Push Security

The Buyer's Guide to Browser Security

The browser is where modern work happens — and where modern attacks play out. This guide gives security teams a practical framework for evaluating browser security solutions: defining requirements, testing vendors, and building the business case.

Contents

01 Why browser security, why now 03
Where work and AI happen · Where attackers strike · The browser attack era · Market momentum

02 Why buy anything new? Why existing tools can't stretch 06
SWGs, CASBs & SASE · EDR · Email security · IdPs, MFA & passkeys · SaaS and cloud logs · The common thread

03 The browser security market landscape 11
Different buyers, different architectures · Full-stack browsers · Managed browsers · Browser extensions · The acquisition wave

04 Defining your use case requirements 15
Ten use cases in three tiers · The threat model trap · Use case requirements · Tier 3 and exclusions

05 Shortlisting, evaluating, and testing solutions 28
Shortlisting · The platform pitfall · 10 evaluation criteria · Running a proof of concept

06 Comparing solutions 36
The comparison framework · Vendor-by-vendor analysis · Putting it together

07 Building the business case 44
Five economic value drivers · The board-level argument · Funding routes · Managing investment risk

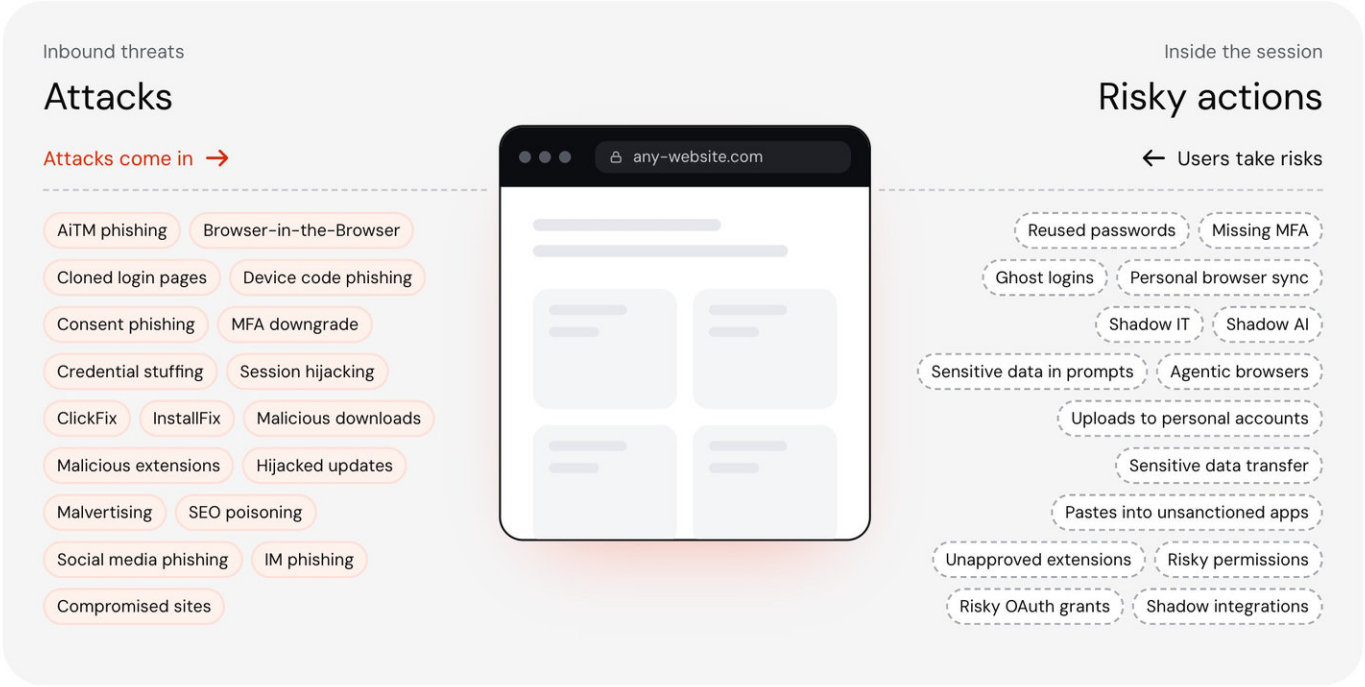
A Appendix: Questions to ask vendors 48
Evaluation questions mapped to each of the ten use cases

01 Why browser security, why now

The browser is where the majority of enterprise work now happens, where AI runs, and where attackers operate. But most security tools still sit outside it. That gap between where users and attackers operate and where the security team can reach is driving preventable breaches and leaving entire categories of risk unmanaged.

Attacks come in through the browser, and data goes out through it. Attackers use the browser to reach your workforce with fake login pages and downloads. Company data leaves through it too, into AI tools and personal accounts.

It's why 88% of organizations now rank browser security among their top five priorities — and why, for most security teams, the browser is the critical gap remaining in their stack.



88%

Top-five priority

of organizations rank browser security among their top five priorities (Omdia)

85%

Of the workday

of employee working time is spent on browser-based activity (Microsoft)

49%

Successful attacks

of organizations suffered a confirmed browser-based attack in the past 12 months (Omdia)

It's where work happens

The applications that run a business are no longer installed on a managed network; they're logged into over the internet, through the browser.

Employees spend around 85% of the workday on browser-based activity across SaaS apps, cloud consoles, messaging apps, and most recently, AI tools.

AI runs in the browser

The vast majority of AI interaction happens in the browser too: Prompts entered, files uploaded to models, agents authorized through OAuth grants. Push data shows the average organization already runs 16 generative-AI apps, 17 AI browser extensions, and 17 AI-related OAuth integrations connected to just Microsoft and Google. AI tools are fast becoming the hub that orchestrates activity across a mesh of cloud apps, and all of it runs through the browser.

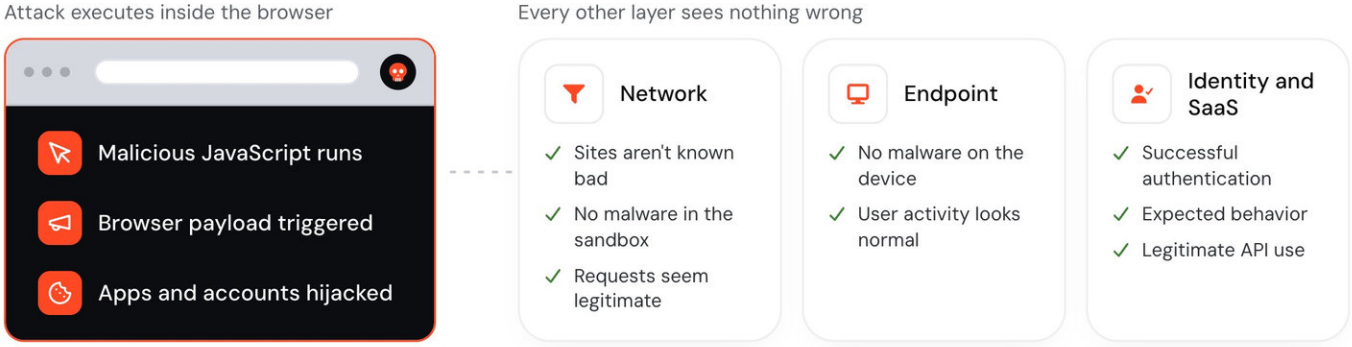
It's where attackers strike

Despite modern work having already migrated into the browser, security has been slow to follow. AI adoption also creates a new opportunity for attackers. Users operate outside the security team's purview and adopt new tools that connect to each other and to company data. The result is a vast attack surface across thousands of web applications. Once attackers compromise those apps, they get the sensitive data and functionality they need to achieve their goals.

Rather than continue targeting the endpoint and network perimeter, both well hardened after years of investment, attackers have adapted their techniques to target users in their browsers. Compromising accounts on cloud applications is now the route of least resistance for achieving the goals they've always had.

This browser-native attack playbook works because it exploits the blind spot left by traditional security tools. Consider an attacker who uses an AiTM phishing kit to steal a session token, deploys a ClickFix lure to manipulate a user, or abuses an OAuth consent flow to gain persistent access. Each of these attacks plays out entirely within the browser security blind spot.

To the rest of the security stack, the traffic looks clean, the authentication looks legitimate, and the cloud access looks normal. These browser-native, identity-first attacks also leave almost no window for detection and response. Attackers can move from account takeover to data exfiltration in minutes (CrowdStrike puts the average breakout time at 29 minutes), often with no lateral movement, no privilege escalation, and next-to-no dwell time.



The browser attack era

The data confirms the scale of the shift. CrowdStrike's 2026 Global Threat Report found that 82% of detections are now malware-free. Verizon's 2026 DBIR found credential abuse in 39% of all breaches, the single most pervasive technique in the dataset. Identity weaknesses played a material role in 89% of Unit 42 investigations, and Google's Mandiant found identity issues were the initial access vector in 83% of cloud-related incidents.

The highest-profile breaches in recent years tell the same story. MGM, Caesars, 2024's Snowflake attacks, Marks & Spencer, Jaguar Land Rover, and the 2025 Salesforce breaches — none of these attacks broke in by exploiting the victim's endpoint or network. Instead, each one hinged on compromised identities for cloud services, obtained through phishing, help desk social engineering, infostealer-harvested credentials, or stolen OAuth tokens. The impact has been devastating: MGM estimated costs at $100 million and later agreed a $45 million class-action settlement covering this and an earlier 2019 breach. Marks & Spencer put the hit to operating profit at around £300 million. The attack on Jaguar Land Rover was severe enough that the UK government guaranteed a £1.5 billion loan to keep its supply chain intact. The disruption was so significant that UK GDP actually contracted in as a result.

The victims of these attacks are all mature enterprises with strong security postures. The problem is that those postures were built for how attackers used to operate, not how they operate today.

Growing market momentum

Security teams are now having to confront this new reality and play catch-up. This is reflected in the fast-emerging browser security market. In Omdia's 2026 research, 88% of organizations ranked browser security among their top five priorities and 86% had already increased their spending, with secure browser tools now accounting for 12.6% of the average security budget. Gartner has also redefined the category around the browsers people already use: It defines a secure enterprise browser as one delivered "through a centrally managed browser extension, and optionally, a full-stack custom web browser."

The market is relatively new and quickly evolving, but buyers are doubling down: 85% expect to increase spending further over the next 12 to 24 months.

> If you're evaluating a browser security investment, it's probably because you've already identified the volume of activity, risk, and attacks that now play out in the browser. You've probably also seen the limits of your existing tools against them. You're in good company: Most organizations that have already invested plan to spend more. The rest of this guide is designed to help you evaluate your options and build the case.

82%

Malware-free

of detections now involve no malware (CrowdStrike)

39%

Identity abuse

of all breaches involve credential abuse (Verizon DBIR 2026)

83%

Initial access

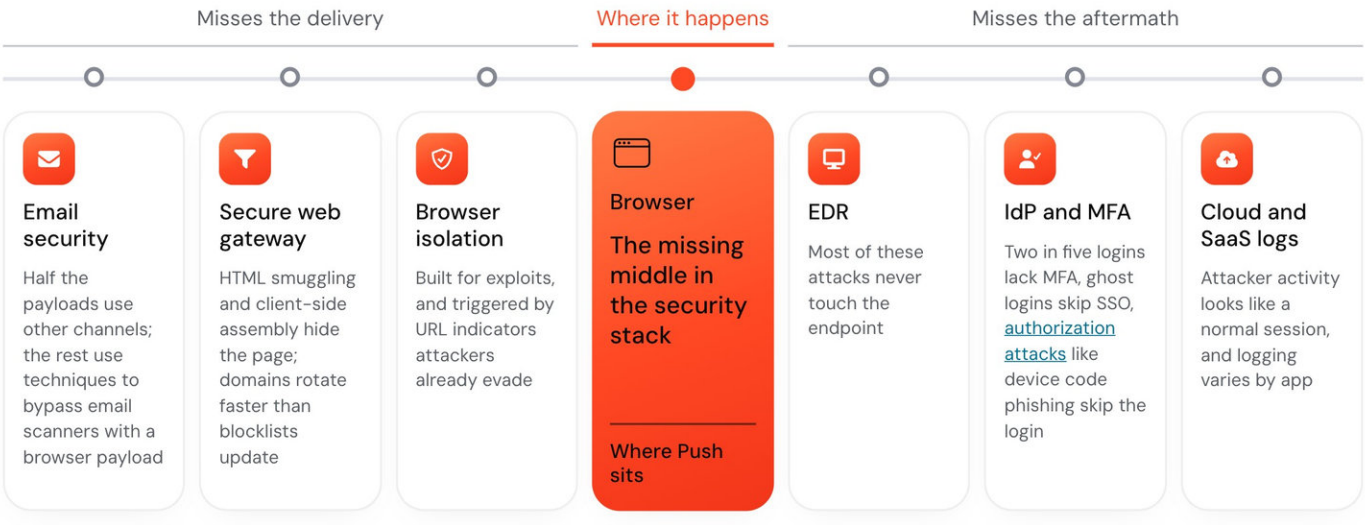
of cloud incidents began with identity issues (Mandiant)

02 Why buy anything new? Why existing tools can't stretch

If you're evaluating browser security solutions, you may have encountered this objection: "Another tool!? We already have a SWG, EDR, email phishing tools, an IdP … can't we just solve these problems with the tools we already have?"

It's a reasonable question — and the response isn't that those tools are failing. They were designed for very different jobs. In fact, they've become effective enough at securing their respective layers that attackers have adapted by rerouting their techniques through the one layer none of those tools were designed to instrument: The browser session.

The gap is structural, not a product shortcoming, and no amount of additional investment in tools operating at the wrong layer will close it. That's what makes browser security a net-new requirement rather than a duplication of existing coverage. This section walks through each category to show where its coverage ends — and how risky user and attacker activity now happens beyond its reach.



SWGs, CASBs, and SASE platforms

Network and proxy-based security tools — SWGs, CASBs, and converged SASE platforms — are effective at what they were designed to do: Securing the wire. They sit inline, inspect traffic, enforce URL filtering, apply DLP policies, and block known-bad destinations. The limitation is that what happens over the wire is only a partial picture of what's truly happening.

Vendors of these solutions still claim to detect and stop these attacks. But the traffic they inspect offers limited visibility, and that limit constrains how well they can do it. Much like traditional anti-virus, they largely rely on matching known-bad domains and URLs against blocklists and reputation feeds, rather than observing how pages behave as they render. Attackers are well aware of this limitation, and they're constantly rotating their infrastructure to stay off blocklists and ahead of defenders.

Modern web applications are complex client-side programs rendered by the browser's JavaScript engine. A significant amount of what happens on the page — DOM manipulation, script execution, credential entry, clipboard operations — never generates traffic that a proxy can observe. When an attacker deploys an AiTM phishing kit, the proxy sees a request to what appears to be a reputable domain, followed by standard web traffic. The phishing infrastructure is hidden behind redirects and conditional logic designed specifically to screen out proxies and automated scanners. From the proxy's perspective, everything looks clean. Inside the browser session, credentials are being intercepted and session tokens harvested in real time.

Solutions relying on traffic analysis are structurally behind the curve, chasing ever-changing blocklists, and struggle to stop the most dangerous attacks.

Some proxies attempt deeper analysis by reconstructing page behavior from decrypted traffic. They run into a fundamental problem: Every web app encodes its HTTP data differently. To analyze what happens inside any given app, the vendor needs custom reverse-engineering that breaks every time the app updates. Even with decoded traffic, correlating specific user actions to specific browser tabs across the volume of disorganized network requests isn't something that can be done reliably or in real time.

The same visibility gap applies to AI governance and data loss prevention. A CASB can categorize and block AI domains, and with TLS inspection it can see what a user eventually submits. What it can't see is what users type or paste into the page before submission, or how they interact with it, because that happens client-side. Telling a corporate account from a personal one on the same AI platform is also brittle from traffic alone, which matters when the goal is to enforce policy rather than block the tool entirely.

Instead of inspecting traffic from the outside, a browser security solution can observe the page's true behavior and how the user interacts with it. Page-level signals — DOM fingerprints, script execution patterns, user interaction metadata — are what make it possible to identify cloned login pages, expose phishing toolkits, and catch credential entry into malicious forms. Those signals only exist inside the browser tab, and traffic metadata can't reconstruct them. Just like we moved from signature-based antivirus to behavioral EDR, the same shift is now needed in the browser.

89%

Short-lived domains

of phishing domains are active for less than two days (Spamhaus, 2025)

6.5%

Survive 15 days or more

by the time a domain is blocklisted, the campaign has moved on

95%

Bot protection

of attacks detected by Push use bot protection to hide malicious content from automated tools

Endpoint detection and response

EDR is the success story that, to a large extent, forced attackers off the endpoint and into the browser. Rather than acquiring EDR-evading malware and spending weeks living-off-the-land, attackers have pivoted to targeting users in the browser, where they leave no trace on the endpoint itself. From the operating system's perspective, the browser process looks the same during normal use as when an attack is happening inside it.

Attackers understand that EDR agents are designed to monitor process execution, file system activity, memory behavior, and registry changes at the OS layer. They can't see which tab is open, what scripts are executing inside the page, whether a login form is legitimate or a proxied clone, or whether a session token is being exfiltrated. The evidence shows attackers are deliberately adapting to work outside EDR's coverage — and these attacks play out over standard web sessions where the host is unaffected as far as EDR is concerned.

Some endpoint vendors acquire browser capabilities, as CrowdStrike did with Seraphic. Even then, they tend to view browser security through the lens of traditional endpoint protection: Detecting browser exploits and sandbox breakouts that could lead to endpoint compromise. That's a meaningful capability, but it's still focused on attacks that ultimately target the endpoint. It doesn't address the modern identity attacks that avoid the endpoint and network altogether, where the browser works exactly as designed and the attacker never triggers an exploit.

Email security and phishing controls

The security industry has historically treated email security and anti-phishing as the same thing, but in recent years, that equivalence has broken down. "Phishing" now covers very different attacks: Malware delivery, business email compromise, and credential phishing.

Credential phishing, specifically the MFA-bypassing AITM variety, is now the dominant form. It doesn't always arrive by email, but it always plays out in the browser. That makes it a browser security problem, not an email security problem.

A phishing lure can arrive via a LinkedIn DM, a malicious Google ad, or a compromised website surfaced in search results. In each case, the SEG sees nothing: No message to quarantine, no sender to block, no way to recall the same lure hitting multiple users. And even for phishing that does arrive via email, the SEG's ability to evaluate the destination page is increasingly compromised. MFA-bypassing AiTM kits are now table stakes, and the latest kits use bot protection, conditional loading, and DOM obfuscation to defeat page inspection.

> The net result is that many attacks bypass email altogether, and even when they're delivered via email, they use detection evasion techniques that exploit the limitations of email scanning tools.

44.6%

AiTM share

of the phishing techniques Microsoft identified were AiTM (Microsoft, 2026)

26%

Fewer password attacks

drop in password attacks year over year, as MFA and passkeys push attackers to other methods (Microsoft, 2026)

1 in 2

Outside email

of phishing payloads intercepted by Push are delivered outside email entirely

4 in 5

Via search

ClickFix pages intercepted by Push arrived via search engines

IdPs, MFA, and passkeys

If you've invested in phishing-resistant authentication, you've addressed the most direct path to credential theft. The problem is that modern attacks increasingly route around the authentication layer entirely rather than trying to break through it. We call these authorization attacks.

Device code phishing is the clearest example. A device code phishing attack can't be prevented with authentication controls, including all forms of MFA and even phishing-resistant methods such as passkeys. The attack targets the authorization layer, not the login. If the user already has an active browser session, they only need to enter a device code and select their account, with no password or MFA challenge. The phishing happens entirely on a legitimate site with no fake login page, no malicious payload, and a genuine URL in the browser bar.

OAuth consent phishing follows the same pattern. **ConsentFix**, first identified by Push in December 2025, hijacks the OAuth authorization flow rather than the authentication flow — passkeys and all other phishing-resistant methods have no impact because the authentication process is never involved. It went from nation-state technique to commoditized criminal toolkit in six weeks.

Beyond post-authentication attacks, IdPs are structurally limited in what they can see. They enforce policies for federated applications, but many logins never reach them. Of the last million logins Push observed, 1 in 4 used a password rather than SSO. The IdP has no visibility into these direct logins.

Ghost logins are legacy local credentials that persist alongside SSO. They give attackers a way around a company's chosen SSO option, and security teams often have no central way to audit or remove them across hundreds of applications.

This SSO gap is especially acute for AI governance. Most AI tools aren't federated, which means the IdP can't see or govern them at all. Push data shows that more than one third of files uploaded go to "shadow" tenants where the employee has logged into a personal account outside the scope of the IdP.

Even where strong authentication is in place, **MFA downgrade attacks** exploit the fact that accounts with phishing-resistant methods remain vulnerable when a less secure backup method is also present. Going fully passwordless across an enterprise's entire application estate would close this gap, but for most organizations that remains an aspiration rather than a reality.

> For a comprehensive reference of the techniques attackers use to target identities in the browser, see the **Browser and Identity Attacks Matrix**.

30+
Device code phishing
unique device code phishing kits tracked by Push this year

1 in 4
Password logins
corporate logins are password-based rather than SSO

2 in 5
Lack MFA
of corporate logins have no MFA protecting them

1 in 5
Weak passwords
use a weak, breached, or reused password

Cloud and SaaS security

If the attacks driving the most damaging breaches target account takeover on cloud services, why not gain visibility into the cloud services themselves? Rather than approaching the problem from the browser, you might consider collecting SaaS application logs, integrating them into your SIEM, or using SSPM tools to surface suspicious activity.

In theory, that's appealing. In practice, you can only collect logs from the SaaS you know about. While 90% of executives are confident they can see the AI tools in use, 52% of employees admit to using AI tools without approval, often through personal accounts. The gap is self-adopted shadow IT that security teams have no visibility into.

Even for known apps, each log integration is brittle and expensive to maintain. Many applications don't expose the telemetry you need even at premium tiers. Most SaaS providers don't offer audit logs at all, or charge extra for the privilege.

And even where logs exist, they can only tell you what happened after an account has been compromised. At that point, in-app attacker activity is largely indistinguishable from legitimate user behavior, your response options are limited, and the attacker has likely already achieved their objective. Detecting the attack earlier, in the browser session where the credential theft or consent grant actually occurs, is the most reliable way to intervene before account takeover rather than investigate after it.

The common thread

Each of these tools operates at a specific layer: Network traffic, the OS, the email channel, the IdP, or the SaaS/cloud API. Modern browser-native attacks are designed to work outside each of their observation boundaries, or only enter them when it's too late for the attack to be stopped.

To every tool in the existing stack, the activity looks normal — the attacker uses legitimate infrastructure, authenticates through legitimate flows, and operates inside legitimate browser sessions. The same structural gap applies beyond attack detection. Users access AI tools through personal accounts, reuse passwords across unfederated apps, and upload sensitive data into ungoverned SaaS. Those actions happen inside the browser session, so they're equally invisible to tools outside it.

This is further compounded by the unmanaged device gap that cuts across all categories. BYOD and contractor devices are far softer targets than corporate endpoints. They typically have no EDR agent, no centrally managed antivirus, no hardened configuration baselines, and no security operations team watching for alerts.

Key takeaway

No combination of tools operating outside the browser session can see what happens inside it, and that's where the attacks are. Browser security closes a structural gap those investments were never designed to cover.

52%

Unapproved AI use

of employees use AI tools without approval, while 90% of executives think they have visibility (Okta, 2026)

32%

Unmanaged access

of users access corporate apps from unmanaged devices at least occasionally (Omdia)

46%

Infostealer infections

of infostealer-infected systems with corporate credentials were unmanaged personal devices (Verizon DBIR 2025)

03 The browser security market landscape

Section 2 established why tools operating outside the browser can't close the gap inside it. The natural next question is what a purpose-built browser security solution looks like. The market offers meaningfully different answers depending on the problem you're trying to solve.

Different buyers, different architectures

IT teams and security teams are both buying solutions in this space, but they're trying to solve different problems with them. IT teams are looking to full-stack enterprise browsers — replacing employees' chosen browsers with a centrally controlled workspace platform for device posture, VDI and VPN reduction, web filtering, and access governance. Security teams are buying browser security solutions for four use cases: Stopping modern browser-based attacks, governing AI, protecting data at the browser layer, and discovering shadow IT and hardening identities.

These are fundamentally different use cases, and each group leans toward different solutions as a result. The market has converged on a single category term, Secure Enterprise Browsers, for all of them, which makes the landscape harder for buyers to navigate.

Within that single category are broadly two architectures:

Architecture A

Browser extensions



Runs inside each browser people already use

Serve the security buyer by deploying into the browsers users already have, extending the security team's visibility and reach into browser sessions without requiring migration.

Architecture B

Full-stack enterprise browsers



Replaces them with one enterprise browser

Serve the IT buyer by replacing the user's browser with a controlled workspace optimized for access governance, device posture, and policy enforcement.

Full-stack enterprise browsers

Full-stack enterprise browsers like Island, Prisma Browser, and SURF Security are best understood as managed workspace platforms rather than browsers in the conventional sense. Different products with different origins, but all converging on the same buyer: An IT organization solving for workspace control.

> **Transforming the browser into a centralized, enterprise-grade platform, eliminating layers of legacy IT infrastructure by building more functionality in the browser.**
>
> **Mike Fey, Island CEO**
>
> Island now positions itself as "the agentic control plane for enterprises."

Full-stack enterprise browsers replace the user's default browser with a managed browser application. That application gives IT teams deep workspace control: Copy/paste restrictions, download policies, watermarking, screenshot prevention, ZTNA, and device posture enforcement at the OS layer. Their strongest use cases are ones where the workspace itself is the control surface. Examples include contractors and BYOD populations on unmanaged devices, where the browser is the next-best control point, and regulated environments like call centers and BPO operations.

The Achilles heel of any full-stack enterprise browser is migrating users onto it. Adopting one means asking every user to switch away from Chrome, Arc, or whatever browser they personally prefer to use. Gartner notes that most buyers start with a single use case like covering contractors and rarely pursue organization-wide deployment. User resistance is consistently cited as the top barrier. As Cribl CISO Myke Lyons puts it: "I'm not going to be able to shove a browser on everybody."

$350K

Per year for 500 users, at list price

UK reseller price lists put Island at $250,000 a year for the management console plus $200 per user per year. That's about $350,000 a year for a 500-person company before discounts, to replace a browser that's free.

And the migration challenge will only get harder as AI adoption grows, with agentic browsers like Comet and Dia pulling users toward AI-native workflows that currently most enterprise browsers can't match.

Managed browsers: A baseline, not a browser security solution

A managed browser is a standard browser that IT configures through central policy, such as Chrome under Chrome Enterprise Core or Edge under Intune. Admins can push settings, enforce updates, allow or block extensions, and block URLs. That's useful hygiene, but it's configuration rather than security monitoring. A managed browser doesn't detect a phishing page, flag a stolen session, or show what users paste into AI tools.

Google and Microsoft both sell a security tier on top. Chrome Enterprise Premium adds real-time DLP, watermarking, screenshot protection, shadow AI controls, and context-aware access to Chrome. Edge for Business comes with Microsoft 365, and its data and AI controls come from Microsoft Purview. It integrates with Entra, Intune, and Defender for Endpoint. If you already pay for either, it's worth turning on. But each one stops well short of what this guide asks of a browser security solution.

Where the built-in security tiers stop

	Chrome Enterprise Premium	Edge for Business with Purview
Browsers covered	Chrome only, in the managed work profile	Edge only
Phishing protection	Google Safe Browsing, which checks pages against lists of known dangerous sites	Defender SmartScreen, which relies on domain reputation and known-bad URLs
AI and data controls	Shadow AI controls that work by URL category	Purview DLP for unmanaged AI apps such as ChatGPT and Gemini works only on Intune-managed Windows devices, and GenAI prompt protection needs Microsoft 365 E5

Why they aren't in the comparison

We've left both out of the vendor comparison in Section 6 because neither is built for the threat model this guide ranks highest. Each one protects only its own browser, so the same app opened in another browser or a personal profile is outside its reach. Both also judge pages by reputation, and 89% of phishing domains are active for less than two days (Spamhaus, 2025). A phishing page on a fresh domain often reaches the user before any list flags it. Attacks on legitimate infrastructure, such as device code phishing on Microsoft's real sign-in page, give a reputation check nothing to flag at all.

Use what you already own. Turn on the controls in your Google or Microsoft license as a baseline, including Safe Browsing's Enhanced Protection setting. Then add a browser security layer that runs inside every browser your people use, Chrome and Edge included.

Browser extensions

Extensions plug into the browsers users already have — Chrome, Edge, Firefox, Brave — without requiring migration, retraining, or abandoning existing workflows. Once deployed, they collect telemetry from inside the browser session, detect malicious and risky activity, and enforce controls to stop it. Extensions are lightweight, with minimal impact on browser stability and performance, but they're a powerful instrumentation point. Because they run inside the browser session, they can access rendered page content, DOM structure, script execution, credential entry, network requests, and user interaction.

Security teams prefer this architecture because it mirrors how they already deploy security agents where they need visibility and control. Just as an endpoint agent deploys to the endpoint, a browser extension deploys to the browser. Extensions can also be deployed to contractor and BYOD machines without MDM, and they work in agentic browsers where full-stack enterprise browsers can't follow. The majority of vendors Gartner places in the Secure Enterprise Browser category are now extensions rather than full browsers. Gartner explicitly notes that extensions have become the preferred form factor, citing easier deployment, minimal training, and multibrowser compatibility.

What matters most, whether extension or full-stack browser, is what the vendor builds on top. This comes down to the threat models they address, the use cases they prioritize, the depth of capability they deliver, and their ability to innovate faster than adversaries. Some extension vendors have oriented toward IT use cases, echoing the enterprise browser model with "make any browser an enterprise browser" positioning. Others focus on security operations: Threat detection, identity protection, and AI governance.

In practice, the choice is rarely either/or. Many organizations deploy a full-stack enterprise browser for the contractor population or regulated workload where IT needs workspace controls. They then deploy a browser security extension across the rest of the workforce for the security use cases that a managed browser isn't built to address. Even within the security-focused extension vendors, different solutions take different approaches depending on which threats they're built to detect — that differentiation is what the next section helps you evaluate.

The acquisition wave

Four browser security companies have been acquired by larger platforms in 2026. CrowdStrike acquired Seraphic Security in January, and Zscaler absorbed SquareX in February. Akamai acquired LayerX (announced in May, completed in July), and Jamf acquired Keep Aware in September. Add Palo Alto Networks' earlier acquisition of Talon, and the browser security market has consolidated faster than almost any adjacent security category before it. That both validates the important role browser security solutions play in the modern security stack, and makes the market harder for buyers to navigate.

What matters most for those already with or seeking a browser security solution is what happens after the acquisition closes. When a browser security product is absorbed into a larger platform, engineering effort shifts toward integration and away from innovation, and capabilities are trimmed to align with the parent's core offering. Four of the five acquired products have been renamed and folded into their parents' access and zero trust portfolios: Prisma Browser (formerly Talon), CrowdStrike Falcon Seraphic Enterprise Browser, Zscaler Zero Trust Browser, and Akamai Workforce Protector. Keep Aware, the most recent deal, still runs under its own name.

Dec 2023

Prisma Browser

Talon acquired by Palo Alto Networks. Renamed and repositioned as Prisma Browser.

Jan 2026

Seraphic

Acquired by CrowdStrike. Now Falcon Seraphic Enterprise Browser.

Feb 2026

SquareX

Absorbed by Zscaler. Now Zero Trust Browser.

May 2026

LayerX

Acquired by Akamai. Now Akamai Workforce Protector.

Sep 2026

Keep Aware

Acquired by Jamf.

04 Defining your use case requirements

This section provides a framework for deciding which security requirements matter most to your organization, whether they're best addressed in the browser, and how to evaluate whether a shortlisted solution can truly address them. It groups use cases into three tiers by security value (how much risk reduction the capability delivers) and browser fit (how well-suited the browser is as the control point, relative to other layers).

The ten use cases fall into four groups: Stopping modern browser-based attacks, governing AI, protecting data at the browser layer, and discovering shadow IT and hardening identities. Buyers come to the category for all four. In Omdia's 2026 research, GenAI security (59%) and data protection (51%) were the top two use cases organizations named.

Ten use cases in three tiers

Tier 1 use cases are uniquely solved in the browser.

Tier 1

Uniquely solved in the browser

High security value, and the browser is the best place to deliver it.

01 Account takeover prevention

02 Advanced phishing detection

03 AI visibility & control

04 Identity posture hardening

05 Shadow IT & OAuth governance

06 Extension security

Tier 2

The browser is one layer

Real value, but other layers share the job or cover parts of it better.

07 ClickFix / social engineering

08 Data loss prevention

09 Investigation & IR

10 Infostealer defense

Tier 3

Low incremental value

Already covered by existing tools, or the risk is narrow.

- Browser exploit protection
- URL / domain controls
- Prompt injection

The threat model trap

Before evaluating any vendor, it's important to understand that there are different threat models that are addressable in the browser, and they aren't of equal security value. The question for any security team evaluating this space is: Which of these threat models is actually causing breaches in the real world?

Many browser security solutions are built around attacks on the browser itself. They scramble the JavaScript runtime to prevent exploits, sandbox file downloads, and stop attackers from breaking out of the browser's sandbox to compromise the underlying device. Technically legitimate capabilities — but they address a problem that's small and getting smaller. Chrome's sandbox architecture and rapid patching cycle mean disclosed vulnerabilities are typically patched within days.

The attacks behind the most damaging breaches in recent years don't target the browser itself. They target what happens in the browser: Users doing their work in cloud apps through their corporate identities. AiTM now accounts for 44.6% of the phishing techniques Microsoft identified (Microsoft DDR 2026). Push has tracked more than 30 unique device code phishing kits this year as attackers find new ways to abuse legitimate OAuth authorization flows.

The economics explain why. Browser exploits are extraordinarily expensive to develop, increasingly difficult to execute against a hardened modern browser, and patched rapidly when discovered. Identity attacks are cheap, scalable, and have a low technical barrier to adoption.

This attacker preference between the two threat models — and the different levels of risk they therefore present — is what the three tiers reflect. The highest-value use cases are all problems that play out inside the browser through its legitimate capabilities, targeting users and their identities rather than the browser as a piece of software. The lowest-value use cases are the browser-as-software problems that many solutions were originally built to solve.

This is why defining your requirements before speaking to vendors matters. The use cases that follow, and the questions in the appendix, are designed to keep your evaluation anchored to the threat model that the evidence says matters most.

The economics of getting in

Hackers don't hack in, they log in



01 / Account takeover prevention: Detecting credential attacks across all vectors

Tier 1

Why it matters

Account takeover is the dominant entry point for enterprise breaches. Its attack surface is far wider than most identity tooling can see: Ghost logins, shadow IT apps outside IdP coverage, accounts without MFA, and credentials already compromised in earlier breaches.

Why the browser

Account takeover is a browser-native attack vector: Attackers target users inside their browsers in order to compromise their identities on cloud services. That makes the browser the natural place to defend against these attacks — and explains why controls operating outside of it can't. SWGs and proxies inspect traffic on the wire, but credential entry is a client-side event. A proxy can't see whether a password is being entered into a cloned login page, or whether it's compromised, weak, or reused.

IdPs enforce policy for federated applications, but Push data shows fewer than 1 in 3 apps even support SAML SSO, and actual federation rates are lower still. The browser observes every login, which gives it visibility of the full identity attack surface and the TTPs that attackers use to exploit it.

What good looks like

01 **Map, harden, and reduce the identity attack surface**

Discover every app and identity from actual login events, including shadow IT the IdP has never seen. Then close gaps at the point of authentication: Enforce MFA enrollment, redirect to SSO, block compromised credentials and unsanctioned apps.

02 **Real-time detection and blocking of attacks that lead to ATO**

Stop attacks early in the kill chain: Phishing pages harvesting credentials, AiTM proxies intercepting authentication flows, device code phishing abusing OAuth consent. Detections should be technique-class and behavioral, not low-level IOCs attackers rotate.

03 **Continuous threat research**

The window between a technique's first appearance and its commoditization is compressing — evaluate whether research drives detection updates proactively rather than reactively.

Tier 1 · Stop modern browser-based attacks

Highest security value: Attacks and risks where the browser is the primary control point and tools outside it have the least visibility. Expect browser security to lead here.

89%

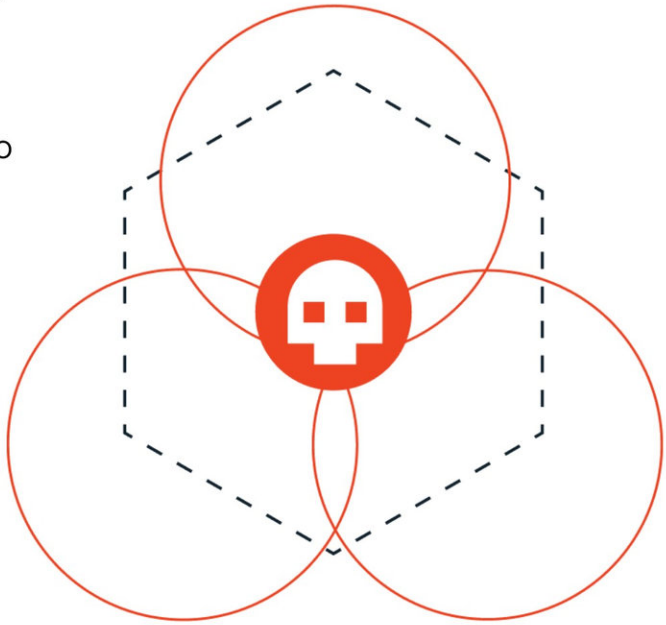
Identity-driven

of 750+ Unit 42 incident response cases exploited identity weaknesses

63%

Compromised logins

of human logins involve credentials already compromised elsewhere



02 / Detecting and stopping advanced phishing: AiTM, multi-channel delivery, and zero-day lures

Tier 1

Why it matters

AiTM phishing — where a reverse proxy intercepts credentials and session tokens in real time — is now the standard technique for bypassing MFA at scale, sold through kits that compete on detection evasion. Traditional controls can't keep up, with 1 in 2 payloads Push intercepts now delivered outside email entirely. And the threat extends beyond credential phishing: MFA downgrade attacks force fallback to weaker factors, while consent and device code phishing bypass authentication entirely.

Why the browser

Modern phishing pages are dynamic web applications rendered by the browser's JavaScript engine, not static HTML. Email gateways analyze sender and content but never the destination page — and see nothing when the lure arrives via LinkedIn DM, malicious ad, or in-app message. Proxies reconstruct page behavior from traffic, but modern kits use DOM obfuscation and bot protection (95% of attacks Push detects) so what the proxy sees bears no resemblance to the page. Blocklists chase the most disposable indicators attackers have. The browser is the only layer where you can observe the rendered page as the user sees it, in real time — the same shift the industry made from signature-based AV to behavioral EDR.

What good looks like

Detection must target attacker techniques, not indicators. Kits fragment constantly but share behavioral mechanics: How AiTM relays proxy credential entry, how cloned pages harvest credentials, how Browser-in-the-Browser pop-ups spoof authentication windows. Technique-class detection fires on zero-day kits regardless of domain or infrastructure age. Credential entry guardrails add a second layer, blocking corporate passwords from unauthorized domains regardless of page classification. Real-time blocking is essential because attackers now exploit access so quickly. Deep context for responders matters too. Many scanners just see a CAPTCHA or Turnstile, and the malicious content can be gone by the time you investigate.

Tier 1 · Stop modern browser-based attacks

Highest security value: Attacks and risks where the browser is the primary control point and tools outside it have the least visibility. Expect browser security to lead here.

63%

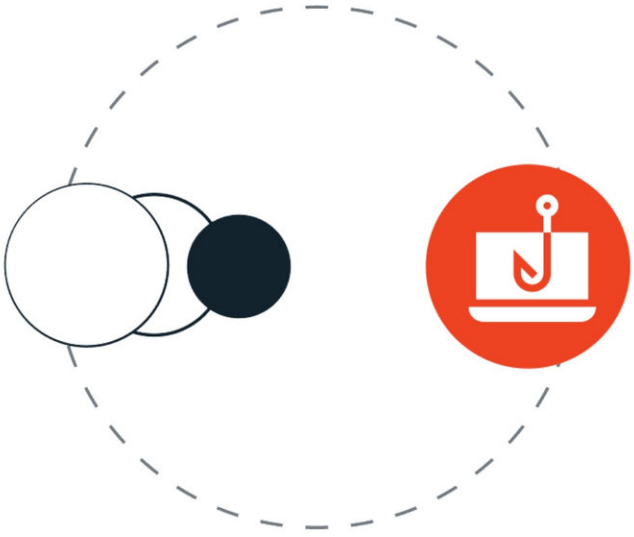
PhaaS compromise

of account compromise incidents attributed to PhaaS kits

75+

Attacker toolkits

unique attacker toolkits documented by Push



03 / AI visibility and control: Enforcing which AI tools employees can use and how

Tier 1

Why it matters

AI adoption is outpacing security governance at nearly every organization. It's now the use case buyers name most often: In Omdia's 2026 research, 59% of organizations named GenAI security as a top browser security use case. Many of the risks are familiar Tier 1 problems in a new form. Shadow AI apps are shadow IT (#5), AI OAuth integrations are OAuth governance (#5), AI browser extensions are extension security (#6), and personal AI account usage is an identity posture problem (#4). What's new is the data: Prompts and uploads carry sensitive content straight into third-party models. Organizations that block AI tools outright push usage underground rather than prevent it, trading visible risk for invisible risk.

Why the browser

The browser is the best single layer for AI visibility and control: It sees the apps, the OAuth grants, the extensions, and whether the user is authenticated via a corporate or personal account. For unsanctioned AI tools, it provides a real-time enforcement point at the moment of use. For AI agents and MCP-connected tools that operate through OAuth grants, the browser is where the consent decisions that authorize them are made. A complete strategy also requires endpoint coverage for local AI tools and IDE-integrated agents — the right primary layer, but not the only one.

What good looks like

Effective AI governance requires graduated controls: Monitor, warn, and block — not binary allow/deny. Monitor first to understand usage, warn to change behavior without disrupting workflows, block only where the risk justifies it. Use the browser to enforce which AI tools employees can access and ensure they reach the corporate tenant, then rely on platform-native controls (Claude Enterprise, ChatGPT Enterprise, Copilot) to govern activity within that environment. Evaluate whether the solution distinguishes corporate from personal usage of the same tool, and surfaces AI OAuth grants alongside standard SaaS integrations. Prompt-level DLP on sanctioned tools is better handled platform-natively.

Tier 1 · Govern AI

Highest security value: Attacks and risks where the browser is the primary control point and tools outside it have the least visibility. Expect browser security to lead here.

45%

Regular AI users

of employees are regular AI users on corporate devices, up from 15% (Verizon DBIR 2026)

58%

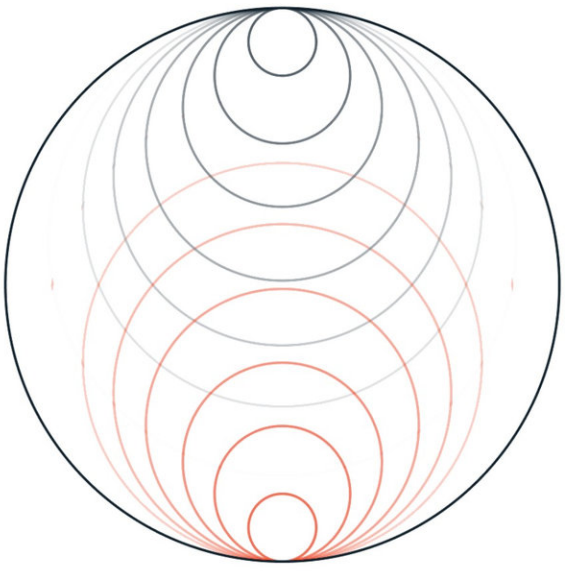
Rely on SWGs

of organizations use a SWG to secure GenAI, which can't see what users paste into a prompt (Omdia)

16

Shadow AI · apps

unique AI apps in active use at the average organization — as many as 40 at the high end



04 / Identity posture hardening: Enforcing security across the apps your IdP doesn't manage

Tier 1

Why it matters

The average employee maintains ~15 identities across ~220 applications, and Push data shows fewer than 1 in 3 of those apps support SAML SSO. Of the last million logins Push observed, 2 in 5 had no MFA and 1 in 5 used a weak, breached, or reused password. Even apps behind SSO often keep a parallel password login, called a ghost login, that bypasses every control the IdP enforces. Ghost logins appear alongside SSO in roughly 10% of accounts. Attackers, regulators, and insurers have all noticed. Criminals actively target weak and unfederated identities. NYDFS has fined firms for MFA failures, including OneMain ($4.2M), National Securities ($3M), and Travelers ($1.55M). The UK's Cyber Essentials scheme now requires MFA on every cloud service where it's available.

Why the browser

IdPs enforce policy for federated applications, but 1 in 4 corporate logins Push observes use a password rather than SSO. Those identities sit outside the IdP's reach. API integrations, traffic inspection, and email sign-up notifications only ever build a partial picture, because they can only see apps they already know about. The browser sees the login itself, discovering the identity at the moment it's created or used: Authentication method, password strength, MFA status, and all.

What good looks like

Discovery without enforcement is just an inventory problem. Being in the browser means acting at the moment of authentication: Guardrails that prompt MFA enrollment, guide users toward stronger credentials, and redirect to SSO close the gap at scale, on every app. This produces the continuous, auditable evidence of MFA coverage and credential hygiene that regulators, insurers, and auditors increasingly require — evidence no IdP-centric tool can provide for apps outside its scope.

Tier 1 · Discover shadow IT and harden identities

Highest security value: Attacks and risks where the browser is the primary control point and tools outside it have the least visibility. Expect browser security to lead here.

15

Identities

maintained by the average employee

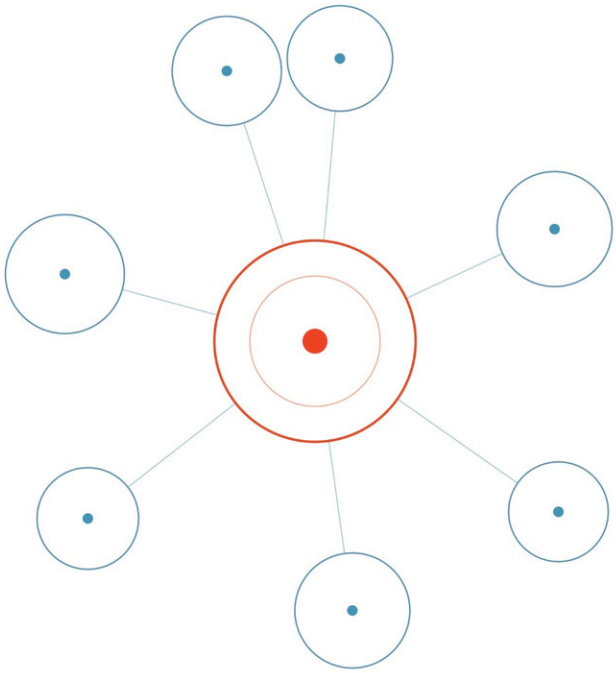
2 in 5

Lack MFA

of corporate logins have no MFA protecting them

The cost of gaps

A Canadian city had an $18.3M insurance claim denied after a ransomware attack exploited MFA gaps in its environment.



05 / Shadow IT discovery and OAuth integration governance

Tier 1

Why it matters

Where identity posture hardening (#4) focuses on how employees authenticate, shadow IT discovery focuses on what they authenticate to — including SaaS apps IT never sanctioned. Many of those apps arrive as self-adopted freemium accounts that generate no purchase event at all, so procurement and expense data never see them.

OAuth connections across these unmanaged estates cause much of the damage. The ShinyHunters campaign against Salesforce and Google Workspace tenants pivoted through OAuth tokens stolen from Salesloft Drift and Gainsight. The Context.ai-to-Vercel chain used stored tokens from a forgotten AI app trial to bridge into Google Workspace. Once issued, a grant can outlive password resets and MFA changes, and fully revoking it requires first knowing it was granted.

Why the browser

Most discovery tools rebuild the app estate after the fact, from sources that each miss part of it. Expense data misses free tools, and IdP logs only cover apps that already use SSO. Email-based discovery misses accounts created with a personal address, and network traffic can't show how a user signed in. The browser observes real logins as they happen, on every app, whether or not IT knows the app exists.

Every OAuth consent grant also transits the browser: The authorization prompt, the scope disclosure, the approval click, and the redirect that completes the grant. That puts the browser in a position to intercept an unwanted grant before the grant completes.

What good looks like

Discovery should capture applications at the moment of login, regardless of whether the app is sanctioned, federated, or known to IT, and must distinguish corporate from personal accounts. For OAuth, the critical capability is real-time consent interception: Evaluating and acting on a grant request before the token is issued, configurable by scope, app category, and user group. Like standard credential phishing kits too, OAuth consent phishing attacks leave detectable traces around the legitimate consent pages and flows that can be intercepted before the user can approve a malicious consent grant.

Tier 1 · Discover shadow IT and harden identities

Highest security value: Attacks and risks where the browser is the primary control point and tools outside it have the least visibility. Expect browser security to lead here.

41%

Expensed apps

of applications are bought through expense channels rather than IT procurement (Zylo, 2026)

1,000+

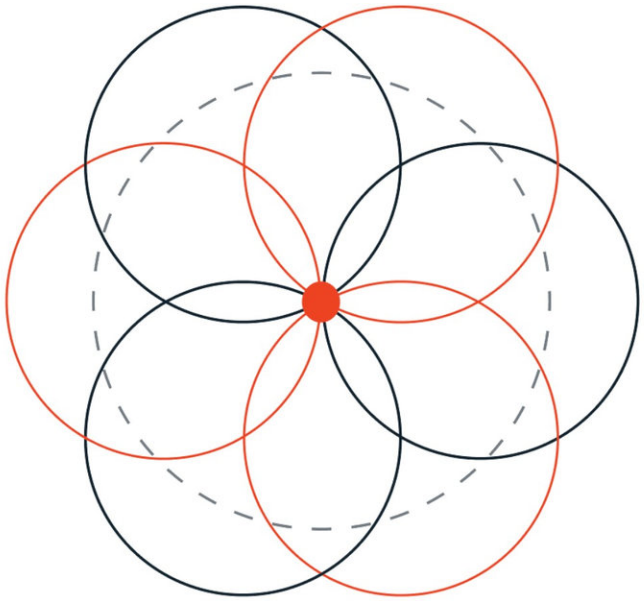
Organizations

hit in ShinyHunters' Salesloft Drift and Gainsight campaign — 1.5B records claimed stolen

17

Shadow AI · integrations

unique AI OAuth integrations connected into just Google Workspace and Microsoft 365 — up to 55



06 / Browser extension security

Tier 1

Why it matters

DarkSpectre, a China-attributed group, ran malicious Chrome, Edge, and Firefox extensions that infected more than 7.8 million browsers. Its extensions earned real users and "Verified" badges for three to five years before DarkSpectre switched on the payloads (Koi Security, 2025). The pattern continues: Poper Blocker kept Google's "Featured" badge and 2M+ active users while it stole AI chat data for at least four months after it was reported to Google (Dark Reading, 2026). The Cyberhaven campaign compromised 35+ Chrome extensions reaching ~2.6 million users after attackers phished developer OAuth credentials. Push's analysis of 20,000+ extensions across its customers found 46.76% hold the permission combinations needed to perform account takeover with no user interaction.

Why the browser

SWGs and network tools are structurally blind here: A malicious extension exfiltrating session tokens generates no anomalous network signal. Endpoint agents have no visibility into extension behavior at the session level. Even the stores' own review consistently fails to catch malicious updates — attackers use dynamically loaded payloads, time-delayed activation, and conditional execution to evade static analysis. If detection can't happen at the store level, it has to happen in the browser at runtime.

What good looks like

01 **Complete inventory**

What extensions are installed and what permissions they hold, across every browser in the environment.

02 **Supply chain change monitoring**

Ownership transfers, permission escalations, developer contact changes, and delisting: The discrete events that actually predict compromise.

03 **Enforcement**

Default-deny allowlisting where only vetted extensions are permitted, plus remote removal of risky extensions across any browser.

Detection of known-malicious extensions is table stakes. The real threat model is legitimate extensions that turn malicious after an ownership transfer, developer account compromise, or malicious update.

Tier 1 · Discover shadow IT and harden identities

Highest security value: Attacks and risks where the browser is the primary control point and tools outside it have the least visibility. Expect browser security to lead here.

7.8M+

"Verified" extensions

browsers infected by DarkSpectre's extensions, which built trust for years before turning malicious

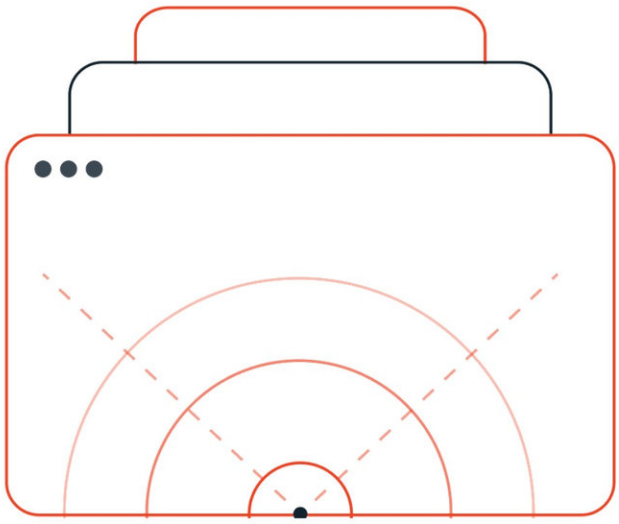
17

Shadow AI · extensions

unique AI browser extensions at the average organization — reaching 163 at the high end

⚠ Why risk scoring fails

Every major extension supply chain breach of the past two years scored as low-risk immediately before compromise. Scores describe the extension as it is today — not what it becomes after the next update.



07 / Blocking ClickFix and social engineering-based malware delivery

Tier 2

Why it matters

ClickFix was the most common initial access vector in Microsoft's 2025 Digital Defense Report, and its 2026 report shows ClickFix attacks grew roughly 23x between December 2025 and May 2026. The technique writes a malicious command to the victim's clipboard and social-engineers them into executing it. The attack is fileless and user-executed, and 4 in 5 of the ClickFix pages Push intercepts arrive via search engines rather than email. A single technique bypasses download scanning, endpoint behavioral detection, and email security. The *Fix family keeps diversifying: InstallFix clones developer tool installation pages, LLMShare hosts delivery pages on chatgpt.com and claude.ai, and ConsentFix is fully browser-native, with no endpoint component. A complete ConsentFix criminal toolkit appeared on the XSS forum within weeks of Push detecting APT29 using the technique.

Why the browser

The browser is an early and effective intervention point. It can detect the clipboard injection and social engineering lure before anything reaches the endpoint in executable form. That avoids the costly reimaging and lateral-movement investigation that follows endpoint compromise. For standard *Fix variants the detection surface is shared with the endpoint; ConsentFix is the exception, where the browser is the only detection layer that matters. When attacks arrive via malvertising, compromised websites, or trusted platform domains, there's no email to scan and no network indicator to flag. The page that loads in the browser is the first, and often the only, detection opportunity.

What good looks like

Detection should be technique-class and generic, applying to any attack that results in malicious copy-and-paste behavior regardless of the *Fix derivative or social engineering wrapper. Once the command has been pasted and run, detection and remediation become endpoint problems. A mature defense requires both layers. Evaluate whether detection fires on novel variants — because the latest samples are quickly adopted by PhaaS vendors and made available to the wider criminal marketplace.

Tier 2 · Stop modern browser-based attacks

Strong security value, shared or complementary with other layers. The browser is one of several relevant control points — expect it to complement other tools rather than replace them.

~23×

Volume growth

growth in ClickFix attacks between December 2025 and May 2026 (Microsoft)

67%

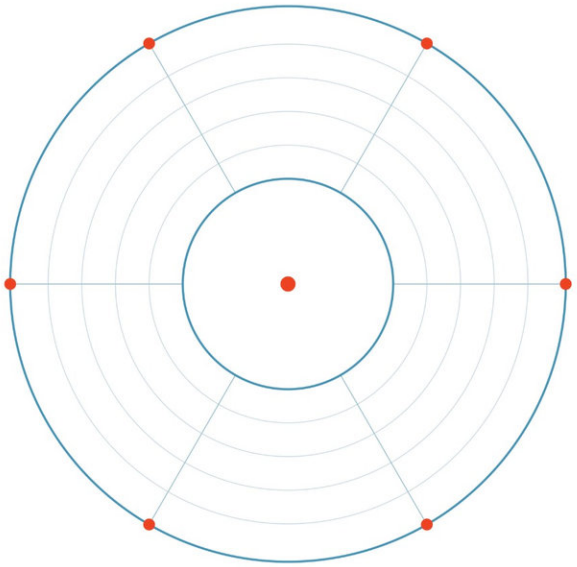
ClickFix share

of Push detections in August 2026, up from a 52% average through Q2

4 in 5

ClickFix payloads

are accessed via search engines



08 / Data loss prevention: A key component of effective DLP, but not the full picture

Tier 2

Why it matters

File uploads to unsanctioned services, sensitive data pasted into AI tools, and exfiltration through personal accounts are genuine risks that traditional DLP tools weren't designed to catch. But DLP is a multi-layer problem: Email, endpoint, cloud API, and browser each cover different exfiltration paths. The browser is an important layer but not a standalone DLP replacement — it can't see email-based loss, endpoint-to-endpoint transfers, or cloud API exfiltration.

Why the browser

Browser-layer controls provide real value here, particularly for BYOD users and contractors, where endpoint DLP agents can't be deployed and the browser is the only available data loss visibility. The browser sees file uploads, downloads, clipboard activity, and AI prompt inputs in a way network and endpoint tools don't.

What good looks like

Start with telemetry, such as file uploads, clipboard activity (especially into AI prompts), and application access patterns. Then build targeted controls from that understanding: Block uploads to unsanctioned destinations, flag sensitive data pasted into AI tools, and restrict risky application categories. The trap to avoid is hyper-granular policies for every user group on every app, which introduces excessive friction and inevitably gets switched off. In the Cloud Security Alliance's 2026 *DLP in the AI Era* survey, 54% of teams had already softened DLP enforcement after business pushback. Integration with dedicated DLP platforms extends the browser's telemetry into existing workflows rather than creating a parallel alert stream.

Tier 2 · Protect data at the browser layer

Strong security value, shared or complementary with other layers. The browser is one of several relevant control points — expect it to complement other tools rather than replace them.

16%

DLP blocked it

of organizations said DLP blocked their most significant incident before the data left (CSA, 2026)

67%

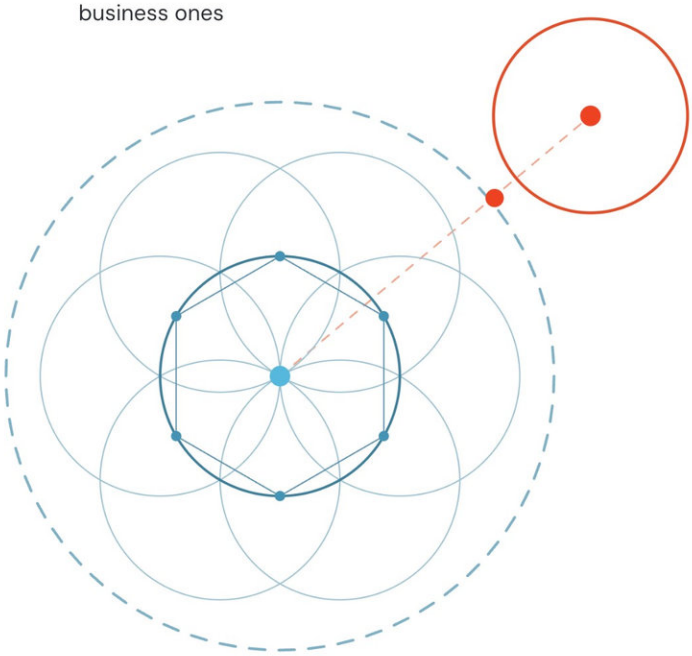
Personal accounts

of GenAI users on corporate devices are using personal accounts (Verizon DBIR 2026)

38%

Shadow uploads

of file uploads to AI apps in the browser are made from shadow personal accounts rather than business ones



09 / Investigation and incident response: Closing the missing middle

Tier 2

Why it matters

Endpoint logs show what processes executed, network logs show traffic destinations, and IdP logs show authentication events. None of them show what happened inside the browser session: The phishing page the user saw, the credentials they entered, the malicious OAuth grant, the data pasted to an unsanctioned service. This is the missing middle of modern incident investigations. Most modern attacks don't leave the endpoint artifacts investigators traditionally depend on, because they play out through legitimate browser sessions rather than malware execution.

Why the browser

Investigations are inherently multi-source: Without browser data the picture is incomplete, but without endpoint and network data it's equally so. The browser fills the session gap with the causal link other sources miss. It bridges "a user visited a URL" and "credentials were submitted to a phishing page that issued a session token now being replayed from an attacker-controlled browser."

What good looks like

User and session timelines should reconstruct the full sequence of events: What users clicked, what pages loaded, what credentials were entered, and what session activity followed. Behavioral query engines that search across browser activity using behavioral indicators (password inputs on unfamiliar domains, JavaScript obfuscation combined with login forms) transform the platform from a detection tool into an investigation platform.

The same telemetry supports proactive scenarios: When a third-party vendor announces a breach, immediately identify which employees had accounts on the breached service, whether credentials were reused, and the blast radius. Integrated with SIEM and SOAR, high-confidence detections enable automated response workflows, and custom detection rules extend coverage to environment-specific threats.

Tier 2 · Stop modern browser-based attacks

Strong security value, shared or complementary with other layers. The browser is one of several relevant control points — expect it to complement other tools rather than replace them.

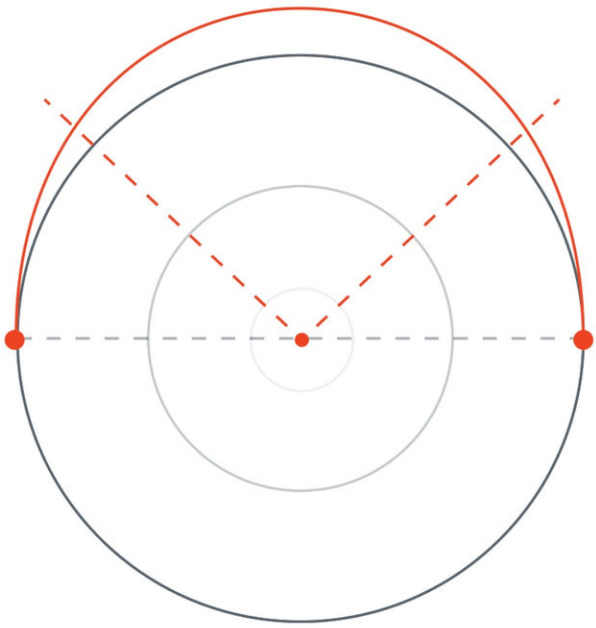
48%

Browser-based

of intrusions involve browser-based activity

The missing middle

For these investigations, the absence of browser telemetry leaves responders reconstructing attack timelines from sources that never saw the attack happen.



10 / Infostealer defense: Detecting exposure and blocking delivery

Tier 2

Why it matters

Infostealers are the upstream supply chain for a disproportionate share of the most damaging enterprise attacks. They harvest credentials, session cookies, and browser profile data en masse. Criminals then sell the outputs for use in credential stuffing, ATO, and ransomware campaigns. The 2024 Snowflake attacks began with infostealer-harvested credentials. The 2023 Okta breach began when an attacker compromised service account credentials saved to an employee's personal Google profile. Those credentials gave the attacker access to support files containing customer session tokens (an example of browser sync exploitation). The chain is self-reinforcing — ClickFix delivers infostealers, infostealers harvest accounts, compromised accounts distribute more ClickFix lures — compounding with every rotation.

Why the browser

The browser is relevant at three points in the kill chain. First, delivery interception: ClickFix is now the primary infostealer delivery mechanism, and the browser is the earliest layer that can intercept it, before anything is executed. Second, detecting stolen factors in use: Breach dataset matching at login and session marker injection for stolen tokens. Third, the browser is itself the credential leakage path: Personal-account browser sync copies saved corporate passwords to a personal cloud account outside the organization's control. In the Verizon 2025 DBIR, 46% of infostealer-infected systems with corporate credentials were unmanaged personal devices. Execution remains an endpoint problem; a complete defense requires both layers in concert.

What good looks like

Stolen credentials should be flagged at the point of login against known breach datasets. Stolen session tokens require a different mechanism: Sessions from instrumented browsers carry a marker, and a token appearing in an uninstrumented browser is a confirmed stolen session. Raw threat intel feeds are overwhelmingly noise — evaluate whether the vendor validates against credentials actually in use in your environment.

Tier 2 · Stop modern browser-based attacks

Strong security value, shared or complementary with other layers. The browser is one of several relevant control points — expect it to complement other tools rather than replace them.

50%

Ransomware link

of ransomware victims had an infostealer or credential event in the 95 days before the attack (Verizon DBIR 2026)

54%

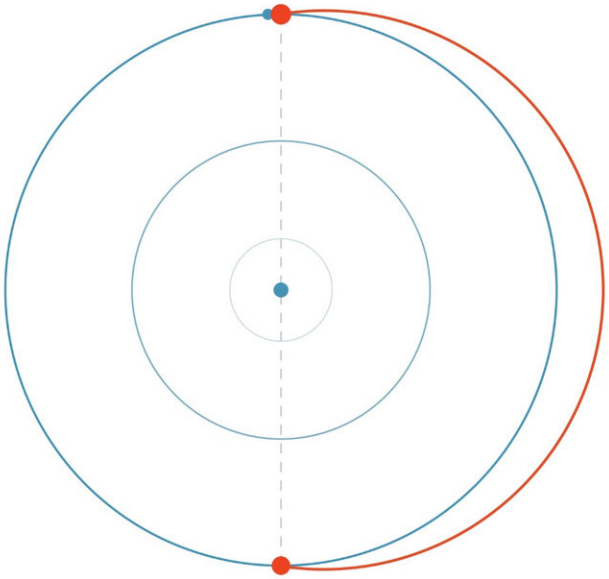
Infostealer-infected

of devices in initial access broker logs had at least one infostealer installed (Verizon DBIR 2026)

<1%

Actionable

of raw stolen-credential alerts are true positives at review (Push data). Push's behavioral corroboration removes 99% of those false positives.



Tier 3: Low incremental security value

Some capabilities that appear on vendor feature matrices deliver limited incremental value. The problem may already be well-addressed by existing tools, the risk itself may be narrow, or the capability isn't yet solving real-world breach scenarios. Understanding where a browser security solution adds little beyond what you already have helps you avoid overweighting these in your evaluation.

Browser exploit protection

Addresses the narrow risk of a remote code execution or sandbox escape targeting the browser engine itself. Browser zero-days represent just 9% of the in-the-wild zero-days tracked by Google, and the overwhelming majority of modern attacks bypass the endpoint entirely. This is a problem for browser vendors to solve, and it's not a big enough problem to warrant enterprises investing in additional mitigating controls.

Domain and URL category controls

Offer genuine browser-layer value but are commoditized by SWG and DNS filtering tools that most organizations already operate. They offer limited security value against modern attacks that quickly rotate these kinds of indicators and are designed to blend in. A browser security solution may include category-based blocking as a convenience feature, but it's not a differentiator that moves the needle on security.

Prompt injection and AI jailbreaking protection

An emerging concern that has been primarily limited to vendor research at the time of writing, but is likely to pose a greater threat as AI agents gain deeper access to enterprise systems and data. The absence of current breach data doesn't mean it should be ignored. Look for a strong and informed point of view on the threat and a credible plan for tackling it effectively, not superficial checkbox features added to a marketing page.

Use cases excluded from scope

Two use cases that frequently appear in browser security vendor pitches are excluded from this framework — not because they lack value, but because they fall outside the scope of this guide.

Access management (ZTNA, VPN replacement, PAM)

An IT infrastructure and access architecture problem, not a security operations problem — it belongs to a different buyer with a different evaluation frame. There are numerous (typically full-stack) enterprise browser solutions on the market that address IT use cases like this well. If your primary need is workspace access governance, evaluate the full-stack enterprise browser category described in Section 3 rather than the security-focused solutions this section is designed to help you assess. Securing contractor and BYOD devices is different. It isn't a separate use case but a deployment advantage: A solution that reaches unmanaged devices without MDM extends every use case above to people your endpoint tools can't cover. Section 5 covers it under deployment and coverage.

Remote browser isolation

A legacy approach to browser security that has been superseded by modern browser security solutions. RBI addresses browser exploit risk rather than the identity-first attacks that represent the majority of current enterprise browser risk, and introduces UX friction that limits deployment at scale. When it triggers, it introduces latency but still fails to detect and stop browser-native attacks that operate through legitimate authentication and authorization flows.

05 Shortlisting, evaluating, and testing solutions

You now have a ranked set of requirements anchored to the use cases that deliver the most security value in the browser. This section helps you move from those requirements to a decision. It covers three stages: Shortlisting vendors, evaluating them before committing to a formal test, and structuring a proof of concept that surfaces real differences rather than confirming vendor marketing.

01

Shortlisting

Where to find vendors, how to filter them against your requirements, and why limiting your search to existing platform vendors is a major mistake.

02

Evaluating

What to assess during vendor evaluation before committing to a formal test.

03

Testing

How to structure a Proof of Concept (PoC) that surfaces real differences rather than confirming vendor marketing.

Shortlisting solutions

Where to start

Most solutions in this space sit within the Secure Enterprise Browser (SEB) category, as defined by Gartner. The Gartner SEB Market Guide (February 2026) confirms that the majority of solutions are now browser extensions, not replacement browsers. This is the best single category to find a representative long list of browser security solutions to evaluate.

From there, the architectural distinctions from Section 3 and the use case requirements you ranked in Section 4 should help you narrow that long list to a shortlist worth evaluating in depth.

Top tip: Feed AI comparisons current data

If you're using an AI application to compare vendors against your requirements, request the latest datasheet from each vendor and feed that into your analysis. This is a fast-moving solution category, and there's a good chance that a vendor's capabilities have moved on since the point at which the AI model's training data was collected.

Don't exclude a vendor because an AI application says it doesn't address one of your requirements. The vendor may support it, and the detail just wasn't on its website when the model was trained. The same risk applies in the other direction: A model may over-credit a vendor based on announced but not yet shipped features, or miss that an acquired product's roadmap has changed since acquisition. Current datasheets give the AI — and you — an accurate picture to work from.

Avoid the platform pitfall

The biggest risk when shortlisting vendors for evaluation is limiting your search to what your existing vendors can offer. Adding browser security tooling through an existing CrowdStrike, Zscaler, or Palo Alto relationship means fewer contracts, a simpler procurement process, and a consolidation narrative that plays well internally.

In most other security solution categories, it's reasonable to accept a lesser capability in return for the ease of not having to onboard and manage a new vendor. That's because the risk exposure left by the capability gap between an incumbent's offering and a dedicated best-of-breed solution is often negligible. But that's different for browser security solutions: The inherent risk from modern threats that play out in the browser is significant, and the difference in residual risk left by different vendor solutions is vast.

The three shortcomings of platform-based browser security tools

Platform-based browser security tools typically have three shortcomings that leave their users underserved and exposed. Understanding them is essential to building a strong shortlist and evaluating it effectively. The pages that follow examine each.

1. Built for the wrong threat model

Section 4's threat model trap established that the highest-value browser security use cases all involve identity attacks that play out inside the browser through its legitimate capabilities, not attacks on the browser itself. The problem for buyers is that most of the solutions now available through platform vendors were architecturally built for the lower-value threat models:

CrowdStrike Falcon Seraphic Enterprise Browser

Built for browser runtime exploit prevention — effectively defending against traditional endpoint attacks that come via the browser.

Zscaler Zero Trust Browser

Built for file-based malware sandboxing in disposable browser containers.

Akamai Workforce Protector

Built for access governance, risk scoring, and AI usage policy.

These are all technically legitimate capabilities, but they aren't the capabilities behind the breaches making headlines. If you select a solution built around one of these threat models, you'll have browser security tooling deployed. But the identity attacks that Section 4 ranks as your highest-priority risk will remain largely unaddressed.

2. Capabilities that work on paper, not in practice

Platform vendors will claim their browser security tools can do everything that dedicated best-of-breed solutions do. But the difference between an effective and an ineffective browser security solution isn't in what they claim to do — it's in how they actually do it. Two examples illustrate what this means in practice.

Phishing detection

Every vendor claims phishing detection. What matters is where on the Pyramid of Pain those detections sit. Most browser security vendors detect at the bottom — matching domains, URLs, and IPs against blocklists. These are the indicators easiest for attackers to change. 89% of phishing domains are active for less than two days, and 95% of attacks Push detects use bot protection to hide from the scanners that feed those blocklists. Every phishing attack is effectively a zero-day for organizations relying on known-bad detection. The shift from static signatures to behavioral analysis is what killed antivirus and gave rise to EDR — most browser security vendors haven't made the equivalent shift.

Browser extension security

Most solutions approach extension security through risk scoring — rating extensions on permissions, store metadata, code analysis, and developer reputation. This bears little relevance to how extension breaches actually happen. Every major extension breach of the past two years — Cyberhaven, DarkSpectre, Trust Wallet — involved extensions that scored as normal or low-risk right up until they were weaponized. The attacks come from legitimate extensions being compromised through supply chain attacks or ownership transfers, not from extensions that looked risky beforehand. The signals that actually predict compromise — ownership changes, permission escalations, developer account takeover — are discrete events that require monitoring and immediate response, not a recalculated risk score.

3. Innovation that stalls after acquisition

When a browser security product is acquired, it gets pared back to fit the acquiring vendor's core platform offering. Focus shifts from innovation to integration.

In the acquisition wave from Section 3, four of the five acquired products have already been renamed and folded into their parents' access and zero trust portfolios. The high-value security use cases established in Section 4 compete for roadmap priority against a much larger product portfolio, and engineering effort turns inward toward platform integration rather than advancing detection capability.

But this is no time for engineering efforts to turn inward. AI is compressing the time to generate a convincing phishing campaign from hours to minutes. Techniques like device code phishing have gone from research curiosity to industrialized exploitation in a matter of months, in large part enabled by AI-powered tools. The gap between an acquired product focused on integration and a dedicated vendor focused on stopping these emerging threats will continue to widen over time.

Why this matters for your shortlist

These three issues — mismatched threat models, superficial capabilities, and slowing innovation — make solution evaluation more challenging, but they also make it more important. The ten evaluation criteria that follow are designed to help you test for them.

10 evaluation criteria to assess before committing to a proof of concept

Once you have a shortlist that includes dedicated best-of-breed solutions alongside any platform offerings, the ten criteria below will help you evaluate them before committing time and resources to a formal proof of concept. Most of these are questions you can answer through vendor briefings, documentation review, and reference checks. The appendix includes specific questions to ask vendors during demos, mapped to each of the use cases defined in Section 4.

01 Start from your own requirements

Use the ranked use cases from Section 4 as your evaluation framework. Define the outcomes you need before speaking to any vendor, and hold every conversation accountable to those outcomes. The use cases that address the largest areas of risk, and for which the browser is uniquely suited, should be the areas you drill into deepest.

02 Identify each vendor's threat model

Every browser security product was built around a specific threat model, and that original orientation typically still shapes the product's strengths and limitations. Ask each vendor which threats their product was originally designed to address, how that focus has evolved, and where their detection depth is strongest. A solution built around browser exploit prevention or file sandboxing may have added identity attack detection to its marketing, but the architectural foundations and detection depth will reflect where the product started. Some products aren't oriented around threat detection at all — they're built around governance, policy enforcement, and risk scoring, and won't deliver detection depth regardless of what their marketing claims.

03 Test capability depth and the expertise behind it

The gap between what a capability does on paper and how it performs in the real world can be considerable. Don't just ask what the solution does, but how it does it — and press on specifics. For threat detection: Ask every vendor whether a detection matches a known-bad indicator or an attacker technique. Then ask them to show you a detection that fires on a page where the domain, URL, and IP are all brand new. For extension security: Is the vendor relying on risk scoring based on static attributes, or monitoring for the ownership changes, permission escalations, and supply chain indicators that actually precede real-world compromise? The quality of detection also depends on the expertise behind it. Detections built by practitioners with real red and blue team experience produce very different results from those that technologists build in reaction to customer requests. Ask who builds the detections and what their background is.

04 Interrogate AI claims with specificity

Every vendor in the category now claims AI-powered detection. The question is what the AI operates on and what infrastructure supports it. Commercial AI models are commoditized. The value isn't in the model itself but in the telemetry it analyzes, the domain expertise that puts its output in context, and the enforcement layer that turns detections into real-time protection. Ask what telemetry the solution collects and where it's collected from. Ask whether the vendor has an in-house threat research team that discovers novel attack techniques, or whether they document what others have already found. Ask how quickly a newly discovered technique goes from identification to a shipped detection. Finally, distinguish between AI that summarizes and triages alerts (useful, but not detection) and AI that proactively hunts novel attack techniques across real customer environments — the latter delivers far more value.

05 Assess the deployment model

The deployment model determines where the solution can and can't operate. An extension deploys into existing browsers with no migration; a full-stack enterprise browser requires mass migration to a new browser; a kernel-level agent requires endpoint access and creates stability considerations. Omdia's 2026 survey found that 48% of organizations cited the ability to deploy to existing browsers as an important factor. Each approach also has coverage implications: Kernel-level agents lose meaningful capability on unmanaged devices, BYOD machines, and contractor endpoints — and 32% of users access corporate applications from unmanaged devices at least occasionally. Consider too whether agentic browsers — Comet and Dia — are emerging in your environment; the solution you choose needs to cover them.

06 Evaluate how the solution can be operationalized at scale

A solution that works in a demo or a small PoC doesn't necessarily work across a full production deployment. Features can become inactionable at scale. Extension risk scoring is easy to act on with ten extensions to review. When a full deployment surfaces hundreds of high-risk extensions, the same feature becomes an unmanageable backlog rather than a useful control. Automated vulnerability remediation workflows need to hold up when applied to thousands of users. Ask about false positive rates at production scale, how much configuration is required to reach an acceptable signal-to-noise ratio, and what the deployment timeline looks like for an organization of your size.

07 Test integration depth — including processes and playbooks

Integration is more than a list of supported SIEMs. Does the tool support how your team raises, triages, and investigates detections? Does it provide the right data in the right format and send it to the right downstream tool? Will it slot into how your analysts already work and into their existing incident response playbooks? A solution that generates high-quality telemetry but delivers it in a format that doesn't match your workflows creates overhead that erodes the value it provides.

08 Evaluate innovation velocity

The shortlisting section explained why acquired products turn inward at the worst possible time. During evaluation, ask for evidence: What novel attack techniques has the vendor discovered and published in the past six months? What new features or telemetry streams shipped in that period? Are new detections releasing continuously, or in quarterly cycles? For acquired products, ask how the roadmap has changed since acquisition — specifics, not reassurances. The most meaningful signal is whether the vendor has published attack techniques the rest of the industry subsequently built detections for: The difference between leading the field and following it.

09 Assess platform lock-in vs. open ecosystem integration

Every capability consolidated into an existing platform vendor reduces your ability to change direction later. A solution that integrates into open ecosystems — feeding browser-layer telemetry into your SIEM, SOAR, and identity tools alongside the rest of your stack — preserves flexibility. One that requires you to be on the parent platform to realize its full value creates a dependency extending well beyond browser security.

10 Think beyond initial visibility — evaluate sustained value

Visibility is often the entry point: Any capable solution will surface problems you didn't know you had. But think past that discovery phase. The problems it surfaces — new attack techniques, identity vulnerabilities, shadow IT sprawl, risky extensions — need to be addressed, not just reported. Evaluate whether the solution can act on what it finds at scale, or whether it will generate a growing list of issues you can't resolve through the product. Then look further ahead: When new browser-based threats materialize, how well positioned is the vendor to address them? A solution that front-loads value but can't remediate, respond, or evolve will plateau while your subscription continues. Ask about customer retention, and ask for references from former customers to understand why they left.

Running a proof of concept that surfaces real differences

Once you have shortlisted solutions based on the criteria above, a proof of concept is where claims meet reality. The following practices are drawn from real-world evaluations and are designed to help you structure a PoC that tests what actually matters.

01

Anchor your success criteria to your threat and risk model

Before the PoC begins, be clear about the threat and risk model you need this solution to address — the use cases ranked in Section 4 and the evaluation criteria above should define that. Build your success criteria directly from it: Which threats must the solution detect, which risks must it reduce, and what does a pass or fail look like for each? A PoC without criteria anchored to a specific threat model will end with each vendor interpreting results through the lens of whichever problem their product was built to solve. Define what matters before the test starts, and measure every solution against the same bar.

02

Deploy to representative user groups from across the business

Evaluating security teams often deploy only to security and IT staff. These are typically your most security-aware and policy-compliant users. Data from them can significantly understate the problems the solution is designed to address: Less shadow IT, fewer unmanaged identities, fewer identity vulnerabilities, fewer risky user behaviors, and less exposure to targeted attacks. Deploy to a cross-section of the business to get an accurate picture of what the solution will likely surface in production.

03

Test detection against real-world attack techniques and evasion methods

Make sure your testing accurately reflects how real-world attackers operate. Test against the TTPs behind actual breaches: AiTM phishing, malicious OAuth consents, malicious or risky browser extensions, personal browser profile sync. Use real AiTM tooling such as Evilginx, or your red team's equivalent, on fresh infrastructure to replicate the techniques that criminal kits like Tycoon2FA and Sneaky2FA use. Don't rely on old phishing URLs whose indicators are already on blocklists by the time you test. Real attackers rotate infrastructure constantly to stay ahead of known-bad detection; your testing should reflect that.

04

Measure telemetry quality, not alert volume

During a PoC, count how many detections are immediately actionable and how many require analyst investigation just to determine whether they represent a real threat. A solution that generates hundreds of low-confidence alerts — suspicious domains, file heuristic matches, anomalous but benign user behavior — creates work without reducing risk. High-fidelity detections tell your team what happened and what to do about it; low-fidelity alerts tell your team to go and find out. If your analysts are spending most of their time triaging alerts that turn out to be noise, the solution is adding to alert fatigue rather than solving it.

05

Test that detections lead seamlessly into blocking and response workflows

Detection is only part of defending your organization. When the solution detects a threat, what happens next? What telemetry does it generate, and where does it go? Can your SOC actually investigate using this data — determine what happened, how far it got, whether it's part of a broader targeted campaign, and what else may be affected? A solution that detects an attack but doesn't block it in real time, or can't support the investigation and response that follows, has limited value for a security operations team.

06

Evaluate the deployment experience — and the vendor behind it

Test whether the solution works in your actual environment — not a lab. Does it change the way your network, endpoints, or applications behave in ways that could cause instability or service desk tickets? Test on the browsers your employees actually use, including any enterprise or agentic browsers in your environment. Rolling out a new solution across your estate will inevitably surface issues, and the PoC is your opportunity to evaluate how the vendor handles them. Do they proactively spot and resolve deployment problems before you report them, or are you chasing support tickets? Do they offer communication channels with people who have the right level of expertise and attitude to work through complicated issues? Are they willing to add support for your specific systems and processes? Are they open to feedback and keen to work with you to evolve the capability? The quality of the vendor relationship matters long after the PoC ends. A strong product with poor support is a liability. A vendor that treats the PoC as a partnership rather than a sales exercise shows you what the ongoing relationship will look like.

07

Test the user experience of controls

Most browser security solutions directly interact with users in a way most security tools don't, and whether users comply with its controls will determine whether the solution actually reduces risk in production. During the PoC, test whether the controls make it easier for users to do the right thing. Do users adhere to them, or find ways around them? Does the solution improve security awareness by providing contextual guidance at the point of risk — in the browser, at the moment it matters? Collect user feedback directly: Do they find the tool helpful, disruptive, or invisible? A solution that generates complaints and workarounds at scale has a hidden operational cost that won't show up in a vendor demo.

08

Test end-to-end workflows, not just individual features

Individual capabilities may look strong in isolation, but the real test is whether complete workflows are practical, intuitive, and actually achieve the outcome you need. This is where the sustained value argument from evaluation criterion 10 becomes tangible. A solution that surfaces a problem but needs a clunky or incomplete workflow to resolve it won't deliver value beyond the initial discovery phase. Test specific workflows end to end. Examples include hardening identities on sensitive applications, guiding users from an unsanctioned app to an approved alternative, and responding to a public breach that involves an app or extension in your environment. A workflow might break down, need manual steps the product can't support, or fall short of what you need. In each case, that's a gap you'll live with for the duration of the contract.

09

Evaluate integration with your existing stack — including processes and playbooks

Test integration with your IdP, SIEM, SOAR, and endpoint tools. Go beyond checking that data arrives: Test whether the data format, event structure, and alert fidelity support your existing triage and investigation workflows. If your analysts have to manually reformat or enrich data to make it actionable, that's a cost the vendor's integration checklist won't surface.

10

Use the PoC to baseline your actual attack surface

A PoC isn't just a product test — it's an opportunity to measure exposure you didn't know you had. Most organizations model risk from conventional network, endpoint, and cloud telemetry, and entire categories of exposure go unmeasured. A browser security PoC can surface environment-specific data that changes the picture. It can show how many phishing attempts reach your users outside of email and how many shadow IT apps and AI integrations exist across your workforce. It can also show how many identities lack MFA or use weak and reused credentials, and how many previously unknown browser extensions carry the permission combinations required for account takeover. That data has value well beyond the product evaluation itself.

For most buyers, a focused PoC is the right approach. Confirm the solution works as intended against your requirements, and validate detection capability against the TTPs that matter. Then verify integration with your stack and capture the baseline data above. Keep it short, keep it targeted, and let the results speak.

In some cases, though, the problem isn't picking between solutions but justifying the purchase in the first place. If you need to surface evidence that you have a set of security problems requiring an unplanned budget allocation, a longer proof of value deployment may be warranted. This type of evaluation generates findings specific to your organization that are far more compelling to a CFO than generic industry benchmarks, but the tradeoff is more time and effort to run.

One 30-day Push proof of value

4,500-employee financial services firm

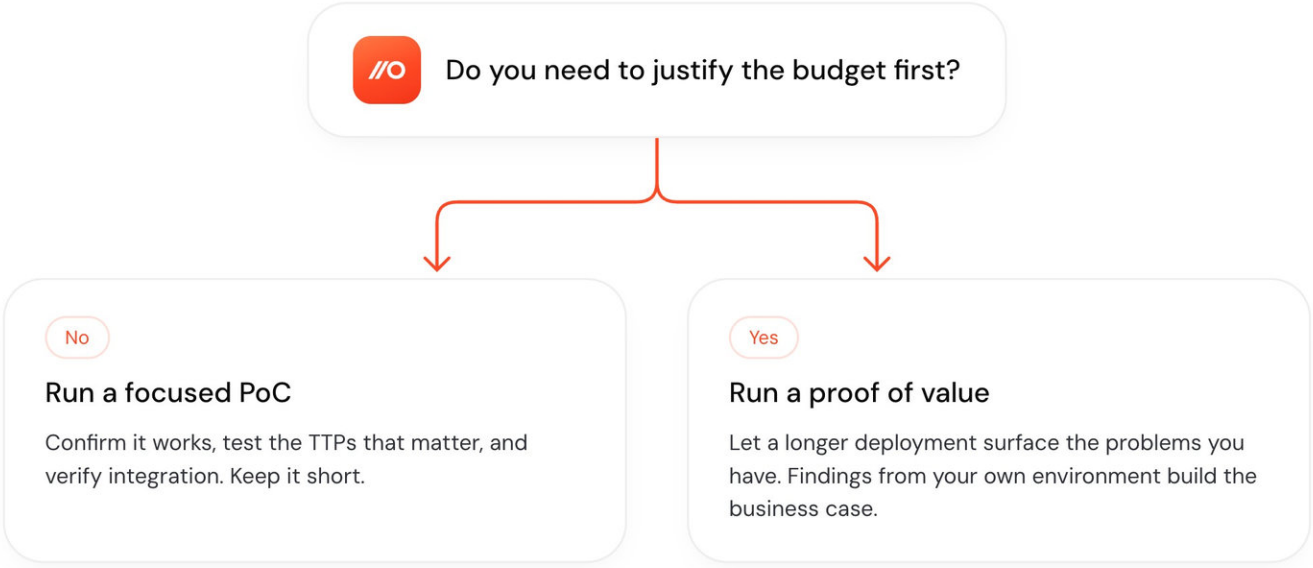
6	10	2,859	203
ClickFix attacks stopped	AiTM phishing attempts stopped	corporate accounts with security vulnerabilities	OAuth apps granted significant permissions without admin consent

No other tool in the stack saw the attacks Push stopped.

Only go this route when you expect significant resistance to budget and need to make the case in organization-specific terms. Section 7 covers how to build the business case from this evidence in detail.

Ready to decide?

If you've followed this section, you've shortlisted solutions on their capabilities rather than ease of procurement. You've evaluated them against the criteria that matter most to your organization, and you've run a PoC that surfaces real and impactful differences. You should now have the evidence you need to make a confident decision. The next section applies this framework to the vendors most commonly encountered in the market today.

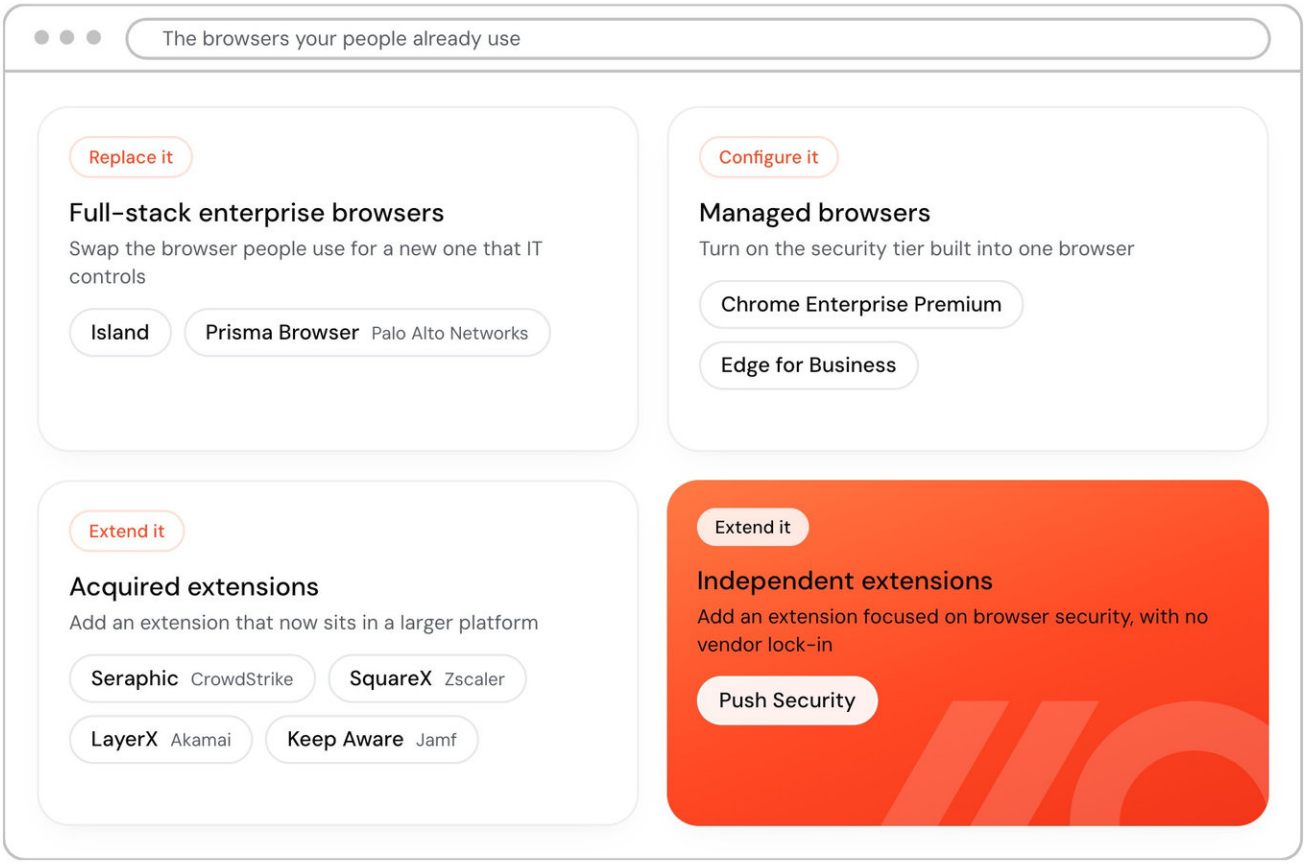


06 Comparing solutions

You now have a ranked set of requirements, a framework for evaluating vendors against them, and a structured approach to testing that surfaces real differences. This section applies that framework to the vendors most commonly encountered in the market today, organized around the six dimensions the guide has already established.

The comparison covers seven vendors across two architecture categories: Full-stack enterprise browsers (Island, Prisma Browser) and browser extensions (Push Security, Seraphic/CrowdStrike, SquareX/Zscaler, LayerX/Akamai, Keep Aware/Jamf). Four of the extension vendors have been acquired by larger platform companies in 2026, and those acquisitions shape what buyers can expect from each product going forward.

Managed browsers such as Chrome Enterprise Premium and Edge for Business sit outside the comparison, for the reasons covered in Section 3.



The comparison framework

Each dimension below maps directly to the use case ranking, evaluation criteria, and architectural analysis the preceding sections established. The framework is the same regardless of which vendor you're evaluating; the outcomes differ because of what each vendor was built to do.

01

Threat model alignment

Does it address the threats and risks that are actually leading to real-world breaches? This distinction has the greatest bearing on the overall risk the solution can address, and the security value it can provide.

02

Capability depth

Does the solution provide deep technical capabilities that allow multiple problems to be addressed now and in the future, or only superficial capabilities that check requirement boxes on paper?

03

Deployment and coverage

How does the solution deploy, what environments does it cover (managed, unmanaged, BYOD, agentic browsers), and what friction does it introduce?

04

Operational maturity

Does the solution work at production scale, integrate into existing workflows and playbooks, and produce actionable output rather than noise?

05

Innovation velocity

Is the vendor discovering and publishing novel attack techniques, or documenting what others have already found? Has acquisition slowed the pace?

06

Strategic independence and sustained value

Does the solution lock you into a platform, and does it deliver value beyond initial visibility, through remediation, response, and evolving capability?

How to read the matrix

The landscape matrix on the next page rates all seven vendors across these six dimensions. A checkmark-style feature matrix can't distinguish between a behavioral detection and a blocklist lookup, or between a shipped capability and an integration roadmap — the cell notes explain the structural reason behind each rating.

Vendor comparison matrix

Solution	01 Threat model alignment	02 Capability depth	03 Deployment & coverage	04 Operational maturity	05 Innovation velocity	06 Strategic independence
Full-stack browsers						
Island	IT workspace control. Policy enforcement and known-bad blocking — not built for identity attack detection.	Known-bad blocking. Policy rules on URLs, domains, file types. Not built around behavioral detection.	Full browser replacement; extension is an on-ramp. Migration resistance limits scope.	Mature platform, named enterprise customers. Strong IT workflow integration.	Rapid expansion, but IT/productivity-focused — not security research.	Independent, but full replacement creates migration lock-in.
Prisma Browser	SASE-derived: URL filtering and sandboxing heritage. Threat model centers on web content.	Advanced Web Protection and Precision AI phishing scans. Strong DLP (1,000+ classifiers).	Full browser, extension, mobile app. PANW advises against its extension on unmanaged devices.	Mature via the PANW stack. Value tied to being a PANW customer.	Repositioned since the Talon acquisition. Effort shifted to platform integration.	Sold as Prisma SASE's last mile. Findings depend on Cortex.
Acquired extensions						
Falcon Seraphic (CrowdStrike)	Exploit prevention heritage. Built for a threat that's rare in practice.	Exploit prevention, Falcon Data Security, and AI prompt monitoring. Test phishing depth in a PoC.	JS runtime injection; stability risk beside the Falcon sensor. Weaker on BYOD and contractor devices.	Some integrations shipped (Fusion SOAR, Falcon Shield). Probe GA versus roadmap.	Repositioned around access. SGNL integration still on the roadmap.	Falcon ecosystem lock-in. Standalone value unclear.
Zero Trust Browser (SquareX)	Malware sandboxing heritage. File-focused isolation in a malware-free landscape.	Sandboxing, extension analysis, and inline DLP. Identity attack detection unproven at scale.	Extension or full browser. Limited pre-acquisition deployment history.	Early-stage ZTE integration. Probe GA versus roadmap.	SquareX Labs silent since February. Roadmap follows Zscaler's network focus.	ZTE lock-in. ZIA overlap on core capability.
Workforce Protector (LayerX)	Governance and compliance model. Governs employee activity, not external adversaries.	Captures full AI conversations, including desktop AI apps and IDEs. Risk scoring, not adversary detection.	Extension-based, lightweight. Integrates with Akamai's private access product.	Standalone product relatively mature. Now Akamai Workforce Protector.	AI governance ahead of market; post-acquisition trajectory uncertain.	Recently acquired. Akamai's track record suggests absorption.
Keep Aware (Jamf)	Known-bad phishing pages, extension risk, and AI policy. Not built around attacker techniques.	Known-bad domains, risk-scored extensions, and policy rules. Limited public technical detail.	Lightweight extension, quick deployment. Agentic browser coverage undocumented.	Covers fundamentals. Enterprise-scale references less established.	Some research output; pace of discovery less visible.	Acquired by Jamf (Sept 2026). Roadmap now follows Jamf's Apple management focus.
Independent extensions						
Push Security	Built for in-browser attacks: AiTM, session hijacking, credential theft, OAuth abuse, ClickFix.	Behavioral phishing detection. One detection covers a technique across many kits and new domains. Session marker injection.	Extension into any browser. MDM, email, or self-enrollment. 100K users in under an hour.	Production-scale deployments. SIEM/SOAR, session timelines, remediation workflows.	Dedicated research team (ConsentFix, InstallFix, SAML-jacking). Agentic hunting pipeline.	Independent. No migration lock-in; complements the existing stack.

Full-stack enterprise browsers

Island

Independent · Full browser, with an extension on-ramp

Built for	IT workspace control and secure access
Strongest for	Contractors and VDI replacement
Ask in your PoC	What works without the full browser?

Island's core strength is workspace control. For IT teams that need to govern a contractor population, enforce DLP at the OS rendering layer, or replace VDI with a managed browser, Island is a capable platform with a clear value proposition. In September 2026, Island raised $400M at a $6.4B valuation to extend it.

That strength doesn't extend to detecting the attacks this guide highlights. Island's security model is built around policy enforcement and known-bad blocking: Controlling what users can do inside a managed workspace. That's a different class of capability from detecting a phishing kit behaviorally as it loads, identifying a stolen session replayed from an unprotected browser, or catching clipboard injection attacks through page analysis. Island doesn't target attacker techniques at the behavioral level, and its positioning reflects this: The company doesn't position itself as a security operations tool. Island's own documentation says it isn't suited to environments that need independent visibility and control across detection and enforcement layers.

The deployment model is the other consideration. Island is a full browser replacement: Every user migrates off their existing browser. Island now offers a browser extension to ease that migration, but the extension is designed as an on-ramp to the full browser, not as the end state. Island comes at a significant price tag, making wall-to-wall deployment prohibitively costly for most organizations; in practice, most deployments are limited to contractors or specific subsets of the workforce. The migration resistance and limited deployment scope Section 3 documented apply directly here.

For organizations that need both IT workspace controls and security operations capability, Island and a browser security extension are complementary — and Island supports extension-based solutions deploying inside it. The evaluation question is whether a single product can credibly serve both the IT buyer and the security buyer, or whether trying to do both means doing neither with sufficient depth.

Prisma Browser

Palo Alto Networks · Browser, extension, and mobile app

Built for	Last-mile enforcement for Prisma SASE
Strongest for	Existing Prisma SASE and Cortex customers
Ask in your PoC	What do findings look like without Cortex?

For organizations already invested in the Prisma SASE stack, Prisma Browser offers clear value through unified policy management. Prisma Browser's integration with Cortex XSIAM, WildFire, and Advanced URL Filtering means browser telemetry feeds into an existing detection and response pipeline without additional integration work. For a PANW customer, this is a compelling consolidation story.

The question is what Prisma Browser delivers for organizations that aren't already Prisma SASE customers. The telemetry value is tied to the broader platform; for teams running a different stack, workflow integration is limited. Prisma Browser does bring real capability into the browser: Advanced Web Protection inspects pages after they load, and it inspects how approved extensions behave, not only whether they're allowed. Precision AI scans for "evasive phishing" and Enterprise DLP brings more than 1,000 classifiers. But detection built around attacker techniques isn't the same capability as web-content inspection, even if vendors market both as “real-time.”

Deployment tells a similar story. Prisma Browser is available as a full browser, an extension, and a mobile app. But Palo Alto's admin guide calls the extension “a foundational layer in a phased deployment” toward the full browser, and its product FAQ says “secure extensions are not recommended for unmanaged devices.” Prisma Browser began as Talon, an acquired product, and Palo Alto has since renamed it and repositioned it around access management. That fits the post-acquisition pattern described in Section 3, where engineering effort shifts from innovation toward platform integration.

The acquired extension vendors

Four browser security companies were acquired by major platform vendors in 2026. CrowdStrike acquired Seraphic (January), Zscaler absorbed SquareX (February), and Akamai acquired LayerX (announced May, completed July).

Falcon Seraphic (CrowdStrike)

CrowdStrike · Browser and extension (runtime injection)

Built for	Browser exploit and sandbox-escape prevention
Strongest for	Falcon customers focused on exploits
Ask in your PoC	How does it run beside the Falcon sensor and how are performance concerns addressed?

Seraphic was built to prevent browser runtime exploits: Randomizing the JavaScript engine environment to defeat memory-corruption attacks without relying on signatures. For the narrow category of browser sandbox breakouts and zero-day exploit chains, this is a differentiated capability.

But as Section 4 documented, browser exploits represent a small and declining share of the threat landscape. The attacks behind the breaches this guide documents — AiTM phishing, session hijacking, ClickFix, credential theft, OAuth consent abuse — don't target the browser runtime at all. They play out inside the browser session, targeting users and their identities on cloud services. Seraphic's architecture was designed for a different threat model, and that original orientation shapes what the product can and can't detect.

CrowdStrike has added data and AI controls around the browser. Falcon Data Security (March 2026) covers browsers alongside endpoints, SaaS, and cloud, and Falcon monitors AI prompts and responses.

Seraphic's architecture works by injecting into the browser's JavaScript runtime: The same approach antivirus vendors used for years, with well-documented stability consequences. CrowdStrike's global outage in July 2024 showed how far a stability failure in a security agent can spread. Buyers should expect CrowdStrike to make changes to an injection-based architecture slowly and conservatively as a result. That means integration work and detection updates for Seraphic are likely to arrive slowly, on top of the stability risk the architecture already carries.

CrowdStrike repositioned Seraphic around secure access management and also made the SGNL identity acquisition. The SGNL dynamic authorization integration is still on the roadmap, so buyers should ask what's generally available today. They should also test deployment stability at representative scale, particularly alongside the existing Falcon sensor and on BYOD and contractor devices, where Seraphic loses capability.

Zero Trust Browser (SquareX)

Zscaler · Extension or Chromium browser

Built for	Sandboxing malicious file downloads
Strongest for	Existing ZIA customers
Ask in your PoC	What is GA, and what is still roadmap?

SquareX's core capability was sandboxing suspicious file downloads in disposable browser containers: A legitimate approach to malware prevention that also supports browser-based isolation for file handling. SquareX also developed extension analysis capabilities and an active research program with talks at DEF CON, Black Hat, and RSA. Some of that research drew strong pushback. Ars Technica called its DEF CON passkey research "pure nonsense," and Perplexity called its Comet findings "contrived." Either way, SquareX Labs has published no new research since the acquisition.

Sandboxing, though, addresses a declining attack surface. The credential theft, session hijacking, and consent abuse techniques behind the most damaging breaches generate no malicious file and never trigger a sandbox. For existing Zscaler customers, the question is particularly pointed: Zscaler already provides sandboxing through ZIA, so the incremental value of SquareX's sandboxing capability is unclear beyond its extension analysis.

Browser-native security could disrupt the legacy SWG category. Inside Zscaler, SquareX now follows Zscaler's priorities, which center on network traffic. Buyers should ask how the roadmap has changed since the acquisition.

The acquisition closed in February 2026. Zscaler has since relaunched the technology as Zero Trust Browser, delivered as a cross-browser extension or a full enterprise browser. The April 2026 extension adds inline DLP for copy/paste, uploads, downloads, and printing. Since the deal, no customer case studies reference the product, and Zscaler's messaging centers on agentic AI rather than browser security. SquareX had limited enterprise deployment history before the acquisition.

Buyers should probe what is generally available versus what remains on the integration roadmap, and should thoroughly test detection depth and quality.

Workforce Protector (LayerX)

Akamai · Extension

Built for	AI governance and compliance audit
Strongest for	AI visibility on managed and BYOD devices
Ask in your PoC	Can you show me how an attack happened, not just that it was blocked or risk-scored?

LayerX is focused primarily on AI governance. It captures full AI conversations, classifies sensitive data automatically, and extends coverage to desktop AI apps and developer IDEs. For organizations whose primary browser security requirement is AI usage visibility and compliance, LayerX fits that use case.

The threat model distinction Section 4 established applies directly here: Protecting the company from its employees and protecting the company — including its employees — against external threats are different problems. LayerX is built around policy enforcement, risk scoring, and compliance audit trails: Controlling what users do and flagging when they violate policy. That solves a real problem, but it isn't the same problem as stopping external adversaries from breaching the organization. AiTM phishing, session hijacking, and OAuth consent abuse don't violate a policy: They compromise an identity. No amount of risk scoring detects them; behavioral detection of attacker techniques does.

On extension security, LayerX relies on the permissions-based risk scoring model that Section 4 identified as structurally ineffective against supply chain compromise — the dominant real-world extension attack pattern.

Akamai acquired LayerX for approximately $205 million, completing the deal in July 2026 and renaming the product Akamai Workforce Protector. Akamai's track record with prior cybersecurity acquisitions (Guardicore, Neosec, Noname Security, Inverse) follows a consistent pattern: Products get absorbed into the platform, independent roadmap development slows, and market presence fades. LayerX lost its name at close, which follows the Neosec and Noname pattern. LayerX's innovation trajectory post-close is unlikely to resemble its trajectory as a standalone company.

Acquired in September 2026

Keep Aware (Jamf)

Jamf · Extension

Built for	Known-bad phishing and AI policy
Strongest for	Quick MDM rollout and SIEM forwarding
Ask in your PoC	How has the roadmap changed under Jamf?

Keep Aware covers the fundamentals of browser security — phishing detection, extension management, AI usage control, and DLP — in a lightweight extension that, like Push, deploys in minutes. Jamf acquired Keep Aware in September 2026 and plans to pair it with its Apple device management and AI governance work, so the roadmap will now follow Jamf's priorities. The considerations for enterprise buyers center on depth and direction. Building and maintaining detections across the full range of Tier 1 attack techniques demands dedicated research capacity that scales with the threat landscape. A small team that covers phishing, extensions, AI governance, and DLP at the same time has to spread that investment thin. That makes detection depth the thing to test.

Browser security is a multi-year investment, so buyers should ask how the roadmap has changed under Jamf. They should also test detection capability rigorously during a PoC, particularly against novel kits on fresh infrastructure where reputation-based detection has no signal. Buyers should then check whether the output supports investigation workflows or only generates policy alerts that need manual assessment.

Other vendors you'll meet

They come up in evaluations and share an architecture with the vendors above.

Sophos Protected Browser Full-stack browser

Powered by Island, with the same architecture and limits. It suits existing Sophos customers.

Citrix Enterprise Browser Full-stack browser

A Chromium browser for apps launched from Citrix Workspace. Its value is tied to an existing Citrix estate.

Netskope One Enterprise Browser Full-stack browser

A managed Chromium browser that extends Netskope SSE policy to contractors and unmanaged devices.

SURF Security Full-stack browser

A smaller Zero Trust Browser vendor focused primarily on GenAI data controls. The full browser needs a migration.

Menlo Security Browser isolation

Remote browser isolation that keeps files and exploits off the endpoint. AiTM and ClickFix deliver no payload for isolation to contain.

Push Security: Independent extension vendor

Push is a browser extension that runs inside the browsers your employees already use. That puts it closer to your users than any other security tool. From there, Push stops modern attacks, governs AI use, protects sensitive data, and hardens identities at the point where people work.

Why Push

Stop the attacks your stack can't see.

Push's research team and agentic threat hunting pipeline look for the techniques and toolkits attackers rely on, not the domains they rotate. New detections ship to customers in minutes. Each detection is built to survive the changes attackers make, so a single detection can cover a technique across many kits, variants, and fresh domains.

See what your users see, and act in the moment.

Push sees the page a user lands on, the password they type, the file they upload, and the prompt they paste into an AI tool. It can warn or block at that moment. It works inside the browsers people already use, with no migration or user disruption.

A browser security program that doesn't add to your workload.

Protections work out of the box, with no rules to write first. When attackers change their techniques, Push's researchers build and ship the new detections. Your team doesn't have to write rules or tune detections to keep up.

One extension, four main use cases

Stop modern browser-based attacks

Behavioral phishing detection stops AiTM and other phishing pages before the user enters credentials. Push blocks device code and OAuth consent phishing and detects malicious copy and paste attacks such as ClickFix. It also detects stolen sessions when they're replayed in another browser.

Govern AI

Push discovers every AI app, AI extension, and agentic browser in use, and the account behind each login. It can block unsanctioned AI tools, require corporate accounts for approved ones, and control what users paste and upload.

Protect data at the browser layer

Copy and paste controls catch sensitive data such as secrets, credentials, and PII, and can redact it. Upload and download rules apply by user group, file type, and browser profile. Push can also block whole categories of apps and domains.

Discover shadow IT and harden identities

Push finds shadow IT and ghost logins from real login events, not network traffic. At sign-in, it flags breached and weak passwords and enforces MFA and SSO guardrails.

3×

the number of new detections Push ships each month, with shipping time for new threats down from weeks to minutes

1,000s

of threat hunts each month, run by Push's detection agents across the customer base

100,000

users deployed in under an hour, during office hours, with zero downtime



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.



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.



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.

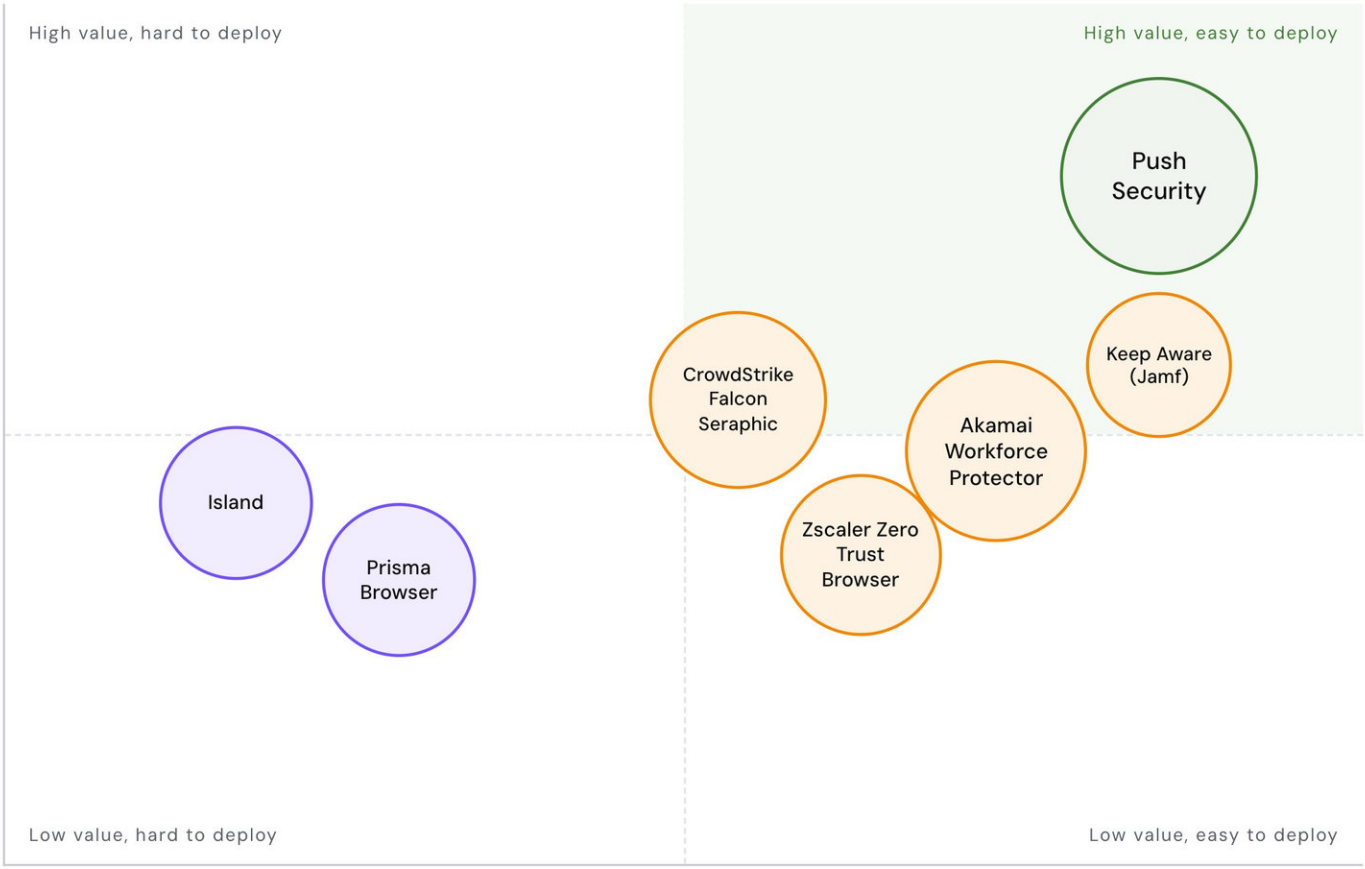


Myke Lyons
CISO, Cribl

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.

Putting it together: Deployment ease × security value

How easily each solution deploys into your environment, against how much security value it delivers on the threats the evidence says matter most.



The chart distills the two factors that matter most: How easily the solution deploys into your environment, and how much security value it delivers against the threats the evidence says matter most.

Full-stack enterprise browsers sit on the left. They require a browser migration that limits deployment scope, and their security value is concentrated in IT workspace controls rather than the attacks that drive the most damaging breaches. The acquired extension vendors land in the center. They deploy more easily, but their original threat models (exploit prevention, malware sandboxing, governance) don't align with the highest-priority use cases. Acquisition has also shifted their engineering focus from innovation to platform integration.

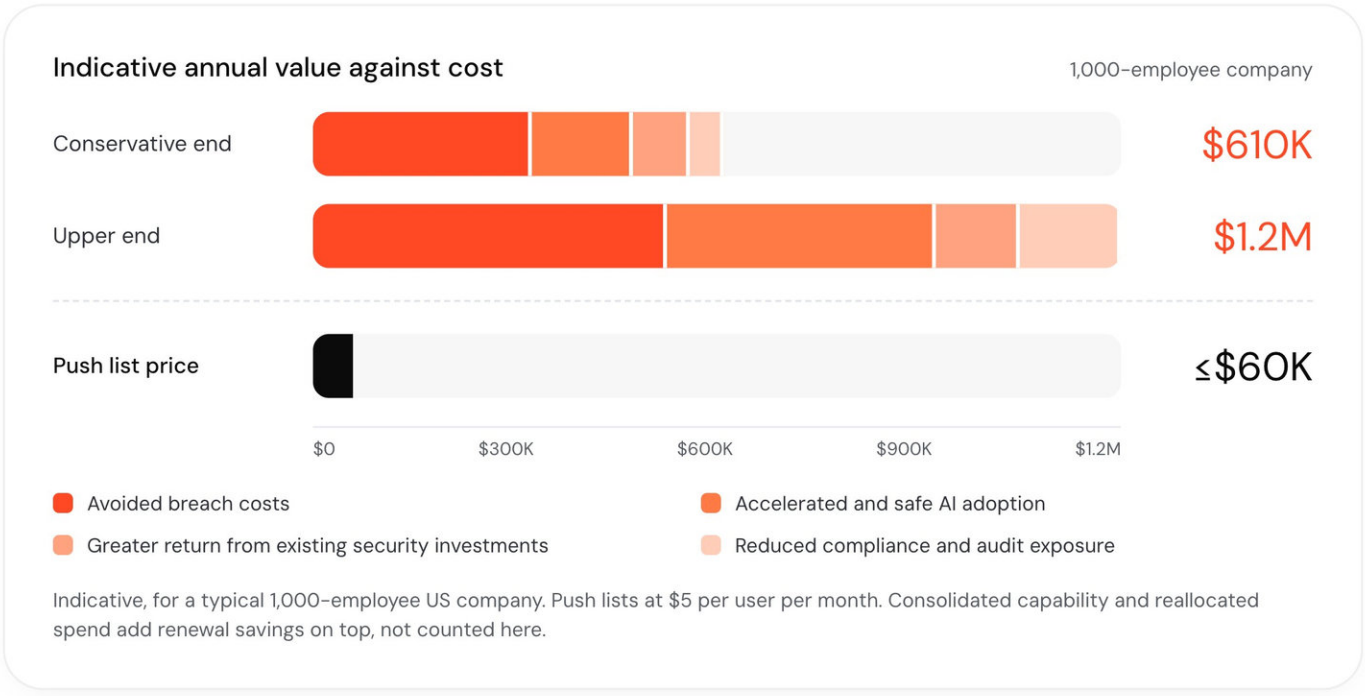
Where Push lands

Push lands top-right: Extension-based deployment with no migration barrier, and security value concentrated on the highest priority attacks. That positioning follows directly from applying the evaluation criteria the preceding sections describe.

07 Building the business case

If you've already selected a solution — or even if you're still evaluating — you'll likely need to build a business case to secure funding. Browser security is predominantly an additive investment. Gartner and Omdia both find that it augments rather than replaces existing security controls, and Omdia found that 80% of organizations expect to deploy it alongside their current stack. You can see this in the acquisition wave from established network and endpoint vendors as they look to broaden their capabilities to address the gap in their coverage. Email security vendors see the same gap: In September 2026, Proofpoint announced Advanced Browser Protection, built in partnership with Push Security, to extend its coverage from the inbox into the browser.

That means there's typically no legacy line item to redirect or renewal to swap out, and you'll need to build the case from scratch. This section provides a framework for doing that. It draws on two strategic imperatives that resonate with non-security executives and five economic value drivers that quantify the return. It also gives practical guidance on where organizations that already made this investment found the budget.



Indicative, for a typical 1,000-employee US company. Push lists at $5 per user per month. Consolidated capability and reallocated spend add renewal savings on top, not counted here.

The economic case: Five value drivers

The board needs to know why something needs to be done, but a CFO wants to see where the money comes from, what existing spend it offsets, and what measurable return it delivers. The economic case draws on five value drivers, each grounded in capabilities specific to operating inside the browser session.

To illustrate the potential impact, each value driver below includes indicative savings for a typical 1,000-employee US company. The assumptions are conservative and the benchmarks publicly available; your own numbers will differ, but the methodology is transferable.

01
Avoided breach costs

The detection gap has a direct financial consequence: Every attack that slips through undetected is a potential breach. IBM's data puts the average US breach cost at a record $10.22M, and breaches involving shadow AI cost about $670K more. The model makes three conservative assumptions: A 5–8% annual breach probability, that roughly 80% of those breaches are identity-based attacks a browser security solution can intercept, and 80% detection and blocking effectiveness. On those assumptions, a browser security solution generates significant expected annual value. The return is defined by the absence of an event, but the methodology is well-established in risk management, and CFOs already accept this logic for insurance and business continuity investments. Leading indicators of success include reduced MTTD/MTTR and fewer attacks progressing to account takeover.

Indicative saving **$325K–$525K**

02
Accelerated and safe AI adoption

Without effective AI visibility and control, your security team becomes either the bottleneck for adoption or allows the risks to go unchecked. Stanford and MIT research found customer support agents with access to a generative AI assistant were 14% more productive on average; the Federal Reserve found generative AI users saved 5.4% of their work hours. A browser security platform removes the governance blocker. When you can see which AI tools employees use, what data they share, and what permissions they grant, the answer to "can our people use this?" shifts from "not yet" to "yes, with guardrails." Accelerating safe adoption by three to six months for 25–40% of knowledge workers captures significant productivity value.

Indicative saving **$150K–$400K**

03
Greater return from existing security investments

Browser security generates direct labor savings in two ways. First, identity hygiene remediation at scale: Data shows that for every 1,000 employees, an organization will typically have over 2,500 identity security vulnerabilities (missing MFA, weak or breached passwords). Without browser-level automation, each takes 5–10 minutes to resolve manually, translating to 26–52 FTE days per thousand employees. In-browser guardrails that prompt users to fix issues at the point of login resolve thousands of vulnerabilities without a single ticket. Second, investigation efficiency: Detecting attacks at the point of initial access via the browser accelerates investigations and reduces the likelihood of incidents escalating to where an external incident response provider is needed. Automated remediation recovers $25K–$35K, investigation efficiency saves $45K–$65K, and reduced analyst attrition from less manual triage avoids $15K–$25K in recruitment costs.

Indicative saving **$85K–$125K**

04

Reduced compliance and audit exposure

Most major compliance frameworks expect strong authentication and visibility into third-party applications, and several (PCI DSS 4.0, NYDFS Part 500, Cyber Essentials) explicitly require MFA. Shadow IT and unmanaged identities create compliance gaps most organizations don't know they have until an auditor finds them. The consequences are increasingly financial. The City of Hamilton had an $18.3M cyber insurance claim denied because MFA wasn't fully implemented. NYDFS has fined firms specifically for MFA failures, including OneMain ($4.2M), National Securities ($3M), and Travelers ($1.55M). A browser security solution discovers every application employees actually use from the login event, and it observes the authentication method, password strength, and MFA status. That replaces weeks of manual audit preparation and provides continuous evidence rather than a point-in-time snapshot. Automated inventory saves $8K–$25K and a 30% reduction in compliance-related financial event exposure represents $45K–$120K.

Indicative saving

$50K–$150K

05

Consolidated capability and reallocated spend

A browser security platform delivers across threat detection, AI governance, identity security, and investigation support — use cases that would otherwise require separate point solutions. AI governance is the most immediate example. Your browser security platform might already deliver app discovery, data sharing monitoring, OAuth consent tracking, and real-time policy enforcement. If it does, the case for a separate AI governance purchase weakens considerably. There's also a reallocation opportunity at renewal. Where browser-native capabilities overlap with SWG or CASB functions, organizations can reduce the scope of those contracts. They can then redirect the savings toward capabilities that address the current threat landscape.

Indicative saving

Renewal reallocation

$610K–$1.2M

Combined annual value, 1,000-employee company

Using conservative estimates: Direct labor savings, risk-adjusted cost avoidance, and accelerated productivity gains. Against that, Push lists at $5 per user per month, so a 1,000-employee company pays at most $60,000 a year.

Lead with what the board already cares about

Two arguments consistently unlock browser security budget. Both work because the board already cares about the problem, so security isn't asking it to take on a new risk.

Argument 01

AI adoption the board has already backed

Boards are pushing AI adoption to stay competitive, and they expect security to make it safe. That usage runs primarily through the browser for most users in the organization, so browser security answers a mandate the board has already set.

Argument 02

The breaches in the headlines

The highest-profile recent breaches started with a compromised identity on a cloud service, not an exploited endpoint or network. These attacks pass through the one layer existing tools don't instrument, so the request fills a gap rather than improving a control already in place.

Funding routes

Organizations that have already invested in browser security have funded it through several established routes:

- Dedicated AI visibility and control projects that needed browser-level instrumentation.
- Post-incident investment following a phishing or session hijacking breach that existing tools failed to detect.
- CASB, SWG, or RBI contract renewals where browser-native capabilities allowed them to reduce scope and cost.
- Hardware migration projects (such as Chromebook rollouts) where the cost savings funded the browser security investment.
- Net-new budget justified by ROI projections alone.

Most commonly, organizations build the net-new case using the framework above.

Managing investment risk

Browser security is a new category, and there will naturally be questions about how safe the investment is. The factors that de-risk it are practical:

- Extension-based deployment via existing MDM means no browser migration, no user retraining, and no workflow changes.
- Findings and detections surface from day one, compressing time-to-value to hours.
- Integration into existing SIEM, SOAR, and IdP workflows means you gain a new capability without taking on a new operational burden.
- The best browser security solutions continuously advance their detection, governance, and remediation capabilities as the threat landscape evolves, so the investment delivers recurring value over the full contract period rather than depreciating.

Proof of value is essential

Run a proof of value before finalizing the funding request. A PoV deployment generates findings specific to your organization: Real instances of employees being targeted, the actual scale of your identity attack surface, and concrete shadow IT and AI usage data. That evidence is often more compelling to a CFO than generic industry benchmarks, and it sharpens the projected value using data from your own environment. The evaluation guidance in Section 5 covers how to structure a PoV that produces this evidence.

A Questions to ask vendors

The questions below map to the use cases and requirements defined in Section 4. Use them during vendor evaluations to test whether a solution can deliver the capabilities that matter most to your organization.

01 Account takeover prevention

- Show me what an identity attack surface looks like in your product. How do you ensure this includes unmanaged shadow identities we're not already aware of — those outside of our SSO coverage?
- What kind of vulnerabilities in this attack surface do you detect: Credential issues, missing MFA, etc.? Show me how your product remediates identity vulnerabilities at scale and reduces our overall identity attack surface.
- Show me how your product detects attacks in the lead up to, and including, account takeover. What types of attack do you cover and how do you detect them?

02 Advanced phishing detection

- Show me a detection firing on a phishing page that has never appeared on any blocklist. What signals trigger it?
- Show me a detection firing on a phishing page delivered via a LinkedIn DM or a malicious search engine ad, not an email link. What changes about your detection coverage?
- Show me a credential entry guardrail blocking a corporate password from being submitted to an unauthorized domain.
- How do you detect phishing attacks that bypass the standard login flow: Consent phishing, device code phishing, or MFA downgrade attacks that force fallback from passkeys?

03 AI visibility and control

- Show me the AI inventory: Apps, extensions, and OAuth integrations. Can it distinguish corporate from personal account usage?
- Show me an OAuth grant to an unapproved AI app being blocked in real time.
- Show me what controls are available for managing AI usage. Can I set different enforcement modes per app, per user group, or per content pattern?

04 Identity posture hardening

- Show me what an identity attack surface looks like in your product. How do you ensure this includes unmanaged shadow identities outside of SSO coverage?
- What vulnerabilities do you detect across this attack surface: Credential issues, missing MFA, ghost logins persisting alongside SSO? Show me how the product remediates identity vulnerabilities at scale.
- Show me how you enforce MFA on applications outside of our SSO deployment.
- Show me the compliance reporting. Does it produce auditable MFA coverage evidence across apps the IdP doesn't manage?

05 Shadow IT discovery and OAuth integration governance

- Show me what a shadow IT inventory looks like in your product. Does it capture authentication context (login method, account type, corporate vs. personal) alongside the app itself?
- Show me an OAuth consent grant being intercepted in real time. Can I configure policies by scope, app category, or user group?
- Show me what controls I can use to crack down on shadow IT and guide users towards supported, sanctioned apps.

06 Browser extension security

- Show me how you detect an extension that was legitimate at install but has since changed ownership and pushed a malicious update.
- Show me the allowlisting workflow. What happens when a user tries to install something not on the list?
- How do you detect a malicious extension that passed the web store's own review process? What signals do you use beyond what the store already checks?

07 ClickFix and social engineering-based malware delivery

- Show me a ClickFix or *Fix-family detection firing. What signals does the detection use, and does it work against novel variants or only known-bad signatures?
- After a detection fires, what contextual information does the security team receive to investigate the incident and understand the nature of the threat or campaign?

08 Data loss prevention

- Can I stop users from uploading files into unsanctioned AI apps?
- Show me a sensitive-data paste being redacted or blocked. Which patterns can I match on?

09 Investigation and incident response

- Show me the session timeline for a phishing detection. Can I see what the user saw, what they entered, and what happened to the session?
- Show me how I can create custom detections that are specific to my organization.
- Show me the SIEM integration. Can high-confidence detections trigger automated response workflows?

10 Infostealer defense

- Show me a stolen session token being detected when it's replayed from a different browser. Does this work for sessions from unmanaged devices?
- Show me a user's compromised credential being detected. How is that detection made and how do you make sure it's a true positive?
- Show me which users have personal browser profiles synced on corporate devices.

Push Security

The Buyer's Guide to Browser Security

Push Security is on a mission to secure every web and AI interaction. Founded by red team and blue team experts, Push sits closer to your users than any other security tool, right inside the browsers they already use. That means your team can stop modern attacks, govern AI use, protect sensitive data and harden identities without a migration, and without adding to its workload.

Book a demo »

Get a free browser security assessment »