

## Re: Reality Check: Session Hijacking

**Source:** <http://coding.derkeiler.com/Archive/PHP/comp.lang.php/2004-05/0459.html>

---

**From:** Daniel Tryba (*news\_comp.lang.php\_at\_canopus.nl*)

**Date:** 05/07/04

Date: Fri, 7 May 2004 12:32:49 +0000 (UTC)

mrbog <dterrors@hotmail.com> wrote:

- > *You still don't get it, and actually, I'm relieved that you*
- > *don't get it because it means you're not an asshole, it's just that*
- > *you're not getting it!*

I have the exact same impression about your persistence to tour broken plan. It's a shame you are not responding to my postings...

- > *So, with my method:*
- >
- > *1. The user is always challenged when he starts to use a secure app,*
- > *and I don't mean just the first time, I mean each and every time he*
- > *STARTS to use the secure app. Subsequent pages of the app don't*
- > *challenge him because subsequent pages are getting his auth info from*
- > *hidden fields. And NOT from the session.*

If someone can read session cookies, that someone can also read the rest of the http response, so he also has the required post data.

- > *3. If the user logs into a secure application, and then leaves it*
- > *(goes and clicks on a flat http page in another window for example),*
- > *he is exposing his session cookie.*

I'm wondering why you are still using a sessioncookie? The session could be hidden with you other authentication fields in a form.

- > *And now, your method:*
- >
- > *2. After logging in, the user wanders away from the secure app (clicks*
- > *on a flat http page in another window, for example), and therefore*
- > *exposes his session cookie over unencrypted http.*

Ehhhh, you know that the domain/path in cookies can be used to not expose the cookie in certain places?

- > *I AM NOT SAYING that you are transmitting name/pw over http. I AM*
- > *SAYING that in your method, after a user has logged in, the single*
- > *only criteria that the subsequent pages of your application require in*

comp.lang.php: Re: Reality Check: Session Hijacking

- > *order to verify that a user is who he says he is, is the session*
- > *cookie. And session cookie themselves aren't secure.*

Everything in a http request/response isn't secure. If someone is sniffing data (kind of man-in-the-middle attack) your screwed unless all data is being encrypted (with something like ssl). But the problem in most cases isn't sniffing... it's crosssite scripting bugs enabling potential harmful (java)script to send data to a 3rd party, this script isn't limited to cookies...

It's a shame you're not responding to my suggestion not to use static data in the post fields... because your solution is nothing more than disabling sessioncookies and putting the session identifier in forms (which php already has support for AFAIK).

--

Daniel Tryba