

Forum:[Hack Forums](#)

Thread: [How Session Hijacking Works | Cookie Theft & 2FA Bypass.](#)

Wake - May 16, 2026, 6:13 pm

? loading encrypted image...

How Session Hijacking Works | Cookie Theft & 2FA Bypass

? loading encrypted image...

Written by [Wake](#)

? loading encrypted image...

Disclaimer

This thread is for informational and educational purposes only. I will not be responsible for any action performed by any reader.

Introduction

In this tutorial, I'll explain how session hijacking works, including techniques such as cookie theft, and how attackers can exploit authenticated sessions to bypass login protections like suspicious activity, social media authentication and two-factor authentication (2FA) (can bypass authentication mechanisms if a valid session is already active).

I'll cover how these attacks occur, why they pose a serious security risk, real-world example, and most importantly, how you can protect yourself against them.

? loading encrypted image...

What are cookies?

Cookies are small pieces of data stored in your browser. They are automatically sent with requests to websites to help maintain state between requests.

Not all cookies are related to authentication some are used for preferences or tracking. However, in most web applications, cookies are used to store session tokens.

Session tokens are what identify your logged-in session and tell the server that a request is coming from an authenticated user. These are the real targets for attackers.

[+] You login to a website.

[+] The website creates a session token.

[+] That token is stored In a cookie In your browser.

[+] Your browser sends the cookie with the token inside every request.

What Is session hijacking

Session hijacking is a type of attack where an attacker takes over a user's authenticated session to

gain unauthorized access to their account, without needing their login credentials.

When you log in to a website, you are given a session token that is stored in a cookie. This allows you to stay logged in without having to re-enter your credentials on every page.

The danger of this is

- [+] No password needed

- [+] Can bypass login systems (and 2FA)

- [+] Gives immediate access to an active session

When an attacker gains access to a valid session token they are able to interact with the website like they are the legitimate user because the server trusts the session, this allows the attacker to perform actions on behalf of the victim.

What an attacker may be able to do

- [+] Access account data (profiles, messages, emails)

- [+] Perform actions as the user (send messages, make changes)

- [+] Attempt further account compromise (e.g. triggering password resets)

The level of access will depend on the websites security. Most modern services require additional verification for sensitive actions like changing passwords or account details which can limit the impact of session hijacking.

How do these attacks happen?

Real world ways this type of attack can happen are

- [+] Malware/malicious software can infect a device and extract browser data, including cookies and session tokens.

- [+] If someone gains access to an unlocked device, they may be able to access active sessions or extract stored cookies.

- [+] Social engineering attackers can trick users into revealing sensitive information or performing actions that expose their session.

- [+] Phishing fake websites or login pages can be used to capture credentials or session data from unsuspecting users.

- [+] Malicious or compromised browser extensions some extensions can access browser data, including cookies, and may expose session information if they are unsafe.

I won't be covering any here but, there are plenty of malware that are built for this exact reason or you can do your homework on your target and create a custom attack your self, there is no one way solves all.

For physical access something like USB Rubber ducky works well or a homemade one most can be made to be FUD (fully undetectable) or to disable Windows defender/Anti virus before running and re enabling It when Its finished extracting the session tokens.

This type of attack has become increasingly more relevant back In the day It used to be about getting passwords but now with modern account security suspicious login detection, SMS or email verification and two-factor authentication (2FA) this has become increasingly more challenging meaning that If an attacker can steal your session cookies or token they can bypass all these protections and access the account as If they where you sometimes without ever even knowing the password.

? loading encrypted image... Real-world example

Linus Tech Tips Is one of the biggest tech focused YouTube channels In the world. It Is run by Linus Sebastian a creator known for making videos about computers and gadgets, In March 2023 they became victim to this attack an employee received what they thought was a completely legitimate sponsor offer In this offer was Malware disguised as a PDF.

This Malware grabbed session tokens which the attackers imported into there session and were instantly logged in as Linus Tech Tips, once inside the attackers renamed the channel and uploaded crypto scams trying to trick users into sending crypto.

This didn't just affect one of Linus Tech Tips YouTube channels It affected 3 Linus Tech Tips, TechLinked and Techquickie all being taken over at the same time during this attack all bypassing 2FA.

Because multiple accounts where accessible from the same machine, a single infected device led to the compromise of multiple high-value channels at once. By the time the breach was noticed, the attackers already had full control and were actively exploiting the accounts.

Examples of this attack

For this example we will be using GitHub I have created 2 throwaway accounts Wake-admin is the victim account logged in on Firefox through Google social sign in with 2fa enabled.

Wake-user is our account we will be using for this takeover.

Spoiler [\(Click to View\)](#)? loading encrypted image...

The first way we will do this is if on Firefox we open inspect element and head over to Storage (right click and inspect) Application if you are on Chrome. Once inside storage find Cookies and the website you are looking for. For this example we are using GitHub the cookies we are looking for in this example are "saved_user_sessions" and "user_session" but each website will have its own name so you may have to search but for GitHub these are all we need.

Spoiler [\(Click to View\)](#)? loading encrypted image...

Go ahead and copy both "saved_user_sessions" and "user_session" now we will move over to our Chrome browser and open inspect and move over to Application. Once inside Application find Cookies and GitHub again now double click on "saved_user_sessions" and "user_session" and replace them with the ones we copied earlier, refresh the page and you will be logged into the victims account bypassing 2fa and social sign in.

Spoiler [\(Click to View\)](#)? loading encrypted image...

The next example I will be using a browser extension called Cookie-Editor that can be found [here](#) this will be playing the role of our Malware or Rubber Ducky for this attack.

I have visited YouTube and logged in on my Chrome browser now I will use Cookie-Editor to steal the cookies like Malware or a Rubber Ducky would, on the Cookie-Editor menu select "export"

Spoiler ([Click to View](#))? loading encrypted image...

Now I will move over to my Firefox browser and import these Cookies, after importing the Cookies and refreshing the page I am able to skip logging in completely and bypassing all suspicious login alerts and MFA (Multi-Factor Authentication)

Spoiler ([Click to View](#))? loading encrypted image...

How can I protect my self from this?

A large number of session hijacking attacks begin with malware or unauthorized access to a device.

[+] Keep your operating system and browser updated.

[+] Avoid downloading untrusted files or software.

[+] Lock your device when unattended.

[+] Do not run unknown browser extensions or scripts.

[+] Remove extensions you no longer use/review extension permissions regularly.

Clear sessions after important activity logging out of accounts can invalidate session tokens on many websites, reducing the chance of an old session being reused.

This is especially important on

[+] Shared devices.

[+] Workstations.

[+] Public computers.

While two-factor authentication (2FA) is still extremely important it is not designed to protect against every type of attack. If an attacker steals a valid authenticated session the website may already trust that session as logged in.

Because of this protecting the device and browser itself is just as important as protecting your password.

Google's solution to session cookie theft is Device Bound Session Credentials (DBSC) released on April 9, 2026.

It works by locking your login session to your physical device. If malware extracts your session cookies the stolen cookies become completely useless on an attackers machine. Coverage is limited to Chrome on Windows but hopefully in the future more browsers and websites opt in.

[Thank you for taking the time to read this thread. If you enjoyed this please check out the rest of my tutorials and write-ups](#)

Mastercarder - May 16, 2026, 11:35 pm

Very informative and valuable information in here. Good stuff.

Pr0m3th3us - May 17, 2026, 12:15 am

Great write up. Definitely bookmarking this one.

DeathPenalty666 - May 17, 2026, 1:00 am

Damn dude did you hijack my session??? Hahaha awesome post and it almost make me freakout because i was learning different kind of payload and method the last week. They can get your cookies with xss like they can keylog or else!!

Go take a look at this repo and try to not be scared after :) [LOOK AT ME HERE](#)

Wake - May 17, 2026, 3:33 am

(May 16, 2026 11:35 PM)Mastercarder Wrote: [\[CITE\]](#) Very informative and valuable information in here. Good stuff.

(May 17, 2026 12:15 AM)Rust-0l3um Wrote: [\[CITE\]](#) Great write up. Definitely bookmarking this one.

(May 17, 2026 01:00 AM)DeathPenalty666 Wrote: [\[CITE\]](#) Damn dude did you hijack my session??? Hahaha awesome post and it almost make me freakout because i was learning different kind of payload and method the last week. They can get your cookies with xss like they can keylog or else!!

Go take a look at this repo and try to not be scared after :) [LOOK AT ME HERE](#)

Thank you all for reading and your kind words.

Sift - September 5, 2026, 8:50 am

Very well written and informative for new users.

Wake - September 5, 2026, 1:58 pm

(Sep 5, 2026 08:50 AM)Sift Wrote: [\[CITE\]](#) Very well written and informative for new users.

I appreciate the read, thank you.

?Border?ineExtr(A) - September 5, 2026, 2:06 pm

Is this similar to intercepting unprotected traffic over Wi-Fi, a technique commonly known as MiTM?

Also, AFAIK cookies are being deprecated across all major browsers.

Wake - September 5, 2026, 2:16 pm

(Sep 5, 2026 02:06 PM)?Border?ineExtr(A) Wrote: [\[CITE\]](#) Is this similar to intercepting unprotected traffic over Wi-Fi, a technique commonly known as MiTM?

Also, AFAIK cookies are being deprecated across all major browsers.

Yeah Device Bound Session Credentials will make session hijacking a lot harder.

Lufisar - September 20, 2026, 8:09 am

This is very useful information!

Wake - September 20, 2026, 1:18 pm

(Sep 20, 2026 08:09 AM)Lufisar Wrote: [\[CITE\]](#) This is very useful information!

I appreciate the read, thank you.

whitefox1013 - October 9, 2026, 7:53 am

The Chrome uses Device Bound Session which uses the device-unique token. The token is unique for each computer device hardware and can't be extracted remotely. And it's almost impossible to steal the session tokens from the cookies because the cookies is encrypted by this device-unique token.

yisuswrt - *October 9, 2026, 8:15 am*

That's really well done, thanks bro.

Wake - *October 9, 2026, 11:50 am*

(Oct 9, 2026 07:53 AM)whitefox1013 Wrote: [\[CITE\]](#) The Chrome uses Device Bound Session which uses the device-unique token. The token is unique for each computer device hardware and can't be extracted remotely. And it's almost impossible to steal the session tokens from the cookies because the cookies is encrypted by this device-unique token.

I covered this a little towards the end of the thread but I should have gone into more detail on it really. Thanks for reading.