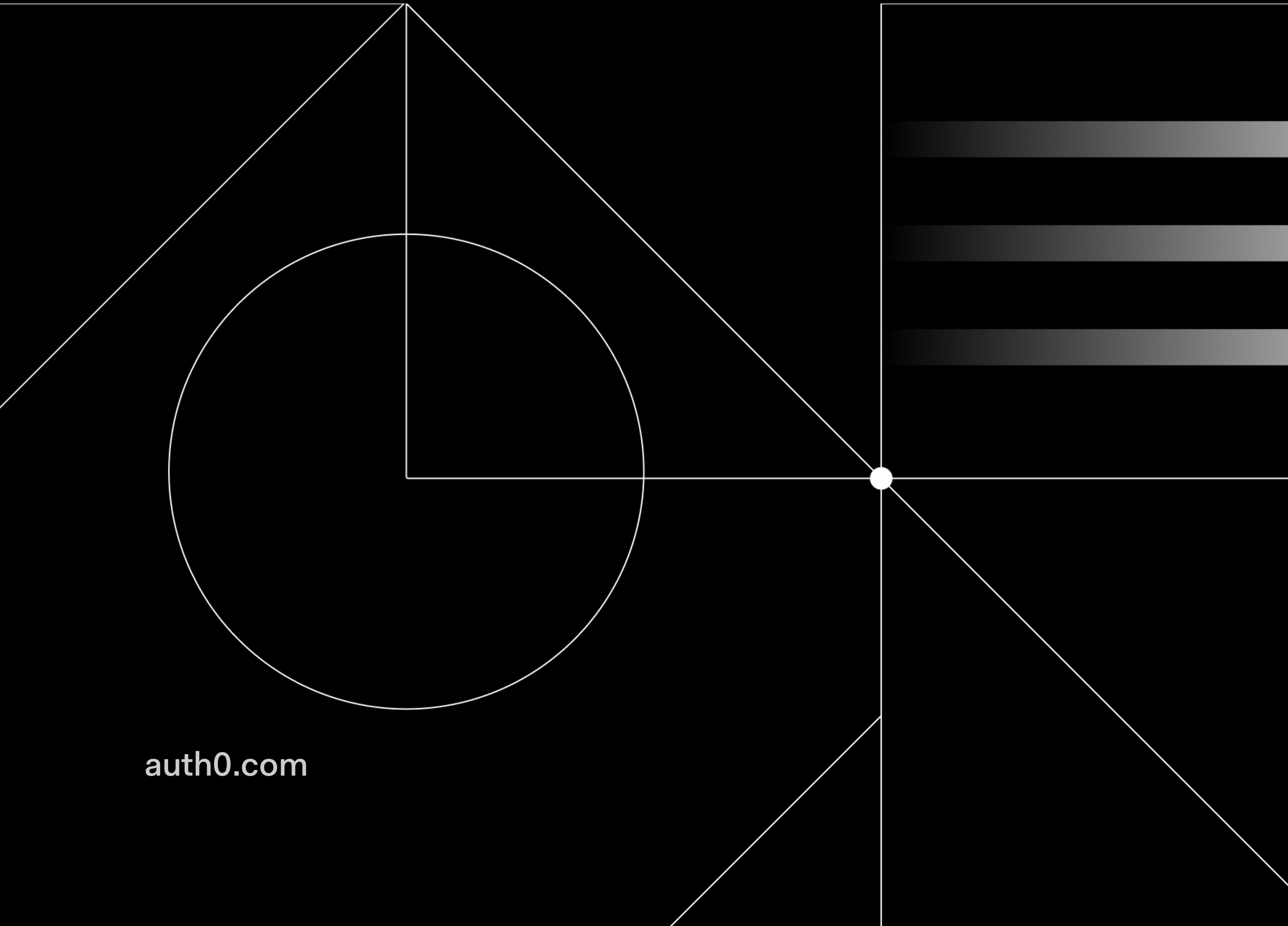
Auth0

Fixing Broken Authentication

Addressing one of the most critical application security risks



auth0.com

Authentication is a cornerstone capability of any application. Ensuring a user is who they say they are is crucial to maintaining data privacy and preventing fraud and data breaches. Consequently, improperly implemented authentication, known as broken authentication, is a potentially devastating application vulnerability. In fact, the Open Web Application Security Project (OWASP) lists broken authentication as the second most critical security risk to web applications.

This whitepaper will provide an overview of broken authentication: why it's so dangerous, the types of threats that can take advantage of this vulnerability, and how Auth0 prevents broken authentication.

What is broken authentication?

Broken authentication is an umbrella term for a variety of vulnerabilities within an authentication system that attackers can exploit to impersonate legitimate users. These vulnerabilities lie primarily with credential management and session management.

Session Management

A session describes all the ways users interact with applications during a set period. Let's say you go to the New York Times website and browse for a while before logging in to your account, doing a crossword puzzle, and then logging out and closing the tab. Everything you did from the moment you arrived was a session.

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

Session management concerns how you define the parameters of that session. For instance, how long can a session last before you automatically log a user out? How do you issue and revoke session IDs? How securely are they linked to a user's IP address?

Credential Management

User credentials are the most sensitive information an authentication system handles. For this reason, they are often a high-priority target for attackers. Certain attacks, such as brute force or credential stuffing, specifically target the login process and aim to guess a person's password through automated trial and error. Credential management encompasses everything from how credentials are stored – using proper encryption, salting, and hashing – to how credentials are transmitted and processed by the system.

What makes broken authentication so dangerous?

Authentication is often the first line of defense against attacks. Not only that, but authentication is often the most significant line of defense, since once a threat has bypassed authentication, they are usually considered to be a legitimate user by the application. There are several aspects of broken authentication that when combined, make it a high-risk vulnerability.

Exploitability

As long as computer passwords have existed, broken authentication has existed. According to Wired, the first computer password might have been used by MIT's Compatible Time-Sharing System (CTSS) in the 1960s. CTSS pioneered many common computer features we see today, including messaging, file sharing, and possibly password-protected access. In 1962, a researcher managed to print off the CTSS master password list and used it to circumvent time limits. This is the earliest documented case of password theft.

Since then, attackers have perfected the art of compromising authentication systems. They have access to billions of breached user names and credentials, default administrative account lists, and automated tools for logging in. Session management attacks are also well established in hacker circles.

Prevalence

According to OWASP, "the prevalence of broken authentication is widespread due to the design and implementation of most identity and access controls." Authentication is a common need in most applications, easy to implement poorly, and commonly exposed to the internet. Developers often try to write their own authentication implementations, which is hard to get right and often leads to vulnerabilities. For attackers, it is easy to detect applications with broken authentication, which they can then exploit on a large scale with automated tools and password lists.

High Impact

Once an attacker is able to log into a user account, most systems consider them a legitimate user. This means they are granted access to most materials without additional security challenges. Once a few user accounts or a single administrator account is compromised, the attacker can obtain wide-reaching access. Depending on what type of application is compromised, this can lead to fraud, identity theft, data theft, or fraudulent money transfers.

What types of attacks take advantage of broken authentication?

Just as there are many types of vulnerabilities associated with broken authentication, there are also many attacks that take advantage of those vulnerabilities. Here are just a few examples.

Credential Stuffing

Credential stuffing attacks are a type of cyberattack where stolen account credentials from a data breach are used to gain unauthorized access to user accounts on another website. This is done through large-scale bot-driven attacks against the login flow.

More than 80 percent of companies state it is difficult to detect, fix, or remediate credential stuffing attacks, and these attacks result in an average of more than $6 million a year in costs per company. At Auth0, credential stuffing attacks account for as much as 67% of all login requests.

Session Hijacking

These attacks occur when an attacker takes over a user session after authentication but before logout. When you login to a web application, the server places a temporary cookie in your browser to prove that you are authenticated. In session hijacking, attackers steal the session ID by stealing the cooking or getting the user to click a malicious link with a prepared session ID. There are simpler session hijacks as well, such as an attacker taking over a public device when a user forgets to log out.

Password Spraying

Password spraying is a subset of brute force attacks similar to credential stuffing, but instead of working from a database of stolen passwords, password spraying uses a set of weak or common passwords to break into a user's account. (A 2019 study found that 23.2 million accounts used "123456" as their password, while millions more used sports names, curse words, and the ever-popular "password.")

Password spraying is a type of brute-force attack, but it often slips by automatic lockouts that block IP addresses after too many failed login attempts. It does this by trying the same password, one user at a time, rather than trying password after password on a single user.

Session ID URL Rewriting

This attack occurs when the session ID appears in the URL of a website. If an attacker can see it, such as through an unsecure Wi-Fi connection, they can rewrite the URL on their own device to piggyback into the session.

Session Fixation

When a web application fails to rotate session IDs after authentication, instead of retaining the same session ID pre- and post-authentication, it leaves users vulnerable to session fixation attacks. In this variation on session hijacking, an attacker takes a legitimate session ID and then tricks a victim into logging in with it. Once the victim has logged in, the attacker copies the ID to impersonate the victim. These attacks would be impossible if the application generated a new session ID after the user authenticated.

How to prevent broken authentication

Use Multifactor Authentication

One of the most effective ways to prevent brute force, credential stuffing, and password spraying attacks is multifactor authentication (MFA). In order to compromise an MFA-protected account, attackers would need access to a set of breached credentials and the device used for the second factor. MFA drastically increases the time and effort needed for the attacker to compromise the account, which makes such attacks infeasible at scale. Where possible, always offer MFA to your users and use it for any administrative accounts.

Do Not Ship with Default Credentials

Default administrative credentials are a major attack vector for a variety of products and applications. Many users do not bother to change default passwords, which makes it trivial for attackers to compromise the account with a basic dictionary attack, which uses a dictionary of keywords to systematically guess passwords.

Enforce Strong Passwords

Many brute force attacks rely on weak or common passwords. Enforce password length, complexity, and rotation based on NIST recommendations or other evidence-based policies. In addition, check passwords against lists of common passwords.

Don't Store Plain-Text Passwords

If your password database is illegible, it's of no value to hackers. Encryption makes your organization a much less appealing target, and it ensures that if a breach does happen, the compromised data won't be used against users in a future credential-stuffing attack. One caveat is that it is essential to stay up-to-date on the latest hashing and encryption protocols for this method to work.

Use Breached Password Protection

Use an identity and access management (IAM) platform with breached password protection. When a compromised credential's cache is discovered, you will be notified if any of your users were compromised. Those users will be locked out until they change their passwords, so their compromised credentials can't be used against you in a credential-stuffing attack.

Use the same message for all failed authentications

Many threats leverage user enumeration attacks to identify a list of users to target. These attacks rely on systems to provide different messages when someone tries to log in with a nonexistent user account or a legitimate username with an invalid password. This gives attackers a complete list of all users in a system, which allows them to conduct more targeted brute force or phishing attacks.

To prevent this, ensure your authentication system displays identical messages for all failed authentication outcomes.

Limit Failed Logins

Brute force, credential stuffing, and password spraying attacks are large-scale techniques that result in hundreds or thousands of failed logins for every successful one. IP addresses responsible for a large number of failed logins should be blocked automatically. In addition, accounts that experience a large number of failed logins should be locked and forced into a password reset flow. Lastly, administrators and security professionals should be alerted to any signs of a credential stuffing or brute force attack.

Use Secure Session Management

Use a server-side, secure session manager that generates a new session ID after login. In addition, do not put session IDs in the URL and ensure they are securely stored and invalidated after logout.

Auth0 makes it easy to prevent broken authentication

Auth0 takes the headache out of preventing broken authentication. It provides secure login flows, session management, and credential management. You can also bring your own identity provider if you want to maintain control of your user credentials. Auth0 also has a strong security program that employs best practices to prevent broken authentication and similar vulnerabilities.

Capabilities such as enforcing strong password policies and defining session length are easily configurable in the Auth0 user interface without the need for code. We also offer several capabilities designed specifically to tackle threats facing authentication systems.

Bot Detection

Bot detection is designed to combat credential stuffing and other forms of bot-driven attacks. It works by correlating a variety of internal and external data sources to identify and mitigate bot-driven attacks before login.

At a high level, bot detection monitors IP addresses for non-suspicious events, such as successful logins; suspicious events, such as numerous failed login attempts across multiple accounts; and IP reputation data, which is used to identify known threat actors. A confidence score is then generated to identify suspicious actors, and when one is found, they are required to solve a CAPTCHA to send a login request. This mitigates the majority of bot attacks targeting the login or registration flow.

Breached Password Detection

This capability identifies users who are logging in with credentials that are known to have been breached and leaked to the public. Auth0 keeps a large, constantly growing database of username-password pairs that are known to be compromised in data breaches.

Auth0 customers can choose to run all logins against this database to determine when users are logging in with compromised credentials. When users are detected using compromised credentials, a number of actions can be performed. An admin can be informed while still allowing the login, the user can be prompted for MFA, or the user can be blocked until they perform a password reset.

Guardian MFA

The key with MFA is to make it as easy as possible to enroll and use in order to promote adoption among your user base. Security doesn't have to mean a poor user experience.

Auth0 provides a variety of easy-to-use, friction-free MFA options. You can choose the MFA option that is right for your users – SMS, one-time password, third-party authenticators such as Google Authenticator, and more. Or choose Guardian, Auth0's proprietary MFA app, and let your users authenticate with the tap of a button.

Guardian facilitates MFA via push notification, enabling users to approve or deny login requests without ever opening the app. In addition to delivering a better user experience, this approach is more secure when compared to SMS-based MFA, which is vulnerable to SIM swapping attacks. Guardian also works with Apple Watch or Android Wear.

The Guardian Mobile SDKs – available for iOS and Android – allow you to build your own white-label MFA app. This can be used to create a custom branded MFA app or embed MFA capability into an existing mobile app. If your mobile app already has a large installed base, you can deploy MFA functionality to all users in an update, which means they won't have to download a new app to enroll.

Conclusion

Broken authentication is one of the most critical and widespread web application security risks. User credentials are considered among the most valuable data an organization has, and poorly implemented authentication places it all at risk.

There are several things to consider when you are designing your approach to authentication:

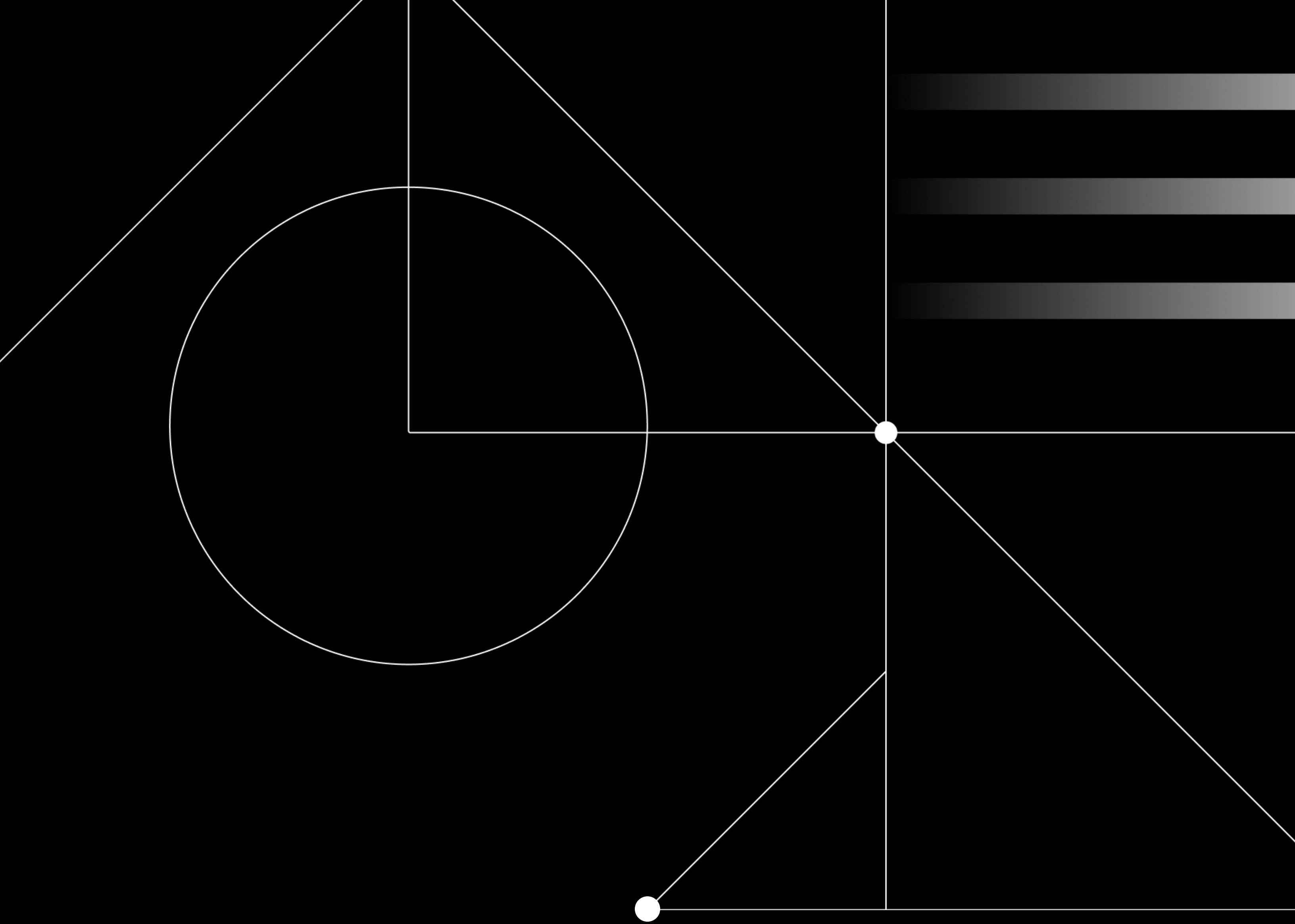
1. Do you have the in-house expertise to implement authentication securely?

2. Do you have the team in place to maintain authentication code, identify and fix vulnerabilities, and respond to attacks?

3. When you want to add new authentication capabilities in the future, such as social login or a new MFA factor, do you have the expertise and development resources to do it securely?

If you answered "no" to any of the above questions, you should consider using an identity-as-a-service (IDaaS) provider such as Auth0. We have staffed our company with some of the best identity and security experts in the field in order to provide an easy-to-implement, secure, and robust solution to our customers.

To learn more about Auth0, **reach out to our team**.



About Auth0

Auth0 provides a platform to authenticate, authorize, and secure access for applications, devices, and users. Security and development teams rely on Auth0's simplicity, extensibility, and expertise to make identity work for everyone. Safeguarding more than 4.5 billion login transactions each month, Auth0 secures identities so innovators can innovate, and empowers global enterprises to deliver trusted, superior digital experiences to their customers around the world.

For more information, visit www.auth0.com or follow @auth0 on Twitter.