Push Security

2026 browser attack techniques

Learn about the browser-based attack techniques behind the biggest breaches today, how they're bypassing cybersecurity controls, and what security teams can do about it.

Push Security

A note from the authors

At the risk of using a few too many buzzwords, we've experienced a massive security paradigm shift in the last few years. Where we used to talk about novel software exploits and advanced endpoint malware, in 2026 we're instead talking about cloud apps and identities as the "patient zero" of modern breaches.

But there's more to it than that. This shift has been accompanied by an evolution in attack paths themselves that many security teams (and vendors and providers, for that matter) haven't adjusted to. And this is leaving them unprepared to deal with modern attacks.

We think the best way to understand this shift is that the browser is the new endpoint. If you start to frame modern attacks as being browser-based, it's easy to see why security tools are failing to intercept many of them.

It's no use just monitoring email if modern attackers can target your users over LinkedIn DM, via paid ads on Google Search, or a (previously) legitimate browser extension. If an attack plays out in the victim's browser, where all an EDR agent sees is a browser process (but not what's happening inside the session), and a web proxy sees some heavily obfuscated traffic (but not what's happening client-side on the web page), it becomes pretty clear that the missing piece here is visibility inside the browser session itself.

We hope you enjoy reading this report, and find it useful in understanding and contextualising the threat evolution that we security professionals are faced with today. If you've got questions, comments, or feedback, let us know!



Mark Orlando, Field CTO
Mark has 20+ years in detection and response, having built, assessed, and managed security teams at the Pentagon, the White House, the Department of Energy, and numerous Fortune 500 companies. He's also a certified SANS instructor.



Dan Green, Security Researcher
Dan has 8+ years in security research and service development. He's designed and implemented security programmes for the public and private sector, testing their cyber defenses and helping to reduce threat exposure.

Push Security

Contents

The browser is the new battleground 1

Browser attacks 101: Methods, objectives, and attack paths 4

TTP breakdown: 6

- Attacker-in-the-Middle attacks 6
- ClickFix and *Fix variants 8
- Malicious OAuth integrations (aka consent phishing) 10
- Malicious browser extensions 12
- Credential attacks (& ghost logins) 14
- Session hijacking (aka token replay) 16

How browser-based attacks bypass traditional controls 18

Threat actor overview: Scattered Lapsus$ Hunters 19

Detecting and responding to browser-based attacks 20

Push Security

The browser is the new battleground

Modern breaches begin in the browser. Often, they never leave it.

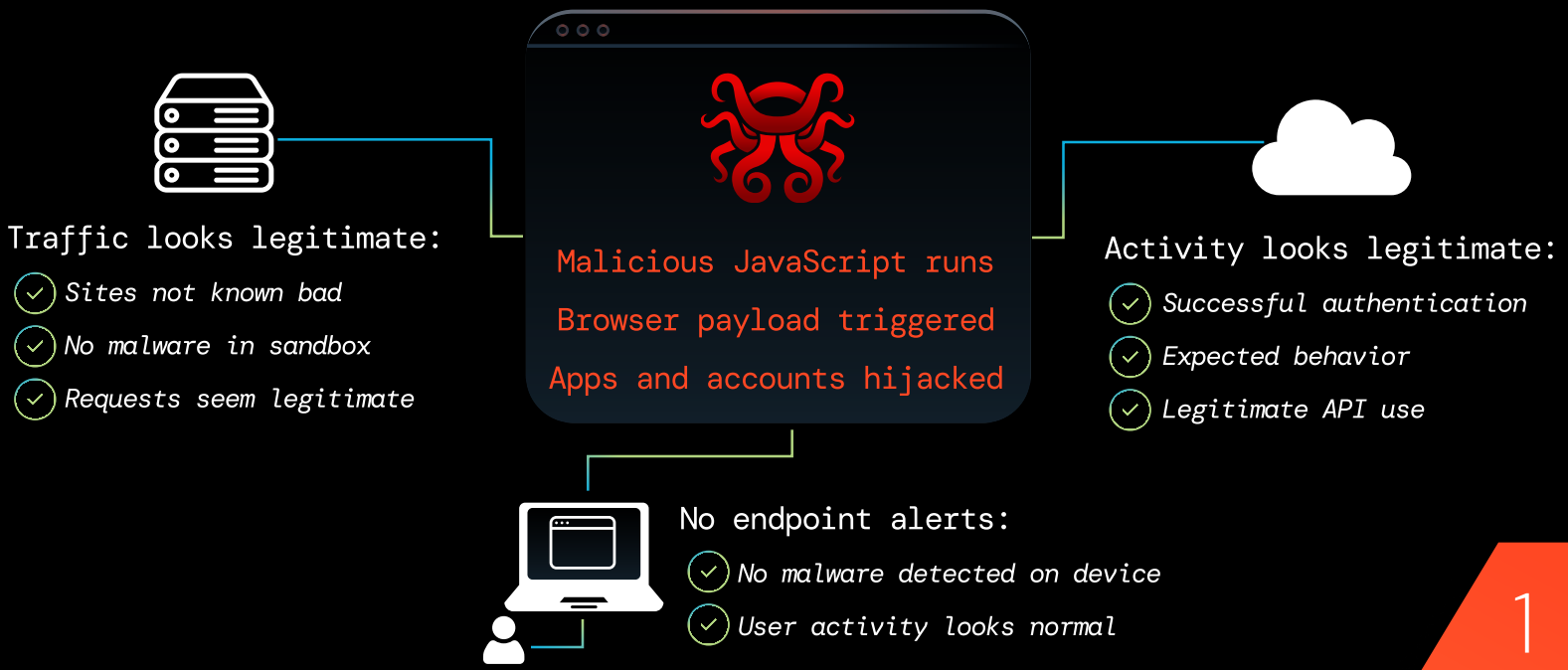
Many modern breaches happen entirely in the web browser. Attackers target your users as they go about their work, intercepting them as they access legitimate, trusted websites.

Attackers are using an ever-expanding kit of browser-based methods to take over victim accounts, hijack their access to business apps, and leverage that access to achieve their objectives — typically data theft, disruption, and ultimately extortion.

These attacks are happening **entirely inside the browser sandbox**, targeting applications directly over the internet, and blending in with legitimate web and network traffic, application access, and user activity.

While it's tempting to see the browser as a gateway — something to be looked at in the context of traffic in, traffic out — it's more like an endpoint. And you need more than a firewall to protect an endpoint.

This is a significant challenge for security teams. Existing security tools can't get visibility of what's happening *inside* the browser. Attackers know this, and are exploiting the browser blindspot. This means attacks aren't being discovered until the damage is already done.



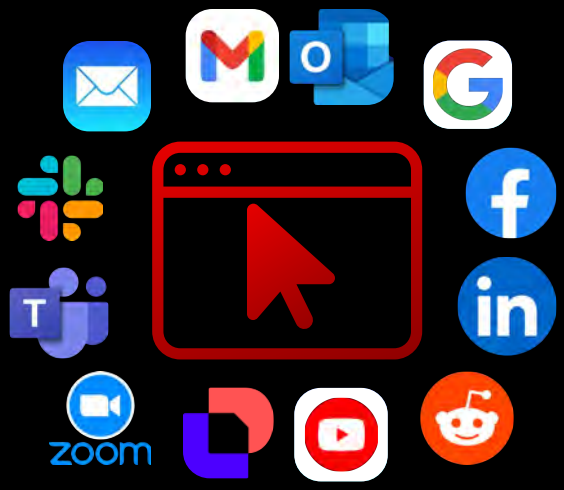
Push Security

It's a jungle out there

Your users can stumble upon malicious content pretty much everywhere they work in the browser. *This is alarming when you consider that 85%+ of modern work happens there!*

Search engines, IM platforms, and social media apps are increasingly popular delivery vectors as an alternative to email. Malicious ads are being delivered across Google Search and Facebook, LinkedIn DM's and job posts are being used to distribute phishing links. These channels have virtually no screening of content or messages and are invisible to security tools.

And when attacks are launched via email, attackers find ways to blend in by abusing legitimate services to send emails from trusted domains — completely bypassing detections.



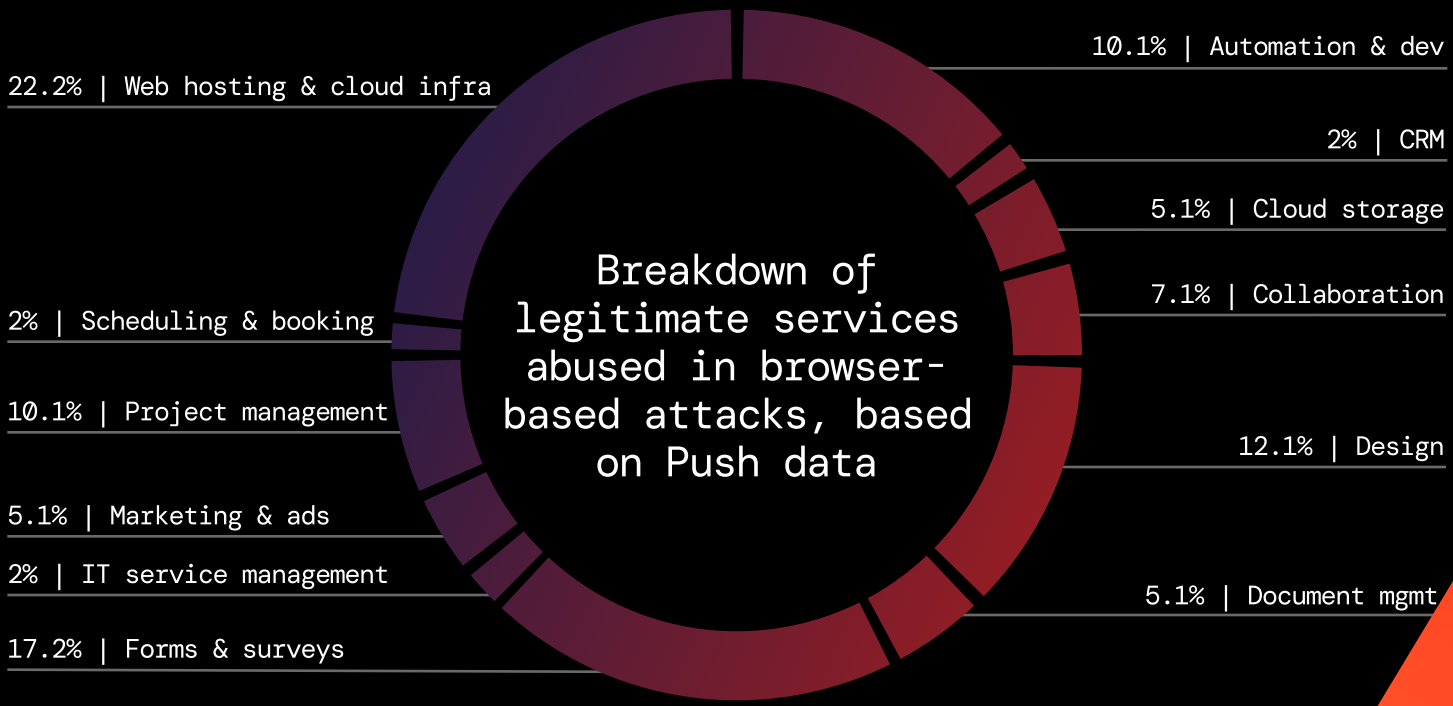
1 in 3 payloads intercepted by Push in 2025 were sent outside of email

Hiding in plain sight

Attackers are abusing a wide range of legitimate sites to host both pages delivering malicious content, and benign pages used in attack chains.

This allows them to blend in with normal, trusted web traffic and fly under the radar.

Services like Microsoft Dynamics, SharePoint, Adobe, Google Firebase, Google Sites, Jotform, Linode, Azure, Cloudflare, Atlassian, and many more are commonly abused by attackers.



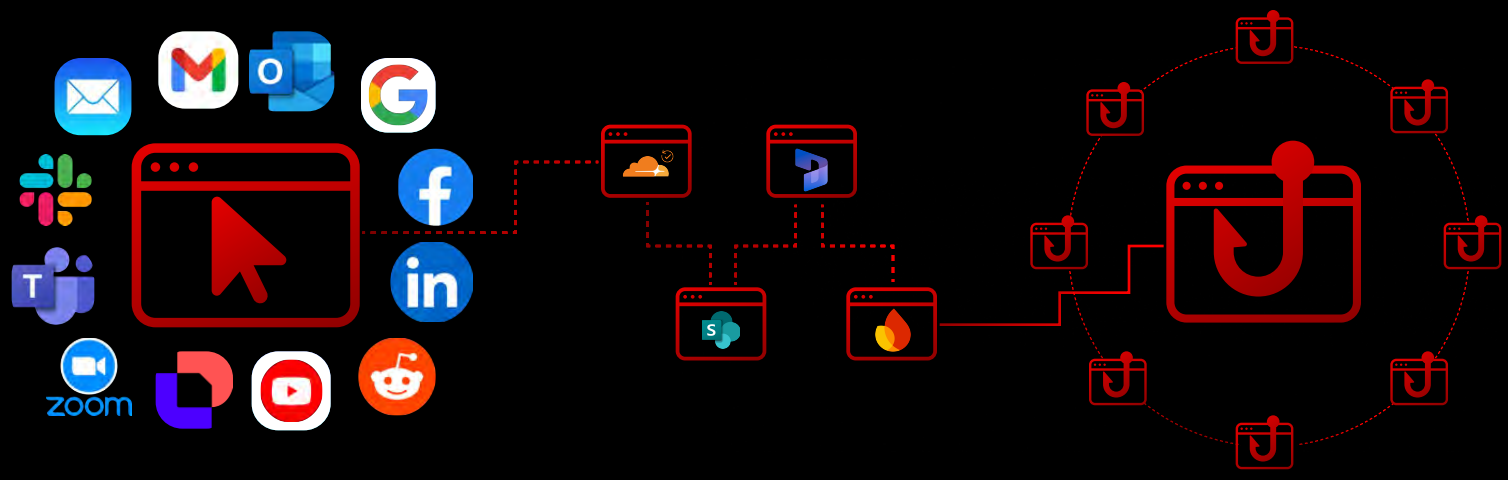
Static indicators can't keep up

Not only are attackers utilizing trusted sites and hosting services, they're combining them with detection evasion techniques like rapid domain rotation and complex redirect chains, as well as email, browser session, and IP-based checks to filter out unwanted visitors.

They're even using ***legitimate security tools*** (bot protection) ***to bypass other legitimate security tools*** (web content scanning and analysis solutions).

This makes it hard for tools that rely on blocking known-bad pages (e.g. network and email solutions) to reliably detect malicious activity. **If you're primarily looking at static IoCs, pretty much every attack is a "zero-day" and will catch you off-guard.**

95% of in-browser attacks detected by Push used some form of bot protection service.



Malicious links delivered over many channels

Redirect chains through legitimate, trusted sites

Malicious domain rotation to stay ahead of blocklists

Attacks in vs. on the browser

It's important to make clear that attacks designed to exploit the browser itself are not the problem. Such exploits are extremely costly to develop or acquire, and this overhead is unnecessary when you consider the viable alternatives. **Attacks *on* the browser are beyond most criminals — but attacks *in* the browser are within reach for all.**

The average price of an identity from a criminal broker, or the tools to compromise an identity yourself, are orders of magnitude below a browser RCE exploit.

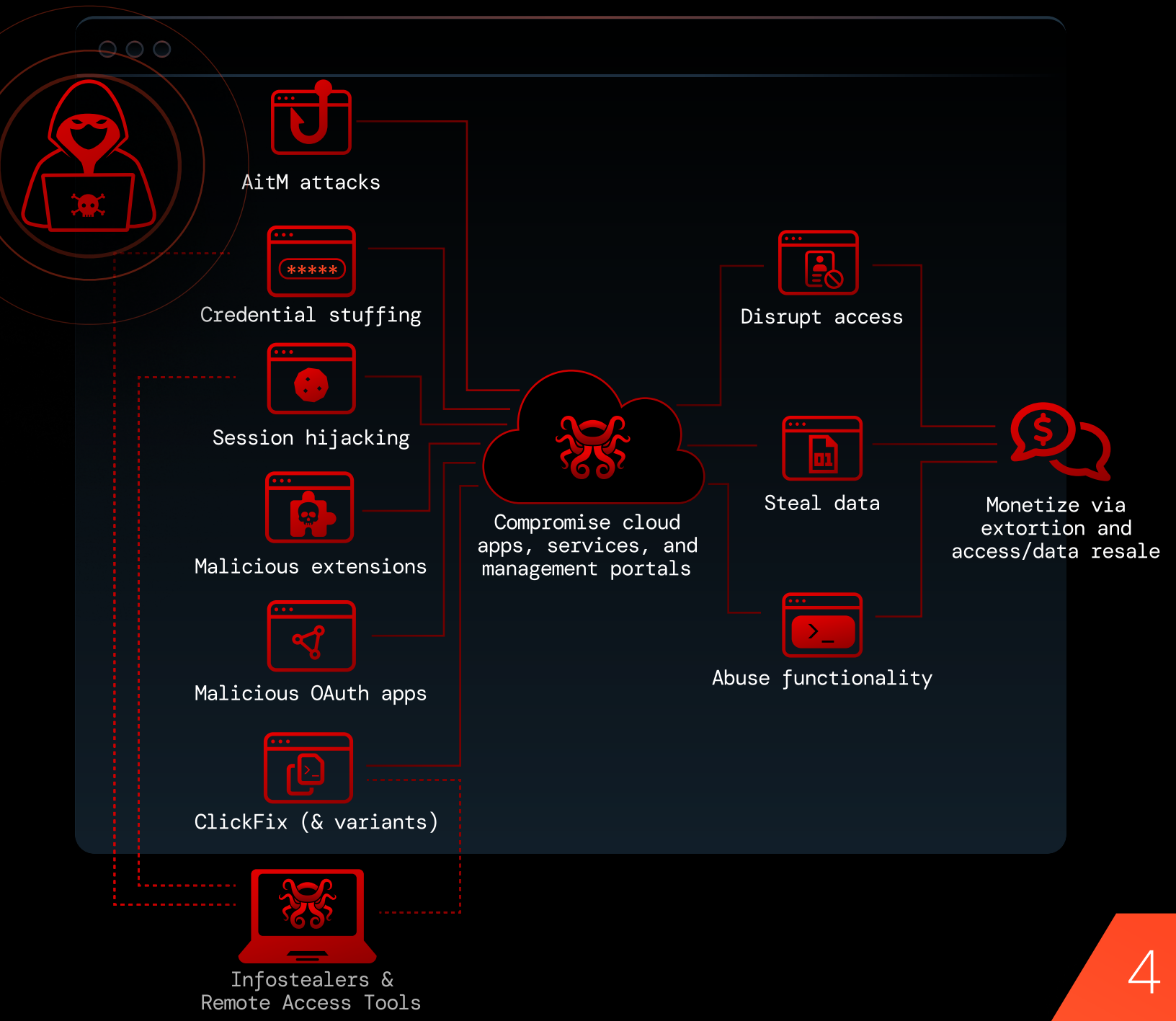
This is where the bulk of attacker activity is happening, reflected in the rapid innovation of new attacker techniques and tools.

$250,000	$3,000	$1,000	$15
Google Chrome bug bounty RCE	IAB-provided IdP admin account	1-year phish kit rental	Stolen cred bulk list

Push Security

Browser attacks 101: Methods, objectives, and attack paths

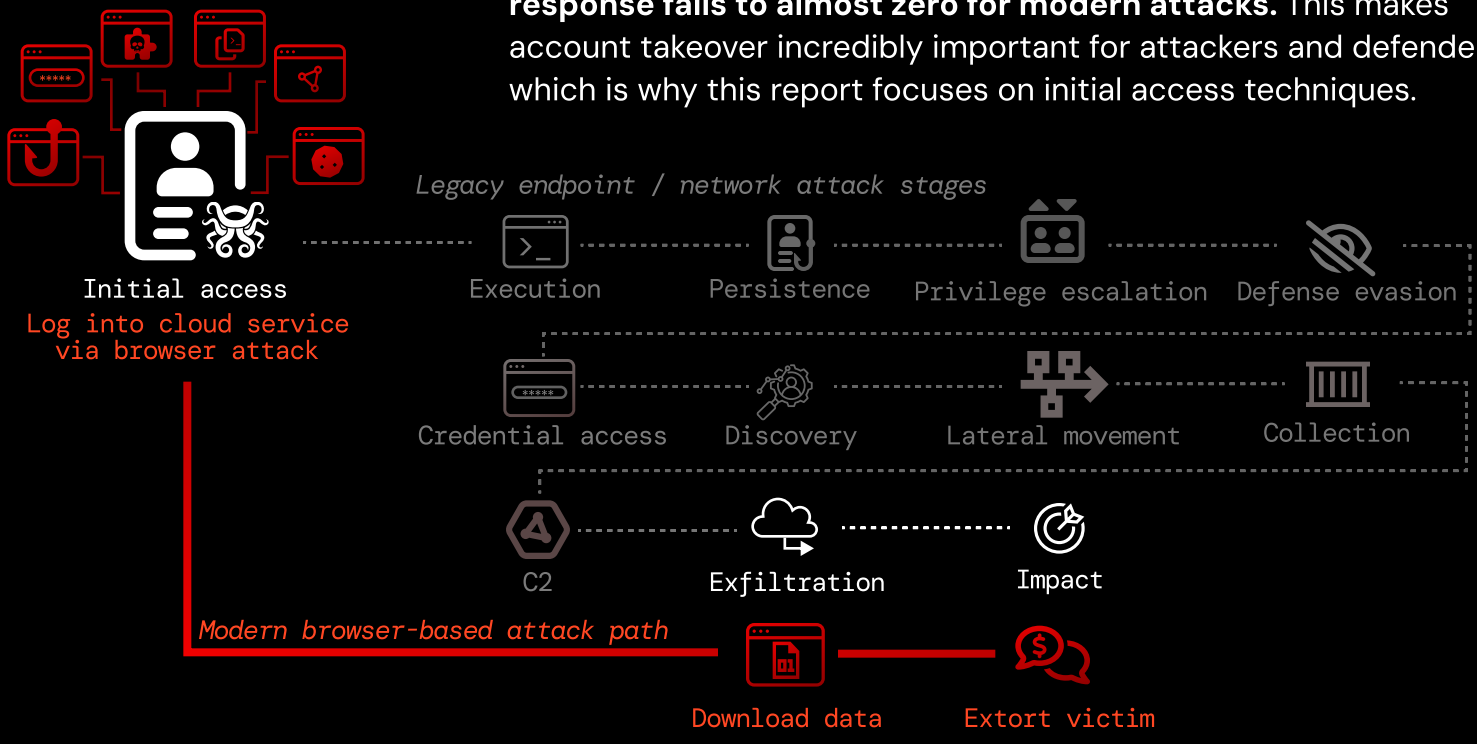
Attackers in 2026 are using a wide (and growing) range of browser-based techniques to achieve a common goal: **compromise cloud applications and services** accessed over the internet, and ultimately profit from data theft, disruption, and extortion. **This is now the primary attack path traversed in the wild today, as seen in the Snowflake and Salesforce mass customer breaches, the attacks on Marks & Spencer and Jaguar Land Rover, and many, many more.**



The modern kill chain looks very different

With many attacks happening entirely over the internet, the kill chain has shifted. You can quickly jump from initial access to a full compromise (often automating the attack entirely). Most breaches today involve the attacker accessing apps and services, and dumping sensitive data for resale and extortion.

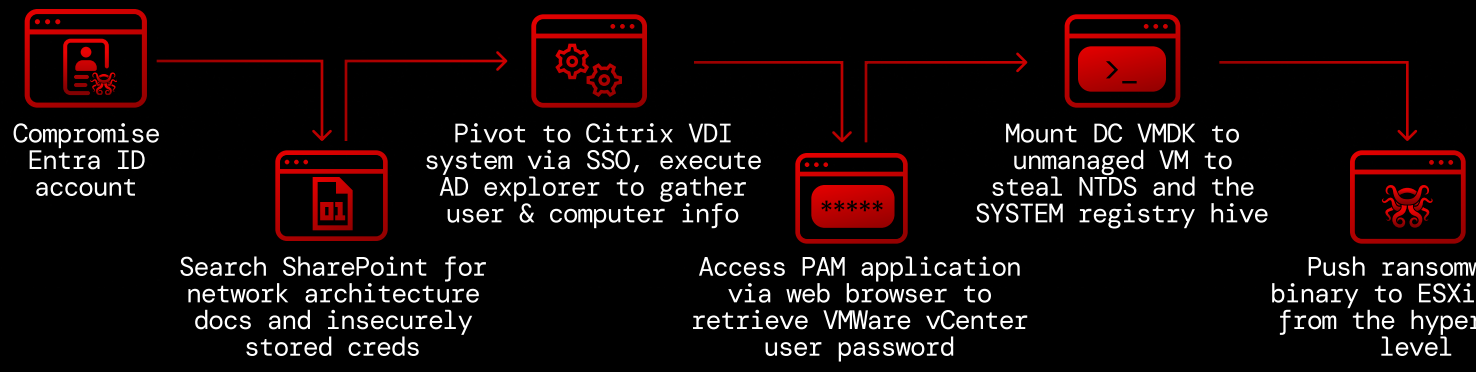
Attacks can go from beginning to end within minutes of gaining access to an account — **meaning the window for detection and response falls to almost zero for modern attacks.** This makes account takeover incredibly important for attackers and defenders, which is why this report focuses on initial access techniques.



29 minutes

average e-crime "breakout time" (time to move from initial access to high value assets) in 2025, *CrowdStrike*

When an attacker does choose to deploy ransomware for extra leverage when extorting a victim, this too is happening over the internet, **from the initial foothold of a cloud account** — finding ways to execute outside the remit of typical anti-ransomware controls.



Typical "Scattered Lapsus$ Hunters" ransomware attack chain. Breakout time = 3 hours. Sources: Google, CrowdStrike

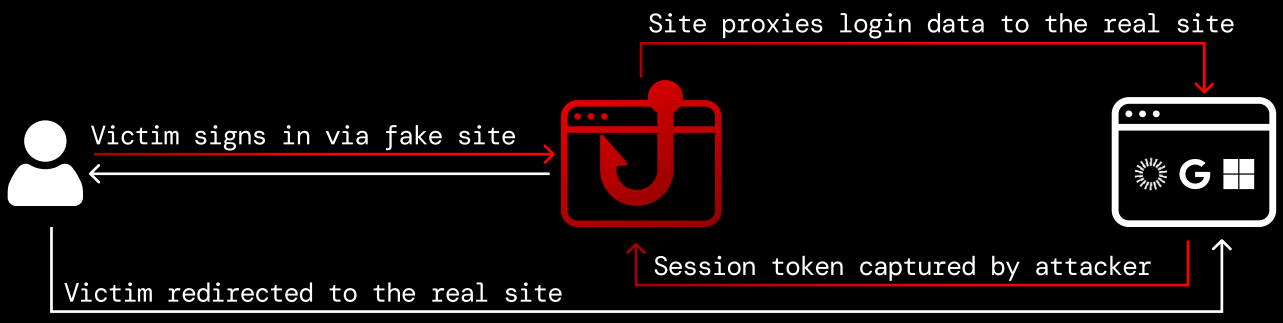
Push Security

TTP breakdown: Adversary-in-the-Middle (AitM) attacks

MFA-bypassing AitM attacks are the standard phishing method today.

AitM is a reverse-proxy based technique that sees the attacker relay traffic from a phishing site to the real site in real time. This means that as the victim enters their credentials, passes MFA checks, and receives a session token from the app, they are all intercepted by the attacker. After, the victim is often redirected to the real site as expected, reducing the risk of suspicion.

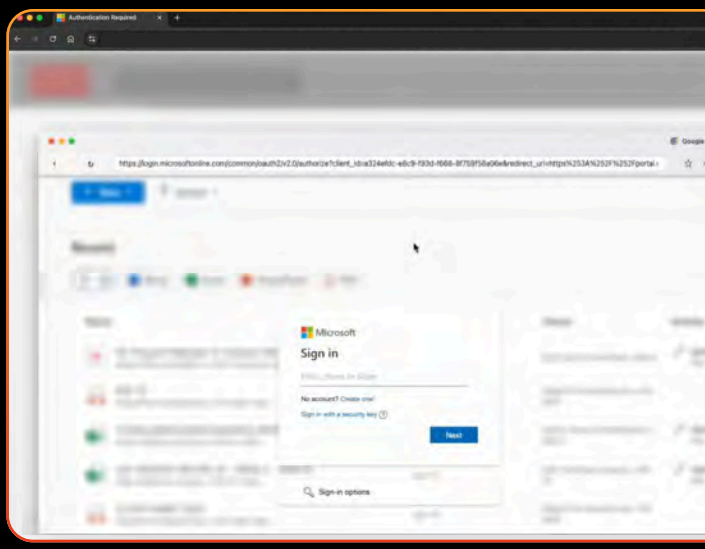
Attackers typically target enterprise cloud and identity provider accounts like Google, Microsoft Entra, and Okta. Not only does this provide access to the suite of enterprise apps and data, but it also allows them to hijack SSO to pivot to downstream SaaS too, increasing the blast radius.



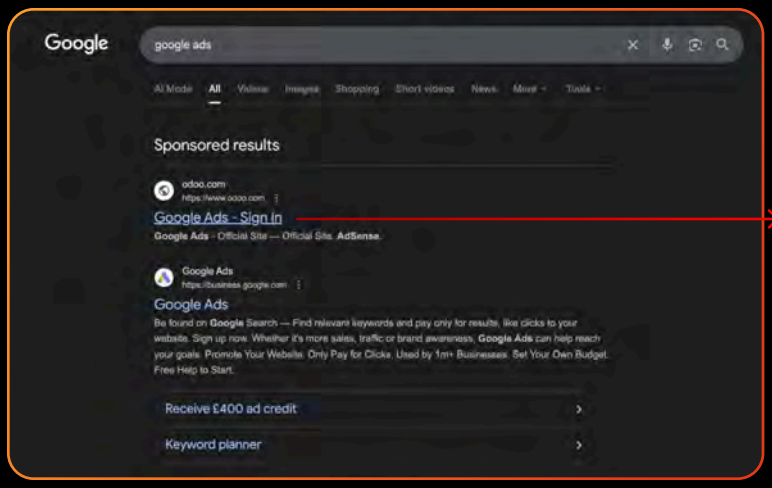
AitM attacks are powered by Phishing-as-Service (PhaaS) kits that attackers can rent on criminal marketplaces. Commonly, we see Tycoon 2FA (59% of detections!) followed by Sneaky2FA, FlowerStorm, Evilginx (*nominally a red team tool, but widely abused by attackers*) NakedPages, Gabagool, and more.

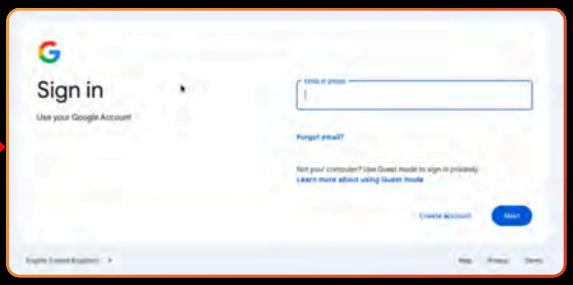
These kits significantly lower the barrier to entry for attackers by adding features and capabilities over time that impersonate new sites, add new detection evasion methods, and generally enhance social engineering with improved look-and-feel.

As mentioned earlier, attacks are also delivered across multiple channels, not just email — with social media apps like LinkedIn, and search engines like Google Search particularly common.



Sneaky2FA phishing kit example detected by Push, featuring a Browser-in-the-Browser pop-up window with a legit-looking URL



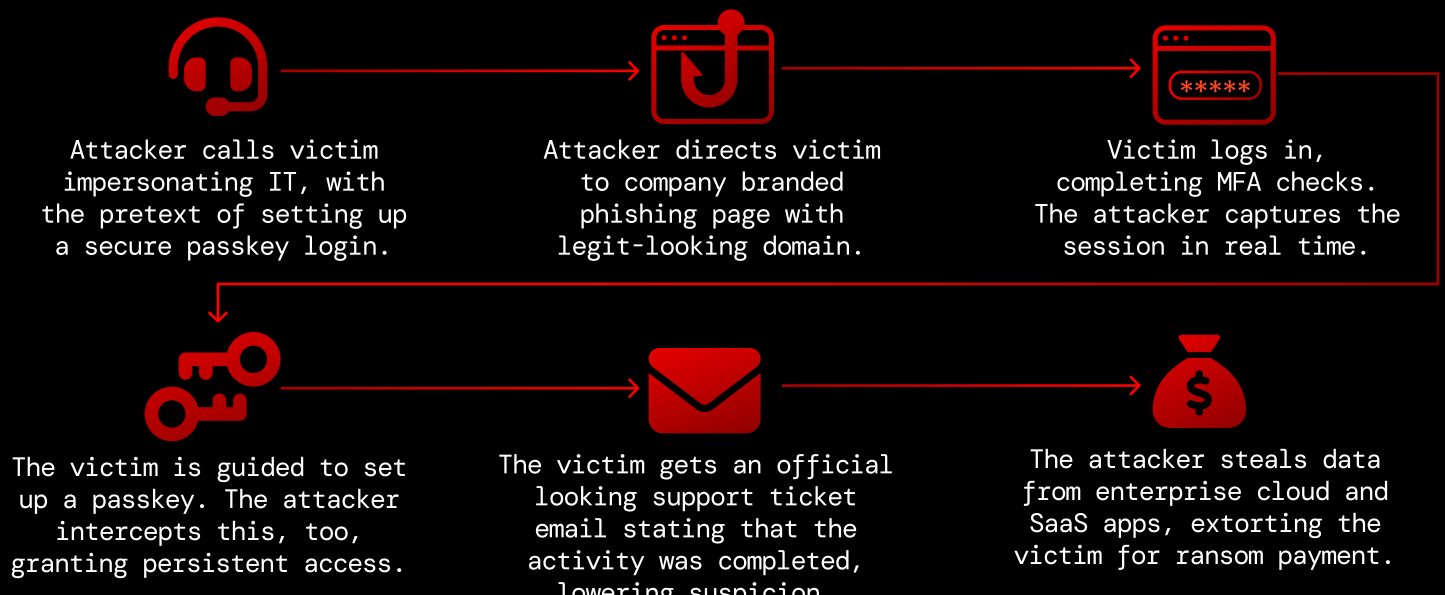


Malvertising attack tricking victims into clicking a fake Google Ad for ... Google Ads — stealing their account via AITM phishing

Case study: How modern attackers are leveraging AitM in highly targeted attacks

Scattered Lapsus$ Hunters, a cybercriminal collective that is an amalgamation of several groups, are known for their use of identity-based initial access techniques. They'll come up a lot as you progress through this report!

Their latest campaign, targeting over 100 companies and claiming tens of millions of stolen records across victims like Betterment, Crunchbase, SoundCloud, and Match Group (so far), is powered by a new kind of criminal AitM kit that is designed to be operated by a human in real time.



This attack is a great example of both **non-email payload delivery** and an **incredibly direct attack path that targets account takeover and data theft from cloud services**. And because the phishing domains will only load if activated by an attacker in real time, it's hard for security teams to proactively find, analyze, and block them.

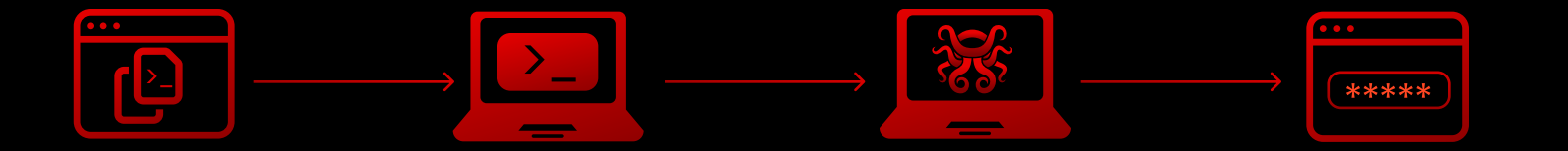
Push Security

TTP breakdown: ClickFix and *Fix variants

ClickFix, FileFix, CrashFix, ConsentFix ... *Fix attacks continue to evolve.

Since late 2024, attackers have been tricking users into performing malicious actions under the pretext of "fixing" an issue for the webpage to be able to load. The most common scenarios relate to "verifying that you are human," styled as a version of the bot protection challenges that we're all used to encountering on the internet today.

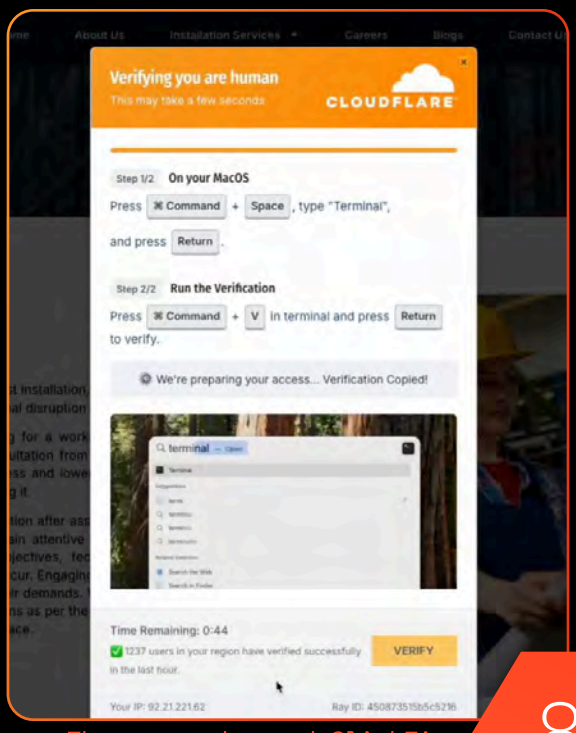
Traditional ClickFix-style attacks are a hybrid of browser and endpoint targeting. While delivered via the browser, the user is required to copy and run malicious scripts on their endpoint, targeting a wide range of legitimate, pre-installed system tools that allow commands to be run, aka Living Off the Land Binaries (LOLBins). This results in the user installing malicious software on their machine — typically Remote Access Tools (RATs) and infostealer malware.



ClickFix really took off in 2025. It was the most common initial access vector reported by Microsoft (47% of attacks!) while CrowdStrike reported that fake CAPTCHA lures were the most common kind of malware download they identified, increasing 563%.

ClickFix is an effective malware delivery vector because it's both fileless and executed by the user, as opposed to a program — and therefore less likely to set off EDR. Combined with alternative delivery mechanisms and some extremely slick lures (*like the one pictured, complete with video instructions and countdown timer*), it's a highly effective social engineering mechanism too.

4 in 5 **ClickFix payloads intercepted by Push were accessed via search engines as the result of malvertising or infected webpages.**

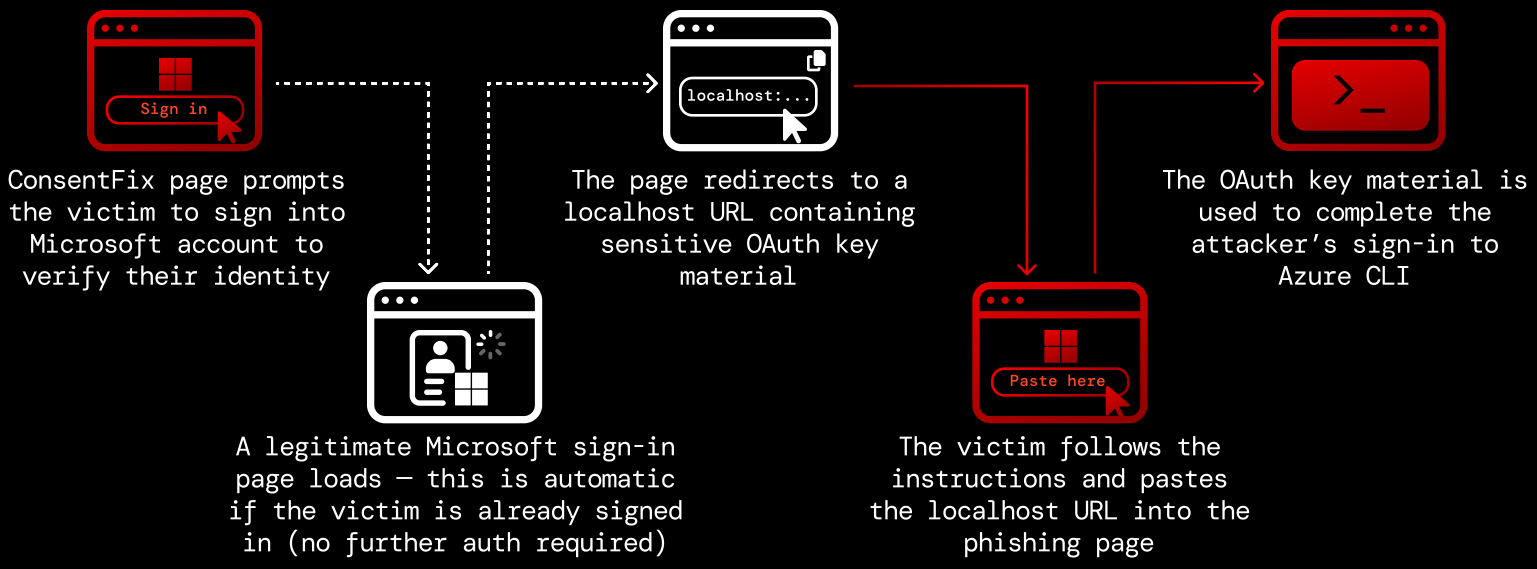


The most advanced ClickFix lure intercepted by Push

ConsentFix: browser-native ClickFix by Russian APTs

In late 2025, Push researchers discovered a brand new ClickFix-style technique that we called ConsentFix. This technique merged ClickFix-style social engineering and a "verify that you are human" page with a fully browser-based attack path, compromising user accounts with a simple copy-and-paste action from a legitimate Microsoft webpage into a phishing site.

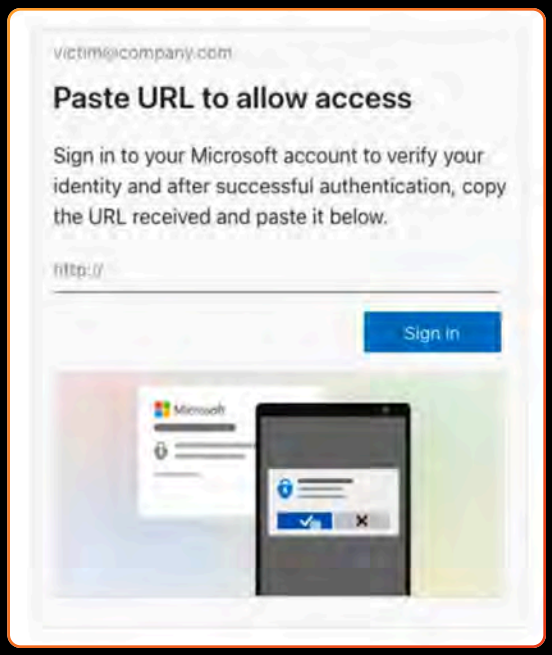
In reality, the victim was unknowingly copying sensitive OAuth key material that would allow an attacker to take control of their account via Azure CLI.



This attack was traced back to a sophisticated campaign **linked to Russian state-affiliated APT29**, involving numerous compromised websites which the ConsentFix payload had been stealthily injected into.

The phishing page fingerprinted your IP and browser so it would only trigger once for a visitor across every site involved in the campaign. It also performed conditional access checks based on the email address provided, refusing to trigger the payload and allowing you to use the site as normal unless the required conditions were met. **This allowed the campaign to fly under the radar for months until Push detected and blocked it.**

By going fully browser-native, ConsentFix overcomes the key weakness of ClickFix-style attacks in that they must also bypass endpoint-layer controls. We should expect to see more browser-native ClickFix variants in the future.



ConsentFix lure

Push Security

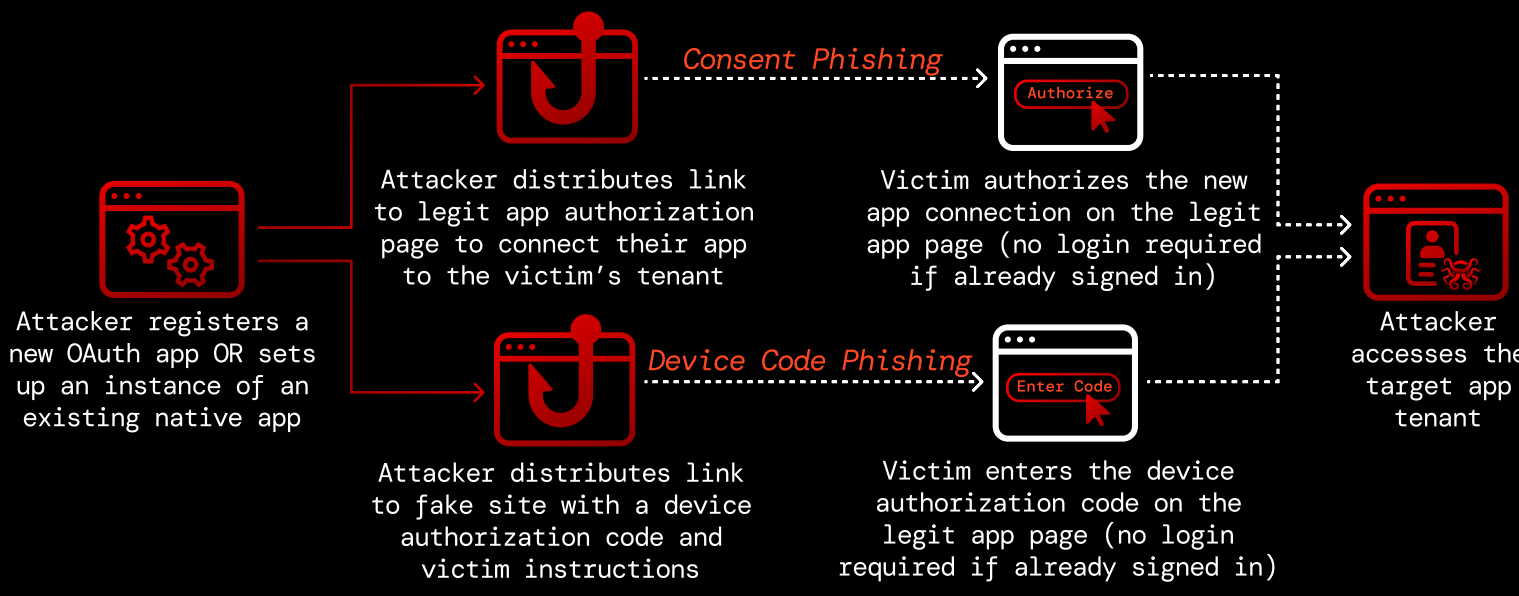
TTP breakdown: Malicious OAuth app integrations

OAuth attacks are on the rise, bypassing the standard authentication flow to hijack apps

Attackers are increasingly launching phishing campaigns via malicious OAuth consent grants. This involves an attacker establishing a malicious app connection to a business app tenant. Historically, these attacks have focused on enterprise cloud platforms like Microsoft 365 and Google Workspace, but also increasingly SaaS services like GitHub and Salesforce.

In some cases, the attacker will create an entirely new app. In others, the attacker will simply set up a new instance of an existing app that is allowed by default in the target application (*similar to how ConsentFix abused first-party Microsoft apps*).

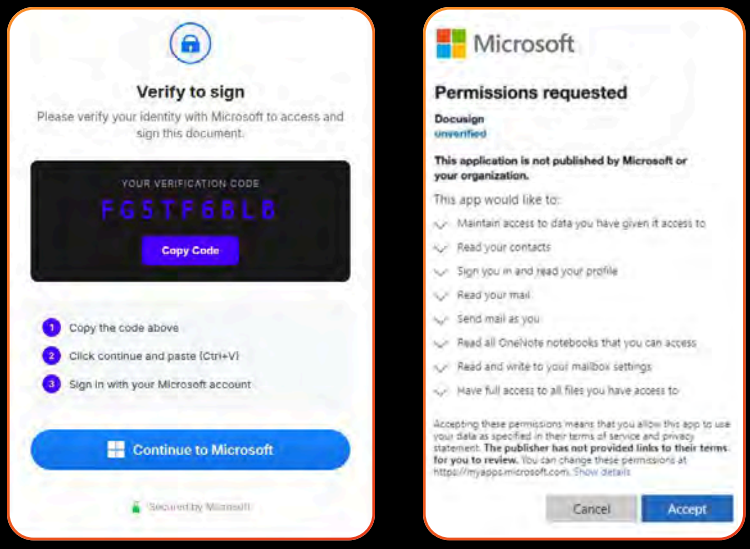
This is achieved through two main phishing techniques: **consent phishing** (*authorizing a third-party app via an OAuth consent grant*) and **device code phishing** (*authorizing a third or first-party app via the device code authorization flow*).



This grants the attacker access to the target app without needing to authenticate in the normal way, such as with an email, password, and MFA factor. This means that normal login protections are circumvented, enabling them to breach accounts that are otherwise difficult to compromise (e.g. those using phishing-resistant auth like passkeys).

Like ClickFix, malicious OAuth attacks aren't a phishing scenario that users are trained or prepared to deal with.

Ultimately, it doesn't feel like you're doing anything as risky as giving up your password or downloading a file — you're simply clicking a button or entering a code on a legitimate webpage!



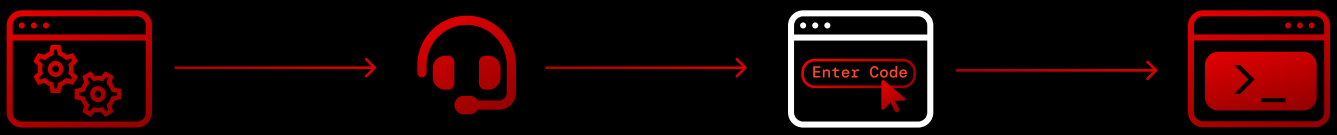
Lures for device code phishing (left) and consent phishing (right).

Case study: Mass Salesforce breaches via device code phishing

Scattered Lapsus$ Hunters (*we told you they would be coming up a lot*) conducted a massive campaign against Salesforce customers in 2025 that resulted in a claimed **1000+ organizations compromised**, with **1.5 billion records stolen and used to extort the victims en masse** (including attempts to extort Salesforce directly).

Similar to the more recent AitM phishing campaign, the attacker combined voice-based social engineering with a browser-based payload, to great effect.

The attacker saw the victim unknowingly connect a malicious "DataLoader" app into their Salesforce tenant (a fake version of a legitimate app) that was configured to request broad OAuth scopes, including full access to the Salesforce API, and the ability to generate refresh tokens without re-authentication.



Attacker registered a malicious Salesforce app called "DataLoader"

Attacker called victims impersonating IT, with the pretext of installing the app

Attacker talked the victim through opening Salesforce and authorizing the new app

Data and secrets were mass exfiltrated from Salesforce via API

By targeting Salesforce directly, the attacker took advantage of limited admin controls and wider default permissions when it comes to OAuth consent grants compared to other enterprise cloud platforms.

Push Security

TTP breakdown: Malicious browser extensions

Attackers are doubling down on malicious browser extensions as their method of choice.

Attackers are increasingly using malicious browser extensions to steal data, intercept credentials and tokens as they transit the browser, log keystrokes, and more.

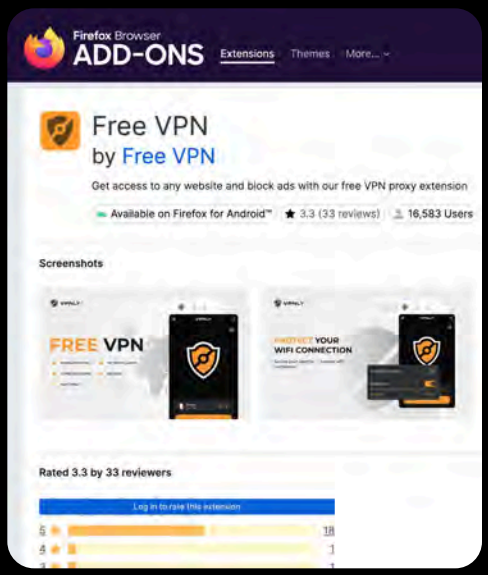
Most malicious extensions didn't start that way. Attackers often begin with a legitimate extension. Then, they bide their time, waiting for the right moment to deploy a malicious update. This usually means waiting until a certain install count is reached to achieve maximum impact.



This makes malicious extensions an incredibly scalable method that enables attackers to compromise millions of users with a single malicious update.

Permissions alone aren't a good indicator of risk — pretty much every extension has permissions that can be exploited by an attacker, even legitimate ones. The most dangerous allow sensitive data, credentials, and session tokens to be intercepted as they transit the browser.

And by using dynamically compiled, stealthily smuggled code that can't be reliably spotted through static code checks or sandbox analysis, a steady stream of malicious extensions are bypassing security checks and making it into extension stores — often even achieving "Featured" or "Verified" status.



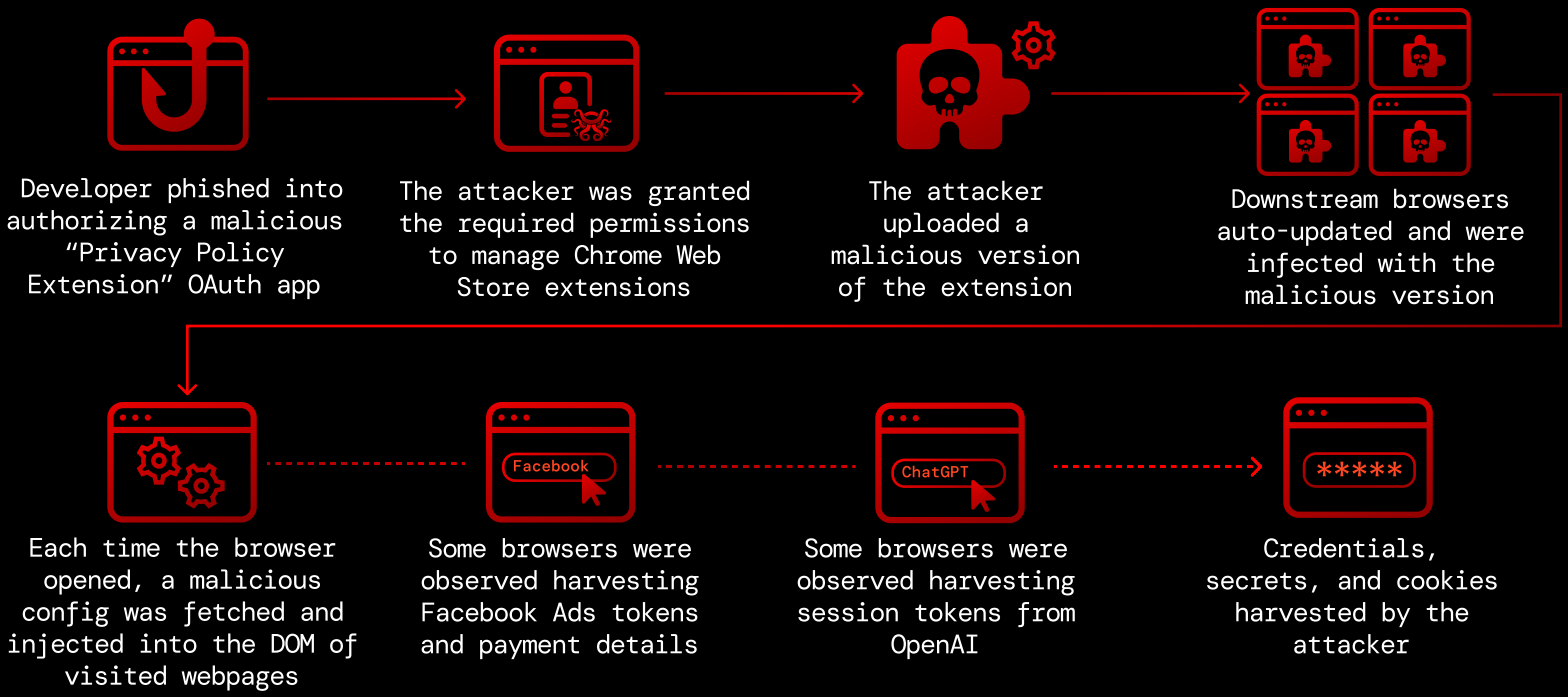
FireFox page for "FreeVPN", recently included in the GhostPoster campaign of 890k installs across 34 extensions

Case study: Cyberhaven extension compromise

In December 2024, attackers compromised the extension of Cyberhaven. The attack was part of a broader, month-long campaign targeting multiple extensions, impacting 2.6m users in total.

The attack was notable for the method used to take over the extension (phishing an extension developer via a Consent Phishing lure generated from a legit Google domain, a technique we discussed earlier) and the stealthy nature of the malicious update.

Every time a browser with the infected extension was launched, it requested a configuration from a remote server. This meant that the attacker could issue different configuration files to each browser. When specific pages were visited, malicious code would execute.



By dynamically loading malicious configurations from a remote server, the exact nature of the malicious code could not be identified just by analyzing extension code itself. At the same time, varying the payload and only triggering on certain pages meant that it was difficult to analyze a consistent set of malicious behaviors to support detection.

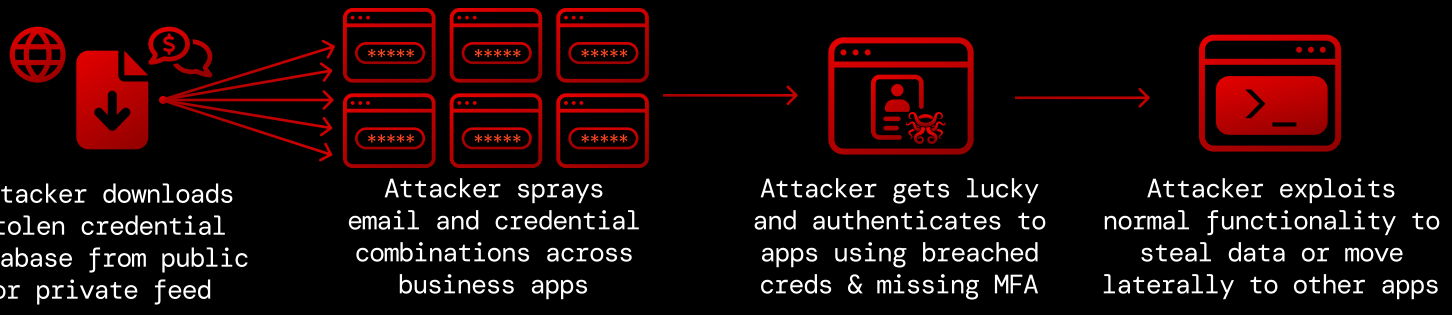
This is a great example of how malicious extensions evade both static and dynamic analysis checks. Another recent campaign dubbed "GhostPoster" (890k installs across 34 extensions) was similarly identified waiting 48 hours between check-ins, and only loading a payload 10% of the time.

Push Security

TTP breakdown: Credential stuffing (& ghost logins)

Credential stuffing is one of the oldest tricks in the book — and it works better than ever.

Attackers have been abusing password vulnerabilities for decades. But password security issues remain one of the leading causes of cyber breaches. In fact, it's arguably worse than ever in the context of sprawling SaaS and cloud services. Our data shows us that users are logging into **tens**, sometimes **hundreds** of apps *each*. And there are now **billions** of stolen credentials available on the internet, fuelled by a steady stream of infostealer infections and data breaches.



Attacker downloads stolen credential database from public or private feed

Attacker sprays email and credential combinations across business apps

Attacker gets lucky and authenticates to apps using breached creds & missing MFA

Attacker exploits normal functionality to steal data or move laterally to other apps

You're probably thinking — *but our users log in via single sign on (SSO), and those accounts are all protected by MFA, right?* **The reality might surprise you. Of the last million logins we saw:**



1 in 4 were password logins, not SSO



2 in 5 were not protected by MFA



1 in 5 used a weak, breached, or reused password

There are many reasons that single-factor password logins haven't disappeared. Sometimes, an SSO login just isn't possible. Apps often charge extra for the privilege of SAML SSO (if they offer it at all). And even when it is supported, an admin needs to be able to configure it. If it's been self-adopted, or IT hasn't gotten to it yet, it's likely that alternative login methods will creep in.

Finally, the vast majority of apps support simultaneous logins by default — while many don't allow login methods to be restricted. This leads to what we call **ghost logins.**

Ghost logins — backdoor auth that sits outside your SSO

Ghost logins are backup logins outside of SSO that you can't see just from looking at your IdP logs.

Ghost logins are created through the normal app adoption process. Many apps historically have required you to set up a credential when you first register an account. Even if you begin to use SSO to log in later, this credential continues to be valid unless specifically disabled or removed.

Because many organizations focus on enforcing MFA at the IdP layer, local app configuration can be overlooked. When you consider that many apps don't allow you to centrally view MFA status, or enforce mandatory MFA for users in your tenant, gaps can go undetected until they're exploited by an attacker.



Case study: Snowflake, 2024's landmark cyber breach

ShinyHunters (now part of the collective known as Scattered Lapsus$ Hunters) breached more than 165 organizations using stolen credentials, logging into their Snowflake tenants and mass-dumping the data via SQL commands directly executed in the Snowflake platform.

80% of the compromised accounts had prior breach exposure, found in compromised credential datasets dating back as far as 2020.

In total, more than 1 billion records were stolen from just 9 publicly named victims. The real impact will be far greater.

The Snowflake breach demonstrates how little control or visibility organizations have over downstream SaaS logins. Not only did these accounts lack MFA — they were still using credentials that had been available online for years. The stolen credentials had not been identified, and the accounts had not had their credentials rotated since they were initially leaked online.

*The Snowflake breach was powered by previous data breaches and infostealer infections. Next, we'll cover how another by-product of infostealer breaches can be leveraged by attackers — **stolen session cookies.***

Push Security

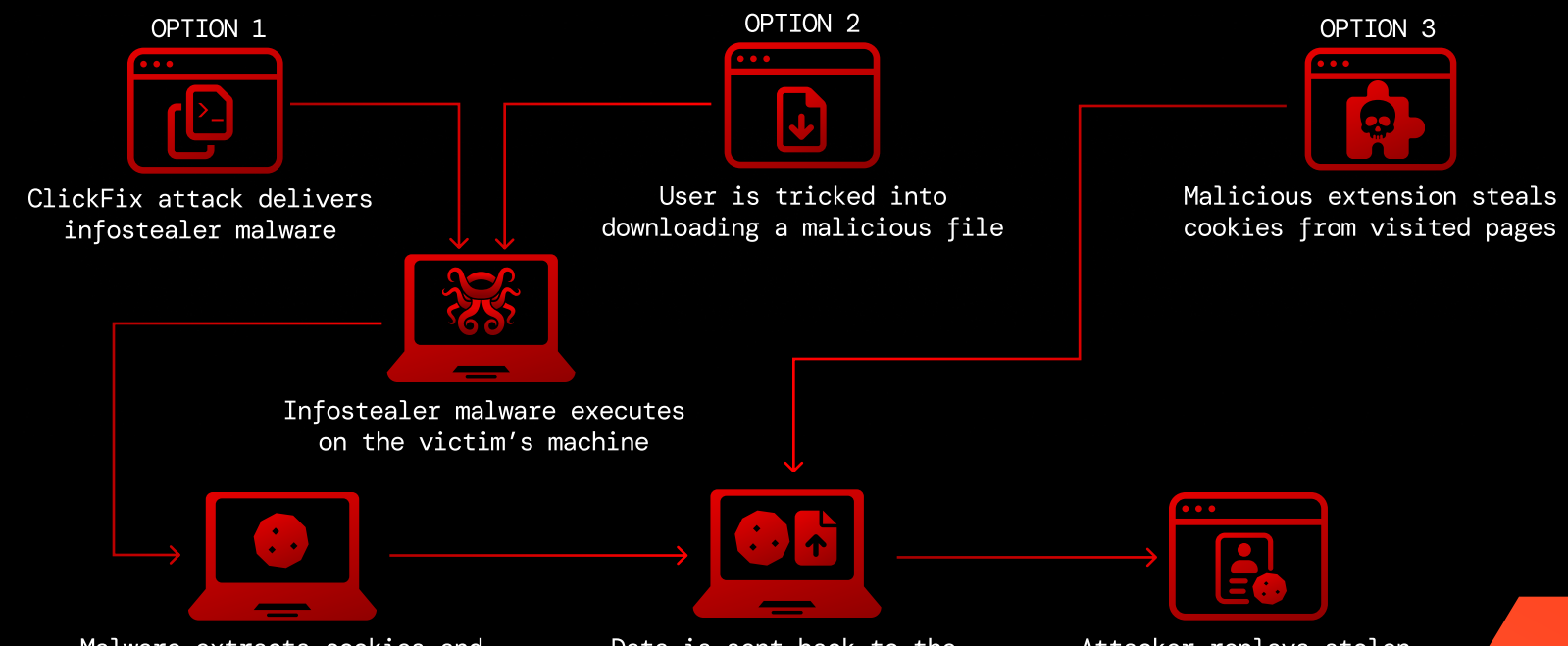
TTP breakdown: Session hijacking (aka token replay)

Sidestepping authentication controls by replaying stolen session tokens.

Session hijacking attacks allow attackers to bypass the authentication process by taking an already-approved session token that they've stolen from the victim's device or browser, and reusing it in their own browser. **This enables them to get around even phishing-resistant authentication controls like passkeys.** This is different to AITM attacks, which see a new session created via the attacker's reverse-proxy connection to the target app.

Sessions can be stolen using a variety of methods, some of which we've already discussed. Malicious browser extensions can extract them from webpages visited by the user, for example. But the most prominent source of stolen tokens is **infostealer malware** (*also the leading source of stolen credentials powering credential stuffing attacks*).

As mentioned previously, **ClickFix is now the go-to method for delivering malware like infostealers**. ClickFix is more detection resistant than a normal file download, which are more likely to be intercepted and analysed by controls like a web sandbox before hitting the endpoint, and are more likely to trigger endpoint alarms during execution.

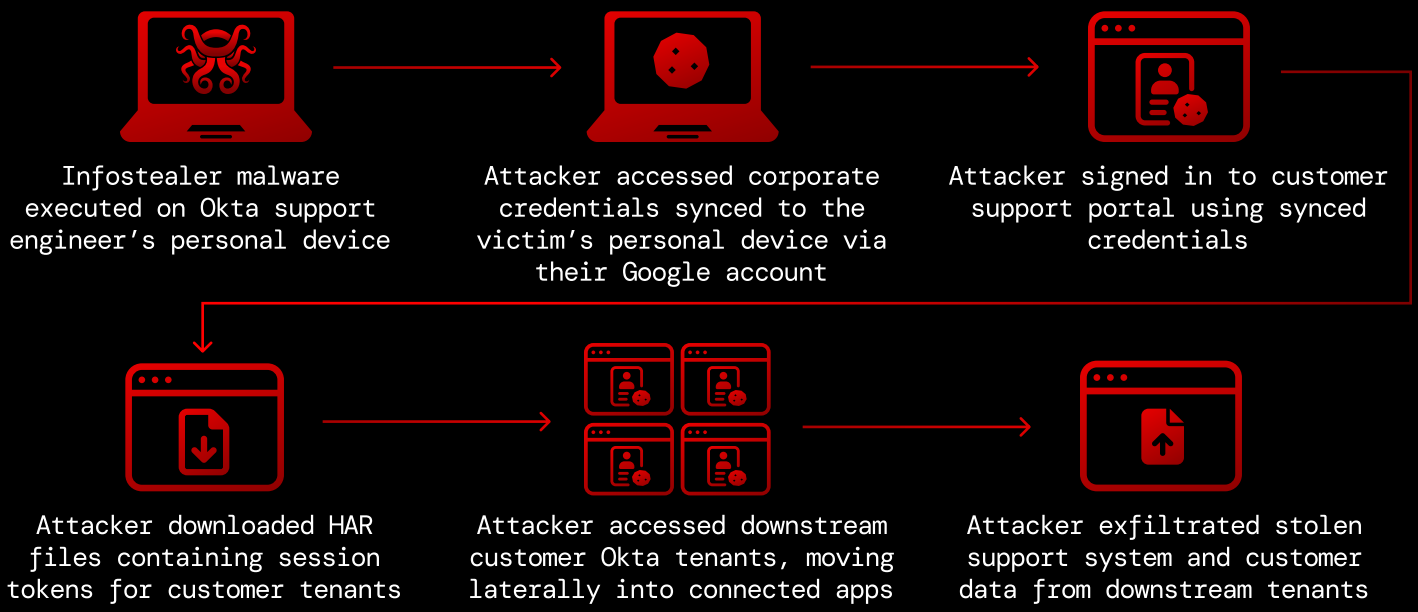


Case study: Okta session theft leads to customer compromise

In late 2023, Okta disclosed a security incident in which one of their support engineers had been compromised with infostealer malware. This allowed the attacker (reportedly Scattered Spider, again) to access Okta's customer support system, where sensitive files could be accessed containing session tokens. In turn, these tokens were replayed to access customer environments.

In total, 134 customers were impacted, with the attacker exfiltrating a report containing the full names, email addresses, and access tokens of all 18,400 support users (i.e. customers who used the case management system).

BeyondTrust, 1Password, and Cloudflare all reported further attacker activity, with Cloudflare in particular impacted by the attacker accessing their internal Atlassian server, including their Confluence wiki, Jira database, and source code in Bitbucket.



Okta later identified that an employee had signed in to their personal Google profile on the Chrome browser of their Okta managed laptop.

The username and password of the service account had been saved into the employee's personal Google account. When their personal device or account was compromised, their corporate credentials synced to the account were also stolen. This shows how sensitive business data and credentials can easily transit personal and managed devices through features like profile syncing, which are easy to enable, but hard for most organizations to track.

Push Security

How browser-based attacks bypass traditional controls

Browser-based attacks are so effective because they find ways around many traditional control points and security tools.





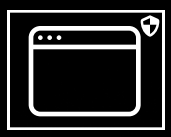
Email

- Deliver over non-email channel
- Send emails from trusted apps
- Don't use known-bad domains — constant rotation



Secure Web Gateway

- Evade malicious download detections through HTML smuggling
- Client-side reassembly of complex encoded web pages bypasses traffic signatures
- Rapid domain rotation effectively bypasses domain blocklists



Remote Browser Isolation

- RBI is designed to protect against exploits, not attacks happening inside the browser e.g. account takeover



File Sandbox

- Evade malicious download detections through HTML smuggling, or fileless malware delivery methods like ClickFix



Endpoint Security

- Most attacks now don't touch the endpoint
- When they do, methods like ClickFix enable stealthy malware execution



Identity & Authentication

- 2 in 5 logins not protected by MFA, "ghost logins" circumvent SSO
- Downgrade attacks can circumvent phishing-resistant auth
- Many attacks are post-auth, meaning they succeed regardless of auth strength



Cloud Security

- In-app exploitation blends in with normal user activity
- Cloud logging is inconsistent from app to app, creating visibility gaps

This isn't to throw shade at these controls — but most of them weren't designed with browser-based attacks in mind. And as any red teamer knows, just because something says it counters an attack on paper, doesn't mean there aren't ways around it in practice. This kind of "list thinking" can be problematic when it comes to defensive controls. It's important that blue teamers re-evaluate whether their tools are providing the protection they thought in the new context of browser-based attacks.

Push Security

Threat actor overview: Scattered Lapsus$ Hunters

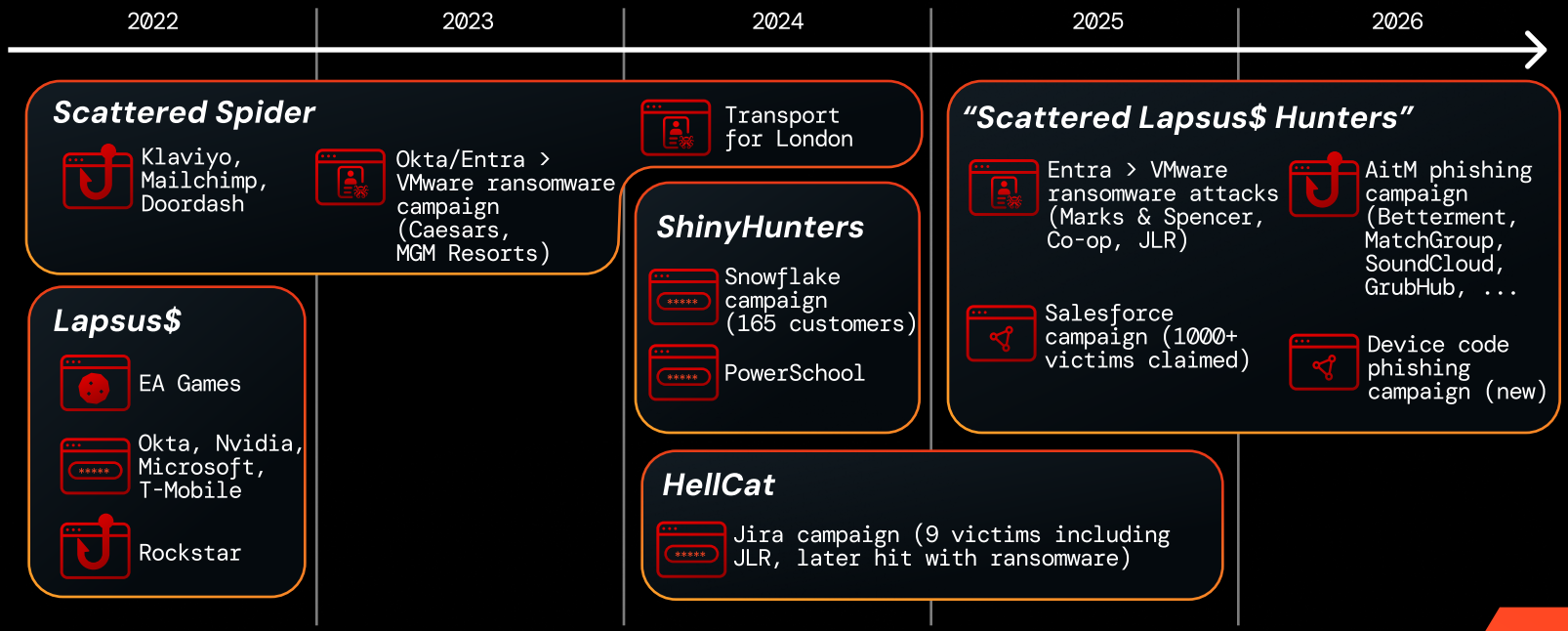
We've mentioned **Scattered Lapsus$ Hunters** a lot throughout this piece. This group has been particularly quick to capitalize on browser-based methods and is a referenceable example of a modern threat actor based on how public their attacks have been.

They are known primarily for targeting cloud services directly over the internet — often leveraging voice-based social engineering in combination with a browser-based payload.

Since 2024, there's been a clear shift to broader campaigns centred around varied payloads, with a willingness to try out new techniques. This variety is exampled in the mass-targeting of Snowflake and Jira tenants with stolen credentials, aiming at Salesforce with a device code phishing lure, launching a mass phishing campaign using a new real-time operated AitM kit, and now recent reports of a new Microsoft-focused device code phishing campaign.

Scattered Lapsus$ Hunters is best understood as a collective affiliated with "the Com", a community of English-speaking cybercriminals with international criminal affiliations. They are also known to collaborate with a range of cybercrime "as-a-Service" vendors for phishing, initial access, ransomware, and more.

The group is so-called because they are a merger of Scattered Spider, Lapsus$, and ShinyHunters members. This name is meant to be intentionally tongue-in-cheek (*after all, they didn't call themselves Scattered Spider, CrowdStrike did*).



Scattered Lapsus$ Hunters & precursor groups — public breaches and campaigns.
Check out our blog post for the full breakdown.

Push Security

Detecting and responding to browser-based attacks

Defenders need to meet attackers where they actually operate — inside the browser.

Push Security was designed specifically to stop browser-based attacks. Our browser security platform enables attacks to be detected and blocked in real-time, while providing deep browser telemetry to support investigations and proactively discover vulnerabilities before they can be exploited by attackers. **Think of Push as being like EDR, but for your browser.**

Using Push, you can counter browser-based attacks by:

- Detecting and stopping phishing attacks by intercepting malicious pages and behaviors in real time, no matter where a link got clicked.
- Remediating SSO gaps, missing MFA, and weak, reused, and compromised passwords currently being used, across every app and login (even Shadow apps).
- Blocking browser-based malware delivery like ClickFix, FileFix, as well as browser-native, malware-free variants like ConsentFix.
- Injecting a unique marker into sessions that originated in Push-protected browsers, alerting when they're used in a different browser.
- Blocking known-malicious extensions from running, and removing risky extensions across any browser in your environment.
- Manage and remove malicious OAuth apps, and deliver real-time warnings to users when making a consent grant to connect an OAuth app.

Push Security is like EDR, but for your browser.

Geoff Belknap

Deputy CISO at Microsoft
Former LinkedIn, Slack, Palantir



Built by practitioners, powered by research

Push Security was founded by veteran red and blue teamers after we observed attackers shift away from targeting users on their endpoints and instead target them in their browsers.

Realizing security teams lacked the visibility and control to counter this threat, we set our mission: to build **the most advanced security tool in the browser**.

As a research-led organization, our work is driven by real-world attacker behavior and anticipating what comes next. Partnering with Push gives you a continuously evolving, threat-informed capability that keeps you two steps ahead of attackers.

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, visit our demo library, or book some time with one of our team for a live demo.

Trusted by security teams doing serious work



Josh Lemos
CISO

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 ramp Sainsbury's