Login/session cookies, Ajax and security

Asked 15 years, 9 months ago Modified 10 years, 3 months ago

Viewed 23k times

▲

43

▼

I'm trying to determine the most secure method for an ajax based login form to authenticate and set a client side cookie. I've seen things about XSS attacks such as this:

How do HttpOnly cookies work with AJAX requests?

and

http://www.codinghorror.com/blog/archives/001167.html

So, I guess my core questions are...

1) Is using pure ajax to set cookies secure, if so, what is the most secure method (httpOnly + SSL + encrypted values, etc.)?

2) Does a pure ajax method involve setting the cookie client side? Is this at all secure?

3) Is setting cookies this way reliable across all major browsers/OSs?

4) Would using a hidden IFrame be any more secure (calling a web page to set the cookies)?

5) If possible, does anybody have code for this (PHP is my backend)?

My goal is to set the cookies and have them available for the next call to the server without navigating away from the page.

I really want to nail down the consensus, most secure way to do this. Eventually, this code is planned to be made Open Source, so please no commercial code (or nothing that wouldn't stand up to public scrutiny)

Thanks, -Todd

ajax cookies security

Share

Improve this question

Follow

edited May 23, 2017 at 12:34

Community Bot
1 ● 1

asked Mar 19, 2009 at 9:50

supertodda
587 ● 1 ● 6 ● 11

1 Answer

Sorted by: Highest score (default)

▲

71

1. The cookie needs to be generated server-side because the session binds the client to the server, and therefore the token exchange must go from server to client at some stage. It would not really be

useful to generate the cookie client-side, because the client *is* the untrusted remote machine.

It is possible to have the cookie set during an AJAX call. To the server (and the network) an AJAX call is simply an HTTP call, and any HTTP response by the server can set a cookie. So yes, it is possible to initiate a session in response to an AJAX call, and the cookie will be stored by the client as normal.

So, you can use AJAX to do the logging in process in the same was as you could have just relied on a POST from a form on the page. The server will see them the same way, and if the server sets a cookie the browser will store it.

Basically, client-side Javascript never needs to be able to know the value of the cookie (and it is better for security if it doesn't, which can be achieved using the "httponly" cookie extension honored by recent browsers). Note that further HTTP calls from the client to the server, whether they are normal page requests or they are AJAX requests, will include that cookie automatically, even if it's marked httponly and the browser honors that extension. Your script does not need to be 'aware' of the cookie.

You mentioned using HTTPS (HTTP over SSL) - that prevents others from being able to read information in transit or impersonate the server, so it's very handy for preventing plain text transmission of the password or other important information. It can also help guard against network based attacks, though it does not make you immune to everything that CSRF can throw you, and it does not at all protect you against the likes of session fixation or XSS. So I would avoid thinking of HTTPS as a fix-all if you use it: you *still* need to be vigilant about cross-site scripting and cross-site request forgery.

2. (see 1. I sort of combined them)

3. Given that the cookie is set by the server in its HTTP response headers, yes it is reliable. However, to make it cross-browser compatible you still need to ensure logging in is possible when AJAX is unavailable. This may require implementing an alternative that is seen only when there is no Javascript or if AJAX isn't available. (*Note: now in 2014, you don't need to worry about browser support for AJAX anymore*).

4. It would not change the security. There would be no need for it, except that I have seen hidden iframes used before to 'simulate' AJAX before - ie make asyncronous calls to the server. Basically, however you do it doesn't matter, it's the server setting the cookie, and the client will accept and return the cookie whether it does it by AJAX or not.

For the most part, whether you use AJAX or not does not affect the security all that much as all the real security happens on the server side, and to the server an AJAX call is just like a non-AJAX call: not to be trusted. Therefore you'll need to be aware of issues such as session fixation and login CSRF as well as issues affecting the session as a whole like CSRF and XSS just as much as you would if you were using no AJAX. The issues don't really change when using AJAX except, except, I guess, that you may make more mistakes with a technology if you're less familiar with it or it's more complicated.

Answer updated September 2014

Share Improve this answer Follow

edited Sep 5, 2014 at 0:00

answered Mar 19, 2009 at 10:13

thomasrutter

117k ● 31 ● 152 ● 170

6 Thank you, that response was just what I was looking for - thoughtful and nothing short of awesome. I appreciate it. -Todd – supertodda Mar 19, 2009 at 11:19