Torbutton and Firefox

Mike Perry
Mozilla Brown Bag
Jan 23, 2008

Topics For Today

1. Torbutton's Adversary Model and Requirements
2. Torbutton Functional Overview and Demo
3. Torbutton Architecture & Major Components
4. Current Firefox Bugs Impacting Tor Security
5. Awkward XPCOM Interfaces and Inconsistencies
6. 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
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
  - 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. Disk Avoidance – Don't write Tor state to disk
5. Location Neutrality – Don't reveal location
6. Anonymity Set Preservation – Mask User Agent
7. Update Safety – No updates via Tor
8. 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

Major Torbutton Functionality (2)

- History management
  - global-history;2 contract hooking
  - Prevent both CSS and JS attacks
- User agent spoofing during Tor
  - user agent prefs and navigator object hooking
- Timezone+Locale spoofing
  - Date object hooking and intl.* prefs
- Session Store Blocking in Tor mode
  - Re-register custom copy of nsSessionStore.js

TorButton Demo

- http://gemal.dk/browserspy/basic.html
- http://gemal.dk/browserspy/css.html
- http://gemal.dk/browserspy/date.html
- http://gemal.dk/browserspy/plugins.html
- http://metasploit.com/research/misc/decloak/index
- http://ha.ckers.org/weird/CSS-history.cgi
- http://www.tjkdesign.com/articles/css%20pop%2

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
  - Recieves 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
- Date hooks use lexical scoping to maintain a reference to original Date class.
  - Create new constructor + prototypes for all methods
  - Prototypes rebuilt inside constructor for each new Date object to provide unique lexically scoped hidden instance for every wrapped instance

Hooked Components

- @mozilla.org/browser/global-history;2
  - Hooks isVisited to lie to Gecko about visted status if Tor is enabled
  - Hooks addURI to prevent disk writes during Tor
- @mozilla.org/browser/sessionstore;1
  - Modified copy of nsSessionStore.js to prevent writing to disk if Tor is enabled
- @mozilla.org/browser/sessionstartup;1
  - Used for notification of crashes via doRestore()
  - Also doubles as an app-startup observer for Torbutton

Additional Components

- @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
  - Searches all windows for tabbedbrowser that owns a content window and caches the result
- @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 Bugs Impacting Tor

- MAJOR: docShell.allowJavascript does not kill all event handlers (Bug 409737)
- MAJOR: Firefox 3 Contract ID hooking issues (Bug 413682)
- docShell.allowPlugins not honored for direct links (Bug 401296, 282106?)
- nsIContentPolicy never receives calls to shouldProcess() (Bugs 309524 and 380556)
- navigator fields ignore some useragent settings
- file:// urls can read and submit local files

Awkward Firefox Interfaces

- Lack of context in nsIConentPolicy, nsIWebListener, and nsIProtocolProxyFilter
  - contentWindow vs tab.. What browser am I in?
  - “getMostRecentWindow” has race conditions and getBrowser() not available from components
- Some components are called only by Class ID
- Some interfaces not suitable for augmentation by hooking
- Difficult to get event just prior to client JS load
- Need for 'Hidden Window'-like hacks

Interface Wishlist

- Individual plugin enable/disable control (done?)
- Timezone config setting (Bug 392274)
- More fine-grained nsISessionStore interface
- 'app-crash-recover' observer event
- nsIProxyInfo member of tabbrowser to allow per-tab proxying

“What can I do to help Tor?”

- Extra bandwidth? Run a node!
  - See Tor source contrib directory for Linux 'tc' prioritization script
  - No need to impact your own traffic flows
- Vote for my Firefox bugs!