Push Security

Are you confident in your MFA coverage?

MFA coverage is being increasingly scrutinized by regulators and insurance providers. Learn how cyber breaches are shaping the future of MFA compliance.

Push Security

Contents

Introduction

Many businesses believe they've solved MFA. It's deployed across their core apps, enforced at the Identity Provider (think Okta, Microsoft Entra, Google Workspace), and covered by policy. But real-world cyber breaches, regulatory fines, and denied insurance claims tell a different story.

Our data shows that **two in five accounts are missing a second authentication factor.** Across the surface of a company's business app suite, unsecured accounts are left exposed to weak and stolen passwords — accessible to even the least skilled and resourced attackers, but useful to all.

Lack of MFA opens the door to data theft, ransomware deployment, and extortion. Attackers are routinely finding and exploiting MFA gaps to breach critical business applications and services accessed over the internet. Modern breaches are exposing the risk posed by unmonitored third-party apps and services, and accounts with missing MFA are the lowest hanging fruit.

Regulators have noticed. Frameworks that once hinted at MFA now mandate it. Enforcement actions treat missing MFA as negligence. Insurance providers are rejecting claims after finding that MFA was missing or misconfigured at the point of breach.

Most organizations assume that MFA coverage is widespread. The truth is: most organizations only have MFA where they're already looking — at their core enterprise logins. Often, assumed coverage doesn't account for the growing ecosystem of business-critical SaaS that teams use daily (hundreds per business) as operations shift from internal networks to the cloud.

This ebook outlines the whole picture. It maps regulatory and insurance expectations today, how the evolution of business IT and cyber attacks in the wild will shape the future of insurance and compliance, and what organizations can do to bridge the MFA gap.

Push Security

Section 1: Mapping the regulatory & insurance landscape

Compliance frameworks haven't just caught up to MFA, they've moved ahead of how most organizations are applying it. What used to be vague best practice, "implement appropriate safeguards," is becoming a clearly stated expectation.

Regulators, auditors, and insurers are no longer asking whether you use MFA. They're asking how it's configured, where it's enforced, and whether it's being validated across all users and accounts. And they're treating missing MFA as a failure to meet baseline security expectations.

Failing to enforce MFA is no longer just a technical oversight, it's a compliance failure. And enforcement actions are catching up with that reality. Healthcare, finance, insurance, and lending organizations have all been fined for MFA-related failures:

Recent fines for MFA non-compliance include:

$500,000 fine
against a children's hospital under HIPAA

$4.2 million fine
for a personal loan provider

$1.55 million fine
against an auto insurer

$3 million settlement
for a financial services company under NYDFS

£3.07 million fine
against a UK software company under GDPR

Across sectors, the message is clear: **missing MFA can cost you.**

Same term, different standards

"MFA" might appear in nearly every security regulation, but what that means and where it applies varies widely.

Some frameworks imply it under broader safeguards. Others mandate it explicitly. A few go further, calling for phishing-resistant methods specifically.

Here's how some of the major frameworks interpret and enforce MFA requirements:

Framework	MFA requirement	Scope
NIST SP 800-63-3	Strongly recommended at AAL2+	Federal systems and organizations adopting U.S. standards
CISA / EO 14028	Mandatory, phishing-resistant MFA required	U.S. federal agencies and contractors
HIPAA*	Implied under "reasonable safeguards"*	U.S. healthcare organizations handling ePHI
PCI DSS v4.0	Mandatory for all admin and remote access	Merchants, service providers handling cardholder data
FFIEC	Recommended for high-risk access at minimum	U.S. financial institutions
GDPR	Implied under "appropriate technical measures"	Any organization handling EU resident data
ISO/IEC 27001 / 27002	Recommended in Annex A.9 (access control)	Any organization seeking certification
CIS Controls v8	Explicitly required for remote, admin, and sensitive access	General-purpose security control framework
NYDFS 23 NYCRR 500	Mandatory for all user access to covered systems	NY-regulated financial entities
Cyber Essentials (UK)	Mandatory for all cloud services accessing org data	Common minimum requirement of doing business in the UK
SOC 2	Expected under common criteria for security	Any organization undergoing SOC 2 audit

*The HIPAA requirement could become more stringent pending planned rule changes.

A shift from guidelines to mandates

Across multiple jurisdictions and sectors, regulatory language is shifting from "should" to "must".

- **PCI DSS v4.0** requires MFA for all non-console administrative access and remote access to cardholder data environments.
- **NYDFS 23 NYCRR 500** mandates MFA for all user access to covered systems, not just privileged accounts.
- **HIPAA** interprets the lack of MFA for remote access to ePHI as a failure to meet the "reasonable safeguards" standard.
- **GDPR** fines now reference the absence of MFA as evidence of inadequate technical protections under Article 32.

MFA is also a minimum requirement for insurance

It's not just regulators getting stricter. Insurers are building in MFA as a minimum condition of insurance coverage.

Organizations are incentivized to have MFA. Roughly 20-25% of cyber insurance premiums are dictated by the security controls in place: MFA, EDR, regular patching, etc.

But if your self-attested MFA coverage doesn't hold up under investigation, your provider may not be required to pay, and you're left footing the bill for IR, recovery, legal fees, and business disruption.

Case study: City of Hamilton ransomware attack

The Canadian city of Hamilton, Ontario fell victim to a ransomware attack in February 2024. Attackers disabled nearly 80% of the city's network and demanded a ransom of roughly $18.5 million in exchange for a decryption tool to unscramble the data.

They attempted to claim $5 million under their cyber insurance policy. After more than a year of dispute, the claim was denied because of MFA gaps — a condition of the coverage. Taxpayers were left to foot the $18.3 million bill, including cleanup, rebuild, and one-time consultancy fees.

Push Security

Section 2: The changes shaping future compliance

The direction of travel is consistent: frameworks are getting stricter, auditors are getting more technical, and enforcement is starting to hit data processors as well as controllers.

It's no longer enough to say you've "deployed MFA." You need to demonstrate that it's enforced, phishing-resistant where possible, and applied comprehensively across your business systems.

But there's more to it than that. In-the-wild breaches are exposing just how much business IT has evolved — and where security controls haven't kept up.

Accelerated app adoption has created compliance gaps

Most organizations now use hundreds of SaaS applications, which translates into thousands of sprawling user identities, login methods, and ways to access company systems and data.

Some of these apps are deeply embedded in your infrastructure and are centrally procured. Many others are added with a credit card and never appear in your CMDB.

The reality is that any app that is connected to a corporate identity and processes business data (and very few apps don't) must be protected by MFA as a minimum viable security control. And any breach that stems from an account without MFA deployed is going to be heavily scrutinized by regulators, insurers, and all manner of third-parties.

Real coverage means all apps, identities, and login methods

True MFA coverage expands beyond your centrally managed, SSO-connected apps or your primary enterprise login to any and every app used by your employees for work. **And you're probably using more apps than you realize.**

Enterprise apps
for productivity, collaboration, and data storage

IM apps
for workforce collaboration and communications

Finance apps
for expenses, invoicing, bank accounts, etc,

Dev platforms
for source control, automation, and deployment

GenAI apps
for content creation, task automation, etc.

Marketing apps
for customer data management, outreach, etc.

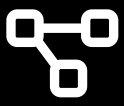
HR apps
for workforce management, recruitment, etc.

Reporting apps
for business analytics, dashboards, etc.

Creative apps
for design, content creation, etc.

Knowledge apps
for internal process management

Security apps
for software patching, vulnerability scanning, endpoint protection, etc.



Network tools
for traffic and infrastructure monitoring

Ticketing apps
for project and task management

Email apps
for internal and B2B communications

And many more...

Unmanaged apps are usually outside the scope of typical audits but inside the reach of attackers. Compromising many, if not all, of these apps could be used by an attacker to cause serious harm to your business.

When an attacker finds an account on an unmonitored app with a breached or weak password and no MFA, it becomes the path of least resistance.

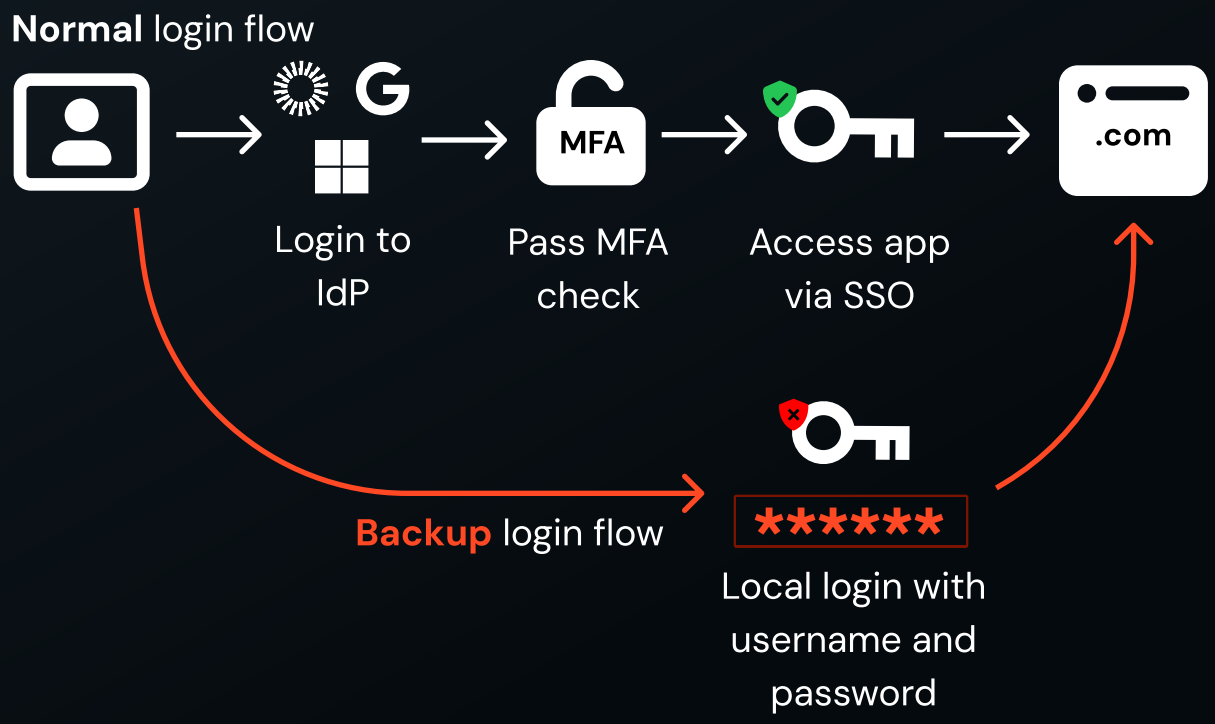
But getting coverage isn't straightforward

It's tempting to think of MFA as binary: enabled or not. The reality is that MFA needs to be enabled across every app and login. This is no small task.

When an app is first used, particularly if self-adopted, a username and password is typically created. Even when an SSO login is created, it's usually added on top. And unless specifically disabled or removed, these login methods can continue to be used.

When most organizations rely on configuring MFA at their IdP login, this means that local logins without MFA can go unnoticed — waiting to be exploited. These "ghost logins" can lead to unexpected MFA gaps that leave accounts exposed to attackers.

"Ghost logins" without MFA can leave accounts at risk of takeover. MFA needs to be deployed at the IdP and app level, OR local logins need to be removed.



App design choices can have a big impact on security

Each app is built differently. Design choices can have a big impact on how authentication and account management is handled.

This means many apps aren't as "secure-by-default" as they could be, while the level to which an app can be configured securely also varies. This leads to situations where, for example, apps allow simultaneous login methods, or don't provide admin-level controls to enforce MFA or make account changes on behalf of users in your app tenant.

Each app provides a different answer to core configuration questions.

- Do you know about the app?
- Is the tenant managed by your security team?
- Can MFA be made mandatory for all users?
- Can you prevent simultaneous login methods?
- Can you audit employee logins methods & MFA?
- Can you enforce changes on behalf of users?

The reality is: most organizations only have MFA where they're already looking — at the IdP layer. This creates a situation where organizations have:

- Accounts on between **200** and **24,000** sites
- **3-100** accounts per employee (average 15)
- **2 in 5** accounts not protected by MFA

Based on Push data, organizations with 1000+ employees when first installing Push

Attackers are exploiting SaaS MFA gaps

Most applications are likely to have user accounts missing MFA, for one reason or another — whether the app doesn't allow MFA to be enforced, you can't manage identities at your subscription tier, or some other configuration nuance. This isn's theoretical — it's playing out in major breaches.

Case study: **Snowflake** customer breaches

165 Snowflake customer tenants were breached via accounts missing MFA. >80% of the compromised accounts had previously had credentials exposed online, and at the time Snowflake did not allow mandatory MFA enforcement. It was also difficult for admins to verify whether accounts had MFA in place.

Hundreds of millions of individuals' data was breached in what has since been touted "one of the biggest breaches ever".

Case study: **Jira** customer breaches

Criminals went on a Jira hacking spree, specifically targeting organizations via their Jira tenants using stolen credentials found online for accounts which lacked MFA. More than 10 organizations were compromised over a 5 month period. 73% of Jira accounts were missing MFA at the point of detection by the Push platform.

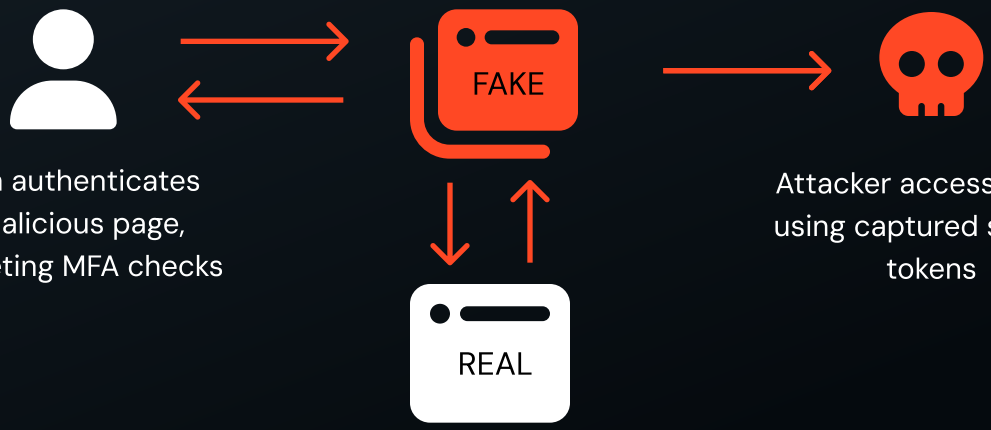
Hundreds of gigabytes of data and thousands of breached records were stolen simply by logging in with a single set of stolen credentials for each victim. One of the impacted companies, Jaguar Land Rover, has now also suffered a major ransomware attack following the Jira compromise.

Not all MFA methods are created equal

If we look at the cyber attacks happening today, it's clear that widespread MFA is the absolute minimum requirement to prevent the lowest-effort breaches.

Less reliable or scalable bypasses like push bombing and SIM swapping have been superseded by modern phishing techniques and tools. MFA bypass attacks are now table-stakes for attackers, meaning that MFA compliance is likely to move beyond a black-and-white question of "was MFA present?" to "what kind of MFA was present?".

MFA-bypassing "Attacker in the Middle" phishing kits are the new normal.



Victim authenticates to malicious page, completing MFA checks

Malicious page proxies information to real site, logging in successfully

Attacker accesses app using captured session tokens

The need for phishing-resistant auth

The rise in MFA-defeating phishing kits is driving the need for phishing-resistant authentication — commonly known as "passkeys". Organizations in the most heavily regulated industries are likely to see phishing-resistant authentication become a compliance requirement in the future.

For passkeys to function as intended, they need to be implemented as the sole login method. Otherwise, they can be susceptible to downgrade attacks. Therefore, detecting and removing backup methods is also recommended — and may also become a requirement for login security in the future.

Push Security

Section 3: Bridging the MFA gap

You can't close gaps you can't see. That's the core problem security teams face with MFA enforcement. The visibility they get from IdPs and SSO platforms doesn't extend across the full surface of SaaS and shadow apps. Policies look clean on paper, but real-world authentication paths tell a different story.

True MFA visibility means knowing:

- Which apps are in use across your environment?
- What login methods are being used for each account?
- Where is MFA missing, misconfigured, or phishable?

App-level integrations can't solve this alone

Posture management tools rely on API integrations with apps or SSO providers to report MFA status. That sounds good in theory. But in practice:

- → Most SaaS apps don't expose MFA configuration through APIs.
- → Many don't support centralized enforcement of MFA.
- → The data returned is often inconsistent across platforms.
- → Free-tier or self-adopted apps often exist outside your configured integrations entirely.

The result? Partial coverage at best, with a long tail of apps where you're limited by each app vendor's development decisions.

Basic coverage is no longer compliant

What used to pass for compliant no longer holds up. Real-world breaches, denied insurance claims, and regulatory fines have exposed the limits of an IdP-centric approach. Surface-level policies aren't protecting organizations from identity-driven attacks, and they aren't satisfying regulators or insurers either.

The bar has been raised. Regulators now expect MFA to be applied consistently and provably. Insurers demand validation, not attestation. Visibility is no longer optional — it's the prerequisite for enforcing MFA where it matters. Without it, organizations can't see what's exposed, let alone protect it.

Getting MFA on every app, login, and identity that processes or accesses sensitive data isn't the ideal; it's the expected baseline. And the longer that gap between perception and reality remains, the greater the risk of attackers exploiting it.

What you need to bridge the gap

To fix the visibility problem, organizations need a solution that:

- → Observes real usage, not just policy or intent
- → Detects login methods and MFA status in real time
- → Surfaces unmanaged apps and ghost logins
- → Provides evidence of MFA status at the app & account level
- → Flags drift or regression over time

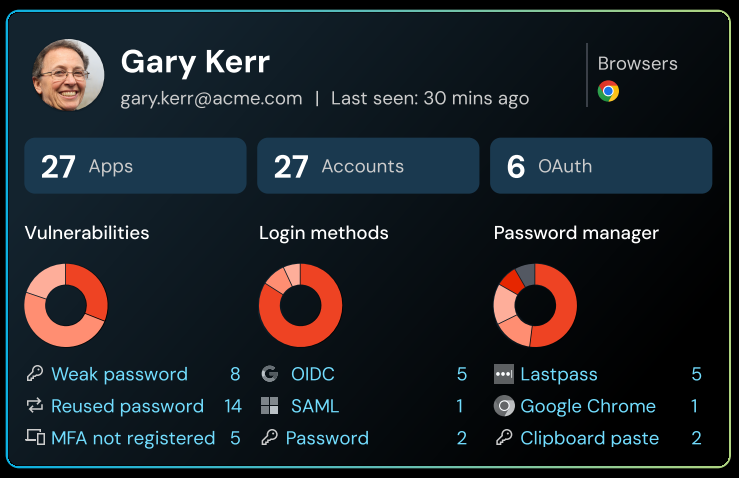
The good news? There's a solution for that...

Eliminate MFA gaps with Push Security

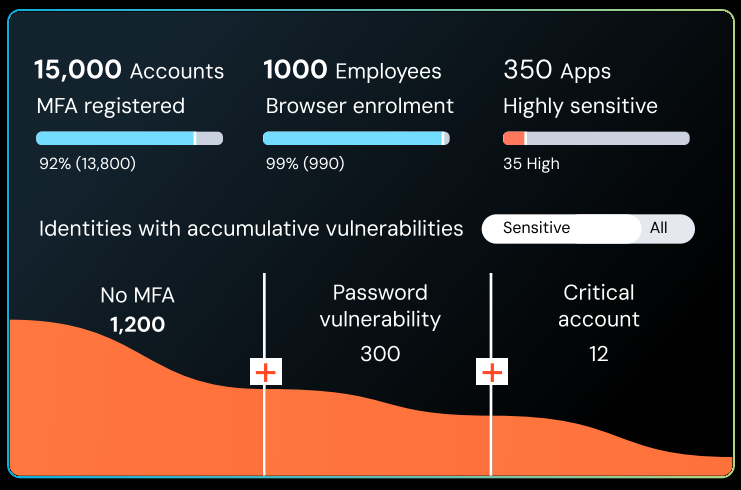
Push Security's browser-based security platform observes logins directly in employee browsers, building a comprehensive picture of user identities and login methods across every app.

Push shows you every app your employees are using (even the unmanaged ones you don't know about), providing detailed information about how users are logging in, and where vulnerabilities exist. This includes accounts missing MFA, where users are logging in with a username and password over SSO, and where a user's password has appeared in a compromised credential feed. You can also use Push to deliver in-browser guidance to users to prompt them to remediate insecure logins.

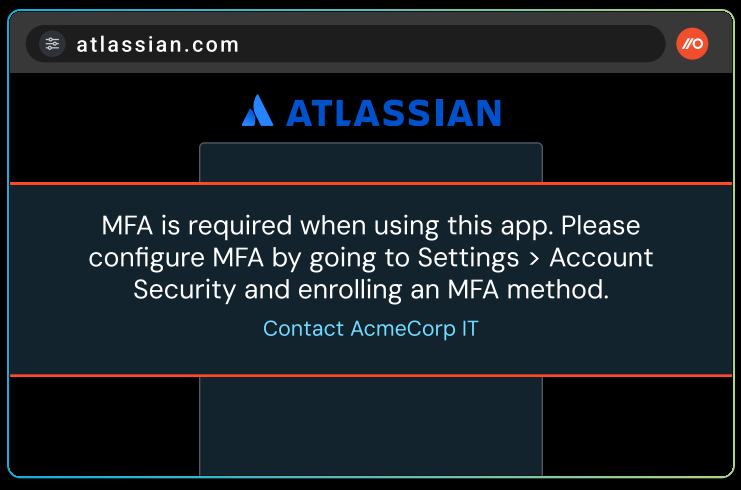
App-level view showing account configuration for all accounts.



User-level view showing all accounts and vulnerabilities associated with an employee.



Dashboard view showing high-level MFA posture and high priority vulnerabilities.



In-browser pop-up instructing a user to register for MFA when logging into an app.

Push Security

There's a lot more to Push than just MFA detection

Push Security's browser-based security platform provides comprehensive detection and response capabilities against the leading cause of breaches.

Push blocks browser-based attacks like AiTM phishing, credential stuffing, password spraying, ClickFixing, and session hijacking using stolen session tokens. You can also use Push to find and fix vulnerabilities across the apps that your employees use, like ghost logins, SSO coverage gaps, MFA gaps, vulnerable passwords, risky OAuth integrations, and more.

If you want to learn more about how Push helps you to detect and stop attacks in the browser, check out our latest product overview or book some time with one of our team for a live demo.

Trusted by security teams doing serious work

Josh Lemos
CISO, GitLab

We looked at the enterprise browser approach, but converging on a single platform was tough. Push gave me the security instrumentation and context I needed without onerous headwinds.

Ash Devata
CEO, GreyNoise

No matter the channel for phishing: email; LinkedIn; text; the employee clicks a link that opens a browser session. We see the main control point moving from the endpoint to the browser.

Jason Waits
CISO, Inductive Automation

Security is only as good as its weakest link. From day 1, Push found the gaps that would allow attackers to circumvent our controls. We use Push every day – they're one of my favorite teams to work with.

Myke Lyons
CISO, Cribl

I'm not going to be able to shove a browser on everybody. I need to be able to support them where they are. Push gave us the visibility and control we needed while allowing people to choose their paths.

GitLab Cribl upvest THINKST CANARY GREYNOISE RISK LEDGER PortSwigger