ATTACKS THAT RESULT IN SESSION THEFT

Contents

Type of attacks

XSS

Brute Force

Backend Data Breach

CSRF

Session Fixation

Software Bugs / Poor Session Flow

JWT Signing Key Stolen

MITM

Malware

Device Access / Social Engineering

Internal Threats

Subdomain Takeover

Rogue Browser Extensions

01 XSS

Procedure

1) User is manipulated into copying JS into the browser console/address bar or

2) lack of input parsing results in actual XSS vulnerability on the website

3) A JS dependency for the website has malicious javascript code that eventually runs in the browser / hybrid apps

4) Downloading a malicious browser extension

Method of prevention

1) For websites using only “httpOnly” cookies to store session tokens.

2) For hybrid mobile apps like react native, the only way to safeguard is to have token theft detection.

Ease of prevention	Do alternative libraries safe-guard against this?	SuperTokens Safeguards against this?	Number of users affected	Occurence of attack
Easy for websites, Hard for mobile apps	Yes for websites, No for mobile apps	Yes	One to all	High

Examples of successful attacks

Microsoft Outlook, Evernote, Wordpress, and using Tesla’s in car infotainment system.

We analysed session management for HDFC bank and found that they

stored the session tokens in the site's HTML, opening it up for theft via XSS.

Sources

1. https://threatpost.com/microsoft-outlook-android-bug-xss/150528/

2. https://nakedsecurity.sophos.com/2018/11/07/serious-xss-flaw-discovered-in-evernote-for-windows-update-now/

3. https://securityboulevard.com/2019/04/wordpress-xss-vulnerability-can-result-in-remote-code-execution-rce/

4. https://www.bankinfosecurity.com/blogs/how-big-rock-revealed-10k-tesla-xss-vulnerability-p-2772"

02 Brute Force

Procedure

An attacker may ‘guess’ various combinations of session IDs until one of them work - thereby hijacking that user’s session/account

Method of prevention

Using session tokens that are long and have high entropy

Ease of prevention	Do alternative libraries safe-guard against this?	SuperTokens Safeguards against this?	Number of users affected	Occurence of attack
Easy	Yes	Yes	One to all	High

Examples of successful attacks

Gitlab’s session tokens were short and non changing making them a perfect candidate for brute force attack.

In general, we have also seen startups use sequential session tokens making them very easy to guess.

Sources

https://threatpost.com/session-hijacking-bug-exposed-gitlab-users-private-tokens/127747/

03 Backend Data Breach

Procedure

Database breach can happen in multiple ways and expose the auth tokens stored in plain text

Method of prevention

Storing only the hased versions of session tokens would mean that even in this event, an attacker cannot use these tokens to hijack any user's account.

Ease of prevention	Do alternative libraries safeguard against this?	SuperTokens Safeguards against this?	Number of users affected	Occurence of attack
Medium	Some of them	Yes	All	High

Examples of successful attacks

Twitter's access tokens exposed from Adobe's hack link for thousands of database hacks"

Sources

1. https://www.youtube.com/watch?v=UM_E-zpTysA
2. https://vigilante.pw/

04 CSRF

Procedure

A malicious site could send a “POST” API request to the target site with malicious intent to mutate the victim’s data. The browser would send the target’s site authenticated session tokens along with the request.

Method of prevention

Use anti-csrf tokens with “same-site” flag (Though same-site is not completely supported by all browsers at the moment).

Ease of prevention	Do alternative libraries safe-guard against this?	SuperTokens Safeguards against this?	Number of users affected	Occurence of attack
Easy	None- but often implemented as a seperate solution	Yes	One	Medium

Examples of successful attacks

Facebook suffered from a CSRF vulnerability that would allow an attacker to “post to the hijacked user’s timeline, change their profile picture, and even trick them into deleting their account.”

Sources

https://nakedsecurity.sophos.com/2019/02/19/facebook-flaw-could-have-allowed-an-attacker-to-hijack-accounts/

05 Session Fixation

Procedure

The attack consists of inducing a user to authenticate herself with a known session ID, and then hijacking the user-validated session by the knowledge of the used session ID

Method of prevention

An application should always change session tokens after user authentication and ideally revoke the older ones

Ease of prevention	Do alternative libraries safeguard against this?	SuperTokens Safeguards against this?	Number of users affected	Occurence of attack
Easy	Yes	Yes	One	Low

06 Software Bugs / Poor Session Flow

Procedure

These are relatively random vulnerabilities which do not fit any one specific method of attack

It could be either incomplete QA or the lack of development time / specialised security knowledge

Method of prevention

Method of prevention would depend on the nature of the vulnerability.

Ease of prevention	Do alternative libraries safeguard against this?	SuperTokens Safeguards against this?	Number of users affected	Occurence of attack
Easy - Medium	Yes	Yes 	All	High

Examples of successful attacks

A Facebook feature had a software bug which exposed 90M user session tokens. Gitlab exposing session tokens via URL + countless startups

Sources

1. https://about.fb.com/news/2018/09/security-update/

2. https://threatpost.com/session-hijacking-bug-exposed-gitlab-users-private-tokens/127747/"

07 JWT Signing Key Stolen

Procedure

Secrets can be leaked in many ways - from database breach, to insider threats, to developer mistakes.

The result of this is that an attacker can essentially “become” any user in the system, potentially making this an extremly serious threat.

Method of prevention

Regular rotation of signing keys, while immediately invalidating the older ones. Ideally without logging out current users

Ease of prevention	Do alternative libraries safe-guard against this?	SuperTokens Safeguards against this?	Number of users affected	Occurence of attack
Hard	None	Yes 	All	Medium

08 MITM

Procedure

1) The website doesn't enforce https and doesn't use secure cookies.

2) MITM attack can be performed, even with https, under a corporate network that monitors all traffic via a proxy setup.

This can happen by monitoring the requests going through the proxy, which is trusted as a certificate authority by all devices within the network.

Method of prevention

1) Using HTTPS with "secure" flag cookies.

2) Certificate pinnig for mobile apps - however, those apps will fail to work in the "trusted" proxy setup.

3) For websites, the only way to mitiagte this attack is to have session token theft detection.

Ease of prevention	Do alternative libraries safeguard against this?	SuperTokens Safeguards against this?	Number of users affected	Occurence of attack
Easy for mobile app. Hard for website	None	Yes	One	Low

Examples of successful attacks

“Several corporates we know of that require devices to connect and approve the proxy as a CA

ESPN was not using https

Indian unicorn with $300M+ had no secure flags for cookies”

Sources

Personally experienced

09 Malware

Procedure

There are multiple ways in which a user can be “infected” by malware on their device that could “steal” the victim’s session tokens.

Method of prevention

One can avoid getting infected by being careful but the only foolproof measure against this is to have session token theft detection

Ease of prevention	Do alternative libraries safe-guard against this?	SuperTokens Safeguards against this?	Number of users affected	Occurence of attack
Hard	None	Yes	One	High

Examples of successful attacks

Youtube influencers accounts compromised via session cookie theft

Sources

https://twitter.com/MarcoStyleNL/status/1192179230341251075?s=09

10 Device Access / Social Engineering

Procedure

Many ways to get hold of a victim's device. Once the device is obtained, if it's a browser based application, the attacker can simply inspect the page and read the session values.

Method of prevention

The only way to mitiagte this attack is to have session token theft detection.

Ease of prevention	Do alternative libraries safeguard against this?	SuperTokens Safeguards against this?	Number of users affected	Occurence of attack
Hard	None	Yes	One	High

Examples of successful attacks

One can physically do this for any service

Sources

N/A. Attackers who exploit this would probably not get caught or covered in a news article

11 Internal Threats

Procedure

Internal employees can misuse JWT keys to create user sessions or view session tokens from databases / logs

Method of prevention

1) Strict internal access control. However, that is generally difficult to guarantee.

2) Storing only hashed versions of session tokens. However, logs may still be an attack vector.

3) Finally, implementing session token theft detection.

Ease of prevention	Do alternative libraries safe-guard against this?	SuperTokens Safeguards against this?	Number of users affected	Occurence of attack
Hard	None	Yes	One to all	Medium

12 Subdomain Takeover

Procedure

Cookies sent to a root level domain may also be sent to a subdomain that is in control of an attacker

Method of prevention

Given that a subdomain has been taken over, one should have token theft detection to detect theft.

Ease of prevention	Do alternative libraries safe-guard against this?	SuperTokens Safeguards against this?	Number of users affected	Occurence of attack
Hard	None	Yes	One to all	Low

Examples of successful attacks

Uber had a sub domain takeover which resulted in session token theft for millions of its users

Sources

https://www.zdnet.com/article/uber-patches-security-flaw-leading-to-subdomain-takeover/

13 Rogue Browser Extensions

Procedure

Browser extensions can access a lot of information in any page, including httpOnly, secure cookies. Which makes it very easy for them to steal tokens

Method of prevention

Token theft detection

Ease of prevention	Do alternative libraries safeguard against this?	SuperTokens Safeguards against this?	Number of users affected	Occurence of attack
Hard	None	Yes 	One to all	Low

Note

Please note - by definition it is difficult to find well sourced public examples of attacks that having single victims. Attacks such as Social engineering, CSRF, Malware, MITM would be difficult to detect and expose through white hat security.

The example of malware was only found cause we coincidentally followed the Youtuber at the time of this account being hacked

The probability of session theft occuring is the sum probability of each individiual attack - which is non trivial