Internet Anonymity

Mike Hoffert
Nathan Abramyk
Jeff Pereyma
Kari Vass

March 24, 2014

Internet anonymity is important

- Supports freedom of speech: governments can't censor you if they don't know who you are
- Skirting surveillance generally requires anonymity (perhaps to avoid government censorship)
- Oppressive governments often attempt censorship
  - Turkey recently blocked Twitter
  - Amnesty International stated that China "has the largest recorded number of imprisoned journalists and cyber-dissidents in the world"
  - In Iran, internet users must promise not to access "non-Islamic" websites

Internet anonymity is important

- Whistleblowers may need anonymity to prevent retaliation
- Undercover military and law enforcement agents often need anonymity for their protection
- Journalists may have to protect their sources
- One line of defence against targeted attacks
- Removes real life consequences for controversial opinions
- Some people feel uncomfortable speaking publicly

Threats to internet anonymity

- IP addresses can be tied to an ISP customer (but insufficient to identify a specific individual)
  - There could be another person using the computer
  - Drive by downloading
- A number of recent American actions would have implications on internet anonymity: SOPA, PIPA, PRISM, CISPA, etc
- ISPs that sell your information
- Tracking services (especially with online advertisers)
- Browsers have lots of identifying information – **this is what our project focused on**

Browser fingerprint

- A browser fingerprint is a way to identify a specific browser, based an unique identifiers that is holds
- Some features include things like:
  - Header Information
  - Installed Plugins
  - Installed Fonts
  - Time Zone
  - Screen Size
  - User Agent
- If you combine these features of a browser together, you will find that every browser tends to have a very unique set of these properties. This is your "Browser Fingerprint"

Panopticlick

- The tool used to gauge the effectiveness of our changes was the EFF's Panopticlick research project
- Panopticlick <https://panopticlick.eff.org/> is an attempt to identify the uniqueness of a browser via some of the previously mentioned techniques
- We picked the most identifying techniques and attempted to thwart them
- The goal was to make the browser as difficult to fingerprint as possible
- Let's go into more detail on how we did that

HTTP Headers

- The fields in your HTTP headers can be used to help to detect your browser fingerprint.
- The header field User-Agent contains identifying information regarding your operating system and version, as well what browser you're using and what version it is.
- The Language-Accept field can also pass identifying information regarding language settings. For instance, your browser might be passing en-CA (Canadian English) as your preferred language.

HTTP Headers: Our Solution

- Our solution to this problem was to look at modifying the header fields before they are sent.
- For instance, in setting the Language-Accept field to accept just the more generic US english instead of trying to accept Canadian english (much less common).
- Consider that with our language preference set to Canadian English, roughly **1 in 220,000** browsers have this value. In passing en-US instead, **1 in 40** browsers have this value.

Available fonts

- Browser fonts can be incredibly identifying because most people have a unique set of fonts installed
- Clever and varying techniques using Javascript can be used to accomplish this
- Primary method we tried to break was measuring font width/height
- 2 Approaches to the problem:
  - Override the specific functions that measure fonts
  - Override or block common DOM methods used when measuring fonts
- Problems arise with page load times and overriding DOM methods

Available plugins

- TODO: Detail how the list of available plugins impacts fingerprinting, how we solved this, and the major shortfall of our solution
- Be sure to detail how the browser/JS implementation could offer a better solution than our hacky fix

Other fingerprinting threats

- Flash is an effective and very common method to detect fonts
- Panopticlick uses Flash for it's font detection
- To prevent these, there are extension like Flashblock, or Ghostery
- You can also change settings in your Flash install
  - File: mms.cfg Line: DisableDeviceFontEnumeration = 1
- Fingerprinting can be combined with the approximate geographical location that an IP address tells us. This is not an accurate location, but neither are other fingerprinting techniques – it's the combination that makes for accuracy

Putting it all together

- Our Chrome plugin, named "Fingerprint Anonymizer", adds several hooks and overrides to prevent or reduce fingerprinting
- The goal was not to flat out block actions that could be used to identify the user, but rather to let the user know that they were taking place and allow them to choose whether or not to allow them
- This goal was ultimately not possible, due to the fact that much of the data was accessed through parameters and not functions – we couldn't add functionality to detect when information was read

Putting it all together

- A whitelist was implemented, which the user could add domains for which the blocking is not activated
- The user can manually add regex for matching domains to the whitelist in the extension's options page
- We also implemented a browser action (a button in the browser's main toolbar that opens a prompt) to add the current domain to the whitelist
- Rewriting the HTTP headers is done for all pages
- TODO: mention end result (panopticlick results)

Drawbacks

- Fingerprinting is not accurate
  - It tracks machines, not individuals
  - Some information, such as browser user agents, can change very frequently
  - Browsers are beginning to implement features to prevent mass gathering of information
  - Firefox, for example, no longer allows enumeration of plugins (but you can still detect individual plugins)
- Chrome extensions are sandboxed. If we modify JavaScript objects in our content script, they won't be modified on the actual page (unlike Firefox)
- Many JavaScript properties are accessed as parameters rather than by functions

Drawbacks

- While parameters can be overridden, you can't change the behaviour of accessing parameters, and thus we cannot tell if a parameter is accessed
- This provides a compelling reason to use getters and setters instead of direct variable access (although JavaScript lacks visibility modifiers)
- The browser itself prevents some types of changes in the name of security – and thus **no** extension can make such modification (sandboxing is an example)
- Thus, some security concerns outside the scope of extensions – up to the browser to implement them

Conclusion

- TODO: Rough conclusion of the things browsers/JS can improve on to prevent fingerprinting, other extensions (like Flashblock, etc), and how effective we think our techniques are at stopping fingerprinting

TODO: Remove me

This slide demonstrates how source code is displayed

```
// Populate the text area with our previously saved array
chrome.storage.sync.get(
    'whitelist',
    function (result){
        // Get the stored whitelist array
        var whitelist = result.whitelist;

        // Iterate over array and populate our page
        for(var i = 0; i < whitelist.length; i++)
        {
            $('#whitelist').append(whitelist[i] + "\n");
        }
    }
);
```