New web privacy system could revolutionize the safety of surfing

October 5 2014

Researchers from UCL, Stanford Engineering, Google, Chalmers and Mozilla Research have built a new system that protects Internet users' privacy whilst increasing the flexibility for web developers to build web applications that combine data from different web sites, dramatically improving the safety of surfing the web.

The system, 'Confinement with Origin Web Labels,' or COWL, works with Mozilla's Firefox and the open-source version of Google's Chrome web browsers and prevents malicious code in a web site from leaking sensitive information to unauthorised parties, whilst allowing code in a web site to display content drawn from multiple web sites – an essential function for modern, feature-rich web applications.

Testing of COWL prototypes for the Chrome and Firefox web browsers shows the system provides strong security without perceptibly slowing the loading speed of web pages. Following its announcement today, COWL will be freely available for download and use on 15th October from http://cowl.ws. The team who developed it, including two PhD students from Stanford (working in collaboration with Mozilla Research) and a recently graduated PhD from UCL (now employed by Google), hope COWL will be widely adopted by web developers.

Currently, web users' privacy can be compromised by malicious JavaScript code hidden in seemingly legitimate web sites. The web site's operator may have incorporated code obtained elsewhere into his or her web site without realising that the code contains bugs or is malicious.

Such code can access sensitive data within the same or other browser tabs, allowing unauthorised parties to obtain or modify data without the user's knowledge.

The research team describe COWL in a paper that will appear in the Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation, a premier venue for operating systems research.

Co-author Professor Brad Karp (UCL Computer Science), said: "COWL achieves both privacy for the user and flexibility for the web application developer. Achieving both these aims, which are often in opposition in many system designs, is one of the central challenges in computer systems security research.

"The new system provides a property known as 'confinement' which has been known since the 1970s, but proven difficult to achieve in practical systems like web browsers. COWL confines JavaScript programs that run within the browser, such as in separate tabs. If a JavaScript program embedded within one web site reads information provided by another web site – legitimately or otherwise – COWL permits the data to be shared, but thereafter restricts the application receiving the information from communicating it to unauthorised parties. As a result, the site that shares data maintains control over it, even after sharing the information within the browser."

Co-author Professor David Mazières (Stanford University Computer Science), said: "Security mechanisms for the web must keep pace with the web's rapid evolution. Current measures, such as the Same Origin Policy (SOP), work by stopping JavaScript programs embedded within one web site – malicious or otherwise – from reading data hosted by a separate web site. This brittle approach doesn't work for modern so-called 'mashup' applications that combine information from multiple

web sites. Essentially, the SOP doesn't fit how many web sites are built today. And prior attempts at weakening the SOP to allow this sort of sharing, such as with Cross-Origin Resource Sharing (CORS), make it trivial for malicious code to leak sensitive data to unauthorised parties."

When building a modern web site, web developers routinely incorporate JavaScript library code written by third-party authors of unknowable intent. The study cites measurements indicating that 59% of the top one million web sites and 77% of the top 10,000 web sites incorporate a JavaScript library written by a third party. The team say such inclusion of JavaScript libraries is dangerous, as although the code includes features the web developers want, it might also contain malicious code that steals the browser user's data. In such cases, the SOP cannot protect sensitive data, as the included library is hosted by the same web site origin (i.e., under the same Internet domain name).

Professor Karp said: "By blocking the building of web applications that synthesize content from multiple web sites, the SOP actually forces web developers to make design choices that put users' privacy at risk. That's a problem we've solved with COWL.

"For example, one useful web application would let users check they're not being overcharged for items they've ordered from Amazon. The app would have to pull in information from the user's bank statement and Amazon, reconcile the two, and present the result in the browser. To do this, a web developer would need to write code that integrated data from the bank's web site with data from Amazon's web site but the SOP would block this, as the two data sources are hosted by different web domain names. Today's web developers get around this by writing an app that asks the user for their bank and Amazon login credentials, so it can log into both services and collect information as if it is the user. This clearly compromises the user's privacy as the provider of the app gains full access to the user's online banking system and Amazon account."

Deian Stefan, lead PhD student on the project at Stanford, said: "What we've achieved in COWL is a system that lets web developers build feature-rich applications that combine data from different web sites without requiring that users share their login details directly with third-party web applications, all while ensuring that the user's sensitive data seen by such an application doesn't leave the browser. Both web developers and users win."

The research team has shown how to use COWL to build four applications previously unachievable with strong privacy, including an encrypted document editor, a third-party mashup application, a password manager, and a web site that safely includes an untrusted third-party library.

More information: Stefan, D., Yang, E., Marchenko, P., Russo, A., Herman, D., Karp, B., and Mazières, D., Protecting Users by Confining JavaScript with COWL, to appear in the Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2014), Broomfield, CO, October, 2014.

Provided by University College London