

# Torbutton and Firefox

Mike Perry  
Mozilla Brown Bag  
Jun 22, 2010



# Topics For Today

1. Torbutton's Adversary Model and Requirements
2. Torbutton Functional Overview and Demo
3. Torbutton Architecture & Major Components
4. Comparison to Firefox 3.6 Private Browsing
5. Current Firefox Bugs Impacting Tor Security
6. Awkward XPCOM Interfaces and Inconsistencies
7. Interfaces that would be really, really helpful

# Adversary Goals

1. Bypassing proxy settings
2. Correlation of Tor vs Non-Tor
3. History disclosure
4. Location information
5. Misc Anonymity set reduction (Fingerprinting)
6. History records and other on-disk information

# Adversary Capabilities (Positioning)

- Can modify content at exit node or its router
- Can insert malicious content into ads or websites they control
  - Can target Tor enabled as well as Tor disabled states
- Can insert malicious content into non-Tor traffic
  - At user's local network, ISP, or other upstream router
- Can seize computers of Tor Users

# Adversary Capabilities (Attacks)

- Can insert javascript into content
  - Attribute-based history disclosure
  - Timezone information, Fingerprinting
  - Browser Exploits
- Can insert CSS into content
  - JS-free attribute-based history disclosure
- Can insert plugins into content
  - Proxy bypass, alternate identifier storage
- Can read and insert cookies
- Can create cached content (unique identifiers)

# Torbutton Requirements

1. Proxy Obedience – Obey Tor settings
2. Network Isolation – Don't mix Tor+Non-Tor
3. State Separation – Keep cookies, cache separate
4. Tor Undiscoverability – Hidden while Tor is off
5. Disk Avoidance – Don't write Tor state to disk
6. Location Neutrality – Don't reveal location
7. Anonymity Set Preservation – Mask User Agent
8. Update Safety – No insecure updates via Tor
9. Interoperability – Don't break other extensions

# Major Torbutton Functionality (1)

- Disable plugins while Tor is enabled
  - docShell.allowPlugins
- Isolate dynamic content per Tor load state
  - docShell.allowJavascript
  - nsIContentPolicy
- Cookie jars/cookie clearing
  - Component based on code from Colin Jackson
- Cache management
  - Cache prefs and clearing on toggle
- Prevent Livemark updates

# Major Torbutton Functionality (2)

- History management
  - global-history;2 and nav-history-service;1 hooking
  - Prevent both CSS and JS attacks + history recording
- Tor-specific warning before launching apps
  - Hook external-[helper-app/protocol]-service;1
- User agent+locale spoofing
- Timezone spoofing
  - Store+set the TZ environment variable
- Session Store Blocking in Tor mode
  - Re-register custom copy of nsSessionStore.js



# TorButton Demo

- <https://www.torproject.org/torbutton/design/#SingleStat>
- <http://ha.ckers.org/weird/CSS-history.cgi>
- <http://www.tjkdesign.com/articles/css%20pop%20ups/5>

# Torbutton Architecture

- Browser overlay
  - Tab tags, plugins, Javascript hooks
- XPCOM contract hooking
  - Register a new class-id that implements a contracted component with one or more interfaces
  - Copies uninteresting members and methods
  - Doesn't work if components are referenced by class-id
- Additional Components
  - Cookie Jar handler
  - Map for content windows -> tabs
  - Content Policy

# Browser Overlay

- Per window observers
  - Receives notification via 'tor\_enabled' pref if Tor state changes
    - Updates UI elements accordingly
- “Master Window” observers
  - 'unload' notification to transfer control on close
  - Receives notification if proxy settings change
    - Updates browser prefs and Torbutton settings accordingly
  - Receives notification if any Torbutton prefs change
  - Tab tags and Javascript hooks deployed from a docloaderservice;1 listener

# Unprivileged Javascript Hooks

- Deployed from a docloaderservice;1 weblistener
  - Needs to receive event before content JS runs, but after window object is built.
- calls evalInSandbox with contentWindow.wrappedJSObject as the sandbox
- Currently only used for window.screen
- Can be unmasked in FF3.0+, need alternatives

# Hooked Components

- [@mozilla.org/browser/global-history;2](#)
  - Hooks isVisited to lie to Gecko about visited status
  - Hooks addURI to prevent disk writes during Tor
- [@mozilla.org/browser/sessionstore;1](#)
  - Modified nsSessionStore.js to prevent disk writes
- [@mozilla.org/browser/sessionstartup;1](#)
  - Used for notification of crashes via doRestore()
  - Also doubles as an app-startup observer for Torbutton
- [@mozilla.org/browser/external-protocol-service;1](#)
  - Warns on external app launch (Firefox fails to do so)

# Additional Components

- [@stanford.edu/cookie-jar-selector;2](https://@stanford.edu/cookie-jar-selector;2)
  - Sends 'shutdown-cleanse' profile change messages to the cookiemanager
  - Writes out current state's cookies, loads new state's
- [@torproject.org/content-window-mapper;1](https://@torproject.org/content-window-mapper;1)
  - Searches all windows for tabbedbrowser that owns a content window and caches the result
- [@torproject.org/cssblocker;1](https://@torproject.org/cssblocker;1)
  - Obtains the contentWindow from node param and uses window mapper to obtain tabbrowser
  - Checks tab tag against current state for allow/deny

# Firefox Private Browsing Mode

- Subset of Torbutton Requirements
  - Not concerned with proxies, anonymity set, location
- Anonymity set issues lead to fingerprinting
- Users can still be tracked via plugins
- Form fill is a problem
- HTML5 protocol handlers a problem
- Certificates+SSL Session Ids are a problem
- DNS prefetching+livemarks a potential problem
- External apps/protocols may be a problem

# PBM vs Torbutton

- Torbutton more flexible in allowing the user to persist state if they want
- This is mainly because of the “Toggle-Model”
  - Google Incognito “Window-Model” may be superior
  - This is also why we build Tor Browser Bundle
  - PBM tab save+restore model dodges a lot of issues
- Torbutton has anti-fingerprinting measures
- PBM handles/clears: clipboard, permission manager, the SDR, and error console



# Combining FF PBM with Torbutton

- Primarily of interest so that other addons know to be private.
- Want to preserve Torbutton's options...
- Wrap `nsIObserver::observe` to block “private-browsing” emit for:
  - `nsCookieService`
  - `nsNavHistory`
  - `NsSessionStore`
- Also need to emit an exit followed by an enter if Tor enabled for startup.

# PBM+Torbutton Integration Issues

- Several components directly query the Private Browsing Service, instead of tracking the emits.
  - This makes fine-tuning behavior difficult
- In particular:
  - Form-fill history cannot be enabled via above hacks
  - History UI is altered. Cannot delete items.
  - Passwords can't be stored
  - Content-type prefs can't be saved
- Clean way to preserve DOM storage?
  - APIs are not developed enough

# Firefox Bugs Impacting Tor

- nsNSSCertificateDB::DeleteCertificate has race conditions (Bug 435159)
- Timezone config/hookable JS Date() (Bugs 419598+392274)
- docShell.allowJavascript does not kill all event handlers (Bug 409737)
- docShell.allowPlugins not honored for direct links (Bug 401296, 282106?)
- Others:
  - <https://www.torproject.org/torbutton/design/#FirefoxBugs>

# Awkward Firefox Interfaces

- Lack of context in nsIContentPolicy, nsIWebListener, and nsIProtocolProxyFilter
  - contentWindow vs tab.. What browser am I in?
  - “getMostRecentWindow” has race conditions and getBrowser() not available from components
- Components.classes & interfaces exposed to content JS. Why? Bug? Allows fingerprinting..
- Some components are called only by Class ID
- Some interfaces not suitable for augmentation by hooking

# Interface Wishlist

- Scriptable nsIPluginManager::register/unregister
- Better scriptable DOM Storage APIs
- More fine-grained nsISessionStore interface
- 'app-crash-recover' event before session restore
- Scriptable control over OOP plugin system calls
  - Or force network IO through proxy settings!
- nsIProxyInfo member of tabbrowser to allow per-tab proxying
- Scriptable hooks for to window.screen and Date

# “What can I do to help Tor?”

- Expose PBM + anti-fingerprinting work as components
  - Torbutton needs finer-grained control
- Help fix Tor-related Firefox bugs!
  - <https://www.torproject.org/torbutton/design/#FirefoxBugs>
- Extra bandwidth? Run a node!
  - See Tor source contrib directory for Linux 'tc' prioritization script
  - No need to impact your own traffic flows