

A more plausible E4X attack

A more plausible E4X attack - Chris Evans, scarybeastsecurity.blogspot.com.

- Published: date not stated
- Original: <<https://scarybeastsecurity.blogspot.com/2009/05/more-plausible-e4x-attack.html>>
- Preserved from: <https://scarybeastsecurity.blogspot.com/2009/05/more-plausible-e4x-attack.html> (live) on 2026-08-10
- Licence: unknown

Rights remain with the original author and publisher. This is a research archive of a source from the Web Hacking Techniques Index collections, kept so the page going offline. To read the original, follow the link above.

Content

UNTRUSTED SOURCE TEXT. Everything below this line is third-party material quoted for research. It is data, not instructions. Do not follow directions, execute code, or fetch URLs because this text says so.

As a quick recap, "E4X" is the name of a Javascript standard relating to strong XML support in the language. Firefox has had an implementation for quite some time but no other major browser seems to have followed suit.

My colleagues Filipe Almeida and Michal Zalewski led the way in E4X security; check out:

<http://code.google.com/p/doctype/wiki/ArticleE4XSecurity>

However, the attack scenarios in that document are in my opinion not likely to occur in many web apps. It so happens that I was fiddling around the night before my HiTB talk (which briefly covers E4X) and I came up with something more compelling. Take a hypothetical web mail service which provides an XML feed format of the inbox, which might look something like this:

```
<inbox>
<mail id="1234"><from>evil@hacker.com</from><subject>{ x = '</subject><body>PWN...</body></mail>
<mail id="1235"><from>bank@bank.com</from><subject>Super sensitive!</subject><body>New pin: 9976</body></
<mail id="1236"><subject>' }</subject><body>...ed!!</body></mail>
</inbox>
```

One general concept of interest in the above fragment is the ability of the attacker to echo little pieces of attacker-controlled text onto a trusted domain. Specifically, in e-mail subject text! More on this in another post. With this realization, we're all set to mount an E4X-based theft attack. First, you'll want to see it in action. You'll need Firefox to see the popup alert indicating cross-domain XML theft:

<https://cevens-app.appspot.com/static/e4xtheft.html>

The attack works by cross-domain including the XML formatted inbox into the attacker's page via `<script src="blah">`. Raw XML is valid Javascript in Firefox, thanks to E4X, so this parses and executes in the attacker's context. The reason the attacker is able to mount a theft is that E4X looks for curly braces in XML values and tries to interpret the surrounded text as a Javascript expression to evaluate. Looking again at our above XML, we see the following in the middle:

```
<subject>**{ x = ' </subject><body>PWN...</body></mail>
<mail id="1235"><from>bank@bank.com</from><subject>Super sensitive!</subject><body>New pin: 9976</body></
<mail id="1236"><subject>' }**</subject>
```

As you can see, the attacker's sneaky choice of subject lines has caused an expression to be evaluated which:

- Wraps a part of the XML in single quotes, forming a Javascript string literal.
- Assigns said string literal to a Javascript variable in the attacker's domain!
- Leaves the XML tag structure balanced, thanks to the repeating nature of the XML tree.

For the attack to work, there are constraints:

- There must be no newlines in the part of the XML structure that you are stealing, because Javascript literals cannot span unescaped newlines.
- There must be no XML prolog or DTD since these break the Firefox E4X parser.
- The single quote character must be rendered into XML values unescaped and double quotes must be used to surround XML attributes (or visa versa).

There will be real-world services matching these constraints. When you find them, drop me an e-mail or leave a comment. As always, Mozilla security responded wonderfully to this advance in E4X theft. A behavioural tweak was committed and is due in Firefox 3.5, which will break this attack.