New system patches security holes left open by web browsers' private-browsing functions

23 February 2018, by Larry Hardesty

Generally, a browser won't know where the data it downloaded has ended up. Even if it did, it wouldn't necessarily have authorization from the operating system to delete it. Credit: Massachusetts Institute of Technology

Today, most web browsers have private-browsing modes, in which they temporarily desist from recording the user's browsing history.

But data accessed during private browsing sessions can still end up tucked away in a computer's memory, where a sufficiently motivated attacker could retrieve it.

This week, at the Network and Distributed Systems Security Symposium, researchers from MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL) and Harvard University presented a paper describing a new system, dubbed Veil, that makes private browsing more private.

Veil would provide added protections to people using shared computers in offices, hotel business centers, or university computing centers, and it can be used in conjunction with existing private-browsing systems and with anonymity networks such as Tor, which was designed to protect the identity of web users living under repressive regimes.

"Veil was motivated by all this research that was done previously in the security community that said, 'Private-browsing modes are leaky—Here are 10 different ways that they leak,'" says Frank Wang, an MIT graduate student in electrical engineering and computer science and first author on the paper. "We asked, 'What is the fundamental problem?' And the fundamental problem is that [the browser] collects this information, and then the browser does its best effort to fix it. But at the end of the day, no matter what the browser's best effort is, it still collects it. We might as well not collect that information in the first place."

Wang is joined on the paper by his two thesis advisors: Nickolai Zeldovich, an associate professor of electrical engineering and computer science at MIT, and James Mickens, an associate professor of computer science at Harvard.

Shell game

With existing private-browsing sessions, Wang explains, a browser will retrieve data much as it always does and load it into memory. When the session is over, it attempts to erase whatever it retrieved.

But in today's computers, memory management is a complex process, with data continuously moving around between different cores (processing units) and caches (local, high-speed memory banks). When memory banks fill up, the operating system might transfer data to the computer's hard drive,

where it could remain for days, even after it's no longer being used.

Generally, a browser won't know where the data it downloaded has ended up. Even if it did, it wouldn't necessarily have authorization from the operating system to delete it.

Veil gets around this problem by ensuring that any data the browser loads into memory remains encrypted until it's actually displayed on-screen. Rather than typing a URL into the browser's address bar, the Veil user goes to the Veil website and enters the URL there. A special server—which the researchers call a blinding server—transmits a version of the requested page that's been translated into the Veil format.

The Veil page looks like an ordinary webpage: Any browser can load it. But embedded in the page is a bit of code—much like the embedded code that would, say, run a video or display a list of recent headlines in an ordinary page—that executes a decryption algorithm. The data associated with the page is unintelligible until it passes through that algorithm.

Decoys

Once the data is decrypted, it will need to be loaded in memory for as long as it's displayed on-screen. That type of temporarily stored data is less likely to be traceable after the browser session is over. But to further confound would-be attackers, Veil includes a few other security features.

One is that the blinding servers randomly add a bunch of meaningless code to every page they serve. That code doesn't affect the way a page looks to the user, but it drastically changes the appearance of the underlying source file. No two transmissions of a page served by a blinding sever look alike, and an adversary who managed to recover a few stray snippets of decrypted code after a Veil session probably wouldn't be able to determine what page the user had visited.

If the combination of run-time decryption and code obfuscation doesn't give the user an adequate sense of security, Veil offers an even harder-to-hack option. With this option, the blinding server opens the requested page itself and takes a picture of it. Only the picture is sent to the Veil user, so no executable code ever ends up in the user's computer. If the user clicks on some part of the image, the browser records the location of the click and sends it to the blinding server, which processes it and returns an image of the updated page.

The back end

Veil does, of course, require web developers to create Veil versions of their sites. But Wang and his colleagues have designed a compiler that performs this conversion automatically. The prototype of the compiler even uploads the converted site to a blinding server. The developer simply feeds the existing content for his or her site to the compiler.

A slightly more demanding requirement is the maintenance of the blinding servers. These could be hosted by either a network of private volunteers or a for-profit company. But site managers may wish to host Veil-enabled versions of their sites themselves. For web services that already emphasize the privacy protections they afford their customers, the added protections provided by Veil could offer a competitive advantage.

"Veil attempts to provide a private browsing mode without relying on browsers," says Taesoo Kim, an assistant professor of computer science at Georgia Tech, who was not involved in the research. "Even if end users didn't explicitly enable the private browsing mode, they still can get benefits from Veil-enabled websites. Veil aims to be practical—it doesn't require any modification on the browser side—and to be stronger—taking care of other corner cases that browsers do not have full control of."

More information: Veil: Private Browsing Semantics Without Browser-side Assistance: frankwang.org/files/papers/wang-veil.pdf

Provided by Massachusetts Institute of Technology

APA citation: New system patches security holes left open by web browsers' private-browsing functions (2018, February 23) retrieved 24 June 2021 from https://techxplore.com/news/2018-02-patches-holes-left-web-browsers.html