Browser Security Track @ OWASP Summit 2011 (17:00 - 18:20 Wednesday Feb 9)

DOM Sandboxing

E Vela Nava == Eduardo Vela Nava, Google
M Heiderich == Mario Heiderich, Uni Bochum
S Di Paola == Stefano Di Paola, Minded Security
D Lindsay == David Lindsay, Cigital
B Sterne == Brandon Sterne, Mozilla
I Fette == Ian Fette, Google
G Heyes == Gareth Heyes
J Nagra == Jasvir Nagra, Google
J Schuh == Justin Schuh, Google
J Wilander == John Wilander, OWASP
L Adamski == Lucas Adamski, Mozilla

[About this document] These are the raw notes from the Summit discussion on DOM Sandboxing. 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, HTML5 Security, and EcmaScript 5 Security – these notes will be the foundation of the forthcoming Browser Security Report.

G Heyes: We need a CSS sandbox, HTML sandbox and JS sandbox. HTML sandbox should be able to sanitize. With a sandbox in the browser it has all the info on what's been rendered.

M Heiderich: A sandbox is not only effective against injections. Every property of the DOM should be under the sandbox policy.

J Nagra: What is the level of granularity we should have?

M Heiderich: We will get separate rules for different browsers. Makes it almost impossible. Then one sandbox per browser too.

S Di Paola: If a developer uses e.g. Google Analytics he/she doesn't know what Analytics needs from the DOM/page. So Google has to provide the white list.

M Heiderich: We might need templates for configuring the sandbox based on attack cases or threat models. "XSS is OK for me but not UI redressing, I choose template X".

J Schuh: We need high granularity along with "reasonable" rule templates.

G Heyes: Content sandboxing is another problem. If CSP was expanded to cover parts of pages how will the sandbox enforce this?

E Vela Nava: iFrame sandboxes are a good idea. Setting top location etc.

J Nagra: If you have support from the language, you can sandbox by choosing the api you pass to a function. Your policy is just that - here is the api you can use. The Caja sandbox is implemented this way - first we identify the subset of JavaScript that denies access to global authority then hand out a subset of the DOM and JavaScript apis to sandboxed code. That api is the policy.

D Lindsay: How do you use the api?

J Nagra: The code doing the sandboxing basically says to sandboxed code “Here's a div you can draw into and here's the api functions you can call.” The rest of the DOM is not available. The sandboxed code itself has to do nothing special - it just calls the api functions it was provided. It might believe that the “window” object it is interacting with is the Real “window” object but that’s immaterial to it.

G Heyes: We had scoping issues with using fake window objects to provide access. I believe in using the real window object. Then prevent access to dangerous objects.

M Heiderich: Only white listing functions is not enough. You want to know what parameters are sent too and how many times a function is called. There was an CSP bypass earlier where you sent two arguments to eval() instead of one and broke out of the sandbox.

S Di Paola: In DOM sandboxes prototypes are often turned off. But a lot of external scripts use prototypes. How do we solve that?

J Nagra: It's possible to support prototypes. We have DOM, HTML, CSS, JS. In eariler versions of Caja we did prevent modifying primodial prototypes, but this is no longer necessary - we’re now able to provide virtualized versions of primodial prototypes. But we can't keep virtualizing. Are there changes needed in browsers? What changes are needed for sandboxing? How do we encourage browser vendors to include that? Strict-Mode in EcmaScript 5 solves a lot of this. Similar efforts are required on CSS and HTML5.

G Heyes: For instance, we could consider cookie-less attributes and make the browser proxy requests for images to not send/leak cookie that way.

J Wilander: So what will it take to get browsers + sandboxes == true?

B Sterne: I was hoping to understand the use cases better. Maybe Gareth could repeat for HTML5 iframe sandbox. Mozilla has plans to implement that feature. We have it on our radar. But I'm having a hard time understanding the drawbacks and what features the iframe sandbox misses.

G Heyes: iframe sandbox addresses the top location. It is a good feature as a website sandbox can be broken then top redirections can be prevented and stop a spammer from redirecting the whole site. Of course it doesn’t prevent exploits but there is no way for a site to provide a top redirection protection.

J Nagra: What authority does a sandboxed iframe start off with? There's still a lot of authority that they inherit. For example being able to sniff history and scan the LAN via `onerror` and `onload`. The first thing that is needed is to have the option to remove all potential sources of authority. Secondly, we should be able to selectively re-introduce authority to the iframe.

B Sterne: I see the Caja case. And even an iframe sandbox can do some bad stuff. Placing it in its own unique origin is one thing.

E Vela Nava: You can mix CSP with iframe sandboxes.

J Schuh: Yeah, iframe sandboxes are blunt. Selectively apply CSP to iframes would be good. That covers the majority of the use cases.

L Adamski: HTML5 sandbox may be a quicker point to build from. Quicker to get accepted.

J Nagra: And how to we reintroduce authority to a sandboxed iframe? Today the only mechanism we have is post message communication with sandboxed code. But the only interaction possible is sending asynchronous serializable messages - essentially programming with strings and callbacks. That seems tedious.

S Di Paola: Tainted data could be a solution. I implemented it in a tweaked FF. Tainted data communicated keeps its tainted status.

J Nagra: How do we handle composition? Two script libraries A & B compose to create a third library C that program D wants to use. What happens? Are A & B loaded into separate iframes? Is C a new sandboxed iframe that has A and B child iframes?

L Adamski: A good starting point is always to ask "Is it possible now?" Extend existing APIs where feasible. Always start by comparing with existing solutions.

G Heyes: If browsers had sandboxes in their engines we will be able to know if it has enforced a policy or not. As long as we're doing this in the client code we will never know. Use the engine to parse the content and inspect before rendering.

B Sterne: There's significant cost to that.

I Fette: We actually considered this. But browsers do things differently and debugging of strange behavior will be a pain for developers – “What goes wrong with event X?”

S Di Paola: I used an internal Gecko class to inject my JavaScript first in head for FF. This allows me to parse the document and inspect it before rendering.