Is it possible for a XSS attack to obtain HttpOnly cookies?

Asked 16 years, 2 months ago Modified today Viewed 37k times

28

Reading this blog post about HttpOnly cookies made me start thinking, is it possible for an HttpOnly cookie to be obtained through any form of XSS? Jeff mentions that it "raises the bar considerably" but makes it sound like it doesn't completely protect against XSS.

Aside from the fact that not all browser support this feature properly, how could a hacker obtain a user's cookies if they are HttpOnly?

I can't think of any way to make an HttpOnly cookie send itself to another site or be read by script, so it seems like this is a safe security feature, but I'm always amazed at how easily some people can work around many security layers.

In the environment I work in, we use IE exclusively so other browsers aren't a concern. I'm looking specifically for other ways that this could become an issue that don't rely on browser specific flaws.

security cookies xss

Share

Improve this question

Follow

edited Oct 23, 2008 at 3:26

Gulzar Nazim

52.2k ● 26 ● 130 ● 170

asked Oct 23, 2008 at 1:07

Dan Herbert

103k ● 51 ● 192 ● 221

6 Answers

Sorted by: Highest score (default)

34

✓

> **NOTE this answer is outdated, and XST is unlikely to be exploitable in almost any modern scenario. Other parts of the answer might be relevant, but there are other more modern techniques.**

First, as some others mentioned, XSS **can** allow other payloads, not just cookie stealing.

But, *is* there anyway to steal httpOnly cookies, with XSS? (ignoring the question of httpOnly support?).... The answer is: Yes.
A subset of XSS is known as Cross-Site Tracing (XST) (or go to the original research paper). This attack has the XSS payload send an HTTP TRACE request to the web server (or proxy, forward OR reverse), which will echo back to the client the full request - INCLUDING YOUR COOKIES, httpOnly or not. The XSS payload can then parse the returned info, and retrieve those delicious cookies...

Btw, yet another "subset" (kinda) of XSS, involves injecting payload into response headers. Though similar, this isnt *exactly* XSS, and Header Injection can even lead to HTTP Response Splitting (HRS) - which is much more powerful, allows near complete control of other clients, cache poisoning, and of course access to cookies, if so wished.

Share Improve this answer Follow

edited Apr 19, 2023 at 22:49

answered Feb 8, 2009 at 1:18

AviD

13.1k ● 7 ● 64 ● 93

2 Wow this is the top answer and its completely wrong. The citation is from a white paper written in 2003. If this is the best XSS attack to obtain HTTP-only cookies, then we are all very safe. – mmla Apr 16, 2020 at 5:10

10 @mmla "wrong", or perhaps you mean "outdated"? As you can see this was written over 10 years ago, that's like a decade in internet time. Back then, this was actually correct, as it predated some of the newer mechanisms. You are welcome to add your own updated answer and I will happily refer to it, or just ask a new question altogether and answer that :-) – AviD Apr 16, 2020 at 8:50

FYI, XST seems to be an ancient APACHE vulnerability and not a general vulnerability. See pentestpartners.com/security-blog/… Please remove this answer because it's misleading. – d2vid Apr 16, 2023 at 3:56

@d2vid see comment above yours - it is outdated, not wrong or misleading. And not Apache-specific, it is a general vulnerability, and one that used to be very common and popular. Strange that PTP dont remember seeing it... IMO it should no longer be the accepted answer, not to mention there are more modern attacks nowadays, but I doubt anyone is interested in editing up a **14 year old question**.
– AviD Apr 19, 2023 at 22:47

9

Using HttpOnly cookies **will prevent** XSS attacks from getting those cookies.

Unless:

- your browser does not support HttpOnly
- there is a hitherto unknown vulnerability in the browser which breaks HttpOnly
- the server has been compromised (but then you're probably hosed anyway).

As another poster has noted: XSS is not the only threat out there, and grabbing cookies is not the only threat from XSS. I'm sure you knew this - I'm just being complete!

Good luck!

Share Improve this answer Follow

answered Oct 23, 2008 at 14:45

AJ.
13.7k ● 21 ● 53 ● 63

4 Agreed, I never understood why there is so much focus on 'stealing' cookies. Isn't it easier and safer for the attacker to just carry out the attack from the victim's own browser? E.g. They can use their malicious script to make a malicious request directly to the server (and the user's valid httpOnly cookie will still be attached). – Jon May 1, 2018 at 7:43

6

If the browser doesn't understand HttpOnly, the attack succeeds. *Edit: okay, you are not concerned. That's fine, but I will leave this notice just for reference. It is useful to state it explicitly.*

Another way of stealing besides sniffing the network would be direct control of user's computer. Then the cookies can be read from a file. If it's a session cookie, it will be of course removed after browser is closed.

By the way, stealing session cookie is not the only possible "payload" of XSS attack. For example it may make your CSRF protection useless. It may alter contents of your site to deceive the user. And many other malicious things.

So better protect yourself in a good way **(escape output),** and think about HttpOnly as **additional layer** of protection.

Share Improve this answer Follow

answered Oct 23, 2008 at 6:32

Paweł Hajdan

18.5k ● 9 ● 52 ● 65

0

HttpOnly cookie can only be stealed if the client reflects the cookie in the response at some point. You can make an XHR request to steal the cookie. Although it is not related to the HttpOnly flag, another way is if the application is using JWT for authentication/authorization, you can read it from Local Storage.

Share Improve this answer Follow

answered 6 hours ago

AZAN SHAHID
39 ● 2 ● 5

-4

Packet sniffing can read the cookies transmitted over http. But it may not fall under the XSS.

Share Improve this answer Follow

answered Oct 23, 2008 at 1:19

Ramesh
13.3k ● 3 ● 54 ● 88

Packet sniffing could, but in my case I'm using an HTTPS connection with a digital certificate, which makes it somewhat harder to sniff packets. – Dan Herbert Oct 23, 2008 at 1:34

-4

JavaScript can modify the HTML on the page, therefore, httpOnly **does *not*** mean you are safe against XSS.

Share Improve this answer

answered May 27, 2010 at 17:47

Follow

Longpoke
12.6k ● 4 ● 52 ● 55

I understand that XSS can exploit other attack vectors. I was specifically asking about cookies because I wasn't aware that there were ways to obtain httpOnly cookies through JavaScript. – Dan Herbert May 27, 2010 at 17:59