When MFA isn't actually MFA

Snir Kodesh 13 September 2023 • 6 min read

On August 29, 2023, Retool notified 27 cloud customers that there had been unauthorized access to their accounts. If you’re reading this and you were not notified, don’t worry – your account was not impacted. **There was no access to on-prem or managed accounts.** Nevertheless, here’s what happened, with the hope that this will help apply the lessons we’ve learned and prevent more attacks across the industry.

What happened

On August 27, 2023, we fell victim to a spear phishing attack. The attacker was able to navigate through multiple layers of security controls after taking advantage of one of our employees through a SMS-based phishing attack.

Several employees received targeted texts, claiming that a member of IT was reaching out about an account issue that would prevent open enrollment (which affects the employee's healthcare coverage). The timing coincided with a recently announced migration of logins to Okta, and the message contained a url disguised to look like our internal identity portal. Almost all employees didn't engage, but unfortunately one employee logged into the link provided by the attackers.

The following is a transcription of the message:

Hello A, This is B. I was trying to reach out in regards to your [payroll system] being out of sync, which we need synced for Open Enrollment, but i wasn't able to get ahold of you. Please let me know if you have a minute. Thanks

You can also just visit https://retool.okta.com.[oauthv2.app]/authorize-client/xxx and I can double check on my end if it went through. Thanks in advance and have a good night A.

After logging into the fake portal – which included a MFA form – the attacker called the employee.

The caller claimed to be one of the members of the IT team, and deepfaked our employee's actual voice. The voice was familiar with the floor plan of the office, coworkers, and internal processes of the company. Throughout the conversation, the employee grew more and more suspicious, but unfortunately did provide the attacker one additional multi-factor authentication (MFA) code.

The additional OTP token shared over the call was critical, because it allowed the attacker to add their own personal device to the employee's Okta account, which allowed them to produce their own Okta MFA from that point forward. This enabled them to have an active GSuite session on that device. Google recently released the **Google Authenticator synchronization feature** that syncs MFA codes to the cloud. As **Hacker News noted**, this is highly insecure, since if your Google account is compromised, so now are your MFA codes.

Unfortunately Google employs dark patterns to convince you to sync your MFA codes to the cloud, and our employee had indeed activated this "feature". If you install Google Authenticator from the app store directly, and follow the suggested instructions, your MFA codes are by default saved to the cloud. If you want to disable it, there isn't a clear way to "disable syncing to the cloud", instead there is just a "unlink Google account" option. In our corporate Google account, there is also no way for an administrator to centrally disable Google Authenticator's sync "feature". We will get more into this later.

We use OTPs extensively at Retool: it's how we authenticate into Google and Okta, how we authenticate into our internal VPN, and how we authenticate into our own internal instances of Retool. The fact that access to a Google account immediately gave access to all MFA tokens held within that account is the major reason why the attacker was able to get into our internal systems.

Getting access to this employee's Google account therefore gave the attacker access to all their MFA codes. With these codes (and the Okta session), the attacker gained access to our VPN, and crucially, our internal admin systems. This allowed them to run an account takeover attack on a specific set of customers (all in the crypto industry). (They changed emails for users and reset passwords.) After taking over their accounts, the attacker poked around some of the Retool apps.

After learning of the attack, we immediately revoked all internal authenticated sessions (Okta, GSuite, etc) for employees, locked down access to the affected accounts, notified the affected customers, and restored their accounts to their original state (with original email addresses), reverting the 27 account takeovers.

As an aside, we're glad that not a single on-premise Retool customer was affected. Retool on-prem operates in a "zero trust" environment, and doesn't trust Retool cloud. It is fully self contained, and loads nothing from the cloud environment. This meant that although an attacker had access to Retool cloud, there was nothing they could do to affect on-premise customers. It's worth noting that the vast majority of our crypto and larger customers in particular use Retool on-premise.

With the attack scope defined, let's dig into what we learned, starting with our biggest point of vulnerability: software-based OTPs for MFA.

Beware of "MFA"

We have an internal Retool instance used to provide customer support; this is how the account takeovers were executed. The authentication for this instance happens through a VPN, SSO, and a final MFA system. A valid GSuite session alone would have been insufficient.

The fact that Google Authenticator syncs to the cloud is a novel attack vector. **What we had originally implemented was multi-factor authentication. But through this Google update, what was previously multi-factor-authentication had silently (to administrators) become single-factor-authentication**, because control of the Okta account led to control of the Google account, which led to control of all OTPs stored in Google Authenticator. We strongly believe that Google should either eliminate their dark patterns in Google Authenticator (which encourages the saving of MFA codes in the cloud), or at least provide organizations with the ability to disable it. We have already passed this feedback on to Google.

Sharing our lessons

Social engineering can affect anyone

Social engineering is a very real and credible attack vector, and anyone can be made a target. If your company is large enough, there will be somebody who unwittingly clicks a link and gets phished. Especially as social engineering evolves (with AI and deepfakes, but also with more personal information being available on the internet), educating, preventing, and testing (via red teams) your employees regarding phishing attacks will become ever more important.

Preventing systematic failures

Even with perfect training and awareness of these attacks, mistakes will happen. Just like preventing a junior engineer from accidentally dropping the production database, there need to be systems in place to prevent human error from impacting the overall system. SMS as a second-factor has been rightly criticized for being too vulnerable to SIM swapping attacks. Time-based one-time password (TOTP), while resilient to SIM swapping, is still vulnerable to the same social engineering threat vectors.

Hardware security keys– using FIDO2 provide resilience to these threats. A code can't be disclosed to an attacker– since there isn't a code to share in the first place!

Defense in depth

We're equipped with many levels of protection, including multiple MFA codes, multiple lines of defense (Okta account, VPN, internal admin system, etc.). That said, technological solutions only go so far, and we believe that adding a human-in-the-loop is necessary (although again, not sufficient, given deepfakes) for important actions. We oftentimes bias towards technological solutions because they're easy to self-serve, but ultimately, anything that can be done by an employee can be socially engineered out of that employee too.

We have already implemented this at Retool internally, and expect to implement this in Retool — the product — so our customers have access to building these kinds of human-in-the-loop workflows too.

Trust as little as possible

This incident only affected a very small subset of Retool cloud customers– no on-prem or managed accounts were impacted. We intentionally architected Retool on-prem such that it doesn't trust Retool cloud. Retool on-prem makes no contact to Retool cloud; it is fully self-hosted (the front-end, back-end, storage, etc. are all within your own VPC). This way, customers are fully in control of their security (as well as when they update), and do not need to trust Retool. Even though our cloud systems were compromised, there was no way for attackers to compromise Retool on-premise. We are fortunate we architected on-premise this way.

The vast majority of our customers in more sensitive industries (e.g. crypto, healthcare, finance, etc.) use our on-premise solution, and we encourage our customers to consider it, if security is important.

Understand your threat model

Retool is in the unique position of being a platform where you can build any sort of software imaginable. However, that also means that customers still need to treat building in Retool with the same care and attention to security that traditional codebases require.

We encourage you to understand your own threat model: if you are operating in an industry where apps have access to dangerous, irreversible actions, we strongly recommend customers to integrate further protections. For example, we encourage customers to require separate MFA tokens to execute actions (a first-class primitive in Retool), or escalation flows that require multiple employees to approve actions above certain thresholds (soon to become a first-class primitive in Retool). When we examined our forensics, customers who built secure apps and understood their threat matrix were effective at repelling the attack, even despite the account takeovers.

Conclusion

This situation was challenging. It’s embarrassing for the employee, disheartening to cybersecurity professionals, and infuriating for our customers. For those reasons, these kinds of attacks are difficult to talk about, but we believe that they should be addressed in the open. There are clear risks here that the industry needs to address (e.g. the dark patterns in Google Authenticator). Our hope is that by publishing these attack vectors we can make the industry overall more aware, and enable cybersecurity professionals to harden their own systems.

Editor's note: We are actively working with law enforcement and a third party forensics firm.

The top animation libraries for React in 2023

From Framer Motion to react-spring and react-motion, we evaluated some of the best React animation libraries based on compatibility, presets, docs, performance, and more.

10 MIN READ

Want to leverage LLMs in your engineering organization? Start here.

If your team is just beginning to leverage LLMs, these learnings and best practices can help you use them effectively to ship thoughtful AI-powered apps, workflows, and features. Learnings and best practices to get developers

7 MIN READ

Retool

Use Cases

Admin panels

Firebase GUI

MongoDB GUI

GraphQL GUI

Dashboards

SQL GUI

React components

Google Sheets GUI

Customer support

Financial operations

Integrations

PostgreSQL

MySQL

DynamoDB

Firebase

GraphQL

Amazon S3

Google Sheets

MongoDB

Developers

Changelog

Documentation

Status

On-prem deployment

API Generator

RegEx Generator

Developer Utilities

Developer Network

Resources

Contact Support

Partners

Community home

Support forum

Join our Discord

Show & tell

Retool for Startups

Internal Tools Report NEW

Engineering Time Report

Guide to building secure internal tools

Company

About

Customers

Careers

Blog