Internet Anonymity

Submitted by:
Group 12

Mike Hoffert - mlh374

Jeff Pereyma - jdp037

Kari Vass - kdv504

Nathan Abramyk - nsa901

Date:
March 28, 2014

Abstract

Internet anonymity is important because it can protect individuals from oppressive censorship and those in positions where tying an identity to their online presence could cause harm to come to them. This project focused on one particular method of undermining anonymity: the browser fingerprint. Browsers are fingerprinted by gathering information about the browser. The combination of this information is often reasonably unique, and can thus be used to track that browser.

Our project was to undermine the harvesting of the types of information which are typically used to create the browser fingerprint. In particular, we found that preventing enumeration (and mass detection) of fonts and plugins to help generalize the browser. Setting the HTTP language headers to be as general as possible also helps reduce fingerprinting. The Panopticlick tool, provided by the EFF, was used in gauging the effects of our project.[1]

Contents

1 Genesis

1.1 Approach and early planning

TODO: Discuss why we chose the topic, very early planning (eg, the array of topics we considered)

1.2 Pre-conceptions

TODO: Early thoughts we had. Including misconceptions such as assuming font detection already allowed enumeration, that everything was done in JS, etc

2 Initialization

2.1 Ideas

TODO: Early ideas for project

2.2 Goals

TODO: End goals for the project (in particular, letting users know when fingerprinting is going on), ultimately, however: just stopping fingerprinting

2.3 Obstacles

TODO: Chrome extension and JS limitations, new ground for us, etc

2.4 Process

TODO: From an SE perspective. I guess it was mostly "concurrent engineering". Not much of a formal process... anyone have a better description?

3 Control

3.1 Documentation

TODO: Informal, results based. Project code is the documentation.

3.2 Planning

TODO: Mention how we identified core tasks (plugins, fonts, etc) and divided them up. Also some sharing of tasks, meetings, and so on.

3.3 Data gathering

TODO: How we identified relevant. What was the relevant data? In particular, fonts, plugins, and HTTP headers role in fingerprinting. Panopticlick is main tool.

3.4 Analysis

TODO: This is more of a meta-analysis of our approach and control, not the results. Basically, how effective was our data gathering, planning, and documentation? Where did we make strides and mistakes? In particular, mention sandboxing issues and workaround. JS language's tendency to use parameters rather than functions and how we didn't realize it's possible to apply a getter function to parameter access (I dunno about the rest of you, but I just learned this the other day).

4 Technical components

4.1 Approaches and methods

TODO: How did we approach the project. Code, browser, extension, etc. The tools we used. Mention the use of panopticlick as a driving tool, particularly in checking our results.

4.2 Progress and effort

TODO: How many of our goals did we meet? What places failed but we applied effort on? Particular parts we're proud of and are central to project, etc?

4.3 Difficulties and limitations

TODO: Hard parts, stuff we couldn't do, and limitations (particularly those created by Chrome and JS).

5 Results

5.1 Results and outcomes

TODO: Put the numbers from the slides here. Also mention the overall effectiveness and other approaches.

5.2 Progress and failures

TODO: Progress stuff from previous section, with emphasis on the implications our progress had on the results. And failures are straightforward enough (in particular, inability to truly let the user know when fingerprinting is going on. Adding sites to the whitelist is entirely up to the user, we couldn't provide advice. As a result, some sites fail silently. I, for example, cannot load gmail's inbox without adding the site to the whitelist.

5.3 Analysis

TODO: General summary of results. Perhaps deeper look at other areas? Ideas here?

6 Recommendations for further study

TODO: Recommendations for how browsers could implement our functionality (but better). Chrome shouldn't allow plugin enumeration, and should warn users if too many plugin details are collected. Also, Flash shouldn't be able to enumerate fonts without user permission. Basically, ask the user more.

References

[1] Panopticlick: https://panopticlick.eff.org/