These new techniques expose your browsing history to attackers

30 October 2018

```
class AttackPainter {
  static get inputProperties () {
    // Use the CSS font-family property as a communication channel from the main
    // script.
    return ['font-family'];
  }

  paint (ctx, geom, properties) {
    // Retrieve the key corresponding to the target URL.
    var targetKey = properties.get('font-family').toString();

    // Abuse registerPaint to perform a 1-bit swap operation in persistent state.
    try {
      // If the painter has *not* been previously executed for the given key,
      // this will set the bit for that key.
      registerPaint(targetKey, AttackPainter);
    } catch (e) {
      // Otherwise, a painter-already-registered exception will be thrown, telling is
      // the bit has already been set to 1.
      try {
        // Set another bit in persistent state marking the fact that the painter was
        // executed more than once for the given key. This will be observable from the
        // main script.
        registerPaint(targetKey + '_visited', AttackPainter);
      } catch (e2) {}
    }
  }
}

registerPaint('attack', AttackPainter);
```

An example of code the researchers used for their attacks. Credit: University of California San Diego

Security researchers at UC San Diego and Stanford have discovered four new ways to expose Internet users' browsing histories. These techniques could be used by hackers to learn which websites users have visited as they surf the web.

The techniques fall into the category of "history sniffing" attacks, a concept dating back to the early 2000s. But the attacks demonstrated by the researchers at the 2018 USENIX Workshop on Offensive Technologies (WOOT) in Baltimore can profile or 'fingerprint' a user's online activity in a matter of seconds, and work across recent versions of major web browsers.

All of the attacks the researchers developed in their WOOT 2018 paper worked on Google Chrome. Two of the attacks also worked on a range of other browsers, from Mozilla Firefox to Microsoft Edge, as well various security-focused research browsers. The only browser which proved immune to all of the attacks is the Tor Browser, which doesn't keep a record of browsing history in the first place.

"My hope is that the severity of some of our published attacks will push browser vendors to revisit how they handle history data, and I'm happy to see folks from Mozilla, Google, and the broader World Wide Web Consortium (W3C) community already engage in this," said Deian Stefan, an assistant professor in computer science at the Jacobs School of Engineering at UC San Diego and the paper's senior author.

"History sniffing": smelling out your trail across the web

Most Internet users are by now familiar with "phishing;" cyber-criminals build fake websites which mimic, say, banks, to trick them into entering their login details. The more the phisher can learn about their potential victim, the more likely the con is to succeed. For example, a Chase customer is much more likely to be fooled when presented with a fake Chase login page than if the phisher pretends to be Bank of America.

After conducting an effective history sniffing attack, a criminal could carry out a smart phishing scheme, which automatically matches each victim to a faked page corresponding to their actual bank. The phisher preloads the attack code with their list of target banking websites, and conceals it in, for example, an ordinary-looking advertisement. When a victim navigates to a page containing the attack, the code runs through this list, testing or 'sniffing' the victim's browser for signs that it's been used to visit each target site. When one of these sites tests positive, the phisher could then redirect their victim to the corresponding faked version.

The faster the attack, the longer the list of target sites an attacker can 'sniff' in a reasonable amount of time. The fastest history sniffing attacks have

reached rates of thousands of URLs tested per second, allowing attackers to quickly put together detailed profiles of web surfers' online activity. Criminals could put this sensitive data to work in a number of ways besides phishing: for example, by blackmailing users with embarrassing or compromising details of their browsing histories.

History sniffing can also be deployed by legitimate, yet unscrupulous, companies, for purposes like marketing and advertising. A 2010 study from UC San Diego documented widespread commercial abuse of previously known history sniffing attack techniques, before these were subsequently fixed by browser vendors.

"You had internet marketing firms popping up, hawking pre-packaged, commercial history sniffing 'solutions', positioned as analytics tools," said Michael Smith, a computer science Ph.D. student at UC San Diego and the paper's lead author. The tools purported to offer insights into the activity of their clients' customers on competitors' websites, as well as detailed profiling information for ad targeting—but at the expense of those customers' privacy.

"Though we don't believe this is happening now, similar spying tools could be built today by abusing the flaws we discovered," said Smith.

New attacks

The attacks the researchers developed, in the form of JavaScript code, cause web browsers to behave differently based on whether a website had been visited or not. The code can observe these differences—for example, the time an operation takes to execute or the way a certain graphic element is handled—to collect the computer's browsing history. To design the attacks, researchers exploited features that allow programmers to customize the appearance of their web page—controlling fonts, colors, backgrounds, and so forth—using Cascading Style Sheets (CSS), as well as a cache meant to improve to performance of web code.

The researchers' four attacks target flaws in relatively new browser features. For example, one attack takes advantage of a feature added to Chrome in 2017, dubbed the "CSS Paint API", which lets web pages provide custom code for drawing parts of their visual appearance. Using this feature, the attack measures when Chrome re-renders a picture linked to a particular target website URL, in a way invisible to the user. When a re-render is detected, it indicates that the user has previously visited the target URL. "This attack would let an attacker check around 6,000 URLs a second and develop a profile of a user's browsing habits at an alarming rate," said Fraser Brown, a Ph.D. student at Stanford, who worked closely with Smith.

Though Google immediately patched this flaw—the most egregious of the attacks that the researchers developed—the computer scientists describe three other attacks in their WOOT 2018 paper that, put together, work not only on Chrome but Firefox, Edge, Internet Explorer, but on Brave as well. The Tor Browser is the only browser known to be totally immune to all the attacks, as it intentionally avoids storing any information about a user's browsing history.

As new browsers add new features, these kinds of attacks on privacy are bound to resurface.

A proposed defense

The researchers propose a bold fix to these issues: they believe browsers should set explicit boundaries controlling how users' browsing histories are used to display web pages from different sites. One major source of information leakage was the mechanism which colors links either blue or purple depending on whether the user has visited their destination pages, so that, for example, someone clicking down a Google search results page can keep their place. Under the researchers' model, clicking links on one website (e.g., Google) wouldn't affect the color of links appearing on another website (e.g., Facebook). Users could potentially grant exceptions to certain websites of their choosing. The researchers are prototyping this fix and evaluating the trade-offs of such a privacy-conscious browser.