Alex Miller
Assignment 5 Writeup
Collaborators: AC Fields (acfields)

Problem One

Schemas:
DB tracking_db
TABLE visit

Visit #	Page	cookie	IP	Window size	UA ID	Cookies enabled?	DNT enabled?	Popups enabled?	Fonts

TABLE ua_string

UA ID	User-Agent Hash	User-Agent String

I stored visit data in the first table as it allowed me to distinctly see all the variables I was tracking as well as associate them to page visits. I could also sort these rows by visit_no, which made going through the database looking for clusters much easier.
I also made a second table for storing User-Agent Strings. This table stores the hash of a string, the string itself, as well as a Primary Key. This made comparing user-agent strings much easier when associating clusters. Rather than having to operate on and sort by long strings when working with the visit table, I could just look at the Primary Keys of the ua_string table. I also didn't have to store multiple redundant copies of the same User_Agent string. I hashed the values when storing them to make comparisons upon insert easier (I had to compare on insertion into ua_string to ensure no duplicate strings were added).

Problem 2

Relevant File Names: track.html, sink.php
Solution: My solution uses a combination of track.html and sink.php to log relevant tracking information to the database. Every HTML page that is visited at the webserver gets document relevant information and sends it to sink.php in a POST request, which takes that data as well as data extracted from the request itself (User-Agent strings, IP addresses, cookies, etc) and adds it to a row of a tracking database that records any given visit to the webserver that trigger sink.php.

Problem 3

Relevant File Names: track.html, sink.php
3.1 - Cookies

1. ‘sink.php’ generates a cookie from the client’s IP (covered later), the time, and User-Agent string (covered later) and hashes it using MD5. This hashed value is the client’s cookie and is logged in the visit record.
2. This type of tracking remains possible because sites want to be able to track users between site visits; this makes logging into password-protected accounts easier as well. This type of tracking can be disabled by disabling cookies in your browser. If cookies were no longer used, sites would lose the ability to track users between site visits; they would lose the ability to keep track of what somebody did on a page and the ability to track them across the web (accomplished by sharing cookies). I would just eliminate cookies; they give websites the ability to track us and monetize user data without any real benefit to the user.

3.2 - IP Address

1. When a client is redirected to sink.php, sink.php gets the ‘Remote Address’ environment variable and associates it with the log of the visit in the record.
2. This type of tracking remains possible because servers need to know where to send back requested data. This type of tracking can be stopped by using a VPN or a Proxy Server. This would stop servers from tracking a user’s real location. If users do this, sites lose the ability to geographically limit and market to users. I would eliminate the use of IP addresses when possible and instead use VPNs and Proxies because I don’t like websites knowing my general location.

3.3 - Window Size

1. When a client accesses a page at the web server, the page (based on track.html) gets the width and height attributes of the window using the ‘window.innerWidth’ and the ‘window.innerHeight’ methods and sends it to sink.php via a POST request, where it is logged in the page visit record.
2. This type of tracking remains possible because knowing a client’s page size allows websites to customize how they display content to the client. This type of tracking could be stopped by browsers blocking access to that type of data (page size). If this type of tracking was prohibited, websites would lose the ability to customize content and the web would be uglier. I would keep this type of tracking possible because it is both varied (making it not reliable) and provides users with a distinct benefit (more appealing websites).

3.4 - User-Agent String

1. When a client is redirected to sink.php, sink.php gets the ‘HTTP_USER_AGENT’ server variable, hashes it with MD5, logs that in a database of User-Agent strings, and associates the User-Agent string PK value with the log of the visit.

2. This type of tracking remains possible because knowing a client’s device info allows sites to customize the pages they serve based on things like screen dimensions and browser capabilities. This type of tracking could be stopped by removing User-Agent strings from TCP requests. If this type of tracking was eliminated, sites would lose the ability to serve tailored content to clients which could negatively impact usability (for example, if one implementation of a page uses features a client’s browser can’t handle, the server should redirect to a different page). I would possibly keep this type of tracking but limit the amount of information traditionally transported in it; it is helpful to know a client’s browser, it is not helpful and breaches a client’s security to know their System OS.

3. 5 - Cookies Enabled

1. When a client accesses a page at the web server, the page (based on track.html) checks whether the browser uses cookies using the ‘navigator.cookieEnabled’ method and sends it to sink.php via a POST request, where it is logged in the page visit record.
2. This type of tracking remains possible because sites and browsers continue to use cookies. This type of tracking would be redundant (be stopped) either if everyone used cookies or no one used cookies. Merely removing the ability to track whether cookies are enabled wouldn’t do much; sites would still be able to set cookies and see if they took to track whether a client had cookies enabled. If everyone used cookies, sites would not lose any functionality. If no one used cookies, sites would lose the functionality discussed in problem 3.1. As stated above, rather than remove a site’s ability to track a client based on whether they have cookies enabled, cookies should just be done away with entirely.

3.6 - DNT Header Enabled

1. When a client accesses a page at the web server, the page (based on track.html) checks whether the browser uses a DNT header using the ‘navigator.doNotTrack’ method and sends it to sink.php via a POST request, where it is logged in the page visit record.
2. This type of tracking remains possible so long as browsers continue to allow DNT headers; since DNT headers, if specified, are included in HTTP requests, there is inherently no way to hide whether a client has DNT enabled or not. Removing DNT headers or making them mandatory, therefore, are the only ways to make DNT header tracking redundant. Either way, this doesn’t do much or anything at all; sites can continue to track clients even if they have DNT headers enabled, one example

being this very assignment. I would just remove DNT headers entirely; they don't do much and only give trackers yet another metric with which to identify clients.

3. 7 - PopUps Enabled

1. When a client accesses a page at the web server, the page (based on track.html) checks whether the browser blocks popups by attempting to open a popup window and seeing if the result is 'null.' If the result is 'null', popups are blocked. If it is not 'null,' popups are enabled. The page then sends this information to sink.php via a POST request, where it is logged in the page visit record.
2. This type of tracking remains possible so long as browsers allow pages means of checking the status of popup windows; as this isn't data that can be determined by PHP, pages need to try and open a popup and see what the result is. Therefore, by removing the ability for pages to read the status of a popup, tracking clients by whether they have popups enabled becomes infeasible. However, by removing this functionality from browsers, sites wouldn't be able to serve popups reliably at all; for example, if a user is blocking popups and a site tries to give them a popup form to fill out, the user wouldn't see the form. Moreover, the site wouldn't be able to tell the user that something went wrong. In such a world, the internet would be better off with no popups altogether; this is also inconvenient since popups are helpful. I would therefore not remove the functionality of popups and, by extension, sites' ability to track clients by whether or not they have popups enabled.

3. 8 - Fonts

1. When a client loads a page from the web server, the page then changes five 'div' sections to the specified fonts and checks whether each div changes in dimension size. If there is a change in size, that means the given font is installed. If there is no change, then the font must still be the default font and the given font is not installed. The page then sends this information to sink.php via a POST request, where it is logged in the page visit record.
2. This type of tracking remains possible so long as either a site's pages have access to such values as HTML tag instance dimensions or browsers allow nonstandard fonts installation packages. This type of tracking could, therefore, be stopped if browsers stopped allowing pages to access document data or if all browsers had the same set of fonts installed. The first solution is not tenable as it would stop sites from serving dynamic and appealing content (I, therefore, won't consider it any further). The second solution, however, is tenable; if browsers agreed to include a versatile standard set of fonts, sites wouldn't find their content much altered and clients wouldn't be able to be fingerprinted by their installed fonts as they would all have the same set of

fonts. I, therefore, think it's worth it to limit the fonts that can be installed on browsers in order to stop this type of fingerprinting.

3. 490: DejaVu Sans, 491: GFS Baskerville, 496: Roboto, 498: Liberation Sans, 499: Abyssinica SIL

Problem 4

Relevant File Names: track.html, sink.php

Lots of the features that we tracked are variable over time; cookies expire, IPs can change if a mobile device is the browser being tracked, the window size can change based on how a client is using it, User-Agent strings can change if users change or update their browser, and browser preferences can change at any time. However, you can still associate changes and maintain a track on a client. Different cookies can be tied to each other based on when one expired and one is issued and if the clients requesting them are similar in other respects. IPs may change day to day but they can still be geographically narrowed down to associate locality rather than distinct IP. User-Agent strings can change in some respects but not others, so new ones can be tied to existing clients. And changes to browser settings don't change other trackable values; enabling DNT headers on your browser doesn't change your cookies. Therefore, even though tracking variables may change over time, you can still associate pre and post-change visits because, for one, changed variables can be tied to their previous values (IP and Cookies for example) and changing one variable doesn't necessarily change all of them, so one change doesn't stop a browser from being associated to pre-change visits by other unchanged variables.

Sources

https://www.w3schools.com/php/php_mysql_insert.asp

https://www.php.net/manual/en/function.mysql-fetch-array.php