that "the indirect leakage of Personal

Identifiable Information via Online Social Network identifiers to third party aggregation servers is happening"[2]. That means in many case, personal information leakage through Request-URI, Referer Header to the third party on social networks. Since social network is using widely and deeply take advantages in many aspect of life, it is seem that if we could collect enough information and link with social network, a normal access to web server could be profiled and identifiable with their information currently showing on social profile. In this cases, it would become extremely critical problem because personal information leakage without users' awareness. Even in case user does not public their profile information, attacker can still easily create an account and send a friend request to the user. And once an user become the target of the attacker, it would be difficult to avoid bad consequences such as losing bank account, credit information or even much more critical.

Defense against fingerprint

After realizing above privacy concerns, the first thing every user wants to know is how to avoid being fingerprinted. In this thesis, tracking system is developed by using Javascript. Therefore, when browsing the Internet, if users turn off Javascript function of browser, it would become definitely unfingerprintable. However, since almost web application currently rent a large set of Javascript library and function, it is very inconvenient when turning off Javascript and browsing the Internet. There is an acceptable plugin in Firefox named NoScript appearing useful in enhancing privacy of user since it is reduce fingerprintable fields. However, in related research in Chapter 3, Whitelist from NoScript is fingerprintables and appear to be identifiable fields against users' privacy[19].

Moreover, User Agent Spoofing and Blocking Flash extensions seem to be an efficient method to reduce fingerprintability, but it turn out to make easier to detect users in some case. For example, if we detect Flash extension in Plugins list but we cannot get list of fonts, we can infer that user has blocked Flash function. Another example, if User Agent appears to be likely User Agent from iPhone access but we still detect Flash Plugins and detect Fonts in practical, we could easily know that User Agent has been forged. These are examples of measurements that might be intended to improve privacy but ineffectively and potentially make users become trackable.

Actually, program version or what kind of plugins and its version being installed have a big impact on software development. However, as we discuss on previous section, they also content a hight number of bit of information that makes availability to be fingerprinted. In conclusion, maybe there is a need to trade off with which part of information should be revealed, which part of information should be transparent from user. Until now, we reach the same conclusion as Electronic Frontier Foundation researchers that, still there is no efficient way to defense against

browser fingerprint. And the debate on privacy concerns should be seen from many point of views, from normal user to software development and security engineering to decide which information is sensitive to be identified but not necessary for user in practical.

6.4 Summary

In this chapter, we showed experimental results from three experiment. Besides showing the unique of fingerprint and availability to track user with browser fingerprint, we also show that fingerprint information can be linked efficiently with our method. Along with each experiment, we also have short discussions about weak points and strong points of our proposed methods. In the end of this chapter, we discuss and give conclusion about some privacy concerns raising with availability of browser fingerprint. We also give some counter measurement of how to defend against to be fingerprinted. We reach the conclusion that currently there is a need to trade off between defending and to be fingerprinted because defensive method may itself become a new field in browser fingerprint.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis proposed and implemented a way to track user online behavior by linking their browser fingerprinting information together in case a user uses a lot of browser on the same computer or on other different devices. The results showed that user online behavior can be followed by using the only certain information sent upon each request connection to server side. Even there are many cases that we can not link fingerprint together, the results is enough to make concerns from security and browser experts.

Even we do not implement a complete user online behavior tracking system, in this thesis, the Information Collecting Modules was designed and implemented as one of the most important part. Logged information has been analysed carefully and the effectiveness and accuracy have been shown in this thesis.

In comparison with cookies, this method is more effectively and prove to be stronger since fingerprint can be linked together while cookies can not. And on the best case, when information can be collected enough, it would become extremely difficult for user to aware the privacy concerns since there is still not much research carry on this fields.

We also discuss about the method to defense against to be browser - fingerprinted. And it seems that currently there is no way to get out of the paradox of fingerprinting information and method to defense against to be fingerprinted. Because, the method that use to defense may become fingerprinting information and vice versa. From point of view of a researcher, we can say that we obtain the good results but from point of view of an user, there is also a need to discuss what information and how information should be transfer to Web server in order to protect user's

privacy.

We leave the open question as the conclusion of this thesis as new open fields for researching later.

7.2 Future work

This section presents future work of the research. These work are focusing on improving the accuracy especially when we need to link browser from different devices and also the performance of the system in order to collect information in acceptable time. We also leave the evaluation in a bigger scope as the future work

Improving the accuracy

Currently, we archived some goals when linked browser on the same devices but it is still difficult when link browsers from different devices together. There is no way except building a hug profile of browsing history of user, and fingerprint can be linked base on that. In this thesis, we present the very simple way. By analysis access pattern, we could divide users in some groups but still difficult to give the conclusion whenever a fingerprint can be linked with previous one. This will be leave as the biggest question of this thesis.

Improving Information Collecting Modules

Currently our system collects fingerprint in average 10 seconds per access. Because this time's experiment was carried on a small scope of research lab, performance time was not noticed but in order to take on a bigger scope, there is also a need to improve the performance of this module in acceptable time of loading a page. We expect reduce loading time to 1 second or less.

Extending the research scope

As we can see in the results, since the fingerprint can be linked together by analysis the relative correlation, maybe the privacy problem only become seriously in case of testing in a bigger scope where fingerprint could be collected from different website, especially from popular social network sites. These experiments will make scenario built up in previous chapters come to the light and user would aware of the serious situation.

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my supervisors: Professor Hideyuki Tokuda, Professor Jun Murai, Associate Professor Hiroyuki Kusumoto, Professor Osamu Nakamura, Associate Professor Kazunori Takashio, Assistant Professor Rodney D. Van Meter III, Associate Professor Keisuke Uehara, Associate Professor Jin Mitsugi, Lecturer Jin Nakazawa.

Especially, my deepest gratitude is to my advisor, Professor Keiji Takeda who has always supported and guided me from the beginning when I had just joined to ISC research group of the Internet Research Laboratory. Professor Keiji Takeda also suggested and gave me the instructions to this thesis's topic, encouraged me to find out the problem and solve it by my own way. His patience and support helped me overcome many cristic situations in finishing this thesis.

I would also be extremely thankful to Mr. Toshinori Usui, Mr.Kunihiko Shigematsu and Mr. Yuki Uehara for their endless help and valuable comments to my thesis. Without their help, I cannot even think of writing this thesis.

I would also like to thank all ISC group members, especially Malware Researching Group mcat members, Mr.Toshinori Usui, Mr.Naoto Somi, Mr.Sekine Fuyuki, Mr.Pham Van Hung, Mr.Tomonori Yamamoto, Mr.Hiroki Yoshihara, Mr.Daido Yoshihara, Mr.Takao Oya, Mr.Reimon Arima, Mr.Hiroaki Kono, Mr.Makoto Komatsu, Mr.Do Trung Kien, Mr.Nguyen Anh Tien, Ms.Asuka Nakajima, Ms.Akane Mitsuki, Mr.Kouki Nakayasu, Mr.Kohei Tsuyuki, Mr.Yu Yoshida, Mr.Akifumi Fukaya, Ms.Urara Ozawa, who have given me numerous support and always cheer me up in my hard times.

I own my special thank to Mr.Kunihiko Shigematsu and Mr.Yuki Uehara for their sincerely help and many advices since the time I first came to the lab.

I would like to thank my friends Mr.Pham Van Hung, Mr.Pham Tien Trung and Mr.Doan Hoai Nam, Mr.Nguyen Gia for their supports in many ways.

I would like to thank to members in Murai and Tokuda Laboratory who have been so friendly and kindly supporting me with my research.

At last, I am heartily thankful to my family, including my grandparents, my parents, my younger sister. Especially Ms.Hoang Anh Huong for her greatest support in order to finish this thesis.This is the first time I have left my family to start my new life in Japan. There are many things happened and yet, by my side always, my family and she have given me strength to achieve my goals and pursue all dreams of my life.

Bibliography

[1] The Wall Street Journal. Latest in web tracking: Stealthy supercookies. http://online.wsj.com/article/SB10001424053111903480904576508382675931492.html, August 2011.

[2] Balachander Krishnamurthy and Craig E. Wills. On the leakage of personal identifiable information via online social network. In *ACM SIGCOMM Computer Communication Review Volume 40 Issue 1*. ACM New York, NY, USA, January 2010.

[3] Electronic Frontier Foundation. How unique is your browsers. http://panopticlick.eff.org/browser-uniqueness.pdf, September 2011.

[4] Facebook. Statistic. https://www.facebook.com/press/info.php?statistics.

[5] Nik Cubrilovic. Logging out of facebook is not enough. http://nikcub.appspot.com/logging-out-of-facebook-is-not-enough, September 2011.

[6] Forbes. Facebook's privacy issues are even deeper than we knew. http://www.forbes.com/sites/chunkamui/2011/08/08/facebooks-privacy-issues-are-even-deeper-than-we-knew/, August 2011.

[7] Wikipedia. Targeted advertising. http://en.wikipedia.org/wiki/Targeted_advertising/.

[8] Wikipedia. Behavioral targeting. http://en.wikipedia.org/wiki/Behavioral_targeting.

[9] Mozilla. Do not track. http://dnt.mozilla.org/.

[10] Wikipedia. Http cookie. http://en.wikipedia.org/wiki/HTTP_cookie.

[11] Katherine McKinley. Cleaning up after cookies. Technical Report version 1.0, iSEC Partners Inc, 2008.

[12] Electronic Frontier Foundation S. Schoen. New cookie technologies: Harder to see and remove, widely used to track you. https://www.eff.org/deeplinks/2009/09/new-cookie-technologies-harder-see-and-remove-wide.

[13] Peter Eckersley. How online tracking companies know most of what you do online (and what social networks are doing to help them. https://www.eff.org/deeplinks/2009/09/online-trackers-and-social-networks, September 2009.

[14] Keaton Mowery, Dillon Bogenreif, Scott Yilek, and Hovav Shacham. Fingerprinting information in JavaScript implementations. In Helen Wang, editor, *Proceedings of W2SP 2011*. IEEE Computer Society, May 2011.

[15] Yuki Uehara. Risk analysis and countermeasures on user tracking by digital information surveillance. Submitted in partial fulfillment of requirement for the degree of Bachelor, Keio University, 2009.

[16] Wikipedia. Fingerprint. http://en.wikipedia.org/wiki/Fingerprint.

[17] Wikipedia. Device fingerprint. http://en.wikipedia.org/wiki/Device_fingerprint.

[18] Wikipedia. List of http header fields. http://en.wikipedia.org/wiki/List_of_HTTP_header_fields.

[19] Scott Y. Mowery K., Bogenreif D. and Shacham H. Fingerprinting information in javascript implementation. http://www.w2spconf.com/2011/papers/jspriv.pdf, September 2011.