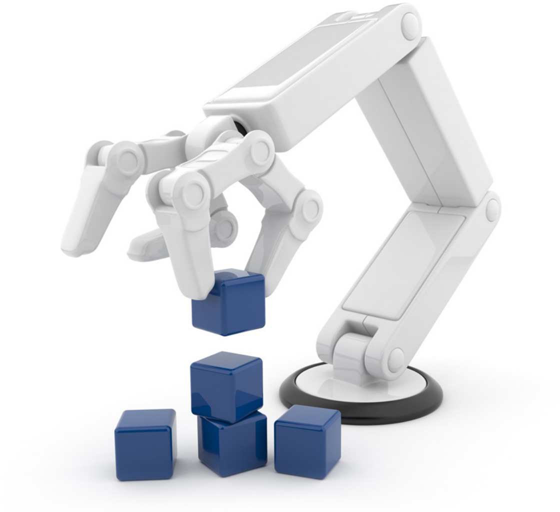
SOPHOS

simple + secure



HTML5 and security on the new web

By **James Lyne**, Director of Technology Strategy

There are lots of changes happening to the key technologies that power the web. The new version of HTML, the dominant web language, offers impressive enhancements for rich web applications. But as HTML5 comes into greater use we'll see new security issues arise.

It's typical for a new technology to have defects and pitfalls. And although the standard is still being defined, it's already being implemented. So how does HTML5 stand up to security scrutiny?

A changing web

The upgrade from HTML4 to HTML5 brings about a wide range of new and amazing possibilities for the web like 3D environments and email clients that work offline. And new products like Sophos' upcoming SSL VPN will be written entirely in HTML5.

HTML5 provides far more access to the computer's resources than its predecessor. It's built to integrate with modern web browsing devices and offers capabilities like location awareness, graphics rendering and access to the webcam and microphone. You can even use the application cache feature to have an application download to your browser for use offline, such as when you're on a plane.

The potential of this language is reinforced by Adobe's announcement that it's working to transition from mobile Flash to HTML5 as the future of rich web media. Although the standard owned by the World Wide Web Consortium for HTML5 is still in draft form, a number of popular web browsers have already adopted it. This means that different browsers are implementing HTML5 in different ways with varying security models.

The enhancements are great, but they radically change the attack model for the browser. We always hope new technologies can close old avenues of attack. Unfortunately, they can also present new opportunities for cybercriminals.

With more local storage and offline chaching in HTML5, the browser is likely to contain much more sensitive data. That makes your browser a direct gateway to your data and an attractive target.

Browser vulnerabilities

Each browser's implementation of HTML5 varies, so it's hard to generalize about their security. HTML5 contains new security features, but it's not a security release—it's a significant update to the standard in every way.

Traditionally, the browser was a kind of thin client with small amounts of persistent data from cookies and cached files for performance. Attackers would use the browser as a means to access the computer, or to try to steal credentials for an online service.

However, with more local storage and offline caching in HTML5, the browser is likely to contain much more sensitive data, such as from your email client or CRM. That makes your browser a direct gateway to your data and an attractive target itself. This subtle change in the attack vector could make trouble, as browser vendors will need to develop a richer security model, much like those for operating systems. A lack of definition of this security model in the standard raises further challenges, as browser vendors are left to make these design decisions independently.

That said, the rollout of HTML5 will at least put us on a path to standardization, replacing insecure third party add-ons like Adobe Flash, which has suffered numerous exploits by cybercriminals.

Privacy concerns

Concerns over privacy have led to numerous regulations focused on cookies. Cookies and associated data can be used to track users across multiple sites, recording their clicks, purchases or preferences. As consumers become more aware of web tracking and data mining, privacy features are growing in importance for browser vendors.

In HTML5 the new local storage mechanisms open up new ways to store information about users that could compromise your privacy or tip off cybercriminals. There's more flexibility for local storage and a relatively liberal access model, depending on site developer implementation. And how to restrict or periodically purge data is less clear than with cookies in HTML4.

Because we do so much of our web browsing from our mobile devices, location data and media tools for mobile devices present additional privacy challenges.

HTML5 can more natively interact with the capabilities of modern web browsing devices. And it defines a series of new helpful application programming interfaces (API) for access to location services, microphones or cameras. However, these services have less-tested security models and have already shown some security weaknesses.

For example, we've seen problems where accessing the location services API cache could enable a caller to identify the last queried location of the user without explicit permission. The media access API for cameras and microphones is also not well defined and varies across browsers, potentially providing cybercriminals with a way to break in and use them for spying. Imagine a web-based attack where hackers turn on the microphone on your iPad and record you unnoticed.

In general, a more flexible and integrated technology carries a greater risk of privacy invasion and data loss. But we should also consider that moving these capabilities into the core language and browser is a step up from the terrible plugins that cybercriminals have targeted over the past 18 months.

Overall, traditional privacy tools and policies will need to catch up to new technologies. We can expect browser vendors to invest in more privacy controls. And we'll likely see some revisions to guidelines from privacy regulators.

Traditional privacy tools and policies will need to catch up to new technologies. We can expect browser vendors to invest more in privacy controls. And we'll likely see some revisions to guidelines from privacy regulators.

Sandboxing and permissions

One question we're asking is how sandboxing and isolation models of browsers will evolve for HTML5. Many browsers work to prevent distribution of malware by isolating themselves from the operating system using sandboxing. But all the new browser capabilities provided by HTML5 create the type of opportunities for data theft that we've had to deal with in our operating systems.

Your browser will be able to access local data, break out of the sandbox, and capture data via your media devices or your location. When your browser does these things at the same time as you visit a website that's been infected with nasty attack code, it's a bad combination. There's not a great deal of definition on how the permissions model will work here, but with the browser becoming more capable, the security model needs to become more multi-dimensional.

We should also be looking at cross document messaging, which attempts to prevent cross-site scripting (XSS) by restricting permissions to a domain. Abuse of DNS and insecure use of the API could also leave websites open to abuse and manipulation in new ways.

Separating individual sessions and the rights of different authors, publishers and sites within the content rendering environment becomes all the more important. There are a few ways these controls could be implemented. But having users set controls by answering a complex set of questions would have a negative impact on security.

Legacy problems

Some of the old security issues of HTML4 and JavaScript remain in HTML5. And cybercriminals who spread malware or steal user information on the web will continue to seek new ways of doing so in HTML5.

Browser vendors have patched many of the security holes that opened the door to clickjacking (tricking a user to click a link or modifying the page to emulate a user click) and phishing websites. Or they've put in place restrictions to minimize the probability of attack, such as requiring a click to occur in the window of focus.

As we launch new technology we have to learn from our past mistakes. As cybercriminals investigate HTML5 they are likely to find new ways of tricking users, spreading malware and stealing clicks.

Developers adopting HTML5 are going to need to change their validation routines and filters. Plenty of websites use web application firewalls or free add-ons like mod_security to prevent attacks like XSS. Better still is for developers to implement proper checks in their code to prevent incorrect use of data.

The traditional focus areas for XSS attacks might widen and change. Media related functions are notoriously problematic—just look at the major attack vectors used by the bad guys over the past couple of years. Tags like <video>, <audio> or <canvas> might open up new possibilities for attackers.

Websites and applications are only as secure as the care the web developer took to make it. If you're a web developer, be sure to follow best practices for filtering data and writing secure code, and borrow from cheat sheets like those produced by the Open Web Application Security Project (OWASP).

What's next

On our journey to greater web usability we're seeing interesting tensions between flexibility, security and privacy. HTML5 introduces some powerful concepts and we should look forward to using more of these web applications. But it's far from perfect.

HTML5 does make progress in usability and security from today's HTML4 and associated components, but it will require a lot of work to be as tried and tested. The web is the biggest vector for distribution of malware. As web technology merges with conventional thick client capabilities and is deployed across a wider range of devices, we can expect cybercriminals to find new ways to attack.

And as HTML5 becomes the backbone of our ecommerce, social media and elearning, the security community will need to diligently identify new attack vectors and revise the security model. We should demand that standards committees and browser vendors be consistent and act responsibly in their work now on HTML5, before the cybercriminals get ahead of us.

James Lyne, Director of Technology Strategy @jameslyne

United Kingdom Sales:
Tel: +44 (0)8447 671131
Email: sales@sophos.com

North American Sales:
Toll Free: 1-866-866-2802
Email: nasales@sophos.com

Boston, USA | Oxford, UK

A Sophos Article 12.11v1.dNA