Executive Summary

The principles of user or device tracking in the online world and the mobile world are basically similar. Identifying and tracking user profiles without Personally Identifiable Information (PII) is something being undertaken by a number of the entities in the delivery of targeted mobile ad campaigns.

A device fingerprint is a set of information collected about a device or user profile for the purpose of audience-based content or ad targeting. Although the cookies are turned off, it is still feasible to use fingerprints for complete or partial identification of individual users or devices.

Device fingerprinting, sometimes known as digital fingerprinting, claims that it can assist businesses to enhance online or mobile advertising and diminish fraud risk.
Introduction

The purpose of this whitepaper is to outline various scenarios and implementation approaches for mobile device fingerprinting, unique user identification, and user tracking to:

- Get insights into mobile campaign performance events such as clicks and installs
- Track which app was installed and which user installed the app
- Track users’ behavior and browsing patterns to enhance your content or ad recommendation engine

Solution Overview

The challenge

The key challenge is to uniquely identify a mobile user. Mobile users access content through both native applications and mobile web. On mobile devices, websites and applications are regarded as disparate domains. These two domains do not use the same identifiers. This means, a single user could possibly appear similar to one or more separate individuals unless measures are taken to tie the multiple identities together.

This whitepaper tries to cover both the scenarios for user identification—mobile web and native apps and how to bring the two together.

The online (desktop) environment and the mobile web share the following analogy. Traditional user identification approach of third-party cookies is used to identify consumers across various mobile websites. The challenge here is to identify the cookie-less approach because cookies are volatile when mobile web browsers are concerned, and a mobile user identification technique cannot rely on cookies.

In the absence of cookies, popular web browsers and mobile web browsers are able to provide adequate information, which results in a unique identity of visitors. The following data is also called configuration information:

- the type of browser
- the operating system
- various plug-ins
- installed fonts

When the preceding data is compiled, a unique identity of almost all visitors can be generated. The preceding data is capable of creating a unique browser fingerprint, but this data fails to actually identify the mobile web user. When a user whose configuration information is available visits other websites, it becomes viable to identify this user.
SCENARIO 1: USER IDENTIFICATION AND TRACKING FOR MOBILE WEB (COOKIE-LESS APPROACH)

The goal here is to develop a user identification system for mobile devices, which can be used as the fingerprint system for the desktop environment also. Due to various cookie policies in mobile web browsers, it's recommended to develop a 'Mobile Tracking with cookie-less mobile device user identification system'.

This scenario covers the proposed cookie-less approach:

1. The scope of the client component is to track mobile user identification for mobile web browser identity.
2. The client component generates a unique user ID on the mobile using the technical approach of JavaScript, header information, HTML5 web storage, or combination of any of these approaches.
3. The logical workflow is depicted in the following image.
4. You can use JavaScript, jQuery, and Modernizr JavaScript library to generate and store unique IDs for mobile devices when cookies are disabled. The client component can retrieve and pass this ID to your existing backend or server component.

5. Even though the cookie check is not required for mobile browsers, it is included (represented as a dotted line) for the future scalability on the desktop environment.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modernizr JavaScript</td>
<td>• To detect if cookies are enabled or not</td>
</tr>
<tr>
<td></td>
<td>• To detect all browser information</td>
</tr>
<tr>
<td>jQuery</td>
<td>• To implement mobile device detection</td>
</tr>
<tr>
<td></td>
<td>• To implement Ajax requests to the server</td>
</tr>
</tbody>
</table>

6. The mobile device fingerprint technique is used to detect mobile users:
   a. The generated unique user IDs are for mobile devices.
   b. JavaScript is used to collect all information necessary to generate unique IDs for mobile devices when cookies are disabled.

<table>
<thead>
<tr>
<th>Information</th>
<th>Uniqueness (approx.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cookies enabled?</td>
<td>1 in 1.3</td>
</tr>
<tr>
<td>User-agent string</td>
<td>1 in 4,184</td>
</tr>
<tr>
<td>HTTP_ACCEPT Headers</td>
<td>1 in 4</td>
</tr>
<tr>
<td>Time zone</td>
<td>1 in 24</td>
</tr>
<tr>
<td>Screen resolution</td>
<td>1 in 11</td>
</tr>
</tbody>
</table>

c. A unique fingerprint is generated with combination of the following:
   I. Earlier mentioned information
   II. Custom identifier

d. The custom identifier can be provided by your server or backend component, and it is stored as a hidden component. The ID is exposed to the mobile user identification module as a JavaScript property.

e. An example of a unique ID is the following:
   Mozilla/5.0-(Windows-NT-6.1;-WOW64;-rv:11.0)-Gecko/20100101-Firefox/11.0-
   <CustomIdentifer>--330-1280X1024

f. Store and retrieve unique ID: The unique ID (“fingerprint”) generated can be stored to the mobile user through HTML5 web storage and retrieved when the user browses a publisher’s webpage having ads delivered by your server component.
g. HTML5 has persistent storage support as mentioned in the following text:

I. Key or Value Pair Storage (Local or Document Object Model (DOM) storage):
   a. HTML5 Storage stores named key or value pairs locally. Although a user navigates away from the website or closes the browser, the data continues to exist.
   b. The data differs from cookies with respect to the fact that it is never sent to the remote web server.

<table>
<thead>
<tr>
<th>Platforms</th>
<th>OS versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPhone</td>
<td>2.0+</td>
</tr>
<tr>
<td>Android</td>
<td>2.0+</td>
</tr>
</tbody>
</table>

II. Web SQL storage (SQLite storage)

<table>
<thead>
<tr>
<th>Platforms</th>
<th>OS versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPhone</td>
<td>3.0+</td>
</tr>
<tr>
<td>Android</td>
<td>2.0+</td>
</tr>
</tbody>
</table>

   a. **Pros:** This storage is supported on almost all mobile web browsers and has easy storage and retrieval.
   b. **Cons:** Its performance depends on the browsers.

**SCENARIO 2: USER IDENTIFICATION AND TRACKING WITH SDK**

An advertiser places an ad for a mobile app in the mobile network. Such ads are displayed to mobile users when they browse any publisher’s website having your ad widget embedded into it. Your ad widget serves such ads to mobile users. Users click such ads and get redirected to the relevant mobile app page on the Apple AppStore or Google Play. You track such clicks and want to further know:

1. How many clicks an ad received?
2. Which mobile applications were installed?
3. Which users installed the app?
This scenario addresses the preceding with implementation of a mobile tracking Software Development Kit (SDK):

1. Development of an SDK for app developers to use your installation tracking in their own applications is the initial step. The tracking module should contact your server and transmit information about the application itself and the user. The user identification should be comparable to user IDs generated by your ad widget in mobile browsers.

2. Proposed approach for Mobile SDK:
   a. The mobile user explores a publisher website having your ad widget. During this request, a unique ID (mobile fingerprint – here: ID A) is generated and stored on the server. The ad widget, including an ad for a certain mobile app, is sent back to the browser.
   b. After clicking the ad, the user gets redirected to the mobile app page on Apple AppStore or Google Play.
   c. In case, the user decides to install the app, the app is downloaded by the user from the AppStore or Google Play.
   d. For the first time the user launches the newly installed app, a request is sent to your server component to let the server know about the installation. During this request, the app creates a unique user ID (ID B) to link the installation to the previously generated ad-click data.
   e. **Approach 1:** After clicking a campaign URL on your ad widget, the user is redirected to the respective app page on iTunes and also a unique ID (A) is generated with the mobile user’s IP address and other device information. This ID A is passed to your server, which in turn can register a click and store the ID A combining Time Zone and Date-time details. At the first time of application launch, the application generates the ID B in a similar manner and passes it to your server. The server compares the ID B received from iOS SDK with ID A received during the click within the time window of 15 minutes.
   f. **Approach 2:** At the first time of application launch, the application can start a web browser with a specific domain site to get the unique ID (mobile fingerprint - here: ID A) stored in the local storage and send the request to the server. After that, the browser relaunches the app again using URL schema. Currently, most app installs are tracked this way if they are installed through a web tracking method.
   g. **Approach 3:** If the mobile web and the native application create the same fingerprint, it is possible for the server to compare it. This approach anonymously makes us match a permutation of attributes to a device. After this matching, if we find that there are two events having a similar fingerprint, we conclude that there is high statistical probability of both these events are related to the same device. Let us see an example. A user taps an ad on a specified mobile web browser. After a short span of time, a similar device registers the app install. In such a case, it is highly probable that these two events are from the same device.
h. **Approach 4:** If we use the same key stored into the local storage of the browser, the server can also compare it. This is doable by reading locally stored data file natively on the device. However, this requires the use of private APIs, which is not recommended from Apple and subjected to the application being rejected for AppStore submission.

i. **Approach 1** mentioned earlier is the most effective of all and the recommended one.

3. **Server component:**
   
a. The Server component compares ID A (generated by a mobile fingerprint along with click tracking) and ID B (generated by SDK using similar logic) to identify the user who clicked the ad and also installed the app from AppStore or Google Play.

b. The Server component finds a match between ad clicks and app installs to help report on effectiveness of various ad campaigns.

**CONCLUSION**

Many Internet advertisers rely on cookies stored on a user’s browser. Some websites place multiple cookies allowing them to track user activity over time. The challenge for marketers is that some users set their browsers to reject cookies, and mobile phones, on which web browsing has increased, do not use cookies. To overcome such a challenge, advertisers and publishers are increasingly turning to device fingerprinting. A plethora of advertisers on the Internet are highly dependent on cookies, which are stored on a user’s browser. A few websites have many cookies that track user activity as the time progresses. In this scenario, marketers have to face the following hurdles:

- Users can set their browsers to reject cookies.
- Web browsing on mobile phones is accelerating at a quick pace, and mobile phones do not use cookies.

To surmount these hurdles, publishers and advertisers are opting for ‘device fingerprinting’ on a large scale.

Device fingerprinting helps advertisers and publishers with the following:

a. Understanding click fraud and fraudulent online transactions

b. Remarketing for mobile

c. Doing unique user (UU) analytics: Tracking and targeting user behavior to improve content or ad recommendations

d. Tracking app statistics with a user’s journey from click to install
Cybage Fast Facts

- SEI CMMI Level 5 version 1.3 and ISO 27001 certified company
- Workforce of more than 5,200 IT professionals
- Multiple industry verticals under a global delivery framework
- Five state-of-the-art software development facilities globally
- Broad software services portfolio with more than 170 global offshore software engagements
- Preferred vendor to leading:
  - 8th Best IT Employer in India by Dataquest
  - Gartner Cool Vendor in the Application Services space
  - Ranked in the IAOP Global Outsourcing 100 List for the last 4 consecutive years
  - Ranked in the Leadership zone as a Global R&D Services provider by Zinnov
  - Featured as a “Specialty ADM Vendor” and “Leaders-Mid-tier Product Development”

Cybage Advantages

- One-stop technology service provider
- Unique business management software system - ExcelShore® model:
  - Operational excellence
  - Value delivery
  - Consistent Return on Investment (RoI)

Centers of Excellence (CoEs)

- Cloud Computing
- Business Intelligence
- Enterprise Mobility
- Customer Relationship Management
- E-commerce
- Supply Chain Management
- Enterprise Content Management