Browser Security for MSPs

Why the browser is now the highest-risk layer in your clients' environments, and how Island helps you cover it

Jason Makevich, CISSP

Executive summary

Most MSPs have spent years hardening endpoints, email, identity, and network controls. That work is foundational, and it's not going away.

The problem is simple: a growing share of business work now happens inside the web browser. Email lives in the browser. File sharing lives in the browser. Accounting, payroll, claims, case management, and CRM live in the browser. GenAI lives in the browser. So do the clicks, logins, sessions, approvals, uploads, downloads, and copy-paste actions that turn routine work into incidents.

That shift creates a gap. Traditional controls can reduce risk around the browser, but they usually can't enforce what users do inside the browser itself, at the exact moment risk happens. That's where phishing kits steal sessions, OAuth tricks capture app access, malicious websites deliver payloads, and unsanctioned SaaS and GenAI become a data leakage lane.

Browser security closes that gap by giving MSPs visibility and control at the point of use: the browser. Done right, it helps reduce session hijacking, strengthens defenses against modern phishing, limits data exfiltration to personal email or unsanctioned apps, and brings governance to browser extensions.

Island is built to deliver that browser security in a way that fits managed services. Island gives MSPs two delivery options under one platform:

- Island Browser Extension for managed endpoints where users keep Chrome, Edge, or Safari
- Island Browser when you want a dedicated work browser, including for unmanaged or higher-risk scenarios, and it can also be used on managed devices

With the Island for MSPs program from PORT1, Island is packaged for how MSPs operate: multi-tenant management, monthly per-user pricing, consumption-based billing with proration, billed in arrears, no long-term contracts, and no minimums. An NFR program is available.

To learn more, visit:
https://port1.io/islandformsps

Or email:
msp@port1.io

The workplace moved into the browser

Your clients didn't "go cloud" as a side project. They rebuilt daily work on top of SaaS and web apps. Identity became the front door, and the browser became the place where work actually gets done.

A useful way to think about this: most client environments now run **a large and growing number of cloud apps**. Okta reports the global average number of apps per customer has topped 100 for the first time. [1] That app count matters because each app adds logins, sessions, and data movement inside the browser.

When the browser becomes the primary work interface, it also becomes the primary attack interface.

GenAI didn't replace work, it blended into it

Most MSPs now support clients where users are using GenAI, whether leadership formally approved it or not. GenAI use isn't automatically a problem. The risk is where users access it and what they put into it.

From a security standpoint, there are three common behaviors that create most of the exposure:

1. **Users use personal GenAI accounts at work**
2. **Users paste sensitive business data into prompts**
3. **Teams adopt new tools before the MSP can govern them**

Netskope's 2026 Cloud and Threat Report highlights how common personal account use is for GenAI and how frequently sensitive data is sent to AI applications. It reports that 47% of GenAI users are using personal AI apps, and the average organization sees 223 incidents per month of users sending sensitive data to AI apps. [2]

This is a browser problem because GenAI usage, personal accounts, and copy-paste are browser behaviors. If you can't see and control them in the browser, you can't reliably govern them.

Why strong security programs still leave a browser gap

MSPs already do a lot to protect clients. Many have solid endpoint protection, good identity hygiene, and reasonable security baselines. That's all necessary.

But browser risk often slips through because many controls were built to observe or stop activity around the browser, not inside it. For example:

- You can block known bad domains, but users can still authenticate to a fake login page hosted on a brand new domain.
- You can enforce MFA, but attackers can steal a session token and bypass it.
- You can prevent malware execution, but users can still leak data into personal SaaS or personal GenAI from a managed browser.
- You can manage apps, but browser extensions can gain broad access to page content and sessions.

This is why browser security is now a default layer, not a specialty add-on. It's the layer where modern identity attacks, data leakage, and user-driven execution converge.

The browser-based threats MSPs need to plan for

The goal here is to explain what these attacks look like in plain terms, why they work, and why "we already have MFA and endpoint protection" isn't enough by itself.

Session hijacking, session token theft, and browser-in-the-middle attacks

If you last worked deeply on IT security in the 1990s, it helps to reset one concept:

When a user logs into a modern web app, they usually don't re-enter their password for every click. The app gives the browser a **session token** (often stored as a cookie or similar artifact). The browser then presents that token on each request as proof the user is already authenticated.

That token becomes the prize.

If an attacker steals a session token, they may be able to impersonate the user without needing the password and without triggering MFA again, depending on how the environment is configured.

One modern approach is the browser-in-the-middle technique, where the attacker places a proxy between the user and the real service, capturing session tokens during a legitimate login. Google Cloud threat intelligence has documented how quickly these attacks can compromise sessions across web applications. [3] MITRE also documents adversary-in-the-middle techniques as a common credential access method. [4]

What this looks like for an MSP:

- A user logs in successfully, but the attacker gets access too.
- The first “real” damage might be mailbox rules, invoice manipulation, funds transfer fraud, or SaaS data access.
- Logs often look like valid authentication because it was valid authentication.

Why browser security helps:

- Browser security can strengthen session handling and reduce token reuse abuse.
- It can help block known phishing infrastructure before the login occurs.
- It can reduce the chance users authenticate through attacker-controlled pages that proxy the session.

Device code phishing

Device code authentication exists so users can sign into services on devices that can’t easily open a full browser login screen, like a TV app or a CLI tool. The user is given a short code and is told to enter it at a legitimate login page.

Attackers abuse this by convincing users to enter an attacker-provided code into the legitimate login page, which ends up granting the attacker access.

Microsoft documented an active and successful device code phishing campaign by a threat actor tracked as Storm-2372, active since August 2024, using lures designed to resemble messaging app experiences. [5]

What this looks like for an MSP:

- The user thinks they are “linking a device” or “verifying sign-in.”
- The login page is real, so users feel safe.
- MFA can be satisfied because the user is performing the approval step.

Why browser security helps:

- Browser security can reduce exposure to the lure pages that drive users into these flows.
- It can add friction and policy around risky authentication steps in the browser.
- It gives you telemetry to spot risky sign-in paths that are hard to see from traditional controls alone.

OAuth consent abuse and ConsentFix

OAuth is a standard way for an app to get limited access to a user's account without asking for the user's password. It's behind many "Sign in with Microsoft/Google" flows and many "Allow this app to access your account" prompts.

OAuth is described in the OAuth 2.0 framework (RFC 6749). [6]

The good version of OAuth:

- A user authorizes a legitimate app to access specific resources.
- The user sees a consent screen describing what access is being granted.
- The user can revoke access later.

The abusive version:

- An attacker registers a malicious app or compromises a legitimate one.
- The user is tricked into granting access.
- The attacker now has durable access through tokens and granted permissions, even if the user changes their password.

ConsentFix is a newer twist that blends browser-based social engineering with OAuth abuse to hijack accounts. Recent reporting describes ConsentFix as combining ClickFix-style guidance with OAuth consent phishing to compromise Microsoft accounts. [7] [8]

What this looks like for an MSP:

- The user thinks they are approving something normal, often during a busy moment.
- Conditional access controls may not block it if the flow looks legitimate.
- The attacker gains access without "traditional" malware on the endpoint.

Why browser security helps:

- Browser security can restrict risky consent actions and reduce exposure to phishing pages driving the approval.
- It can give you visibility into unusual consent activity and suspicious sign-in steps.
- It can help reduce the time between a consent event and your awareness that it happened.

ClickFix and copy-paste driven execution

ClickFix is a social engineering technique where a web page convinces the user they need to complete a “verification” or “fix” step, then guides them into copying and pasting content that results in malicious execution on the device.

Microsoft has documented ClickFix as a growing technique, targeting thousands of devices daily, exploiting users’ tendency to follow instructions to resolve issues, with the end goal of information theft and exfiltration. [9]

MITRE documents this broader behavior as “User Execution: Malicious Copy and Paste.” [10]

What this looks like for an MSP:

- It begins as a browser interaction, not a downloaded file.
- Users follow what looks like a legitimate support step.
- The end result is malware, remote access tools, or credential theft.

Why browser security helps:

- Browser security can reduce access to known malicious pages used in these campaigns.
- It can add controls around risky user interactions that are frequently used in these attacks.
- It can help you identify and respond faster when this behavior occurs.

Malvertising, redirects, and “normal browsing” that becomes an incident

Malvertising is malicious advertising delivered through ad networks or embedded on sites that monetize aggressively. Users click a result or visit a site, an ad chain redirects them, and they end up at a page designed to deliver malware, steal credentials, or capture data.

Microsoft reported a large-scale malvertising campaign detected in early December 2024 that impacted nearly one million devices globally, with redirection chains ultimately delivering information-stealing payloads. [11]

What this looks like for an MSP:

- Users don’t have to install “a suspicious app.” They may only be browsing.
- The payload often targets browser data, including saved credentials and session artifacts.
- The first symptom might be account takeover rather than a clear malware alert.

Why browser security helps:

- Browser security can reduce exposure to risky categories and known malicious destinations.
- It can enforce safer handling of downloads and prevent risky data movement at the last mile.
- It can add another layer of defense in the exact place the redirect chain operates.

Browser extensions: malicious, compromised, and “legit but risky”

Extensions are one of the most overlooked risk surfaces in most SMB and mid-market environments.

A browser extension isn’t just a bookmark. It’s code running inside the browser, often with permissions to read page content, modify pages, capture inputs, and interact with sessions. Chrome and browser vendors warn users during install and update because some permissions effectively grant access to everything the user does in that browser. [12] [13]

There are three categories of extension risk MSPs need to account for:

Malicious extensions
These are created with harmful intent, then distributed through stores or side-loading.

Compromised or hijacked extensions
These start out legitimate, then get acquired or updated with malicious functionality. GitLab reported a cluster of malicious Chrome extensions impacting at least 3.2 million users, spanning diverse “useful” functionality like screen capture, ad blocking, and keyboards. [14]

Legit but risky extensions
Many popular extensions request broad permissions such as the ability to read and change data on websites. A coupon extension like Honey explains that Chrome’s permission wording is broad and that its intent is limited to shopping workflows, but the permission scope itself is still expansive. [15] Writing assistant extensions can also introduce risk in regulated environments because they handle sensitive text, and you may not want that running across every web app your client uses.

The point isn’t “ban all extensions.” The point is that unmanaged extension sprawl becomes an ungoverned code execution surface inside the browser.

Why browser security helps:

- It turns extension governance into a manageable policy problem instead of a user-by-user cleanup project.
- You can default-deny and explicitly allow approved extensions.
- You can keep users productive while reducing the odds that a single extension becomes a data leakage or session theft path.

Shadow SaaS, personal cloud accounts, and shadow AI

Shadow IT used to mean a rogue server in a closet. Now it usually means:

- A user signs up for a SaaS tool with a personal email
- A team adopts a web app with no formal approval
- Users move files through personal storage, personal email, or personal GenAI

The risk isn't that the tool is "bad." The risk is that the MSP and the client can't govern it:

- No consistent access controls
- No consistent logging
- No retention or legal hold
- No DLP controls
- No reliable offboarding

Netskope reports that personal cloud app instances are a major insider risk, involved in 60% of insider threat incidents, and that 47% of GenAI users use personal AI apps. [2]

Why browser security helps:

- It gives MSPs visibility into what cloud apps and AI tools are being used on managed devices.
- It helps enforce data handling rules, including blocking uploads and copy-paste to unsanctioned destinations.
- It gives the MSP reporting that can be used to align the client's leadership on what's happening and what needs to be governed.

Browser vulnerabilities and the patch gap

Browsers are among the most complex and frequently targeted applications in any environment. Even with good patch management, there is always a window between:

- a vulnerability being exploited
- a patch being released
- the patch being deployed everywhere

Google's Chrome stable channel updates regularly include fixes for actively exploited vulnerabilities, including cases where Google explicitly notes exploitation in the wild. [16]

Why browser security helps:

- It provides additional guardrails for risky browsing and data handling while patch cycles catch up.
- It reduces exposure to the most common exploit delivery paths, including malicious pages, redirects, and risky downloads.

What browser security should deliver in a managed services stack

If you're going to treat browser security as a default layer, it should do real work. The outcomes MSPs tend to care about fall into four buckets:

1. **Session hardening and safer authentication**
   Reduce session theft risk, add controls around risky sign-in behaviors, and improve visibility into session-related abuse.
2. **Phishing and malicious site defense where the click happens**
   Reduce exposure to malicious destinations and modern phishing lures that bypass older filters.
3. **Last-mile data protection**
   Control copy-paste, uploads, downloads, and other data movement paths so sensitive data stays in approved places.
4. **Extension governance**
   Prevent extension drift, reduce the risk from malicious or overly permissive extensions, and keep policy consistent across clients.

A fifth bucket is often the difference-maker for MSP adoption: **reporting**. If the MSP can show the client what's happening, it becomes far easier to drive governance decisions and security upgrades.

How Island fits: two delivery options, one platform

Island is designed to put browser security into a form MSPs can deploy, manage, and standardize.

Island Browser Extension (managed endpoints)

For company-owned, managed devices where users keep their current browser, the Island Browser Extension helps MSPs add browser-layer controls and reporting. It's commonly deployed via RMM, Intune, or GPO.

In practice, MSPs often start with a report-only posture to establish baselines and then enable enforcement where it's useful and appropriate.

Island Browser (managed or unmanaged endpoints)

Island Browser is a dedicated work browser when you want the strongest control set and a consistent work environment. It's particularly useful when you need a governed work surface on devices you don't fully manage, but it's also valuable for certain high-risk users and workflows on managed devices.

The key point for MSP operations is that you can use the extension broadly across managed fleets, and use the browser where the use case demands stronger control or a dedicated work environment.

Island for MSPs program (built for how MSPs operate)

Island for MSPs is packaged to match the operational and billing reality of managed services:

- Multi-tenant management
- Monthly billing
- Per-user pricing
- Prorated, consumption-based billing
- Billed in arrears
- No long-term contracts
- No minimums
- NFR program available

Making browser security affordable, standard, and easy to justify

The biggest adoption blocker usually isn't deployment. It's the belief that browser security is optional.

In 2026, it isn't optional if you sell "cybersecurity" as part of your managed services. The browser is where users authenticate to business systems, where sessions live, and where data gets moved into SaaS and GenAI. Leaving that layer ungoverned creates a gap that attackers and accidental data leakage both take advantage of.

The MSP economics are straightforward

Browser security tends to pay for itself in a few ways MSPs recognize immediately:

- **Fewer preventable incidents that turn into after-hours work** (phishing-driven account takeover, consent abuse, session theft, malware delivery through browsing)
- **More governable environments** because you can see risky behavior and drive a decision instead of guessing
- **Better QBR conversations** because you can show what apps are in use, where data is going, and what risky actions are happening
- **More leverage when you recommend additional controls** because your client can see the problem and the progress

Clients will pay for what they can see

A lot of security value is invisible. Browser security is different.

When users and client leadership can see that there is protection in the browser, it reinforces that security is being handled every day. It also keeps security top of mind without relying on fear-based messaging. The goal isn't to create a false sense of safety. The goal is to reduce risk while making security visible enough that clients understand what they're paying for.

This is especially true when the MSP uses browser reporting in the right situations, such as showing leadership what tools are being used on company-owned devices, where personal accounts create risk, and what shadow SaaS or shadow AI is in play.

PORT1 enablement is part of the program

One reason enterprise-grade platforms often struggle in the channel is that MSPs are left to figure out packaging, deployment, and go-to-market on their own.

PORT1 exists to prevent that. The Island for MSPs program includes real enablement so MSPs can deploy and support Island with confidence, including technical enablement and go-to-market support aligned to managed services delivery.

MSP partner testimonials

“When we deployed Island to all our managed endpoints, we were able to use the reporting from Island to help support some other recommended solutions that ended up more than covering the cost of adding Island to our entire stack really quickly!”
— James S., MSP Owner

“We were about to raise our per-seat pricing across all our clients. But when we decided to add Island to our stack, we positioned the increase as a measurable security upgrade. We got zero pushback from clients, and now we’re delivering far better security outcomes than before.”
— Brandon P., MSP Owner

Next steps

If you want to make browser security a default layer in your stack, Island is built for that job, and Island for MSPs is built for your operating model.

Learn more and reach out here:
https://port1.io/islandformsps

Or email:
msp@port1.io

References

1. Okta, “2025 Businesses at Work Report.” https://www.okta.com/resources/whitepaper-businesses-at-work/
2. Netskope Threat Labs, “Cloud and Threat Report: 2026.” https://www.netskope.com/resources/cloud-and-threat-reports/cloud-and-threat-report-2026
3. Google Cloud (Mandiant), “BitM Up! Session Stealing in Seconds Using the Browser-in-the-Middle Technique.” https://cloud.google.com/blog/topics/threat-intelligence/session-stealing-browser-in-the-middle
4. MITRE ATT&CK, “Adversary-in-the-Middle (T1557).” https://attack.mitre.org/techniques/T1557/
5. Microsoft Security Blog, “Storm-2372 conducts device code phishing campaign.” https://www.microsoft.com/en-us/security/blog/2025/02/13/storm-2372-conducts-device-code-phishing-campaign/
6. IETF Datatracker, “RFC 6749: The OAuth 2.0 Authorization Framework.” https://datatracker.ietf.org/doc/html/rfc6749
7. BleepingComputer, “ConsentFix debrief: Insights from the new OAuth phishing attack.” https://www.bleepingcomputer.com/news/security/consentfix-debrief-insights-from-the-new-oauth-phishing-attack/
8. Push Security, “ConsentFix: Browser-native ClickFix hijacks OAuth grants.” https://pushsecurity.com/blog/consentfix
9. Microsoft Security Blog, “Think before you Click (Fix): Analyzing the ClickFix social engineering technique.” https://www.microsoft.com/en-us/security/blog/2025/08/21/think-before-you-clickfix-analyzing-the-clickfix-social-engineering-technique/
10. MITRE ATT&CK, “User Execution: Malicious Copy and Paste (T1204.004).” https://attack.mitre.org/techniques/T1204/004/
11. Microsoft Security Blog, “Malvertising campaign leads to info stealers hosted on GitHub.” https://www.microsoft.com/en-us/security/blog/2025/03/06/malvertising-campaign-leads-to-info-stealers-hosted-on-github/
12. Chrome Web Store Help, “Permissions requested by apps and extensions.” https://support.google.com/chrome_webstore/answer/186213?hl=en
13. Chrome Developers, “Declare permissions (Chrome Extensions).” https://developer.chrome.com/docs/extensions/develop/concepts/declare-permissions
14. GitLab Threat Intelligence, “Tech Note: Malicious browser extensions impacting at least 3.2 million users.” https://gitlab-com.gitlab.io/gl-security/security-tech-notes/threat-intelligence-tech-notes/malicious-browser-extensions-feb-2025/
15. Honey Help Center, “Why does my browser indicate that Honey’s extension can ‘read and change’ all my data?” https://help.joinhoney.com/article/28-what-is-the-read-and-change-data-permission-in-chrome
16. Chrome Releases, “Stable Channel Update for Desktop” (mentions CVE-2025-2783 exploitation in the wild). https://chromereleases.googleblog.com/2025/03/stable-channel-update-for-desktop_25.html

