Browser Security Track @ OWASP Summit 2011 (15:40 - 16:45 Thursday Feb 10)

HTML5 Security

B Sterne == Brandon Sterne, Mozilla
E Vela Nava == Eduardo Vela Nava, Google
G Heyes == Gareth Heyes
M Heiderich == Mario Heiderich, Uni Bochum
J Schuh == Justin Schuh, Google
J Hodges == Jeff Hodges, PayPal
I Fette == Ian Fette, Google
D Lindsay == David Lindsay, Cigital
L Adamski == Lucas Adamski, Mozilla
J Nagra == Jasvir Nagra, Google
G Hogben == Giles Hogben, Enisa
C Hofmann == Chris Hofmann, Mozilla

[About this document] These are the raw notes from the Summit discussion on HTML5 Security. All the cited people have had the chance to edit these notes but there may still be errors and misunderstandings in here. Along with the notes from the other browser security sessions – Site Security Policy, DOM sandboxing, and EcmaScript 5 Security – these notes will be the foundation of the forthcoming Browser Security Report. If you have any questions regarding this document, please email john.wilander@owasp.org.

B Sterne: Please consider subscribing to the public-web-security@w3.org list

[The session starts with an HTML5 intro à la Star Wars scroll]

G Heyes: Back compatibility of HTML has given us a mess.

M Heiderich: Part of the good old browser war in late 1990s early 2000s was to "parse anything that came your way". Both W3C and WHATWG are specifying HTML5. Which one are browser vendors following?

J Schuh: Well, the WHATWG one has more detail.

I Fette: We probably shouldn't discuss that one here.

I Fette: HTML5 is more precise than the preceding spec which is good.

G Heyes: Compare ES5 and HTML5. The attack surface of HTML5 is increasing while ES5 at least is getting better.

I Fette: HTML5 tries to make things possible that previously required a plugin. Such feature expansion will always increase the attack surface. HTML5 is essentially taking over plugin attack surface.

D Lindsay: There are new tags that are interesting from an attacker's point of view. How do you draw the line? If we're opening up for many more XSS attack vectors?

L Adamski: Examples of what you mean? It helps to have specific examples. General concerns are hard to address.

D Lindsay: We have video and audio, for instance
`<video poster="javascript:alert(1)">`.

J Nagra: OK, so HTML5 means new functionality being introduced and old being cleaned up. But what should security people look at? It would help to go through the spec and highlight "New apis but implementable in terms of old apis" and "New apis + adds attack surface". That way security-minded people could just focus on those features that create new attack surfaces and we could get out of the way of people creating much needed new features.

G Hogben: Enisa is working on a threat model for HTML5.

B Sterne: If you search the spec for security sections you'll find a few. Some people want those collected in a separate section while others don't want that. It's a very large corpus of text. Not everyone will be satisfied.

J Hodges: The spec should not only contain security considerations for web app developers but also for developers who implement HTML5 support in browsers.

L Adamski: We do want consistency in APIs and if you do find odd behavior in a browser just file a bug. Sometimes there is careful consideration behind a certain behavior but not always.

C Hofmann: When we look at HTML5 there are both security vulnerabilities introduced by the spec and vulnerabilities introduced by the implementation. If you find bugs in the spec you need to give that feedback to the spec authors in WHATWG.

L Adamski: Having statistics of how popular a certain API or property is would be a good starting point for deciding to pull things.

D Lindsay: Regarding the attack surface in the spec – it's often not really in the spec but rather in the *assumptions* of the ones implementing the spec.

I Fette: When we introduce new things there's always the possibility we break people's assumptions of how things are. But are we opening up fundamentally new attack surfaces?

G Heyes: One new attack surface is putting logic into the HTML for instance via regexps in forms. The attacker can create a side channel leaking form content.

M Heiderich: That could be seen as markup-based XSS without JavaScript.

I Fette: We want to take the web forward. We want new features. Therefore there will be new

security risks.

C Hofmann: And as always, find the exploit and get the bounty. We invite you to both Mozilla and Google.

J Schuh: There's need for regexps in password fields and that's already out there, currently done in JavaScript.

M Heiderich: For HTML4 we have a lot of tools such as HTML Purifier, AntiSamy etc. But for the moving spec of HTML5 we have nothing. What should we tell the developer?

I Fette: We have the HTML5 iframe sandbox. That's how we're trying to make it easier. Developers want to include user content on their site.

M Heiderich: But how about legacy browsers. And all the partial implementations of HTML5?

J Schuh: Well, it's HTML5 we're talking about so legacy browsers don't support it. But we're all guilty of implementing partial specs.

I Fette: It's always going to be that browsers have implemented parts of a spec. HTML5 is a moving, versionless spec. We cannot wait until it's "done".

E Vela Nava: Is there a way to sanitize SVG?

M Heiderich: We're working on it.

E Vela Nava: Browsers could implement and define that SVG is SVG and not HTML.

B Sterne: I'm 90% certain SVG images are not scriptable whereas SVG files are.

E Vela Nava: Yes, and you can surf directly to the file which will make the script execute. People will link that way for sure.

B Sterne: Got ya. In general I can say we rarely do security reviews of the spec as it evolves. But we do when we start implementing.

J Schuh: Right. Things always pop out when you have the first implementation.