# 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 many users have a somewhat unique set of fonts (often installed by other applications)
- 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

# Available fonts

- Problems arise with page load times and overriding DOM methods
- In the case of panopticlick, Java and Flash were used to try and detect fonts (much easier and faster when fonts can be enumerated rather than having to check the existence from a very large list)
- Thus, the use of Flashblock and preventing Java applets from running is the most useful approach against sites that use installed fonts for fingerprinting

# Available plugins

- Browsers use plugins to extend functionality
- Aside from font detection, plugins are the largest source for fingerprints
- Browsers store information about the plugins in an array that is accessible to outside scripts
- Enumerating this array is useful for fingerprinting
- These plugins can also be specifically targeted by name and version if need be (i.e. 'Silverlight Plug-In', 'Quicktime Plug-in 7.73')
- Thwarted by injecting a script that removes access to the plugins array

# Other fingerprinting threats

- Flash is an effective and very common method to detect fonts (and possibly other information about the system)
- Preventable by changing settings in Flash
  - File: `mms.cfg`
    Line: `DisableDeviceFontEnumeration = 1`
- To prevent these, there are extension like Flashblock
- 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 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

# Results

Panopticlick measures the data collected in terms of "one in x browsers" having certain features

| Parameter | Before | After |
|---|---|---|
| User Agent | 2042.95 | 2042.95 |
| HTTP_ACCEPT Headers | 22704.77 | 39.95 |
| Browser Plugin Details | 799203.2 | 825.63 |
| Time Zone | 30.76 | 30.76 |
| Screen Size and Color Depth | 19.41 | 19.41 |
| System Fonts | 1998008 | 6.37 |
| Are Cookies Enabled? | 1.35 | 1.35 |
| Limited Supercookie test | 1.9 | 1.9 |

# 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)
- Many JavaScript properties are accessed as parameters rather than by functions
- This is a shortfall of the language, which does not provide control over visibility

# Drawbacks

- Thus, while we could block access to a parameter, we could not change the behavior of accessing the parameter
- Our original goal was to alert the user when a page appears to be collecting too much information
- We could only silently thwart the fingerprinting page – they get nothing (but neither do legitimate pages)
- Chrome sandboxes extensions. In order to modify JS variables, we had to make our injected content script further inject code into the page's Document Object Model (DOM)

# Conclusion

- Browser fingerprinting has potential of tracking browsers based on their discernible features
- Features that are largely unique are much more trackable
- Preventing enumeration of these features is the best defence against fingerprinting
- Due to limitations of the JavaScript language and browser extension capabilities, preventing enumeration would be better done by the browser
- Preventing detection of fonts and plugins hinders legitimate sites – better to prevent mass detection

## Demo

Let's now look at a demo of our project