2022

Investigation report: Broken Authentication

10-01-2022

STIJN VAN EEKELEN (459346)

Inhoud

Introduction

I'm going to do this research rapport about broken authentication because it's a issue that is really big in the web development world and mismanagement of this can lead to really big data breaches within an organization.

I'm going to first explain the difference between a hacker and the lesser know term cracker. Many people think by a malicious person who attempts to break the system for their own benefit is a hacker but this is wrong. A hacker is a good person who's job it's to hack a system for knowledge purposes for example in order to test vulnerabilities. A cracker is a person who will attempt to break into a system for their own benefit. Meaning the cracker is the bad person not the Hacker that's why I will be using in this Research rapport the term cracker instead of hacker.

What is broken authentication?

Broken authentication is a umbrella term for several vulnerabilities that attackers exploit in order to impersonate a user. Broken authentication mainly refers to two areas session management and credential management.

Broken authentication is one of the most important security vulnerabilities and has caused many of the most major data breaches both in the past and in the present. Because of this security experts have alerted the OWASP since 2017 with this problem and it is put in the OWASP their top 10 security vulnerability list, on top of that in 2020 broken authentication reached the top 2 spot in the list of vulnerabilities. This indicates that the problems are increasing and the importance of this issue is increasing.

Below I will give an explanation what exactly is a session and how it's used.

What are sessions?

A web session is a sequence of network transactions associated to the same user within a period of time. Let's say you go to a social media website and browse for a while before logging in to your account. You find an interesting post where you want to comment, which requires you to log in. Once you leave a few comments, you log out and close the web browser window or tab. Everything you did from the moment you arrived to the website was a session. Web applications can track sessions before and after authentication.

Web applications issue every user a unique session ID for each visit, which allows the web app to communicate with the user as they move through the site. These session IDs commonly take the form of cookies and URL parameters.

What can happen when you’ve got broken authentication?

Session management attacks?

1. Session Hijacking

Without appropriate safeguards web applications are vulnerable to session hijacking. Attackers use in this case a stolen session ID to impersonate the users identities. A simple example of this is a user who forgets to log out of an application a cracker can then continue on their session. The application will be thinking it’s dealing with the user who walked away allowing the cracker to do the same thing as the user.

2. Session ID URL Rewriting

Another common scenario for session hijacking is “URL rewriting”. In this scenario an individual’s session ID appears inside the URL of a website. Anyone who can see the session ID via for example an insecure Wi-Fi connection can piggyback into the ses sion.

3. Session Fixation

One commonly overlooked best practice is to rotate session IDs after a user logs in, instead the user will be given the same id each time they login this can make it vulnerability to session fixation attack which is a variation of session hijacking.

The main idea behind this is that an attacker can send a URL to the victim with a predetermined session ID, then the user will login using this link to use the predetermined session ID to impersonate the victim. Developers can prevent the attack by granting a new session ID after login, making the predetermined session ID useless.

Credential management attacks?

In the recent years attackers have figured out that the easiest way to access off limit data is to use another users credentials. The main methods used for this are phishing and stolen credentials which now belongs to the two most common methods to execute a data breach.

1. Credential Stuffing

When attackers access a database filled with unencrypted emails and passwords they will frequently sell or give away this list of combinations to other attackers. These other attackers then use botnets or brute-force attacks that test these stolen credentials from one different site on a different account. This works because many users tend to use the same password.

Currently there are billions of compromised credentials available to attackers.

2. Password Spraying

Password spraying is a little like credential stuffing. It makes use of a set of weak passwords and instead of using those passwords for a single user password to password it does most of the time a password from user to user.

3. Phishing Attacks

Attackers typically phish by sending users email or phone calls pretending to be from at trusted source. They are trying to trick users in giving them their account information this is mostly done in broad daylight in comparison with the more hidden methods. Most of the time they will attempt to hit a group of users belonging to a specific organization.

Following the report of CrowdStrike 35% of successful network breaches were started with a spear phishing attack in 2019. 19% of these methods used attachments and 15% included a malicious link, and 1% employed spear phishing using a service.

How to prevent broken authentication?

Update session management

1. Control session length

Every web application automatically ends session at some point, either after logout, a period of non-activity or after a specific expiration time. By determining your session length depending to the type of user and the application they're using. For example a video streaming site might want their active for weeks long but a bank site just a few minutes.

2. Rotate and invalidate session IDs

As previously mentioned in the explanation about phishing attacks the best way to prevent session fixation is to rotate a session ID meaning when one session ends the token should no longer be valid.

3. Don't put session IDs in URLs

Just don't store session IDs in URL because there are too many ways to expose them so the safest option you can use is to store you're session in a cookie generated in a secure session manager.

Tighten password policies

1. Implement multi-factor authentication (MFA)

Passwords are way too vulnerable to credential stuffing and password spraying, it's clear that they are no longer adequate to secure an account. MFA provides an extra level of security demanding an additional credential that's harder for attackers to fake, such as a biometric scan or one-time code.

2. Don't permit weak passwords

One important thing to force on your users is to prevent weak passwords this will make the risk of data breach significantly less likely.

3. Don't store passwords in cleartext

When storing passwords don't do it in cleartext. Use a hashing algorithm for this to make them unrecognisable by using a salt on top of the hash you can guarantee that each user has a unique combination.

4. Use breached password protection

Obtain breached password connections forcing you're users to get logged out until they change their passwords so attackers can't use their compromised accounts.

Guard against attacks

1. Conduct workplace phishing training

Conduct workspace training for phishing in order to prevent it within the company from happening.

2. Implement brute-force protection

Protect against brute forcing by for example limiting amount of requests by Ip-address.

3. Employ anomaly detection

By detecting anomality's you can alert yourself when suspicious behaviour is happening to your system.

Conclusion

My conclusion is that there are a lot of risks that can happen with broken authentication. There are also many ways to prevent it especially on the session management side of it. But the credential management side of it is a lot more difficult because it's more user related than application related and extensive training is needed in order to prevent and inform the users of the risks. I'm positive that with the steps I found within this research about this subject a lot can be prevented but it takes a lot of management to perform this. For this reason I've chosen Auth0 for my project which is a all in one package for authentication, registration and authorization.

Bibliografie

Hdiv. (2021). *What is and how to prevent Broken Authentication | OWASP Top 10 2017 (A2)*. Opgehaald van hdivsecurity: https://hdivsecurity.com/owasp-broken-authentication

OWASP. (2017). *OWASP Top Ten 2017 | A2:2017-Broken Authentication | OWASP Foundation*. Opgehaald van https://owasp.org: https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication

OWASP. (2021). *A07 Identification and Authentication Failures - OWASP Top 10:2021*. Opgehaald van https://owasp.org: https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/

Poza, D. (2020, august 20). *What is broken authentication?* Opgehaald van auth0: https://auth0.com/blog/what-is-broken-authentication/