auth0

The State of Secure Identity

Protecting Identity and Access Management Services Against Online Threats

REPORT

auth0.com

Contents

Foreword

Traditionally, the information security domain has examined identity through the lens of the corporate enterprise. In this context, user lifecycle management is about onboarding and managing the identity of employees who need access to corporate systems. For consumer-facing businesses, customer identity and access management (CIAM) is just as critical.

And, this is why I'm so excited to share this report with the community. Auth0 has a unique position in the CIAM space, handling billions of logins each month for consumer-facing businesses around the world, which gives us the visibility to quantitatively explore the state of secure identity in 2021.

One of the key takeaways for developers and security professionals is that managing CIAM is messy, not only because your applications are likely to be exposed to large-scale internet attacks, but also because of the ins and outs of managing customers' identities. Consumers are a varied group and automatically distinguishing between a confused user and an advanced attacker is not straightforward.

Securing your customers' identities is made more difficult by the industry-wide failure to protect data. The prevalence of breached passwords and the availability of automated attack tools makes the humble password a protective measure from the past.

We're also in a time of transition where traditional enterprises are starting to look more like a set of consumer-facing applications, which means enterprises don't have the luxury of ignoring CIAM's security problems. Consequently, identity should be top of mind for CISOs — pragmatism and limited budgets require prioritization, and securing identity should be number one.

At Auth0 we obsess about making identity easy for application builders and our Security and Product teams obsess about keeping those identities secure. I'm very excited to pull back the curtain on what we encounter every day.

—DUNCAN GODFREY, VP SECURİTY ENGİNEERİNG

Executive Summary

Within the overarching domain of identity and access management (IAM), customer identity and access management (CIAM) focuses on managing the identities of customers who need access to websites, e-commerce services, and other online applications.

With changes in digital working models and consumer habits, application providers are facing new threats when it comes to managing these digital identities.

In a recent webinar, **Top Trends and Themes Shaping the IAM Market**, we explored five major benefits of a comprehensive digital IAM strategy:

- Improved business agility
- Enhanced customer experience
- Risk mitigation/reduction
- Cost reduction via process automation
- Generate new revenue streams

In this report, Auth0 shares insights from our observations and analysis to increase awareness of threats and provide strategies for mitigation. What we've seen shows:

- **Fraudulent registrations are an expensive danger**. Rates vary by industry vertical, but roughly 15% of all attempts to register a new account can be attributed to bots. Much more than a mere nuisance, puppet accounts controlled by threat actors are a costly problem that negatively impacts applications and their users, and contributes to larger problems like money laundering.
- **Credential stuffing is a high-volume threat**. A large credential-stuffing attack can account for more than 90% of login attempts within a particular vertical on a given day. In the aggregate, credential stuffing accounts for 16.5% of login traffic on the Auth0 platform, with daily peaks reaching higher than 40%.
- **Large-scale use of breached credentials is a major risk**. For example, this report reveals an attack from February 2021 in which 72% of the credentials came from the 2019 Zynga breach — despite the stolen credentials being salted and hashed, they are now being used in the wild.

Identity services with an agile, secure-by-design, defense-in-depth approach can dissuade threat actors by disrupting the economics of attacks. In the context of IAM, a layered approach means employing defensive measures before and throughout the authentication workflow:

- **Recognizing the telltale signs of scripted, bot-enabled attacks** — like a high volume of requests coming from a limited set of IP addresses — and then rate-limiting their attempts, or challenging them with a CAPTCHA, creates friction for attackers while leaving genuine users unaffected.
- **Comparing user passwords against lists of breached credentials** allows application providers to warn users that they are at risk. Paired with an effective and user-friendly password-reset functionality, breached password detection is a valuable addition to a strong defensive posture.
- **Integrating with social identities** is another user-friendly way to boost security. Ensuring safe session management practices is a strong defense against attacks that target session IDs.
- **Encouraging multi-factor authentication (MFA) use increases identity security** by requiring additional factors for authentication, but also introduces friction — application providers can limit friction while preserving security by employing **Step-up Authentication**, **Adaptive MFA**, and **WebAuthn-enabled biometric methods**.

These capabilities are valuable assets in the fight against data breaches, account takeover, credential stuffing, identity theft, privacy abuses, and other risks. The challenge for application builders is to develop and correctly implement security measures that strike an appropriate balance between increasing friction for attackers while respecting and preserving a positive user experience.

Methodology

This report is based on data from Auth0 customers, retrieved by running simple and anonymous queries against our aggregate database. In many cases, we segmented the data by industry vertical, as self-defined by each customer. Unless otherwise noted, this report presents and analyzes data from the first 90 days of 2021.

Introduction to Identity Security

Digital identities control access to an ever-growing number of applications and services. Eventually, digital identity will impact — and perhaps even govern — all aspects of modern living, making authentication and authorization vital to preserving trust and security.

> **A digital identity is the set of attributes that define a particular user in the context of a function that is delivered by a particular application.**

The New Paradigm: Identity is Trust

For all the attention given to "zero trust," the reality is that identity is trust. Securing identity has taken on even more importance in recent years because as security perimeters dissolve, attackers are focusing efforts on identity — with important consequences for identity and access management (IAM).

IAM services are cornerstones of our connected world, ensuring that only authorized users — employees, contractors, partners, customers — can access particular resources. Conceptually, IAM is very simple: a user proves their identity and is permitted access to a resource to which they are entitled. In practice, however, several factors introduce complexity:

- Today's digital world includes many users
- There are countless ways to express a digital identity
- An individual user may have many digital identities
- There's an ever-growing list of client devices and applications
- Different digital identities have different rights and authorizations with respect to resources

To enable the scope, scale, and convenience of online activity we enjoy today, industry consortiums created open standards to manage authentication, user management, and authorization. These protocol specifications:

- Define canonical roles apps and identity providers play
- Rigorously define how different roles talk to each other to achieve common identity-related tasks

- Allow apps to talk to providers (and each other) without knowing implementation details
- Prevent lock-in with particular vendors and service providers

Crucially, these standards account for:

- Sourcing users from different origins
- Managing accounts, attributes, etc.
- Working with different app types, languages, and resource types
- Customizing the authentication experience and workflow

Identity and Access Management in Action

Federated Identity

Federated Identity Management is a method of transferring authentication data without violating the same origin policy, generally by using an external authorization server.

Single Sign-On (SSO)

SSO is a type of Federated Identity Management. SSO occurs when a user logs into one client and is then signed into other clients automatically, regardless of differences in platform, technology, or domain.

Enterprise Federation

Enterprise Federation is Federated Identity Management with enterprise connections such as Active Directory, LDAP, ADFS, SAML, Google Apps, etc.

Locking the Front Door

When something goes wrong with identity, it has the potential to go catastrophically wrong — which means securing identity is critical both to maintaining a strong cybersecurity posture and to preserving an application provider's reputation.

In fact, the Open Web Application Security Project (OWASP) lists broken authentication as the second-most critical security risk to web applications (with the closely related broken access control placing fifth), citing the exploitability of authentication systems, the prevalence of authentication, and the high impact of compromises.[1]

There are a number of ways in which a user can demonstrate that they are who they claim to be. Broadly, these mechanisms can be grouped into three categories:

- **Something a user knows**, like a password or other shared secret
- **Something a user has**, like a specific device
- **Something a user is**, for instance a biometric quality

An IAM system can challenge a user to produce one or more of these proofs before allowing access to a resource. In general, the more challenges passed, the more trustworthy an identity — but creating a great identity system is about more than hardening security.

Focusing on CIAM, the ease with which customers can change application providers means that application builders must be aware of a tradeoff between maximizing security and minimizing user friction. In practice, the appropriate balance varies based upon the user, the use case, and the potential consequences of a breach of trust.

Even experienced IAM professionals are often unfamiliar with the details of threats that leverage customer identities or target the services that manage them; consequently, developers inadvertently deploy CIAM systems that are vulnerable to attacks.

With this report, we aim to increase awareness of these customer identity-related threats and what can be done to safeguard against them.

1. See the OWASP's **Top 10 Application Security Risks** [OWASP]; you can also learn more in **What Is Broken Authentication?** [Auth0]

Identity-as-a-Service (IDaaS)

Implementing identity and access management (IAM) from scratch can prove challenging, leading to lengthy projects that consume valuable development resources, and that may result in vulnerable implementations. These issues led to the development of Identity-as-a-Service (IDaaS) solutions that act as universal translators between applications and identity providers.

IDaaS allows developers to secure applications without needing to become security or identity experts, enabling a range of use cases (e.g., business-to-consumer, business-to-business, business-to-employee) by using standard identity protocols to connect applications — written in any language or stack — with external identity providers and integrations that are needed.

Threats Against Customer Identity

Attacks that leverage or target CIAM services come in many forms, from small-scale, highly manual efforts to large-scale brute-force methods. Ultimately, threat actors want to gain access to an account (and its rights, privileges, and information) for direct use or resale.

There are two main ways that threat actors can pursue their goals, and identity threats take the form of techniques to achieve either of these outcomes:

- Fraudulent Registration involves a threat actor creating puppet accounts.
- Account Takeover (ATO) is when an attacker gains access to an existing user's account.

Account takeover is a particularly dangerous threat to application providers. In the consumer space, compromised accounts can provide attackers with valuable demographic and personally identifiable information (PII), plus access to resources (e.g., loyalty points) and privileges (e.g., ability to make purchases, especially of products in limited supply). In the corporate world, even a single compromised account can be used

as a vector to gain Initial Access,[2] as part of an impersonation operation, or to assist with intrusion activities. In both environments, a user who is victimized by ATO will likely feel violated and lose trust in the service.

In accordance with economic principles, the more lucrative the potential return, the more time and effort a threat actor is willing to invest. This expense/reward relationship has implications for defensive strategies, so it's important to understand the motivation behind each type of attack.

Let's now explore some of the most common threats against identity services, and against the applications and users who rely upon them.

We'll start the examination with the first thing attackers and users encounter: the registration/login box.

Breaches: an Expensive and Widespread Problem

A recent survey of more than 500 security decision makers, conducted by the Identity Defined Security Alliance (IDSA), found that 79% of organizations have experienced an identity-related security breach in the last two years.[3]

The impact of such incidents is not trivial: the Ponemon Institute warns that, *"A data breach can have far-reaching consequences, causing financial losses and affecting an organization's operations and compliance in the short term. And a major breach in the headlines can potentially damage reputation for years to come, leading to lost business and a competitive disadvantage."*[4]

The *Institute's Cost of a Data Breach Report 2020* calculates the global average cost at $3.86M USD, with breaches in the United States averaging more than double that amount, at $8.64M. The report also

2. See **Initial Access** in the MITRE ATT&CK framework.
3. See **Identity Security: A Work in Progress** [IDS Alliance].
4. See **https://www.ibm.com/security/data-breach** [IBM]

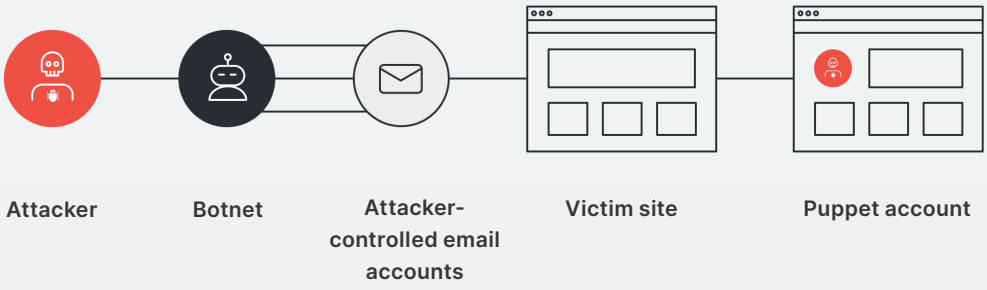
highlights that costs vary by industry, citing healthcare as the most expensive ($7.13M, on average).

Given the potential rewards for threat actors, the rapid rise in remote working, and the well-documented migration to cloud-based apps, we expect this challenge to increase.

Fraudulent Registrations

In a fraudulent registration attack, also known as a fake account creation attack, a threat actor abuses the account registration process to create puppet accounts.

Figure 1: Anatomy of a fraudulent registration attack



There are a number motivations for doing so, including:

- **Harming the application provider's ability to deliver services** by exhausting the namespace of potential users, and thereby preventing legitimate users from registering
- **Gaining disproportionate access to something valuable** (e.g., limited edition sneaker drops, new video game consoles in short supply, etc.)
- **Receiving awards or incentives that are associated with account creation** (e.g., gift cards, cryptocurrency tokens, etc.)
- **Spamming, disinformation, or hacktivism campaigns** (e.g., by leveraging accounts to participate in comment threads or amplify messages)

- **Creating a large number of accounts to resell or to use directly** (see *Example: Attackers target an online marketplace*)

The attacker may use only a relatively small number of puppet accounts or could employ a botnet to automate the creation of thousands or even millions. In the latter case, the operation may be aided by lists of common usernames.

The impact to the application provider varies, but may include:

- Loss of legitimate users and the associated benefits
- Reputational damage
- Direct financial loss
- Operationally expensive clean-up (and the opportunity cost of doing so)

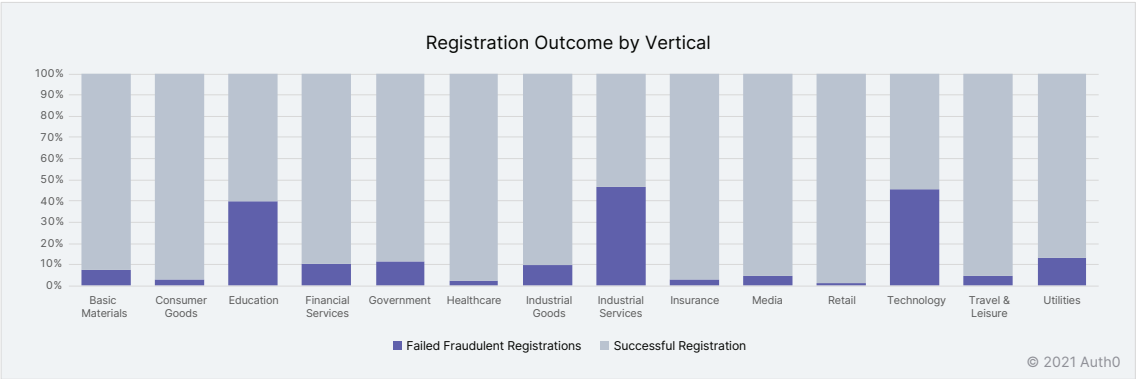
Aggregate Observations

Figure 2 shows both that fraudulent registrations are an ever-present threat across all industries and that there is considerable variation by vertical.

While there are legitimate reasons why a genuine user might experience a signup failure, automated scripts exhibit behavior that is fairly distinct. For example, to contribute to the Failed Fraudulent Registrations in Figure 2, the IP associated with the signup must have experienced more than ten failures on that day — a fairly conservative threshold that is unlikely to be crossed by a genuine user.

Industrial Services, Technology, and Education have particularly high rates of fraudulent registration attempts, around 40% and higher, with a considerable gap between those three and Utilities (~13%).

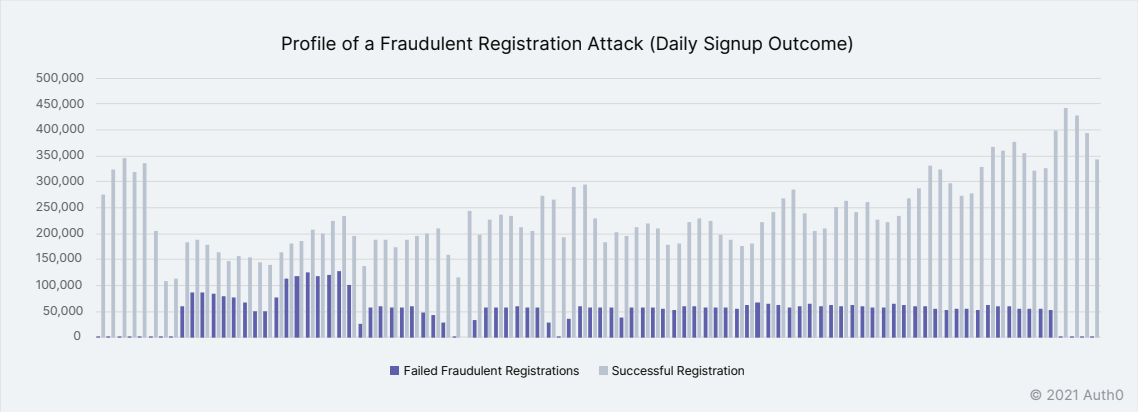
Figure 2: Fraudulent registration threatens application providers in every industry



Example: Attackers Target an Online Marketplace

Figure 3 shows a fraudulent registration attack against a global reseller marketplace; the attack began in December 2020 and carried over into 2021, before the threat actor abandoned the attack in mid-March.

Figure 3: At its peak, this attack generated more than 125,000 failed signups each day



On a typical non-attack day, there are 100 to 200 failed signups and roughly 250,000 successful signups. During the attack, the successful signup rate remained largely unchanged (as it should), but the number of failed signups due to fraudulent registration soared: one day there were 105 failed signups and the next there were nearly 60,000.

For about two weeks, the threat actor configured their automation infrastructure, before settling into a robotically consistent steady state for a little over two months — during which time the attack averaged greater than 50,000 fraudulent registration attempts each day.

During the testing phase, the attack peaked with more than 125,000 failed signups representing almost 40% of registration attempts that day; during the steady state period, the attack was responsible for roughly 18% of signup traffic. One corollary of this observation is that the vertical-wide averages in Figure 3 are heavily influenced by scripted fraudulent registration attacks.

Further investigation discovered that the failed signups all originated from IP addresses in Russia. While intent is impossible to determine with the limited information available, online marketplaces are a known mechanism for money laundering, so this motivation is a distinct possibility.

Fighting Fraudulent Accounts

Application builders can tailor the level of authentication friction to the potential rewards of account creation by employing a number of techniques, including:

- **Using rate limiting (throttling) to counter brute force attacks by imposing restrictions on the rate at which a particular client can access the login interface:** When a client exceeds a prescribed threshold, they may be required to complete a CAPTCHA, or may be restricted from accessing the login interface until a 'cooling off' or 'penalty' period has passed
- **Applying pre-signup rules and actions to further reduce the chances that a new user is illegitimate:** Email reputation scoring is a common approach, and some applications only allow users from paid email services to register

A sudden surge in failed signups is a strong indicator that your application is under attack. In this situation, you may wish to take a closer look into the registration traffic to see if thresholds or rules should be modified.

Credential Stuffing

MITRE's ATT&CK Framework explains that, "Adversaries may use credentials obtained from breach dumps of unrelated accounts to gain access to target accounts through credential overlap."[5]

Broadly, the primary motivation for credential stuffing is account discovery/validation, the goal of which is to develop a high-quality list of credentials that can be sold (e.g., to sell streaming accounts at a lower price than the subscription rate).

Credential stuffing attacks take advantage of the entirely too-common practice of password reuse. When a user reuses the same (or similar) passwords on multiple sites, it creates a domino effect in which a single credential pair can be used to breach multiple

5. See **Brute Force: Credential Stuffing** [MITRE]

The impact of credential stuffing attacks is significant. When the Ponemon Institute investigated the subject in depth, they determined that the average cost per impacted organization was $6 million; this figure incorporated the expenses from a number of consequences, including (in order of cost):[6]

1. Application downtime
2. Costs to remediate
3. Lower customer satisfaction
4. Loss due to fraud
5. Customer churn
6. Damaged brand equity

applications. Unfortunately, research suggests that attackers have plenty of fuel to power the credential stuffing engine:[7]

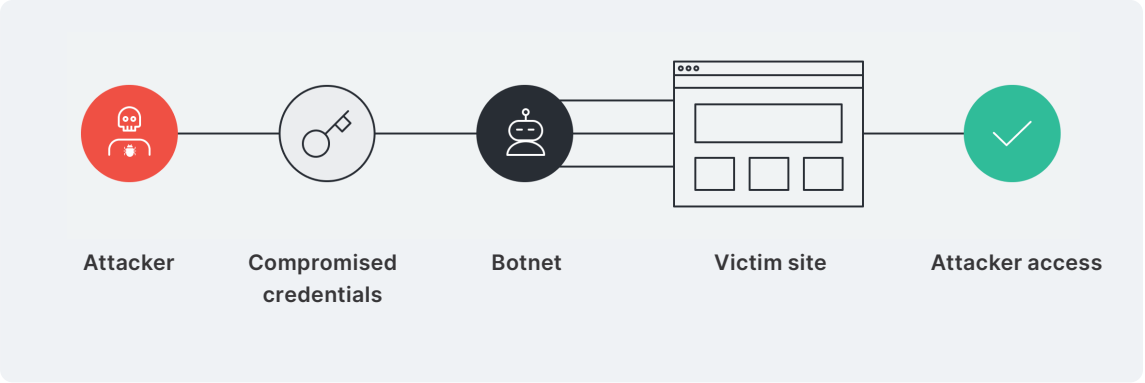
- 73% of online accounts use duplicated passwords
- More than half (54%) use five or fewer passwords across their entire online life — and 22% use just three or fewer
- Almost half (47%) of consumers rely on a password that hasn't been changed for five years

Most credential stuffing attacks use brute force to process long lists of breached credentials (Figure 4).[8] Unfortunately, the barrier to launching such attacks is very low:

- Aggregated lists like Collections #1-5 are readily available[9]
- Renting a botnet is easy and cheap
- Rotating IP services are plentiful
- Automating the components into an attack is straightforward

6. See **The Cost of Credential Stuffing** [cio.com]
7.See **Telesign Consumer Account Security Report** [TeleSign]
8. Credential stuffing attacks aren't alone in using breached credentials —Verizon's **2020 Data Breach Investigations Report** [Verizon] states that, "*Hacking and even breaches in general (at least in our dataset) are driven by credential theft.*"
9. See **Hackers Are Passing Around a Megaleak of 2.2 Billion Records** [Wired]

Figure 4: Anatomy of a credential stuffing attack



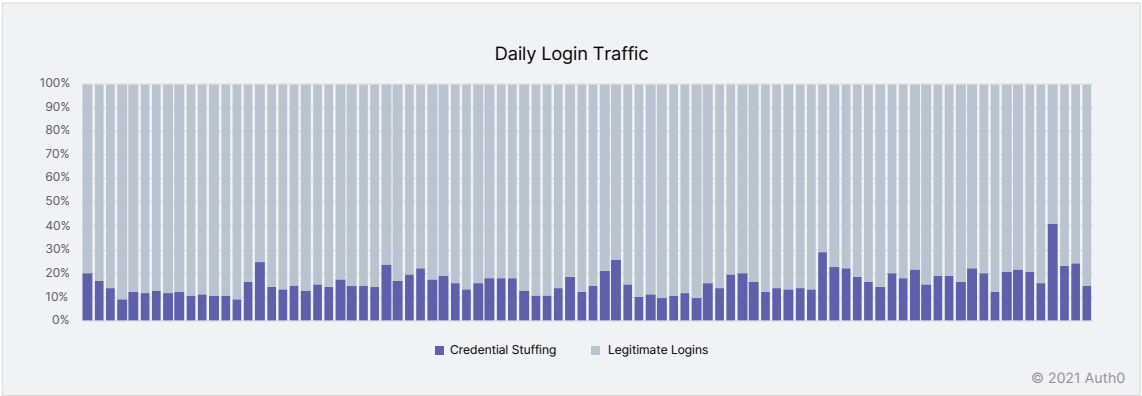
Threat actors employ a number of tactics when conducting credential stuffing attacks:

- **Bursting:** Attempting anywhere from a few dozen up to hundreds of credentials in a short period
- **Trickling:** Operating at a much lower rate, on the order of only a few attempts a minute
- **Sprinkling:** Occasionally interspersing known valid credentials into the stream to try to throw off automated detections

Aggregate Observations

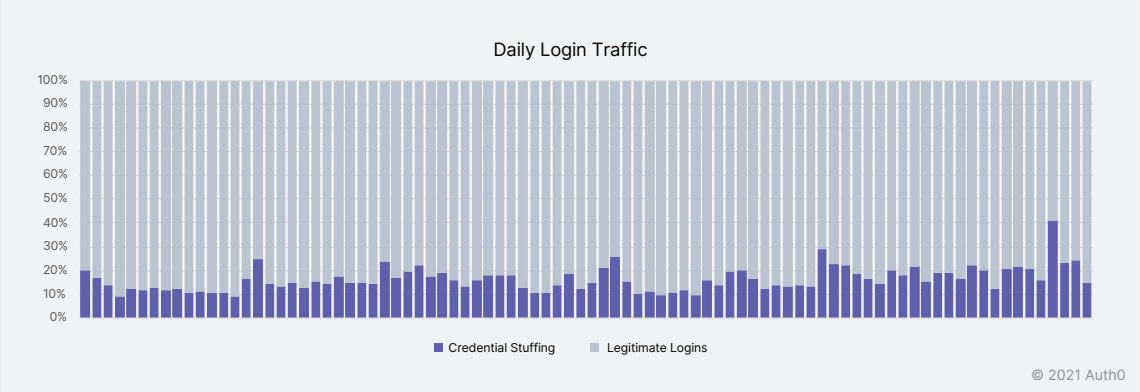
Credential stuffing attacks are the most common threats directly observed by Auth0. In the first 90 days of 2021, credential stuffing accounted for 16.5% of attempted login

Figure 5: Credential stuffing generally accounts for 10% to 20% of login traffic on the Auth0 platform



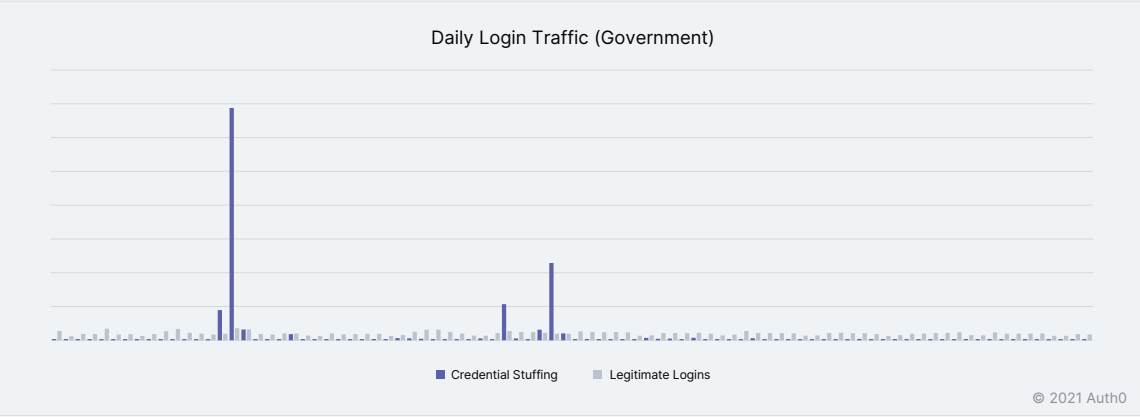
traffic on our platform, with a peak of just over 40% near the end of March (Figure 5), although the general pervasiveness clearly varies by vertical (Figure 6).[10]

Figure 6: The general pervasiveness of credential stuffing attacks varies considerably by vertical



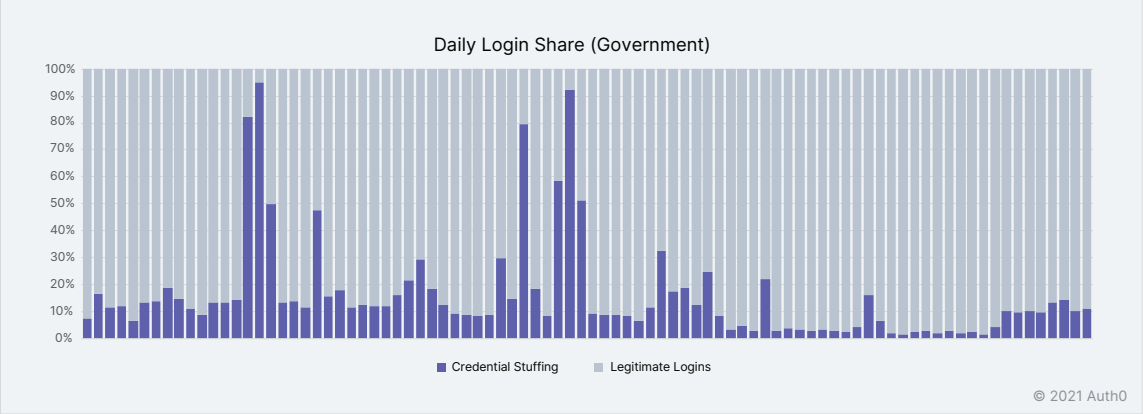
Particularly large attacks can exert enormous influence on the averages. For example, consider Figure 7 and Figure 8, which show the rate and share of login traffic in the Government verticals, respectively. This 90-day period captures a handful of large credential stuffing attacks — including two in which attacks made up more than 90% of login attempts — which is why credential stuffing accounts for more than 44% of Government login attempts in Figure 6. An average taken over just the latter 45 days would be in the 5% to 10% range.

Figure 7: 90-day view of the daily rate of credential stuffing and legitimate logins in the Government vertical



10. It should be noted that in this analysis we have applied a fairly conservative credential stuffing detection threshold, so these values should be considered as minimum real-world rates

Figure 8: 90-day view of the relative share of credential stuffing and legitimate logins in the Government vertical



For the Financial Services vertical (Figure 9 and Figure 10), the 90-day window captures several attacks. In one, credential stuffing exceeded 70% of login attempts on back-to-back days.

Notably, the ebb and flow of legitimate login traffic in Figure 9 clearly shows weekdays and weekends. Credential stuffing attacks seem to follow the same pattern — perhaps suggesting either that threat actors are employed on a similar workweek or that their botnet contains work computers.

Figure 9: 90-day view of the daily rate of credential stuffing and legitimate logins in the Financial Services vertical

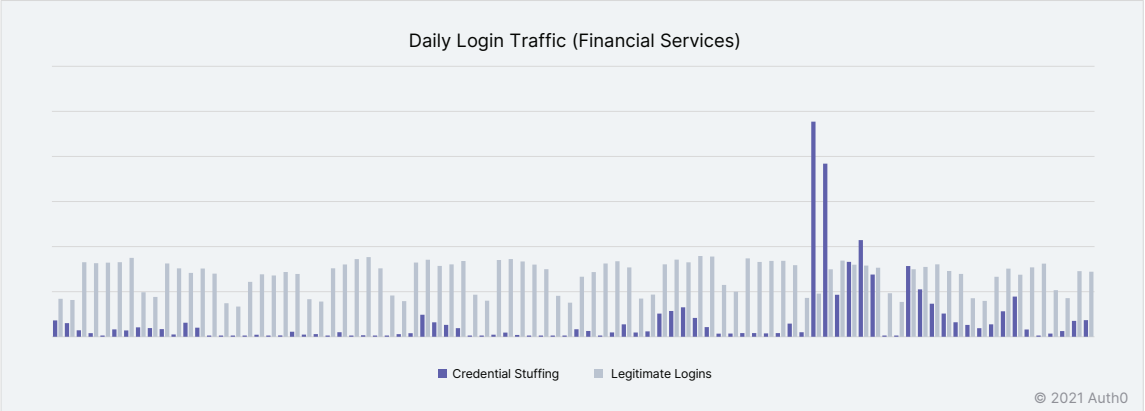
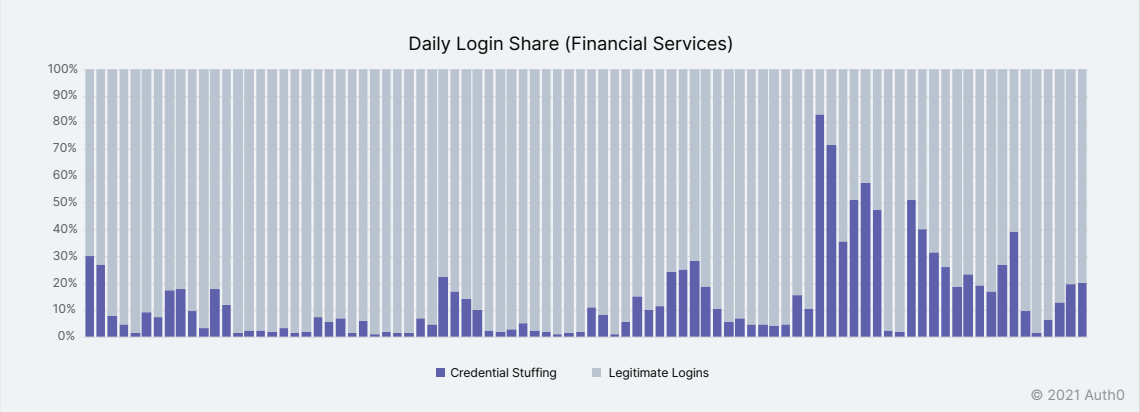


Figure 10: 90-day view of the relative share of credential stuffing and legitimate logins in the Financial Services vertical



Spotting Bots

Allowing a threat actor to enter credentials into a login interface runs the risk of providing valuable intelligence — especially if you use more than a single generic error message.

By correlating a variety of data sources, it's possible to create friction for scripted attacks like credential stuffing and password spraying by detecting when a request is likely to be coming from a bot.

For example, a bot detection algorithm can incorporate past events associated with an IP address, recent login history, IP reputation data, and other factors to generate a confidence score; based upon this score, you can show the login screen or first challenge the visitor to complete a CAPTCHA.

In Auth0's direct experience, such an initial defensive layer can reduce the success rate of a credential stuffing attack by as much as 85%.

Blocking IPs

While multi-factor authentication is the most effective way to prevent account takeover, another way to create friction for threat actors is to detect the telltale signs of an account-focused attack and then implement countermeasures. For example:

- A user experiences 10 consecutive login failures from a single *IP address*; or
- A user experiences 10 consecutive login failures from any *IP addresses*

When a condition is triggered, the system can respond by notifying the affected user, blocking the offending IP addresses for this user account, and notifying the administrator. The blocks should remain in place until the affected user changes their password or confirms that the activity was their own, or an administrator intervenes.

Multi-factor Authentication (MFA) Bypass

Application builders (and many users) understand that multi-factor authentication (MFA) is an effective way to increase identity security — it's especially effective at preventing account takeovers, whether from a credential stuffing attack or from some other attack vector.

Overcoming MFA drastically increases the time and effort needed for the attacker to compromise the account, which makes it infeasible to do at scale. In fact, Microsoft suggests that MFA, properly implemented, is effective at blocking 99.9% of account hacks.[11]

11. See **Microsoft: Using multi-factor authentication blocks 99.9% of account hacks** [ZDNet]

To compromise an account protected by a strong MFA implementation, attackers would need:[12]

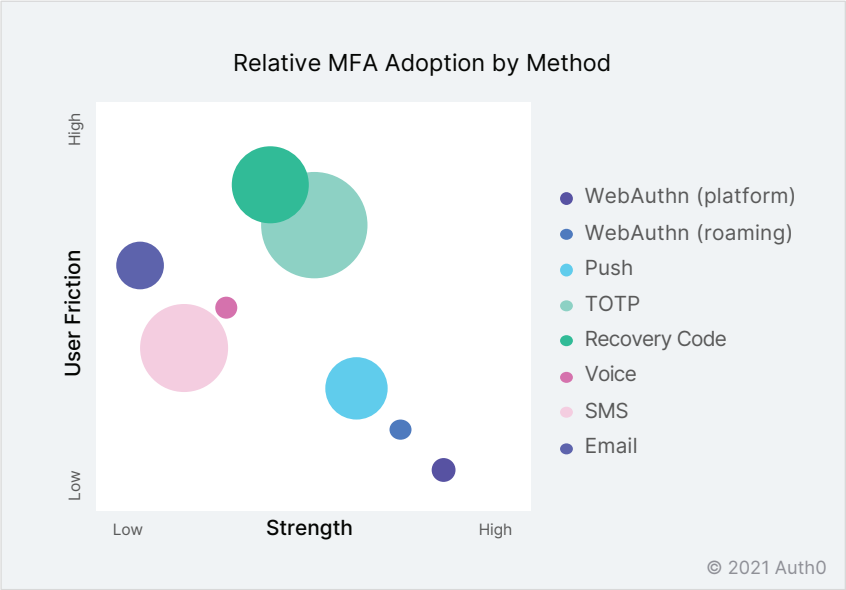
- The account credentials (e.g., from a breach or guessed before triggering an automated defense)
- To pass the MFA challenge as a secondary proof of identity

> **Two critical factors that contribute to the success of an MFA program are enrollment desire and ease of use. There's plenty of room for improvement — in a recent survey of IT and security professionals, the Ponemon Institute found that only 35% of respondents required MFA.[13]**

Aggregate Observations

Auth0 makes a number of MFA methods available to customers:[14] Figure 11 plots these methods on a grid based upon their relative strength and friction; the size of each circle represents the relative adoption within the Auth0 customer base as of May 2021.

Figure 11: Mew MFA methods based on WebAuthn offer a great combination of strength and low user friction



12. It's important to note that it's not uncommon for highly motivated and well-resourced threat actors to know (and to offer for sale) workarounds to MFA — particularly for corporate targets. These bypass mechanisms often leverage legacy authentication protocols, so it's important to disable such systems and to require administrator approval for OAuth and similar applications.
13. See **Cost of a Data Breach Report 2020** [IBM]
14. More details are available at **https://auth0.com/docs/mfa/mfa-factors**

WebAuthn-enabled device biometrics (e.g., Apple Face ID, Apple Touch ID, Windows Hello), shown as *WebAuthn (platform)* in the figure, offer the best combination of high security and low user friction, closely followed by WebAuthn-enabled security keys (e.g., YubiKey, Feitian, Titan), shown as *WebAuthn (roaming)*.

Push notification via the Auth0 Guardian[15] app (*Push*) also provides strong security and ease of use. The most widely adopted MFA methods are a time-based one-time password delivered to an authenticator app, like Authy or Google Authenticator (*TOTP*), SMS-delivered one-time password (*SMS*), and the use of a recovery code provided to a user after they enroll in MFA (*Recovery Code*).

> **WebAuthn is a big step forward for security and user experience**
>
> **Implemented via a WC3 Web API, WebAuthn allows browsers to authenticate using a public/private key pair generated for each user/device/website, instead of shared secrets. Importantly, because it guarantees that credentials are only valid for the websites where users actually registered, the method is not vulnerable to phishing.**
>
> **WebAuthn is relatively new, so adoption is fairly limited at this time; nevertheless, WebAuthn holds tremendous appeal for both users and application providers, so enrollment is expected to grow substantially.**

Using email (*Email*), typically to deliver/receive a one-time password or link, as an MFA method enjoys moderate support, while delivering an OTP via Voice (*Voice*) is near the bottom of the adoption rankings.

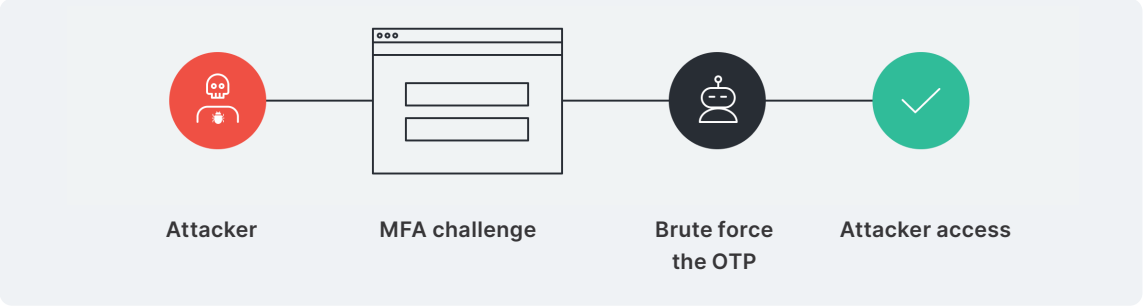
As more organizations offer or require MFA, threat actors are forced to try to bypass such security measures. Some of the different techniques range from manual efforts that often combine a number of tactics (e.g., SIM swapping, social engineering) to simplistic and highly automated approaches.[16]

15. Auth0 Guardian is a mobile app that can deliver push notifications to a user's pre-registered device, or generate one-time passwords if that factor is preferred; to learn more, see **https://auth0.com/docs/mfa/auth0-guardian**

16. For example, the July 2020 cryptocurrency scam that leveraged Twitter employed a combination of social engineering, SIM swapping, and insider threats to gain access to high-profile accounts while disabling MFA

The most common attack vector is to apply brute force in an attempt to 'guess' the authentication code (i.e., the one-time password, or OTP) used in several MFA methods. In the first four months of 2021, Auth0 logged more than 87,000 attempts to brute force an OTP.[17]

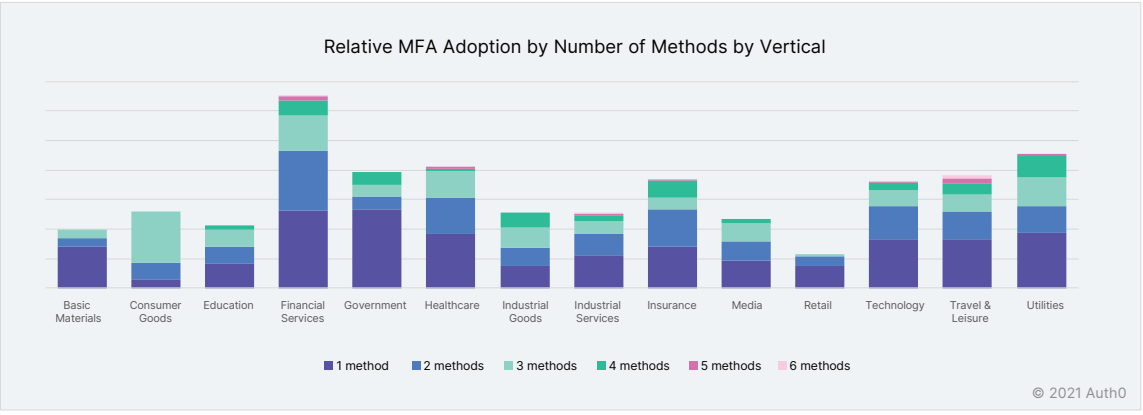
Figure 12: Anatomy of an MFA bypass attack



Application providers can remove enrollment barriers by offering multiple MFA methods for users. This can increase identity security — provided MFA is properly implemented.

Figure 13 shows the relative adoption of one or more MFA methods by Auth0 customers within each industry vertical — six is the maximum number enabled by any customer so far — with the height of each column corresponding to the vertical-wide adoption of MFA. Financial Services leads the way and, along with Industrial Services and Travel & Leisure, is one of three verticals featuring customers who have enabled six different MFA methods. Utilities has the next-highest adoption, followed by Health Care.

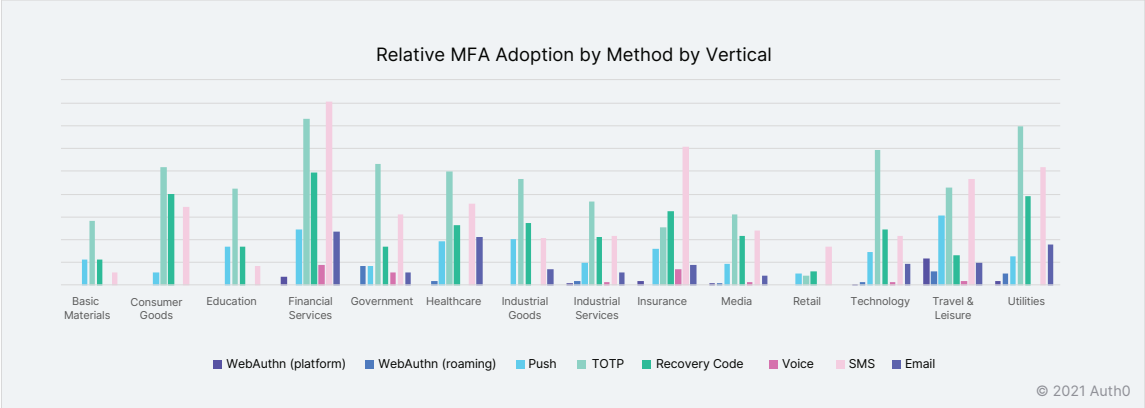
Figure 13: MFA adoption varies considerably by vertical



17. To be considered a brute force attack against MFA, during signup or authentication a user must enter an incorrect OTP more than the limit prescribed by the application provider — note that this is distinct from simply abandoning the login attempt

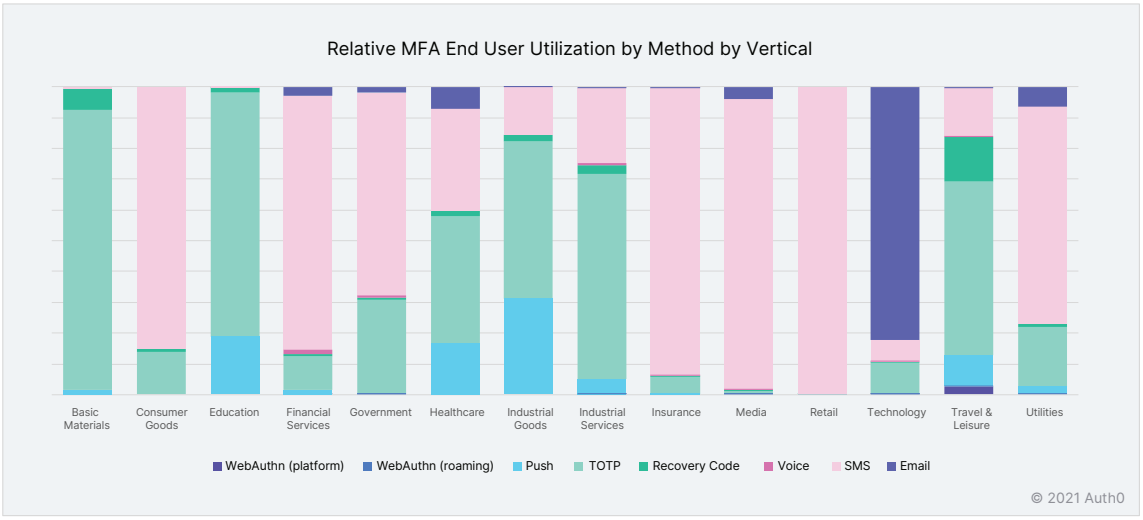
Figure 14 examines MFA adoption slightly differently, this time showing the adoption rate of each of the eight methods. Time-based OTPs have the widest adoption in general, but SMS is favored by Financial Services, Insurance, and Travel & Leisure.

Figure 14: The adoption of different MFA methods varies significantly by industry vertical



When we look at the rate at which end users adopt MFA (Figure 15), we see that SMS is favored, in general, followed by time-based OTPs. Of course, there are variations by vertical: for example, both Basic Materials and Education shun SMS, and Technology is alone in embracing email.

Figure 15: SMS and time-based OTPs have widespread user adoption



Achieving Balance with Adaptive MFA and Step-up Authentication

While multi-factor authentication is the most effective way to prevent account takeover, another way to create friction for threat actors is to detect the telltale signs of an account-focused attack and then implement countermeasures. For example:

- A user experiences 10 consecutive login failures from a single *IP address*; or
- A user experiences 10 consecutive login failures from any *IP addresses*

Achieving a balance between security and usability is vital for creating a positive user experience. Two ways for fine-tuning that balance are:

- Adaptive multi-factor authentication
- Step-up authentication

Traditional MFA as outlined above is incredibly effective in preventing attacks, but it comes with a usability cost, since it requires additional steps that a user must complete in order to continue with the interaction. Adaptive MFA is a technique that only engages MFA when a user interaction is deemed risky based on behavioral data:

- **Unknown device:** A user attempts to log in from a new device
- **Impossible travel:** The location from which the user is attempting to login is too far aware from their previous login location for them to have made the trip
- **IP reputation:** The user is attempting to log in from an IP address that has a poor reputation

By reserving MFA for risky scenarios, adaptive MFA maintains security while preserving the frictionless experience for the majority of users.

To strike a balance between security and friction, step-up authentication is a technique that adapts identity requests to the importance of the resource and the risk level if it were to be exposed. It ensures users (or whomever might be posing as a user) can access some resources with one set of credentials but will prompt them for more credentials (e.g., MFA) when they request access to sensitive resources. Here are three emblematic scenarios where step-up authentication is a practical solution:

1. Users want seamless access to certain resources, but organizations want to verify their identities before they access anything more sensitive
2. Employees need access to data to complete everyday work, but occasionally need access to private data that would cause damage if exposed
3. An organization has or wants to deploy a membership model that limits complete access to their site or service to paying users

The risk with step-up authentication is in the implementation — effective implementations require careful planning about to whom you grant access and whom you ask to step up.[18]

Breached Password Usage

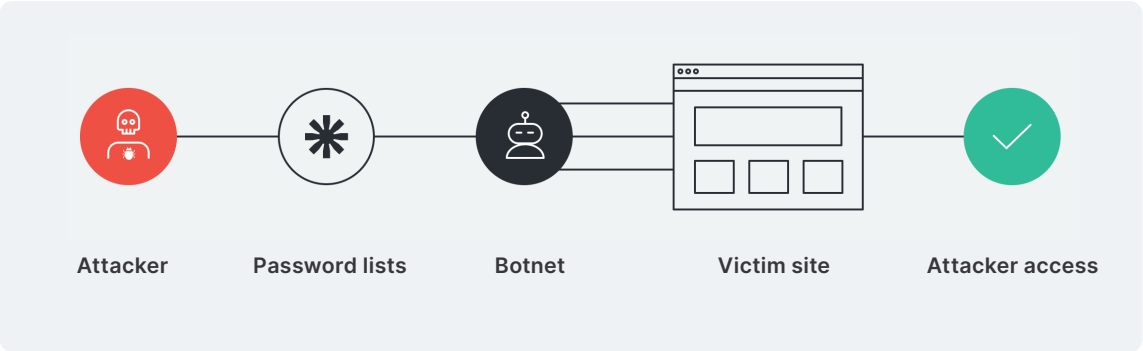
Within the domain of customer identity and access management, poor password management practices on the part of users, and the ease with which threat actors can purchase breached passwords, has the effect that password data is no longer secret.

Today, threat actors can simply purchase breached credentials and use them directly to gain access into accounts. This risk can be somewhat managed by leveraging these

18. To learn more about step-up authentication, in general, please see **What Is Step-Up Authentication, and When Should You Use It?** [Auth0]

same credential lists to detect when users are employing a password that has appeared in a breach. Upon detection, an application provider can warn the user and encourage or require some mitigating action on their part.

Figure 16: Anatomy of breached password usage



Aggregate Observations

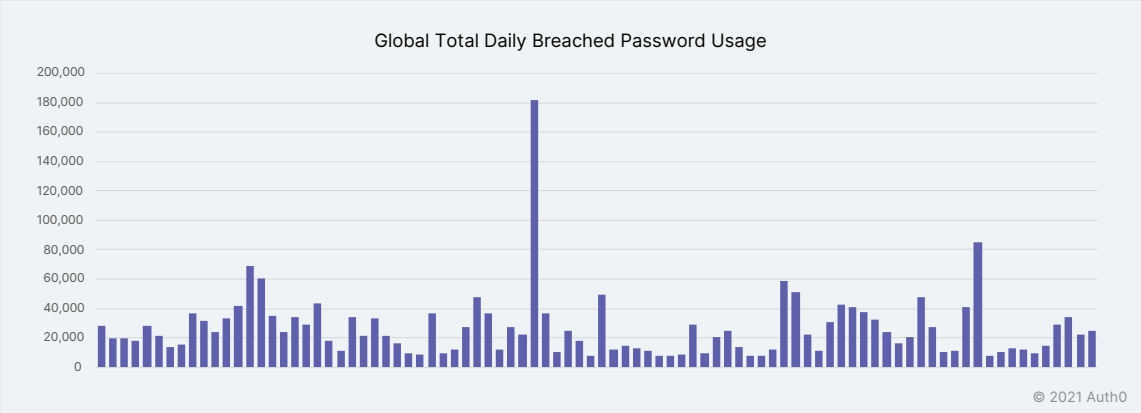
Auth0 maintains a large, constantly growing database of username-password pairs that were known to be compromised in data breaches, which allows us to determine when users are logging in with compromised credentials.[19]

> **When users are detected using compromised credentials, a number of actions can be performed. For example, an administrator can be informed while still allowing the login, the user can be prompted for multi-factor authentication (MFA), or the user can be blocked until they perform a password reset.**

As Figure 17 shows, the use of breached passwords is a constant threat against identity services. In the first 90 days of 2021, the Auth0 platform detected breached passwords at an average of more than 26,600 per day, with a minimum of just under 7,300 and a high on Feb. 9, 2021 exceeding 182,000.

19. To learn more about how to use the Breached Password Detection feature, please visit **https://auth0.com/docs/attack-protection/breached-password-detection**

Figure 17: Volume of breached passwords observed by the Auth0 platform in the first 90 days of 2021



Example: Attackers Use Breached Zynga Credentials

Investigation by the Auth0 Security Engineering Team discovered that the Feb. 9, 2021 incident targeting a customer in the Travel & Leisure vertical had leveraged credentials stolen in 2019 from the online gaming company, Zynga.

Rarely can we attribute the majority of an attack to a single breach, but in this attack 72% of the credentials are attributed to the Zynga breach. Reports note that almost 173 million unique email addresses, along with usernames and passwords were exposed.

> **Done properly, hashing and salting passwords is considered a best practice. But faulty implementations can cause more than headaches. At least employ multiple layers of password defense.**

When the breach was revealed in September 2019,[20] much of the reporting emphasized that the passwords were stored as salted SHA-1 hashes, which would make them harder to monetize. However, the Feb. 9, 2021 attack indicates that this breach has now been cracked and used in the wild.[21]

20. See Zynga's official acknowledgment at **Player Security Announcement** [Zynga]
21. See **Brute Force: Password Cracking** [MITRE]

Improving Password Management

In addition to implementing breached password detection, some simple — but effective — ways to enhance identity security are to:

- Require strong passwords
- Prevent users from repeating their passwords
- Compare potential passwords against a dictionary to prevent use of common passwords
- Implement a good password reset process

Password reset is a necessity for any app. But building a good password reset process is more than asking security questions. If your password reset process makes life harder for your customers, you'll be giving them a reason to stop using your service.

Good password reset processes do two things:

- **They minimize friction for the customer:** It shouldn't take your customer more than a minute to reset their password, and the process should only require information customers are comfortable entering, like email addresses
- **They make sure the customer's information is secure:** Providing safeguards against things like multiple failed logins and only sending information via secure channels

Email is most commonly used for password reset because it satisfies both these criteria. It minimizes friction as typing in an email address is quick and easy for a customer, and it will protect their information as only the customer should have access to their inbox.

Mistakes to Avoid

A single misstep in password reset can ruin your customer's entire experience with your product. These mistakes often come in the form of:

- **Security questions:** Static information is easy to obtain — where you went to school, your mother's maiden name, even your pet's name, are probably available somewhere on the internet, making them available to attackers
- **Passwords in plaintext:** Instead of resetting the password, some sites send the original password back to the customer, which is a massive vulnerability — for a password to be sent in plaintext, it must be stored in plaintext, which means that the chances of attack are increased
- **Error messages:** If an application says whether or not an email address is registered, an attacker could potentially know if a customer has an account — this gives them one more piece of information to use against your customer
- **Requiring unnecessary information:** Security must be balanced with usability — asking customers for a photo ID is a safe practice, but its overall effect on the customer experience is a negative one

Using Social Identities to Combat Password Reuse

Social login provides single sign-on for end users. Using existing login information from a social network provider like Facebook, Twitter, or Google, the user can sign into a third-party website instead of creating a new account specifically for that website. This convenience simplifies registrations and logins for end users and enhances security because a user is more likely to recognize the importance of protecting — and to take the extra effort to protect — their critical social accounts.

Application providers enjoy benefits, too, including:

- **Increased registrations:** Many users prefer reusing an existing account over creating another new one
- **Verified email:** The social network provider is in charge of verifying the user's email. If the provider shares this information, then you will get a real email address rather than the fake addresses often used to register in web applications. Social providers will also handle the password recovery process.
- **Greater personalization and customization possibilities:** Social network providers can give you additional information users have consented to share, such as location, interests, birthday, and more, which you can use to enhance your services
- **One-click return experience:** After users register in your application using Social Login, their return experience will be very simple, as they will probably be logged into the social network, and just one click will be enough to login to your application.

Across the entire Auth0 customer base, Facebook is by far the most-used social identity, followed by Windows Live, LinkedIn, Twitter, and Apple; of course, different verticals have different preferences (Table 2).

Table 1 — The top five social identities used by each vertical

	#1	#2	#3	#4	#5
Overall	**Facebook**	**Windows Live**	**LinkedIn**	**Twitter**	**Apple**
Basic Materials	Facebook	Windows Live	LinkedIn	Twitter	Apple
Consumer Goods	Facebook	Apple	Windows Live	Instagram	GitHub
Education	Facebook	LinkedIn	Windows Live	Twitter	GitHub
Financial Services	Facebook	LinkedIn	Windows Live	Twitter	Apple
Government	Facebook	Windows Live	LinkedIn	Twitter	GitHub
Health Care	Facebook	Windows Live	Apple	LinkedIn	Line
Industrial Goods	Windows Live	Facebook	LinkedIn	Apple	Twitter
Industrial Services	Facebook	LinkedIn	Windows Live	Twitter	GitHub
Insurance	Facebook	Windows Live	Twitter	LinkedIn	GitHub
Media	Facebook	Twitter	LinkedIn	Windows Live	Apple
Retail	Facebook	Twitter	Apple	Windows Live	LinkedIn
Technology	Facebook	Windows Live	GitHub	LinkedIn	Twitter
Travel & Leisure	Facebook	Apple	LinkedIn	Twitter	GitHub
Utilities	Facebook	Windows Live	LinkedIn	Twitter	Apple

Other Common Identity Attacks

While the threats outlined previously represent the vast majority of the attacks we observe, there are several others that warrant brief examination.

Password Spraying and Password Guessing

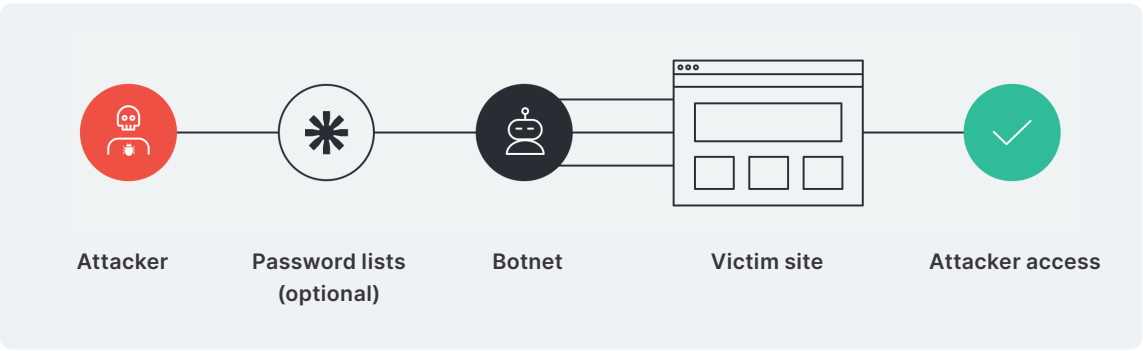
Password spraying is a brute-force attack method in which a threat actor uses automated tools to try common passwords across many different accounts.[22]

Password guessing is a cruder approach:[23] where password spraying tries relatively few passwords across relatively many accounts, password guessing tries many passwords across any number of accounts.

22. See **Brute Force: Password Spraying** [MITRE]
23. See **Brute Force: Password Guessing** [MITRE]

Because of insecure password habits (e.g., password reuse, using common words, etc.), a small number of optimizations — including leveraging lists of breached passwords and dictionaries of words that are frequently incorporated (yes, like "password") — can dramatically improve an attacker's likelihood of trying the correct password.[24]

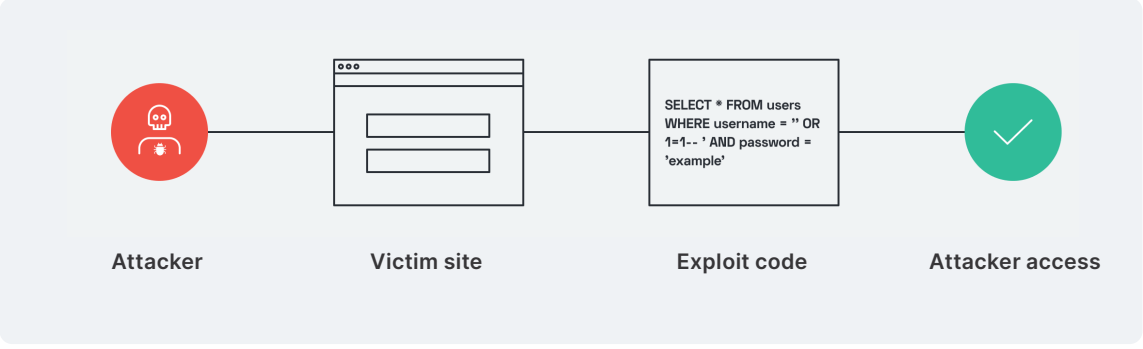
Figure 18: Anatomy of a password spraying attack



Injection

Injection attacks — familiar to every fan of the webcomic xkcd[25] — insert code into a field, like a username, to exploit poorly implemented systems that fail to sanitize inputs. For instance, the code might instruct the backend to ignore the password check and automatically log the attacker into the first account in the database of users, which is often an administrative account.

Once an attacker has administrative access, a wide range of intrusion actions become available.



24. Technically, the attacker does not need to try the correct password for an account, only one that hashes to the same value as the correct password — for an authoritative explanation, see **Birthday Attacks, Collisions, And Password Strength** [Auth0]
25. See **Exploits of a Mom** [xkcd]

Session Hijacking

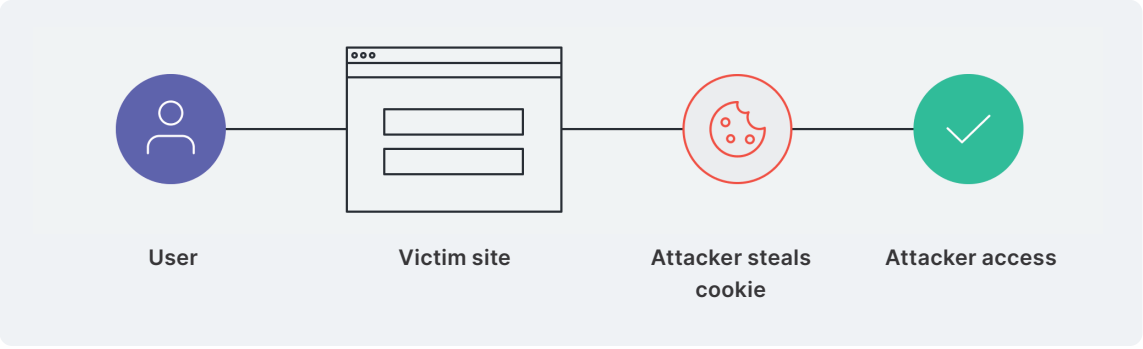
In a session hijacking attack, an attacker gains access to an active session without having to provide a password.[26] Two ways to achieve this outcome are:

1. After a legitimate user logs in, the attacker steals the user's session cookie
2. The attacker tricks the user into logging in through a malicious link with a prepared session ID

Both approaches can be scaled somewhat, but session hijacking is more likely to be used as part of a targeted attack against particular users.

The attacker maintains access as long as the session remains active (a period that varies by application provider).

Figure 20: Anatomy of a session hijacking attack



Session ID URL Rewriting

Like session hijacking, session ID URL rewriting is an attack that provides a threat actor with account access; in this case, the attacker steals the session URL — which can be achieved in a number of ways, including:

- Sniffing an insecure Wi-Fi connection
- Seeing the URL in person (e.g., looking over someone's shoulder)
- Using spyware/malware to grab screen images

As with session hijacking, the attacker maintains account access for the duration of the session.

26. For more information, see **Session hijacking attack** [OWASP]

Figure 21: Anatomy of a session ID URL rewriting attack

Securing Sessions

Here are three ways to improve session security:

- Avoid putting session IDs in the URL
- Use a server-side, secure session manager that generates a new session ID after login
- Securely store session IDs and invalidate them after logout

Conclusions and Recommendations

Robust and resilient customer identity and access management (CIAM) capabilities are critical in the fight against data breaches, account takeover, credential stuffing, identity theft, privacy abuses, and other risks. Identity services have an agile, secure-by-design, defense-in-depth approach that can dissuade threat actors by disrupting the economics of attacks.

Traditionally, defense in-depth referred to the use of multiple security products or solutions operating together at different layers or locations (e.g., endpoint, network, cloud). Today, we extend the meaning to include multiple layers of defense within a single solution.

In the context of identity and access management, this layered approach corresponds to employing defensive measures before and throughout the authentication workflow.

The challenge for application builders is to develop and implement security measures that strike an appropriate balance of increasing friction for attackers while respecting the user experience. Whether you are developing your own in-house solutions, or relying on an identity-as-a-service provider, here are some fundamental recommendations:

Implement and encourage MFA

MFA is one of the most effective ways to disrupt attacks — implement multiple methods and encourage user adoption.

Embrace WebAuthn and enable it on supported devices.

Use the same failure messages

Detailed failure messages can assist threat actors by providing information about users that exist in the system.

Keep attackers in the dark by providing generic failure messages.

Limit failed login attempts

Brute force, credential stuffing and password spraying often trigger many failures for each successful login.

Use this behavior to detect attacks and trigger countermeasures.

Implement secure session management

Use a server-side, secure session manager that generates a new session ID after login.

Don't put session IDs in the URL, and ensure they are securely stored and invalidated after logout.

Don't ship with default credentials

Default admin credentials are a major attack vector because many users leave them unchanged, leaving systems vulnerable to dictionary attacks.

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.

Monitor for breached password use

Many users reuse the same or similar passwords across multiple sites, so a breach in one service can threaten many others.

Force users to change breached credentials.

Don't store plain-text passwords

If your password database is truly illegible, then it's of value to hackers.

Encryption makes your organization a much less appealing target, but the implementation must be sound.

Afterword

We often hear “zero trust” as the paradigm of now, of the solution to a wide range of security threats — but what’s missing from conversations is the fact that zero trust hinges on identity. Identity not only makes zero trust possible, but I’d go as far as to say that identity is trust.

Identity is a constant participant in security but it takes on many forms and exists in many different dimensions. From basic username and password combinations, to fingerprinting and browser-based behavioral profiling, who we are as users is always being redefined. And depending on to whom we are trying to identify ourselves, our identities are constantly reimagined: Is who we are just the combination of a social media provider voucher and a passed CAPTCHA challenge? Are we the embodiment of a username, IP Address, and historical behavior?

I think we are each much more than all of that. We are complicated, complex, and interesting — and online transactions are only just starting to tap into that complexity by utilizing what distinguishes one identity from another to enhance security.

While the web has historically generalized identity as a simple combination of usernames and passwords, this approach no longer withstands the test of time. As attacks continue to grow, attack surfaces expand, and attackers gain sophistication and motivation, securing identities is critical to the future of authentication and authorization in an online context.

—KİM BERRY, PRİNCİPAL SECURİTY THREAT INTELLİGENCE RESEARCHER

Learn More About Identity Management with Auth0

Identity is vital to enabling online applications and will become even more important as the zero trust paradigm gains wider adoption.

Identity is also difficult — even seasoned pros find creating effective and efficient implementations to be challenging.

Auth0 takes on the burden of identity and access management, so you can focus effort and energy on delivering core business value.

Auth0 is an easy-to-implement, adaptable, and secure authentication and authorization platform. Built on a set of composable building blocks exposed through APIs and protocols, the Auth0 identity OS provides multiple solutions to address any identity use case without forcing a compromise between convenience, privacy, or security.

Learn more at **auth0.com/identity-os**.

Signs you need to move from DIY to an identity management solution

- You need a standards-based solution, such as OpenID Connect, SAML, WS-Federation, and/or OAuth
- You have users who authenticate with various identity providers but lack a way to link their accounts
- You have applications on different domains and require users to log in separately for each
- Your best developers spend their time building and maintaining identity management and authentication instead of building core business applications
- Your company has experienced any type of data breach or you are concerned with a data breach
- You're being asked for industry certifications that you haven't considered/addressed

Auth0's modern approach to identity enables organizations to provide secure access to any application, for any user. The Auth0 platform is a highly customizable identity operating system that is as simple as development teams want, and as flexible as they need. Safeguarding billions of login transactions each month, Auth0 delivers convenience, privacy, and security so customers can focus on innovation.

For more information, visit **https://auth0.com** or follow **@auth0** on Twitter.