History Stealing

Ramin Sadre

History Stealing

- Also called: History Sniffing
- Goal: Website X wants to know what other websites user Y has visited before
- This is an attack against the privacy of the user
- Reasons:
  1. Website X wants to learn about its competitors (“Are Proximus users also sometimes visiting the Base website?”)
  2. Preparing a phishing attack (“Which bank is Y using?”)
  3. Improve user experience (“If our users are often using facebook, let’s copy their user interface!”)

History Stealing: How to do it?

- Well, it's not possible *in theory*. Javascript on a website X is not allowed to access
  - the browser history
  - the cache contents (unless objects from X)
  - cookies stored in the browser (unless cookies from X)
- So, there is no way, right?
- Actually, there are a lot of ways
  - Some of them have been fixed years ago
  - Some are still there and are very difficult to fix
  - Some are "intentional" and technically not stealing... https://nakedsecurity.sophos.com/2018/07/06/chrome-and-firefox-pull-history-stealing-browser-extension/

Stealing through CSS

- Demonstrated in 2010
- https://dbaron.org/mozilla/visited-privacy
- Based on the fact that browsers can display *visited* and *unvisited* links differently, controlled by user settings and CSS files:

```
:link, :visited {
      text-decoration: underline ;
}
:link {
      color: blue ; /* unvisited links */
}
:visited {
      color: red ; /* visited links */
}
```

Stealing through CSS (2)

- Attack: A website can check the display style of a (hidden) link

```
var links = document.links;
for (var i = 0; i < links.length; ++i) {
  var link = links[i];
  if (getComputedStyle(link, "").color == "rgb(0,0,128)") {
      // we know link.href has not been visited
      ...
  } else {
      // we know link.href has been visited
      ...
  }
}
```

Stealing through CSS (3)

- This vulnerability has been fixed in all major browsers:
  - Fix 1: `getComputedStyle` always returns the unvisited style for links
    - But that's not enough: by using different font sizes etc., the website could check differences in layout of visited and unvisited links!
  - Fix 2: websites are only allowed to change the *color* of links (no impact on layout)

Timing Attack

- General: Timing Attacks exploit the fact that some operations take longer than others
- The website X executes one of the below operations and measures the time to complete it
  - Define a very complex style for visited links, such that they take more time to be displayed.
  - Request an object from website Y. If the user already visited Y before, the object is in the cache and is retrieved very quickly
  - Request an object from website Y. If the user already visited Y before, the IP address is still in the DNS cache of the browser and no DNS query is sent.
  - Load website Y into an iframe

Timing Attack (2)

- Countermeasures:
  1. Modern browsers have optimized render engines where the display speed difference between styles is very small.
  2. Disable the cache ☹
  3. Disable DNS ☹
  4. Disable Javascript ☹
- Timing attacks based on the cache were invented 22 years ago: https://pr.princeton.edu/news/00/q4/1205-browser.htm
  - Fixed only 3 years ago: https://www.jefftk.com/p/shared-cache-is-going-away
  - Solution: cache isolation. If a website www.ABC.com loads an object from a different domain www.XYZ.com, the cache will be ignored.

Geo Inference

- History stealing techniques can also be used to find the geographical position of the user
  - Check whether the user has visited the weather website of city A, B, C, …
  - Check whether the user has accessed a certain tile of Google maps
  - …

- http://w2spconf.com/2014/papers/geo_inference.pdf

History stealing in practice

- When the first timing attacks were first presented in 2000, researchers were not sure whether they were used in practice
- In 2010, researchers tested 50000 popular web sites for history stealing
  - More than 40 websites were using history stealing techniques to check visits to competitors!
  - *D. Jang, R. Jhala, S. Lerner, H. Shacham. 2010. An empirical study of privacy-violating information flows in JavaScript web applications. In Proceedings of the 17th ACM conference on Computer and communications security (CCS '10)*
- People still discover new history stealing techniques. From 2018:

https://www.usenix.org/system/files/conference/woot18/woot18-paper-smith.pdf

Tracking

User tracking

- Many people are interested in tracking you or your Internet usage
  - Website operators (“Is this a returning user?”)
  - Advertisement (“Personalized ads”)
  - Data brokers (collecting information about you and selling it to advertisement companies)
  - Governments

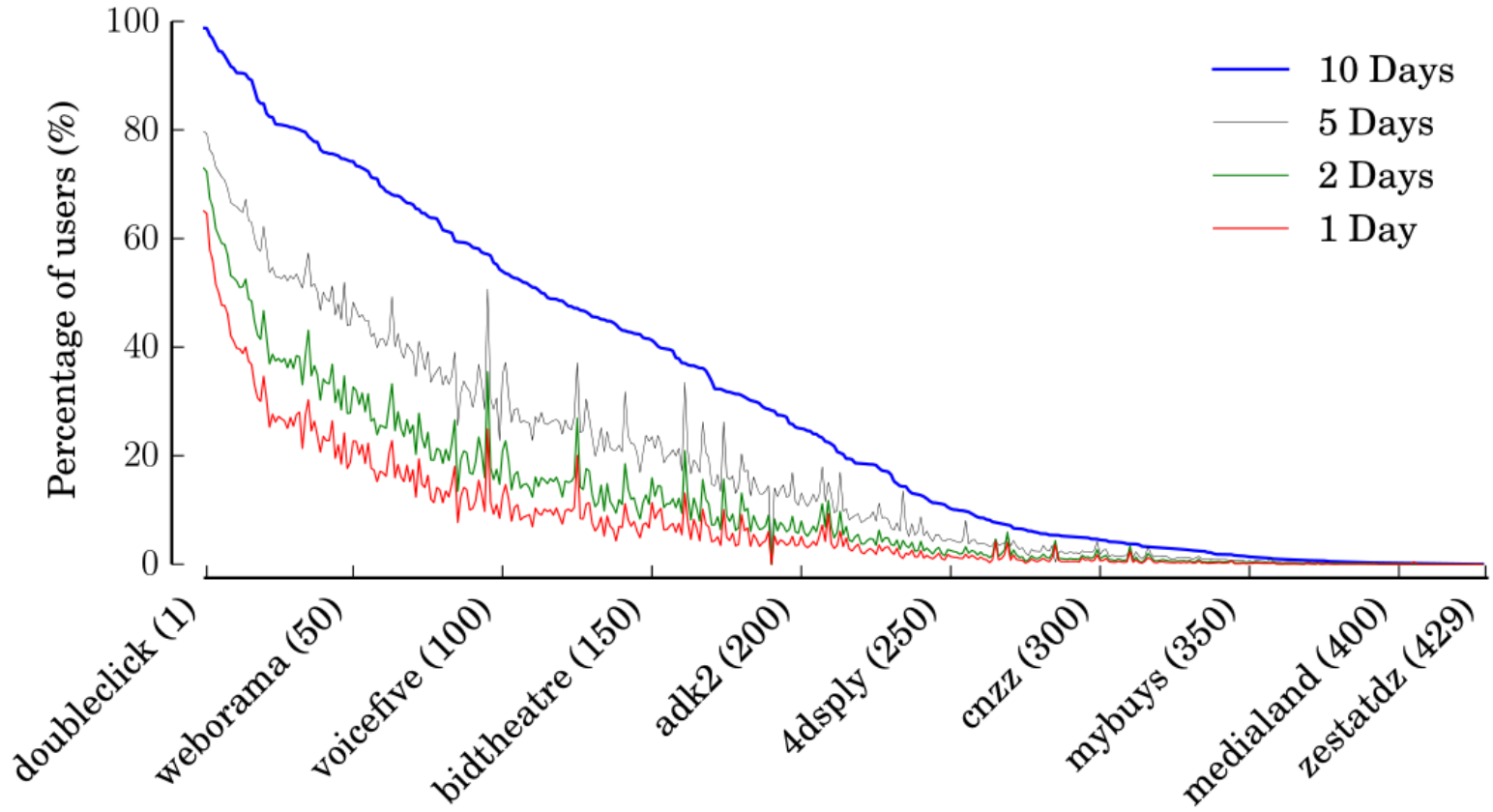
- In the web, your browser has a simple mechanism for that: Cookies

The tracking industry

- Nowadays, websites do not track you manually
- Instead, they embed links to tracking services on their web pages
- When you visit two different websites, it is very likely that they will share one or more of these trackers:
  - Doubleclick
  - Google Analytics
  - Google Syndication
  - Weborama
  - Voicefive
  - …
- Never heard of Weborama or Voicefive? There are more than 400 tracking services in the web!

The tracking industry (2)

- http://porto.polito.it/2602582/1/2602582.pdf



Fingerprinting

- So, just block cookies?
- Unfortunately, there are other means to track you
- The information that your browser is sending to web servers in the HTTP-header fields is unique enough to identify you among millions of users
  - Browser and OS version
  - Language
  - Version number of plugins
  - List of installed fonts
  - Support of special features (GPU,...)
  - ...

- https://panopticlick.eff.org/
- https://amiunique.org/

Perma-cookies

- In 2016, Verizon was fined $1.35 million for tracking their users with perma-cookies

- Perma-cookie (or Supercookie) = a cookie injected into the HTTP requests of users *by their Internet Service Provider or their mobile phone operator*
  - Not visible to the user
  - Clearing your browser cache&cookies does not help
  - Can be also done with other protocols, of course
- Fortunately, many website nowadays use HTTPS