User Profiling Through Browser Finger Printing

Murad Ali, Zubair A Shaikh, Muhammad Kashif Khan, Taha Tariq
{ali.murad, zubair.shaikh, kashif.khan, k102158}@nu.edu.pk
National University of Computer & Emerging Sciences, FAST, Karachi, Pakistan.

***Abstract*—The modern world is changing rapidly. Now a day the use of internet is categorized as essentials for a common man. Everything is put on the web as it is bringing ease of access to the user. The web usage is doubled by the paradigm shift i.e. the smart phones. Online shopping has become a basic and ease of access unit of the user. Because of the huge traffic on the websites the marketing and the advertisement companies have a huge task of displaying relevant advertisements or products to users. To do this they need to track online habits of users and creating their fingerprints or a persona of the respective user. This research has different examinations on user and web-based fingerprinting. The main concept of browser fingerprinting is to gather the data even if the cookies are cleared or disabled by the user. In this paper, we examine how to track user on the basis of his/her system's profile. In this research we have extracted different hacks and exploit the browser. We have discussed that how the browser can be exploited to identify the user. The application of this work is in sales, marketing and promotions etc.**

Keywords: User Profiling, User Fingerprinting, Cyber Security, User Tracking, Browser Fingerprinting

I. RELATED WORK

Many marketers , internet advisors and other websites use different techniques to track and identify users then to make a user persona that can be later used for advertisement and search results Another reason for user tracking is that it helps sites to know about their paid members whether the user is same or someone else is using his account . It also helps websites to know about their popular pages to increase site traffic.

The most popular and commonly used technique is to track. User these days is use of HTTP cookies or browser cookies, they are small pieces of data sent between your browser and web site according to Ayenson Wambach et al.2011, They have detected HTTP cookies on all top 100 websites. Now the problem arise for websites is that what if user delete the cookies or set the browser to reject cookies ,or user is using cookie less browser, Secondly mobile phones are rapidly increasing their part in website usage , some mobile phones don't support cookies (Google has discontinued support for such mobiles) . Marketers now days have also come up with a new version of tracking cookie the super cookie (also known as "the super cookie"). It's difficult to locate or delete flash cookies [1] because they are located in different portions in your system, like in any file which is being used by a Flash plugin. But the problem remains as we are depending on flash plug in, what if the user has disabled or don't have flash plug in as Wikipedia has already advising users that "It is important that super cookies [2] are blocked by browsers, due to the security holes they introduce" .According to McDonald et al. 2011 20% of top 100 websites are using flash cookies[1], but that six of the top 100 sites had Flash cookies that were not unique. If you compare the results of Studies of Soltani 2009, McDonald 2011 and Ayenson Wambech et al.2011 it clearly show that the use of web tracking via both flash and https cookies [3] has been increased, but they all can be deleted, disabled or blocked.

To tackle this situation we have to use some other and modern tracking techniques like fingerprinting This technique helps websites to track user by looking at the characteristics or attributes of a browser like plug-ins, time zone , fonts and many other features . So users can be tracked even if they erase their cookies [2] .Many Studies and researches are going on User fingerprinting exploring the new and efficient techniques for user fingerprinting like Panoptclick, Blue Cava, Threat Metrix, Iovation Reputation Manager, they have used different attribute to make a user unique from others [4]. The Table Below shows the features used by studied Fingerprinting providers [6], features which are shaded are new or acquired through new or modified method in comparison with Panoptclick.

Fingerprinting Category By Panapticlick
Browser Customization
Browser-Level User Configuration
Browser Family and Version
Operating System and Application
Hardware and Network

**TABLE [1.0] Categories of Fingerprinting by Panoptclick*

Even many websites have been started implementing it according to their purpose and need.

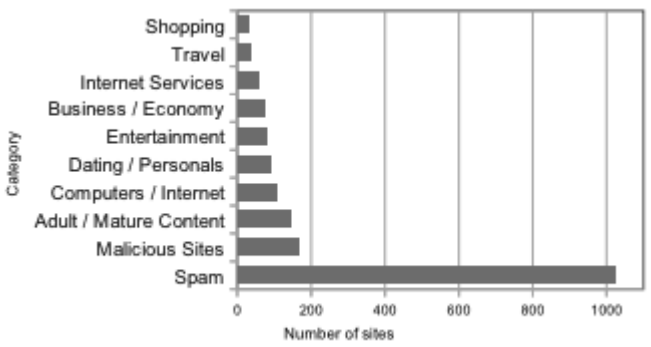

TABLE [1.1]Top 10 categories of websites utilizing fingerprinting

According to the stats in the above figure, it clearly show that Shopping i.e. E commerce is the making the least use of fingerprinting , the most are spams , even though E-commerce can make the most out of it by identifying user and their trends.[5]

The basic key of user fingerprinting was to identifying the returning user on the same web. To detect this Eckersley (2010) formulated a simple algorithm comprising of about eight attributes of the user. Listed below:

Variable	Source	Remarks
User Agent	Transmitted by HTTP, logged by server	Contains Browser micro-version, OS version, language, toolbars and sometimes other information, e.g. details of installed plugins.
ACCEPT headers	Transmitted by HTTP, logged by server	
Cookies enabled?	Inferred in HTTP, logged by server	
Screen resolution	JavaScript AJAX post	
Timezone	JavaScript AJAX post	
Browser plugins, plugin versions and MIME types	JavaScript AJAX post	Sorted before collection. Microsoft Internet Explorer offers no way to enumerate plugins; the PluginDetect JavaScript library was used to check for 8 common plugins on that platform, plus extra code to estimate the Adobe Acrobat Reader version
System Fonts	Flash applet or Java applet, collected by JavaScript/AJAX	Not sorted. Note in 200 cases Mac OSX periodically changed the sort order of the "Lucidia" family
Partial supercookie test	JavaScript AJAX post	Tests for Flash LSO cookies, Silverlight cookies, HTML 5 databases and DOM globalStorage were not implemented.

TABLE [1.2] Attributes of Eckersley Algorithm

A threshold was set for the User-agent, fonts and plugins to at most 15% to match the record if all the attributes got same. The Panopticlick project received over a million of hits and gave a result of returning users also when the browser plugin was also included to an accuracy of 99.1%.

This research and algorithm was further extended by Broenink (2012) in which he adds one different approach in which he identified the different browsers on the basis of attributes that were changed and the things that do not change. For example browser name cannot be changed but browser version can be changed like if the user updated it or the fonts installed can also get changed.

Property	Assumpted Rule
Accept	does not change
Accept-Language	does not change (*)
Accept-Encoding	does not change
Accept-Charset	does not change
Connection	does not change
User-Agent	browser name does not change, browser version does increase
DNT (Do Not Track)	does not change (*)
JavaScript enabled	does not change
JavaScript version	does not decrease
Platform	does not change
Charset	does not change
Language	does not change
Cookies Enabled	does not change (*)
Java support	does not change
Screen resolution	does not change (**)
Timezone	does not change (Daylight Saving Time corrected)
Plugin versions	do not decrease
Font List (All)	order does not change, fonts are not removed, fonts may be added between

TABLE [1.3] External Monitors Being Plugged in for Additional Attributes

K. Boda(2011) further experimented by adding some more attributes to the database. He tracked the user on the basis of IP address and through its derived attributes like locality. He further closes the user down by extracting the time-zone. The list he populated is give below.

locality	Hungarian or international
short user ID	user ID in a shorter, hashed format
created	time of fingerprint creation
ip	visitor IP address in a hashed format
UAS	the user agent string of the browser
os	operating system
screen	screen resolution
timezone	time zone
basic fonts	standard font list for user ID generation
all fonts	all detected installed font list stored for analysis

TABLE [1.4] Time Zone Adds more variables

But the results were not that efficient enough as for instance, if the two users are using the same IP address then this algorithm made only identity of the user on the basis of the IP-address. Further, the user moved from place to place so the IP is not constant of an individual. But they found a very particular key to give the user a unique key that is the fonts. He ran a specialized query that extracted fonts installed on the OS. The results were very upright for the MAC and Android Oses but did not help for the UNIX users.

II. APPROACHES

Starting with the conventional data gathering of the users as previous through a website we hosted a website taha.codinghazard.com. Our main motivation was to get the returning user efficiently. The main constraint that we put was we make sure that the user was accessing our web-site through a smart phone. Because of this we had our resources limited but to extent we populated about 31 attributes on which we were identifying the users. We extracted several libraries of Java-Script such as Navigator and an API named WURFL, this API helped us in getting the data from the smart-phones. The populated list as follows.

VARIABLES	PANOPTCLICK	BLUE-CAVA	THREAT-MATRIX	USER-FINGERPRINTING
PLATFORM				√
USER AGENT	√	√	√	√
TIME ZONE	√	√	√	√
COOKIES ENABLED	√			√
SCREEN –RESOLUTION	√	√	√	√
SCREEN-WIDTH				√
SCREEN-HEIGHT				√
SCREEN-COLOR DEPTH				√
SCREEN PIXEL DEPTH				√
BROWSERS- LANGUAGE		√	√	√
AGENT HEADER				√
BROWSER- CODE NAME				√
BROWSER –VERSION				√
IP ADDRESS				√
HOST NAME				√
CITY				√
REGION				√
COUNTRY				√
LOCATION(longitude-latitude)				√
ORGANISATION(network service provider)				√
DEVICE TYPE				√
BRAND				√
MODEL				√
OPERATING SYSTEM				√
OS VERSION				√
FONTS	√	√	√	√
PLUGIN	√	√	√	√
HASH				√

TABLE [2.0]

We built our base on several strong attributes like the fonts, plugins, screen-resolution, smart-phone model, IP address and the time-zone.

By combining all the attributes we generated a hash of that particular user. This hash comprised all the attributes that we gathered of the user. If someone accesses our web-site from a computer than we displayed a message that kindly access this web-page from your smart-phone.

We gather the accessible data from the user's smart-phone as the user hit our web-page for the first time, after this the user hits us for the second time we caught that the user is a returning one and hence we displayed the message to the user that you are a returning user.

We managed to gather a data set of about 2000 unique users. Once we had our hashes generated and the users were identified the next part we did was the profiling of user on an online-shopping website. This profile comprises of the track record of the user on the website. This profile is like a persona which leads us to personalize the website to the user for next user visit on the website. The user's click on the products and the user's searches (key words) are saved in the respective persona of that user.

So this is final working, when the user visits the site the system checks it that if the user is in our database or not if no then the user's profile is created if yes then that user's respective profile is retrieved and the web-site is personalized on the previous tracked results of the user.

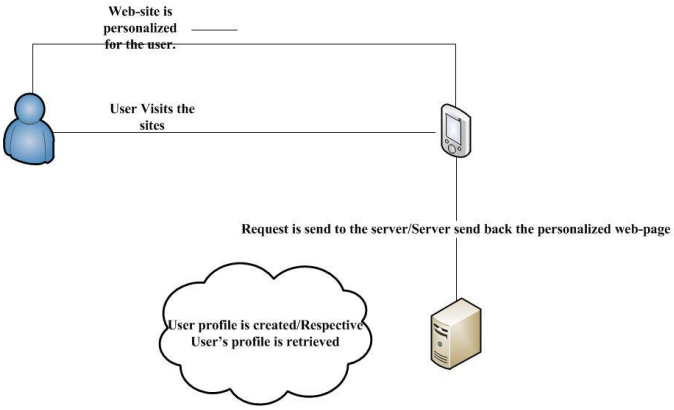

Figure [1] Over All Working

Out of 800 users 99.0% of the users were identified as returning. The key attributes for generating the hash was the IP address. If the user is on move than the IP will not be constant and for this we used a hash of the user with no IP and this hash classified the user to its respective set. On the basis of IP address several other attributes are derived such as region, country, location and organization. The big data set is broken down using this hierarchy.

The second main part of the hash was the fonts and plugins installed on the user's smart phones. But this does not help in getting the hash more unique. The fonts installed on Android, Apple and Windows devices were not been differentiated on the basis of this attribute. Because of this, results were quite low as the smart phones are rarely updated on the basis of plugins and fonts.

The screen-resolution played a little significant role in the making of the user more unique. As the smart phones have different screen sizes. Through our experiments about 20% of the users were identified by their unique screen resolutions.
The maker of the smart phones and model version combined with their operating systems running on them also made some unique input in the generation of the hash. For Android and Windows smart phones these attributes worked and contributed about 60% in the results. But Apple devices were very poor in the results not more than 10% results showed up as Apple devices were not easy to penetrate as it only showed the IOS version and the device runs on MAC. So this attribute was almost useless for the MAC devices.

The profiling part of the user has ups and downs. The user profile is a part of a set. This set contains different variations of the profile. Because of these variations the results dropped to about 60%. This percentage is the systems efficiency of throwing the relevant data when the web-site is personalized per the user's persona.

III. METHODOLOGY

From the beginning towards the end two main prototypes were shown. The feedback was taken and recorded and also workings were done based on this.

1) The first prototype developed in which we identified the user whether the user is new or a returning user on the basis of different attributes like fonts , time zone , location and other hacks from browser without using cookies
2) The second prototype shown included the integrated version of what was shown previously with the improve algorithm and more attributes for identification, Secondly The prototype included the profiling of user based on his history and showing him relevant content of interest on his next visit .we created an ecommerce website "kapray.com "for the implementation of this prototype To test the gathered attributes it was necessary to build a platform on which we could collect the data set and to review our experiments on the attributes selected. We have to determine the results of each attribute and have to see which attribute can be modified to enhance our results.

To do all this we made up a web-site and hosted it, we started our surveys to gather the data set. We asked people to visit our website two times, so as to test our system and algorithms for the returning customers. Our system is only tolerating the access of the website through a smart phone only. After completing the system of identifying the user we created user persona in which we saved the trend and history of user so we were able to push relevant content to the returning customer next time when user visits the visit.

Figure 2 shows all the steps in the same sequence as followed.

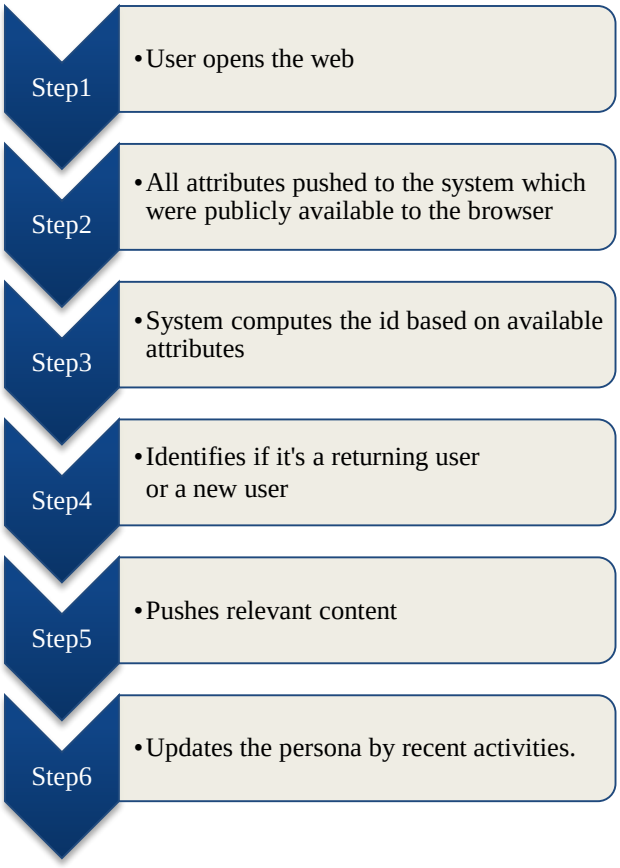

Figure [2] Step by Step How to Develop the Persona of a User

Below is table of variables which we used to identify the user

VARIABLES	DESCRIPTION	REFERENCE
PLATFORM	It tells the user platform for e.g. Android smart phones works on Linux platform.	Java-navigator library.
USERAGENT	Returns a string of browser name and browser version.	Java-navigator library.
TIMEZONE	Tells us that in which time zone of the world the user lives.	Java Script.
COOKIES-ENABLED	Returns a string which tells us that the cookies are enabled or not on the client Smartphone.	Java-Script.
SCREEN-WIDTH	Returns a value which tells the screen width of the smart-phone.	Java-Script.
SCREEN-HEIGHT	Returns screen height of the smart-phone.	Java-Script.
SCREEN-COLOR-DEPTH	Returns screen color depth of the smart-phone.	Java-Script.

TABLE [3]

SCREEN-PIXEL-DEPTH	Returns a value which tells the screen pixel depth of the smart-phone.	Java-Script.
BROWSERS-LANGUAGE	Tells the installed language e.g.firefox-enu,firefox-enfr.	Java-navigator
AGENT HEADER	Used for extraction of parameters.	
BROWSER-CODE-NAME	Build of the browser on which it is based upon. e.g. Firefox is built upon Safari	WURFL Api.
BROWSER-VERSION	Returns the current browser version installed.	Agent Header.
IP-ADDRESS	Returns the IP address of the user.	HTTP HEADER.
HOST-NAME	Derived from IP, the host-name. User's connection is directly connected.	Break down from IP-Address.
CITY	Derived from IP, returns the city where the host is placed physically.	Break down from IP-Address.

TABLE [4]

REGION	Derived from IP, returns the region which host-name covers.	Break down from IP-Address.
COUNTRY	Derived from IP, gives the country where the client is situated.	Break down from IP-Address.
LOCATION (longitude-latitude)	Derived from IP, returns the longitude and latitude of the client.	Break down from IP-Address.

ORGANISATION(network service provider)	Derived from IP, returns the organization name from which the user is connected.	Break down from IP-Address.
DEVICE-TYPE	Tells device type of user e.g. Smartphone or a tablet.	WURFL Api.
BRAND	Returns the device brand name the user is using. For e.g. Samsung, HTC.	WURFL Api.
MODEL	Returns the model number of the device.	WURFL Api.
OPERATING SYSTEM	Tells the operating system is installed on the device.	HTTP-HEADER.

TABLE [5]

OS-VERSION	Returns the operating system version of the running OS on the device.	HTTP-HEADER.
FONTS	Returns the total fonts installed in the browser.	Java-Script.
PLUGINS	Returns the plugins installed in the browser.	Java-script
HASH	Key on which user is identified.	Self-generated.

TABLE [6]

IV. DEPLOYMENT

To test this research we implemented it on a distance learning system and a library management system. The results are encouraging enough so that we are working now to develop a flexible IDE based solution to work in parallel with any management system. To give a brief idea here see a scenario with the result. Figure 3.1 shows the home page when any user hits the web link for the first time. You can see various items of mix categories on home page.

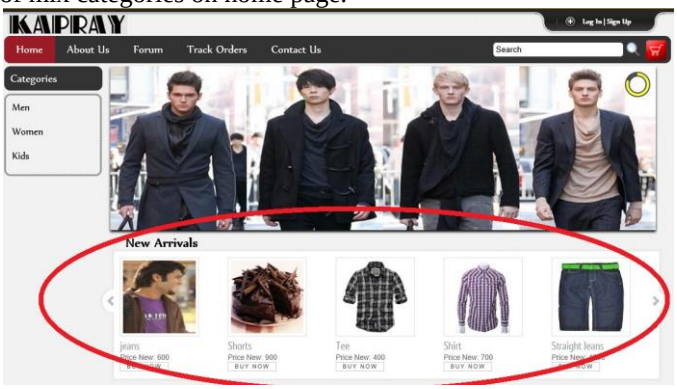

Figure [3.1] Home Page, New User

Later the user search for the red products. From the list of all red products user clicks on a particular red product.

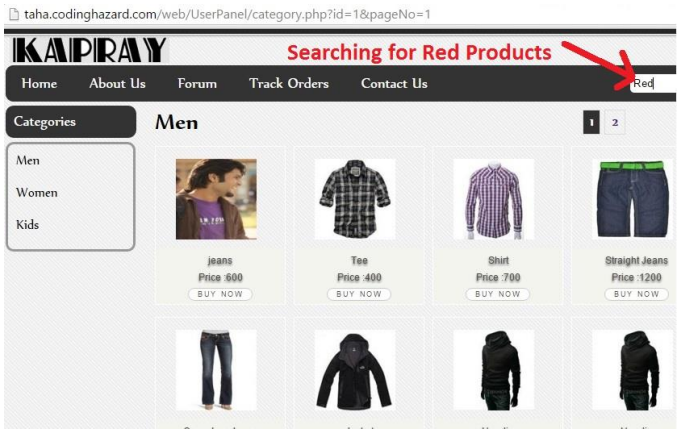

Figure [3.2] User is searching for Red Products

Again same user hits the home page and now the available product is showing before any other product as the system has recognized the taste of the user with the help of user finger printing. Even the user clears his/her cache but still there will be no impact on results.

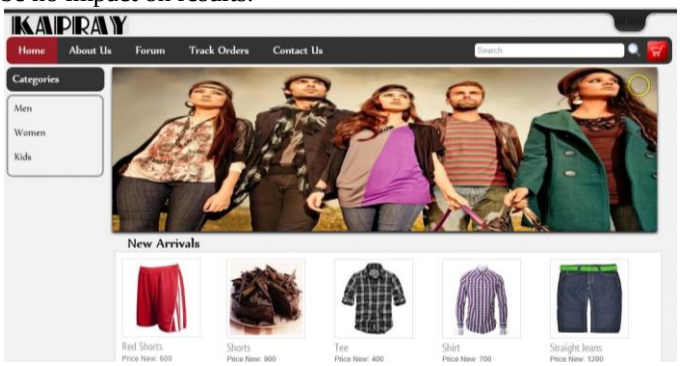

Figure [3.3] Old User Hits Home Page and Gets His interest on Home Page

V. FUTURE WORK

This system will get mature as more users interact with the system. As the data set will increase more efficient results would be determined and efficient algorithms could be made. The attributes can be also increased on the basis of different hacks of the browser. User profiling could be made more reliable and efficient. This profiling will certainly help a lot in the marketing perspective of an online shopping website as well as sales of the products. This was our part of the project but we could not implement it efficiently because of the required capabilities in the time available. Our system will take time so as to learn the patterns of the user. Hence efficient algorithms of machine learning can be engineered so as to get better results.

VI. CONCLUSION

The first objective of our research was to get as much attributes of the browser so as to identify the user more efficiently. We achieved this through our extensive research, also by the review of Panopticlick, BlueCava and Petportal.
We gathered some more attributes by using different CSS hacks and Java-Navigator library and also using WURFL API. The second objective of our research was to get the returning user identified. We were able to identify about 75% of the

returning user the remaining 25% comprises different variables which did not helped us in the results these were fonts, plugins, IP address and Mac issues.

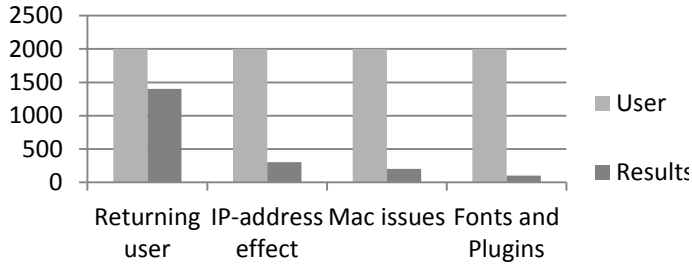

**Figure [4.0] Results of the Experiments*

The third objective of our research was to get design a system to efficiently personalize the web-site as per the user's profile. We took two attributes of the user that is User searches the products and User clicks on the products.

REFERENCES

[1] Ashkan Soltani, Shannon Canty, Quentin Mayo, Lauren Thomas, and Chris JayHoofnagle, *Flash Cookies and Privacy*

[2] Ayenson Wambach et al,Flash Cookies and Privacy II: Now with HTML5 and ETag Respawning

[3] McDonald, A. M., & Cranor, L. F., *A Survey of the Use of Adobe Flash Local Shared Objects to Respawn HTTP Cookies*, CMU-CyLab

[4] Eckersley, P. (2010) 'How Unique Is Your Web Browser?'

[5] Boda, K., Földes, Á., Gulyás, G. and Imre, S. (2011) 'User Tracking on the Web via Cross-Browser Fingerprinting', NordSec'11

[6] Broenink, R. (2012) 'Using Browser Properties for Fingerprinting Purposes'