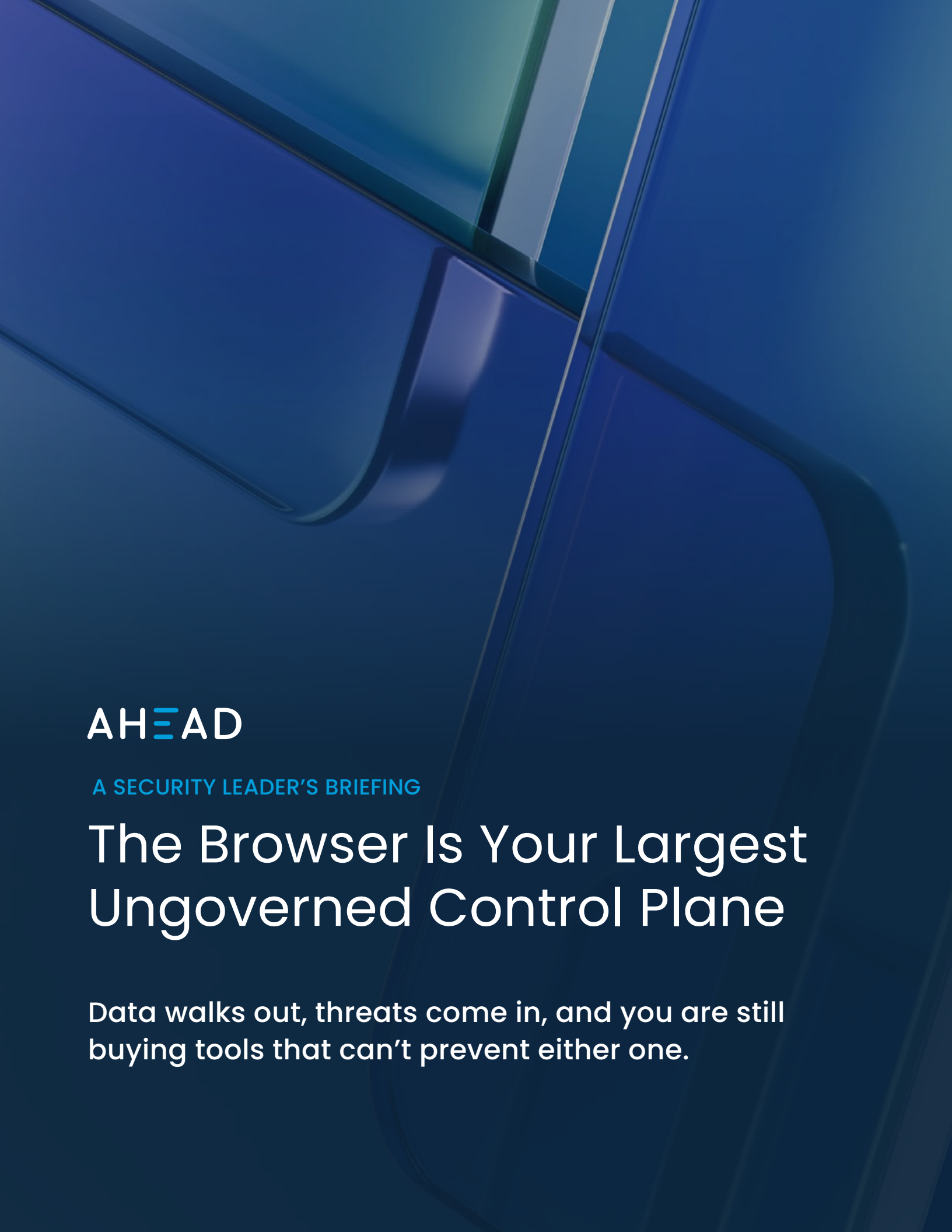




AHEAD

A SECURITY LEADER'S BRIEFING

The Browser Is Your Largest Ungoverned Control Plane

Data walks out, threats come in, and you are still buying tools that can't prevent either one.

Three Numbers That Reframe the Browser

The vast majority of work happens in the browser, so that's where attackers go for credential theft, and where the data leaks out.

WHERE INTRUSIONS BEGIN

48%

of investigated incidents
involved a web browser

UNIT 42, 2025

HOW ATTACKERS GET IN

52%

of basic web-app attacks
used credentials, typed in

VERIZON DBIR, 2025

HOW DATA LEAVES

67%

of employees using GenAI at
work sign in with personal,
unmanaged accounts

VERIZON DBIR, 2025

One session, three problems. The browser is the only layer positioned to see and stop all three.



In This Paper

The problem. Six gaps exist in every stack running on an ungoverned browser:

- AI tools nobody can name
- agents acting with user credentials
- data leaving through paste and upload
- intrusions that hide in the browser
- extensions nobody has inventoried
- a decryption tier inspecting less traffic than you realize

The argument. The browser you use isn't a matter of personal preference anymore. Network telemetry sees traffic. Endpoint telemetry sees processes. But neither one can effectively protect against modern evasive threats that land in the browser or prevent the moment sensitive data leaves the browser. Foundational controls belong wherever risk concentrates, which is why the browser can and should now serve as an essential control plane for your security.

The test. AHEAD deployed Prisma Browser in a controlled lab. Our internal Red Team performed various stress tests in order to validate what we could see, what we could stop, and where the control boundary ended.

The big question. Can you replace SSE with the browser itself?

The scorecard. Four numbers to report to the board after rollout, one for each outcome.

YOUR NEXT STEP:

A Security Workshop with AHEAD

If anything you read here sounds familiar, the next step is a working session with our specialist solutions security team. You'll leave with three things:

- **A read on your exposure.** Which of the six gaps are open in your environment today, and which ones your existing stack already covers.
- **An architecture view.** Where a secure enterprise browser sits alongside your SSE, DLP, and identity investments, what it complements, and where it may reduce operational complexity.
- **A recommendation you can act on.** Product and architecture guidance for your environment, grounded in AHEAD's lab evaluation and customer deployment experience.

Executive Summary

Every CISO is feeling pressure to enable AI adoption, without surrendering control over where enterprise data ends up. Boards can't wait for a perfect control framework. Regulatory obligations are only increasing. Meanwhile, the workforce is already moving faster than company policy, using AI tools the security team has no visibility into. Speed and sovereignty have to hold.

Most security teams have already accounted for these issues on their roadmaps:

Real-time visibility into which AI tools employees actually use, not just the apps you have corporate accounts with.

Data loss prevention that stops sensitive information from reaching an unauthorized destination, including customer records pasted into a chatbot.

A governance posture for agentic AI, as autonomous agents begin executing activities and transactions using employee credentials.

Identity-centric access that travels with the user into whatever SaaS or AI application they open next.

And yet, most security teams are still falling behind on all four.

That's because every single one of these problems lives at the same layer: the browser. Gartner identifies the browser as the primary access method for most modern corporate applications. The browser is where SaaS work happens, and where sensitive data is getting pasted into services nobody has eyes on. It's where agents will soon be acting on a user's behalf as well. This all compounds into a larger problem, because it's the one layer your enterprise security stack wasn't designed to govern. Network telemetry sees traffic. Endpoint telemetry sees processes. But there's a massive blind spot when it comes to what users are presented with inside an authenticated session.



WHAT EACH LAYER SEES AT THE MOMENT OF ACTION

Scenario: an employee pastes a customer list into a consumer chatbot.

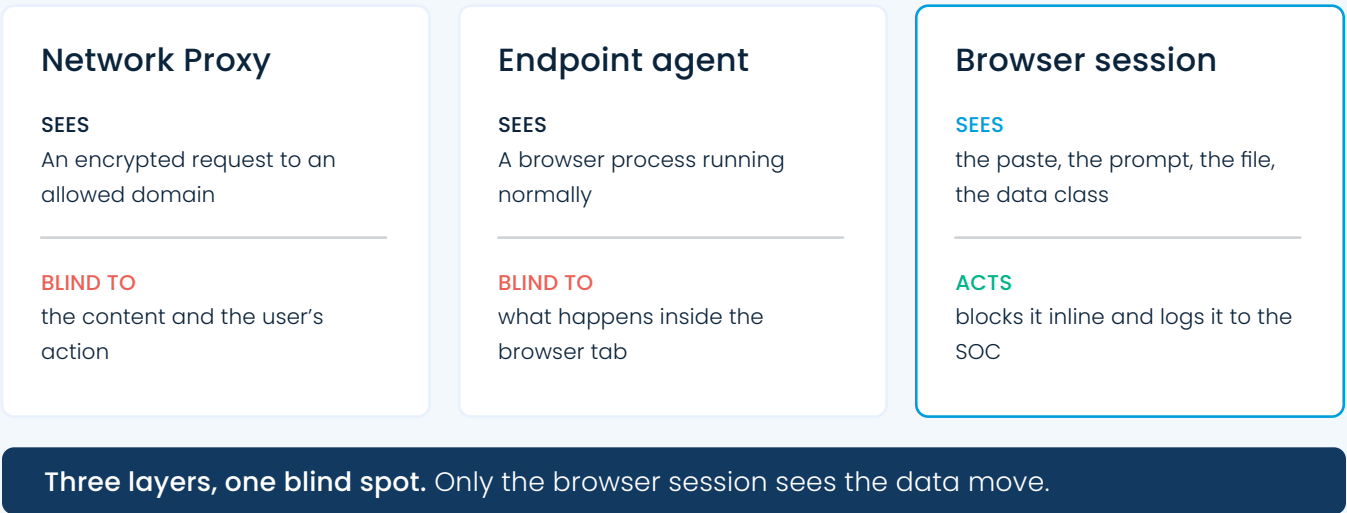


Figure 1. Network and endpoint tools see only an encrypted request and a running process. The browser session is the one layer that sees the user action itself, and can stop it inline.

Foundational controls are defined by where the greatest risk concentrates. Email earned that status when business communication and threat delivery converged there. Identity earned it once access decisions left the perimeter. The browser now belongs in the same category.



This is a fundamental industry inflection point. Identity and the enterprise browser are the defining constructs of next generation architecture, and they are precisely how we have engineered our own security program.

Grant Sewell, Chief Security Officer, AHEAD

Managed Chrome or Edge, layered with existing CASB, DLP, and SSE controls, can take you a long way. Most teams already run a similar set-up and assume that means they're covered. Which they are, for the most part. What they actually have is a regular Chrome or Edge browser, with centralized control for IT that allows them to push policy and run configuration for extensions, sync, updates, and URL or category restrictions. Where it falls short is governing user activity inside the session itself, and protecting against threats that target the browser. Teams lack full control over and visibility into browser extensions, and can't inspect each AI prompt being submitted, or supervise agentic activity. That's where your data moves, and that distinction, between configuration and action, is what the rest of this paper is about.

BRING BROWSER CONTROLS TOGETHER

Prisma Browser brings sensitive actions into one browser policy framework.

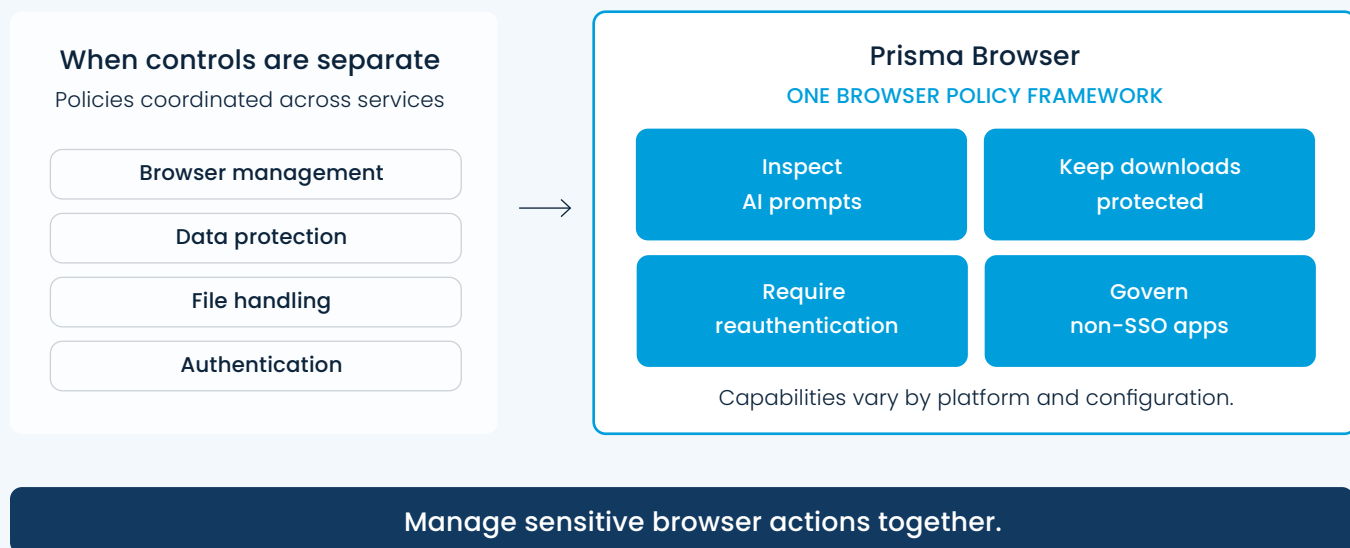


Figure 2. Prisma Browser brings controls for sensitive browser actions into one policy framework.

AHEAD set out to explore this gap, by evaluating Palo Alto Networks Prisma Browser against the questions a CISO would ask:

- What can we see?
- What can we stop?
- What evidence reaches the SOC?
- Where does the control boundary end?

Where the capability is strong, we'll say so. Where the outcome depends on configuration, we'll call it out. The enterprise browser is an essential control plane missing in your current security architecture, and Prisma Browser is the recommendation we can personally stand behind.



KEY FINDINGS FROM OUR EVALUATION

- **AI Tool Visibility and Control.** Policy is enforced on data submitted to both sanctioned and unsanctioned AI tools, at the point of user action and before reaching its destination.
- **Agentic AI Defense.** Human-in-the-loop approval workflows are validated, and additional capabilities to delineate agentic actions from human actions are upcoming.
- **DLP at the Application Layer.** Prisma Browser blocked the tested unencoded sensitive data transfers, while the Base64-encoded test content was not blocked. That result is expected as a general bypass to DLP controls, not just within Prisma Browser, and can be mitigated with the proper configuration.
- **Browser Delivered Threat Prevention.** Phishing, malicious downloads, and C2 callbacks are intercepted at delivery, before anything writes to disk or your EDR has to play backstop.
- **Extension Supply Chain Governance.** Only approved extensions are allowed, and controls govern what data those extensions can read.
- **No Decryption Dependency.** Session-layer enforcement works without TLS decryption, which dramatically reduces the operational overhead of certificate management, breaking applications, and performance concerns. Additionally, modern threats that evade traditional network inspection can be blocked by the browser.

All of this depends on getting the browser to the full target population. The Chromium familiarity and ease of migration remove some of the most common reasons user adoption tends to stall (see Section 3.6).



Table of Contents

- 1.1 Project Team & Scope 09
- 2. Six Gaps That Exist Without a Governed Browser 10**
 - 2.1 The AI Tool Blind Spot 11
 - 2.2 Agentic AI as a New Attack Surface 13
 - 2.3 Data Exfiltration at the Application Layer 15
 - 2.4 The Browser as a Primary Attack Surface 16
 - 2.5 The Extension Supply Chain Gap 17
 - 2.6 The Decryption Gap 19
- 3. What We Tested and What We Found 21**
 - 3.1 AI and Data Security Controls 22
 - 3.2 Agentic AI Defense 24
 - 3.3 Forensics and the Limits of the Control 26
 - 3.4 Threat Prevention 27
 - 3.5 Extension Governance 28
 - 3.6 End User Adoption and Onboarding 30
- 4. Can the Browser Replace Your SSE Stack? 32**
- 5. Conclusion 33**
 - 5.1 Deployment Considerations 35
- About the Authors 37**
- Appendix 38**
 - Test Environment 38
 - Sources 38

1.1 Project Team & Scope

We designed this evaluation to answer questions your CISO will ask: Can browser security close the gap in AI governance, data protection, and threat prevention without sacrificing operational efficiency?

AHEAD's testing covered AI and data security controls, human-in-the-loop workflows, DLP enforcement under simulated exfiltration conditions, threat prevention, extension governance, and end user onboarding and adoption experience.

Explicitly out of scope: Unmanaged and BYOD device scenarios, mobile browser variants, and zero day vulnerability research in the browser engine itself.

Test environment details are documented in the Appendix.

SIX SECURITY GAPS, ONE CONTROL PLANE

Six security priorities. One surface where they can finally be governed together.

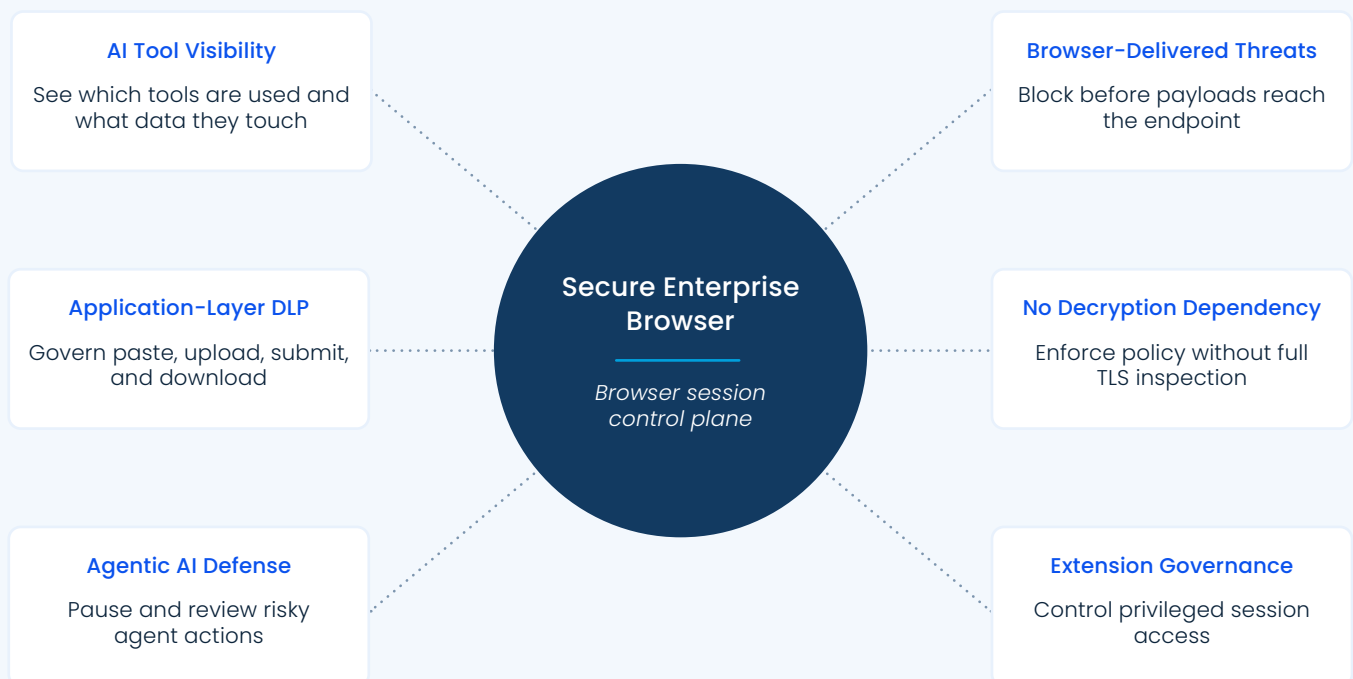
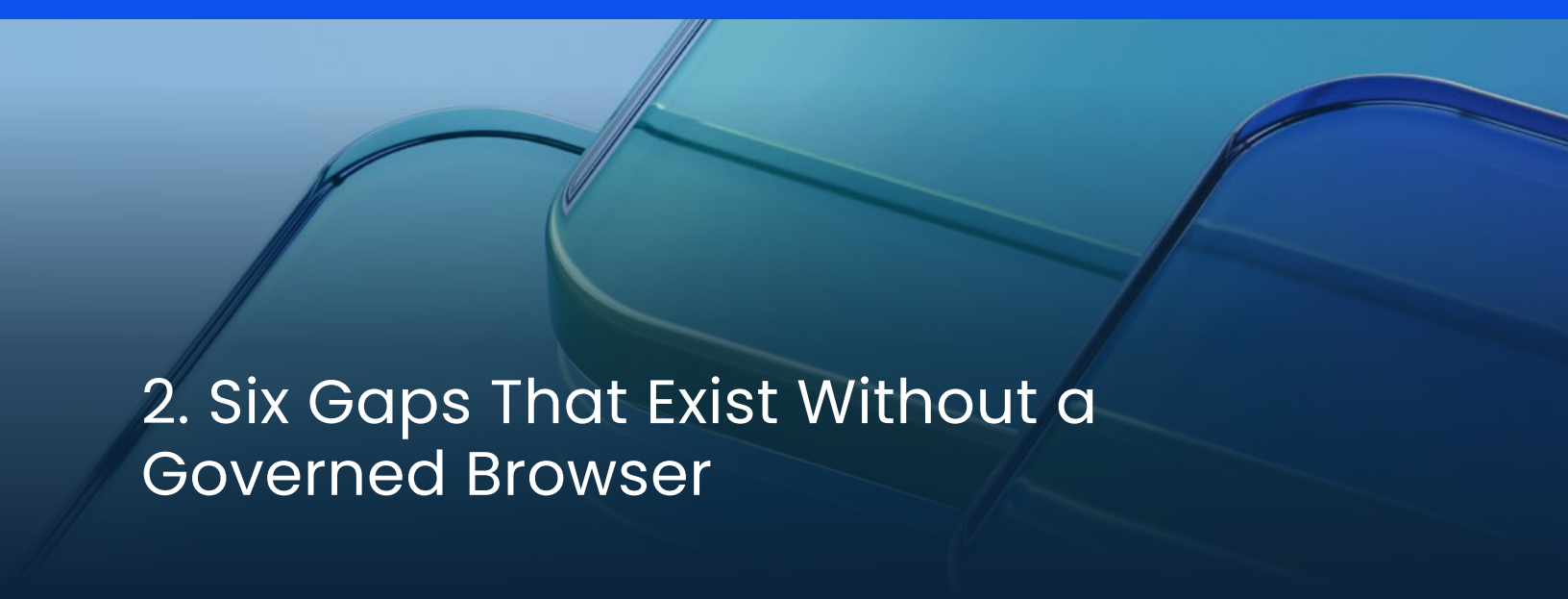


Figure 3. Six security gaps converging on a single control surface.

An abstract background image featuring several overlapping, rounded rectangular shapes in various shades of blue, creating a sense of depth and modernity.

2. Six Gaps That Exist Without a Governed Browser

Most organizations are trying to accelerate AI adoption. At the same time, they're struggling to improve data protection and workforce productivity, while reducing the complexity and cost of security enforcement.

Can you answer the following six questions right now?

1. Name every AI tool your employees used this week, including the ones IT never approved.
2. Can you stop an AI agent before it acts under an employee's credentials?
3. Can you control what data employees share with SaaS or AI tools before it leaves the browser?
4. Can you block phishing and malicious downloads before they reach the endpoint?
5. Name every browser extension running across the workforce, and what each extension can read inside an authenticated session.
6. Can you replace your SSE and SASE decryption challenge with the browser itself?

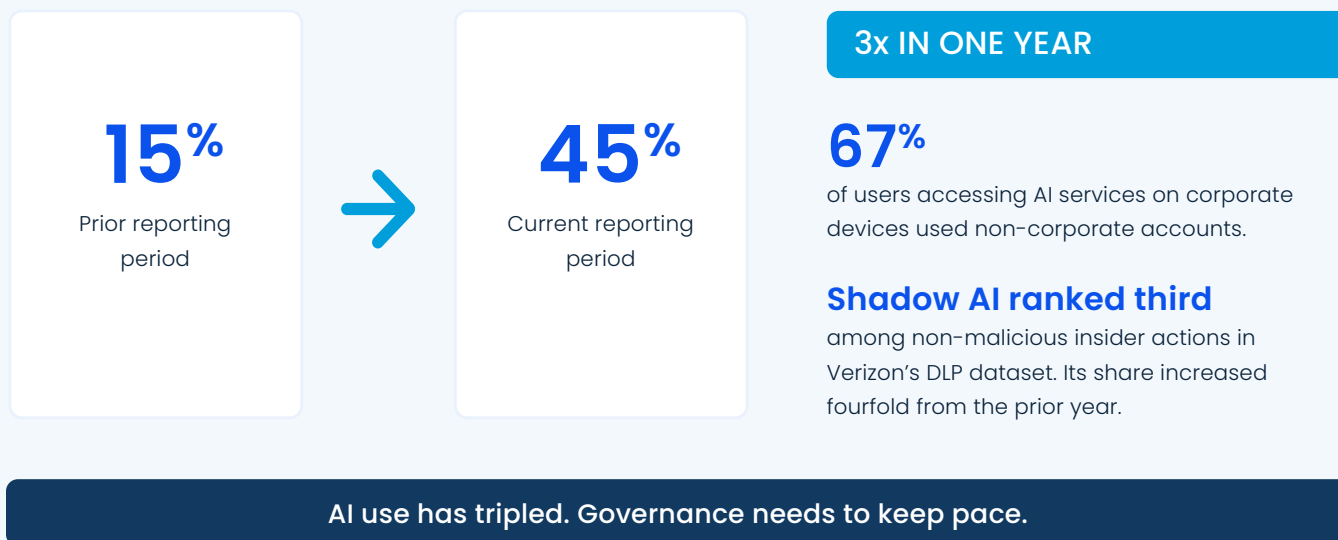
Each of these six gaps converges on the same architectural surface — the browser, naturally — with each underserved by the controls you already have in place.

2.1 The AI Tool Blind Spot

Most organizations can't name which AI tools their employees are using this week, let alone what data has been submitted to them. AI tool adoption has outpaced governance in almost every enterprise environment. Employees are sharing everything from customer records to source code to internal and proprietary data with public chatbots and open-source models. Verizon's 2026 DBIR found that in a single year, regular AI use on corporate devices has roughly tripled, with activity running on personal accounts the security team never provisioned, and therefore can't see into, govern, or revoke (Figure 4). It's not low-stakes data being shared either. Across DLP events, Verizon found that source code was the most common item fed into AI tools, by a wide margin. That's your IP leaving the building through a paste buffer.

REGULAR AI USE HAS TRIPLED

Regular AI use on corporate devices, including authorized and unauthorized tools.



Source: Verizon, 2026 Data Breach Investigations Report.

Figure 4. Regular AI use on corporate devices tripled from 15% to 45% compared with the prior reporting period, including both authorized and unauthorized use. (Verizon 2026 DBIR)

Inline network DLP can inspect and block sensitive data before it reaches an external service. For encrypted web traffic, that depends on decrypting it, routing the traffic through inspection, and supporting the application's protocols. Exemptions and unsupported traffic can leave gaps. Browser-based controls provide another enforcement point inside the session, removing the decryption complexity without depending on network decryption for those browser-session controls.

ON GARTNER'S CALL TO BLOCK AI BROWSERS (DECEMBER 2025)

Gartner's December 2025 guidance recommends blocking AI browsers until enterprise-ready versions are generally available. AHEAD agrees that agentic browsing needs clear controls before it reaches production. We recommend keeping unapproved AI browsers restricted while providing approved ways for employees to use AI. A governed browser can support that approach, but its agentic features still need to be evaluated against the organization's security requirements. An enterprise label alone does not establish that an agent can safely act with a user's access.

None of this lets anyone off the hook on fundamentals. Verizon's findings suggest that the most essential changes are the most basic ones: asset inventory, access control, and monitoring.

AHEAD agrees completely. Our only point is that the browser is where those fundamentals should be enforced.

For your CISO, the bigger issue isn't unsanctioned AI usage. Lack of visibility means your security team won't be able to answer a board, regulator, customer, or legal when they ask what enterprise data has been submitted to external AI systems, by whom, and under what policy.



2.2 Agentic AI as a New Attack Surface

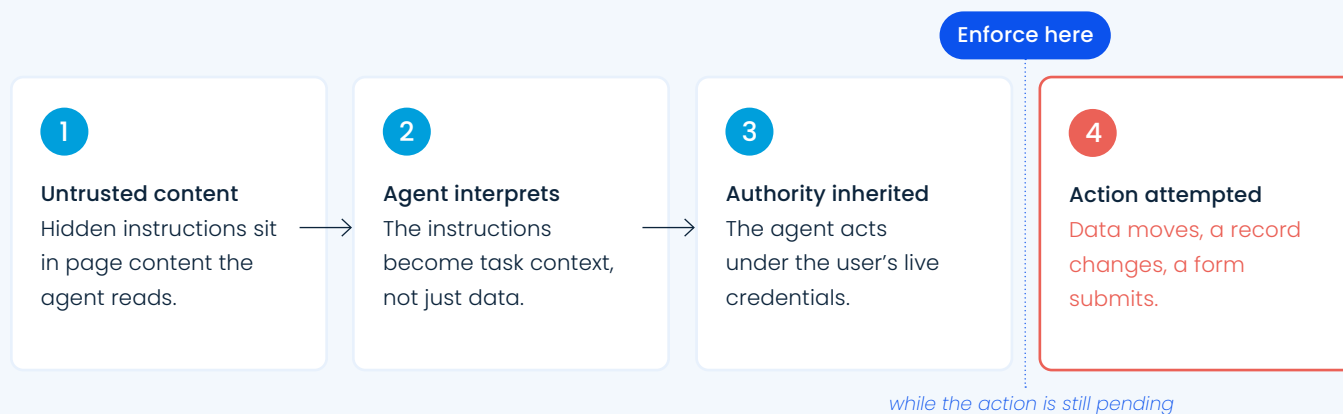
Agentic AI doesn't just respond to requests, like chatbots. Agents act, using the user's credentials to navigate web applications, update records, move data, and trigger workflows inside authenticated sessions.

In a controlled experiment, Palo Alto Networks Unit 42 found that AI-assisted attacks cut the time to data exfiltration to 25 minutes. Human review only helps while the action is still pending, not after the agent has already moved the data.

Traditional controls assumed a reasonably stable chain between user intent, authenticated identity, and executed action. But an agent can read untrusted page content, interpret it as task context, inherit the user's active session, and attempt an action before a human even has a chance to review.

HOW UNTRUSTED CONTENT BECOMES AN ACTION

Scenario: an agent reads a page with hidden instructions, inside an authenticated session.



Detection after the agent acts is forensics, not prevention. The control has to interrupt the step before it completes.

Figure 5. Untrusted content becomes an agent action in four steps. The only reliable intercept point is before the action completes, not after.

Prompt injection is the most obvious example, but it's not the whole problem. The deeper issue is delegated authority: What is the agent allowed to do, and when should it be interrupted? When does a material change in plan require reauthorization? Can an audit trail prove if the action came from a user or from an agent acting with that user's authority? These questions apply to agentic browsers, but also to agentic extensions on any browser. Another growing concern is desktop applications, like Claude Code, that can initiate actions within the system's web browser.

Just a few of the risks:

Unintended agent actions: Agents can modify records, submit forms, move data, or trigger workflows without real-time human review.

Identity blind spots: Existing logs may show which user was authenticated, but not whether the user or an agent acted.

Plan drift: An approved task slowly becomes riskier over time, as the agent changes path based on page content, prompts, errors, or application state.

Tool and extension exposure: Agentic capability often arrives through browser extensions, copilots, plugins, or tool integrations that behave like privileged software inside the session.

Persistent state risk: Some agents retain memory or workflow state across sessions, which means poisoned or unattributed context can influence later actions.

This is why the browser matters architecturally. For browser-mediated agentic activity, it's one of the few places where user context, application state, data movement, and attempted action can be observed together. It's also where policy can be enforced before the action completes, rather than after the activity has been reduced to network traffic, endpoint process telemetry, or SaaS audit logs.

This doesn't mean browser controls will solve every agentic AI problem. Server-side agents, API-native agents, persistent memory stores, and back-end orchestration require separate controls. But for agentic activity that moves through the browser, security needs to operate inside the session, where authority is being exercised.

Prisma Browser addresses this specifically for browser-mediated workflows by inserting a human-in-the-loop approval.



2.3 Data Exfiltration at the Application Layer

Data loss prevention only works if it can reach the places where data actually moves. The majority of modern exfiltration no longer happens through the channels traditional DLP was built to inspect. Employees copy sensitive data into webmail, paste it into consumer SaaS, upload files to personal cloud storage, and print to PDF, all from inside authenticated browser sessions. IBM's 2025 Cost of a Data Breach Report found that organizations with high levels of shadow AI experienced higher average breach costs.

The data moves through a path where nothing is watching, and the detection clock starts running before anyone knows the tool is in use.

SHADOW AI AND BREACH COSTS (IBM COST OF A DATA BREACH REPORT, 2025)

- 20% of studied organizations reported a breach due to shadow AI.
- Organizations with high levels of shadow AI had average breach costs \$670,000 higher than those with low levels or none.
- The global average breach cost in the study was \$4.44 million.
- Customer PII was compromised in 65% of shadow-AI security incidents, compared with 53% overall. Intellectual property was compromised in 40%, compared with 33% overall.

Network DLP needs decryption to see any of this. Endpoint DLP watches the file system, not the inside of a browser tab. That results in partial coverage of what's now the primary exfiltration surface. AI tool submission is just one example of it.



2.4 The Browser as a Primary Attack Surface

Verizon's 2026 Data Breach Investigations Report analyzed more than 22,000 confirmed breaches, its largest breach dataset to date. Vulnerability exploitation became the leading identified initial-access vector, at 31 percent. Credential abuse remained significant: the report found it at some point in the breach progression in 39 percent of breaches. These findings support continued investment in vulnerability management and credential protection. Attackers are also increasingly stealing session tokens, which lets them take over an already authenticated session without needing a password or MFA. Because those tokens are stored in the browser, token theft reinforces the need for a hardened browser.

Unit 42's 2026 Global Incident Response Report found that browser-based activity featured in 48 percent of investigated incidents, while identity-driven techniques accounted for 65 percent of initial access.

Credential phishing, malicious downloads, and C2 callbacks all flow through browser sessions as well. Most of that traffic is also moving over HTTPS, where network inspection depends on decryption that's difficult to sustain at full scope. Endpoint detection often acts as a backstop only after payloads write to disk or processes spawn.

That makes the browser session layer the earliest possible interception point: before a credential is submitted, before a payload executes, before a C2 channel stands up. Most enterprise stacks still have no enforcement layer here at all.

THE EARLIEST INTERCEPTION POINT

One browser delivered intrusion, and where each control layer can act on it.

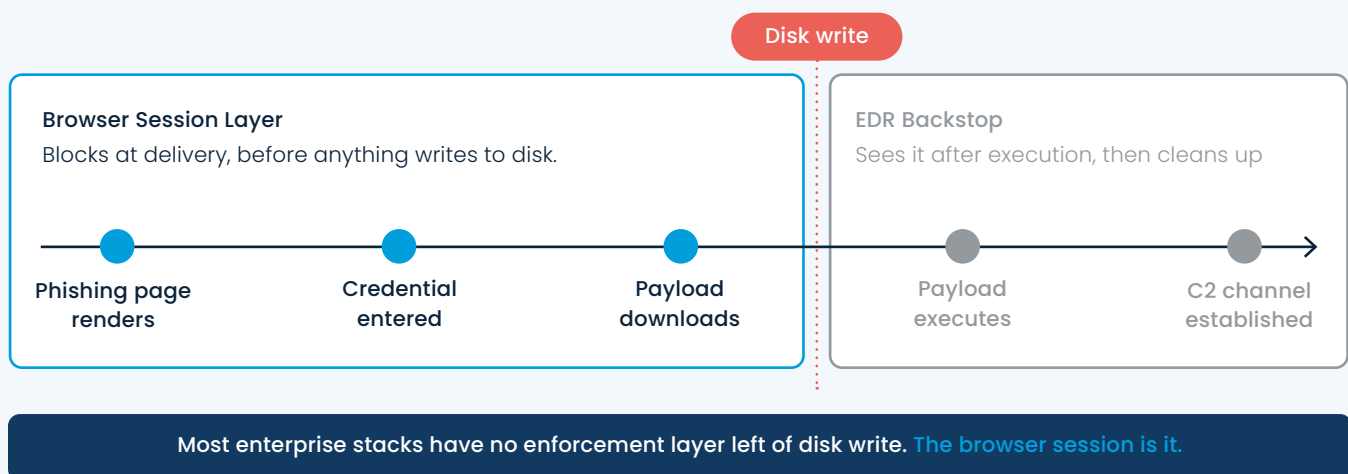


Figure 6. The browser session layer acts at delivery, before disk write. EDR sees it after execution.

2.5 The Extension Supply Chain Gap

Browser extensions operate with broad permissions inside authenticated sessions. They read webpage content, access clipboard data, and interact with SaaS applications on a user's behalf. Many enterprises have no systematic inventory of what extensions are installed across their workforce, much less a classification of what those extensions can access, or an allowlist policy. Endpoint and network controls weren't designed to control or secure this surface.

LayerX's Enterprise Browser Extension Security Report 2026 lays it out by the numbers.

EXTENSION RISK, BY THE NUMBERS (LAYERX, 2026)	FINDING
Users with at least one extension installed	99%
Users now running an AI extension	1 in 6
AI extensions reaching cookies and session, vs. other extensions	3x more likely
AI extensions with scripting permissions, compared with extensions overall	2.5x more likely
AI extensions carrying a known CVE	16.3%
Users with at least one extension that changed permissions in the past year	34%
Chrome Web Store extensions with no published privacy policy	71%

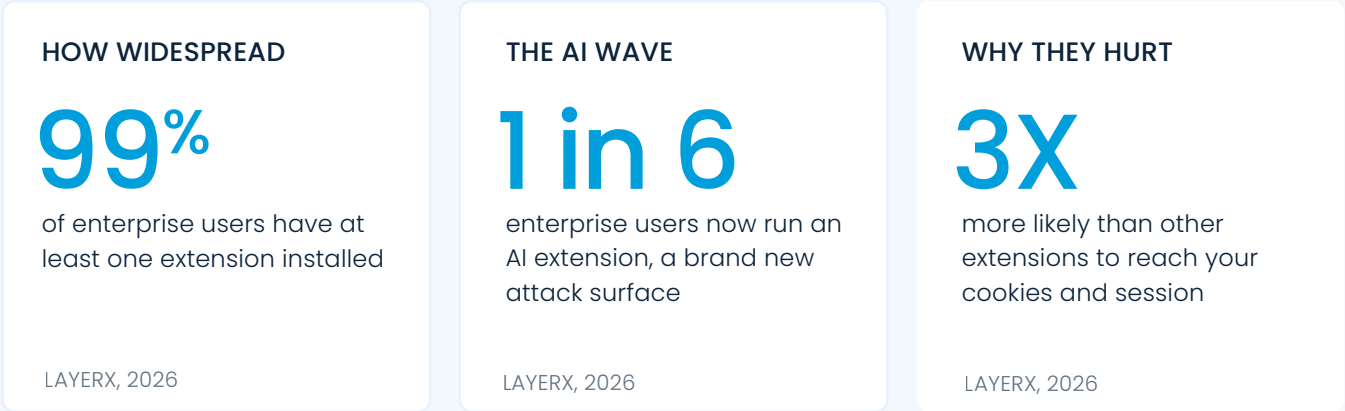


LayerX found that one in six enterprise users in its dataset had an AI extension installed. These extensions were more likely to request cookie access and scripting permissions than extensions overall. Their permissions can also change after installation. Chrome requires user acceptance when an ordinary extension update adds a permission that triggers a warning; administrator force-install policies handle permissions differently. Updates can also change how an extension behaves within permissions it already holds. Extension approval needs ongoing review of both access and behavior.

The December 2024 attack on Cyberhaven shows how an extension supply-chain compromise can reach users through software they already trust. A phishing attack gave the attacker access to publish a malicious version of Cyberhaven’s Chrome extension. The code could exfiltrate cookies and authenticated sessions for targeted websites. Cyberhaven’s final analysis linked the incident to a broader campaign involving more than 30 other potentially compromised extensions. The practical lesson is that approving an extension once does not remove the need to review its updates and access over time.

THE SURFACE NOBODY INVENTORIES

Extensions are privileged software inside every authenticated session, and AI made it worse.



You cannot allowlist what you have never inventoried. And these read the page before any of your controls get a vote.

Figure 7. The extension surface by the numbers: near universal install base, a fast-growing AI extension wave, and AI extensions far more likely to reach session data.

2.6 The Decryption Gap

Decryption is one of the biggest weak points in the current model. Browser-layer enforcement removes that dependency instead of having to work around it. TLS decryption is still the main way network tools inspect browser traffic content, and at full scope it provides meaningful coverage. “Full scope” being the key phrase there, which includes:

- A decryption tier sized for peak session volume
- Ongoing maintenance of certificate infrastructure across your SaaS estate
- Application exemptions, for when certificate-pinning breaks under decryption
- Performance-driven exemptions for high bandwidth categories like video and large file storage
- Cases where the protocol being used doesn’t fully support decryption

Once these realities enter the picture, coverage becomes partial, as does visibility into traffic that’s fallen outside the inspection scope.

HOW FULL SCOPE DECRYPTION ERODES

Each exemption shrinks the slice of traffic network inspection actually sees.

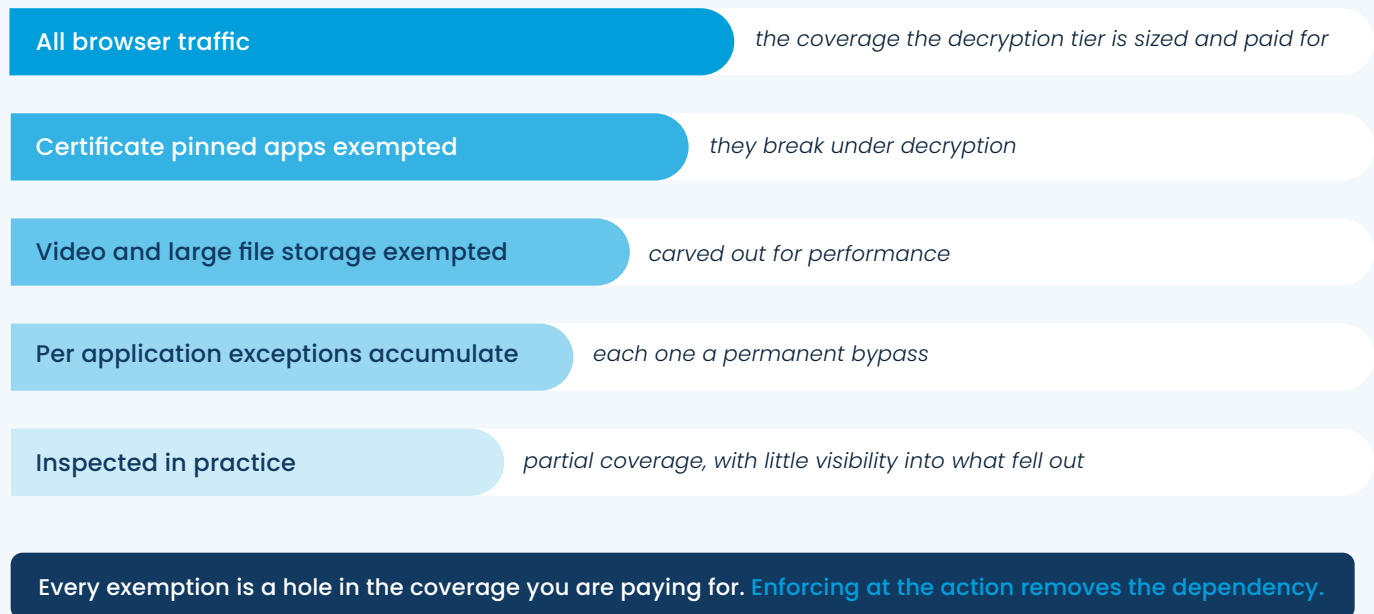


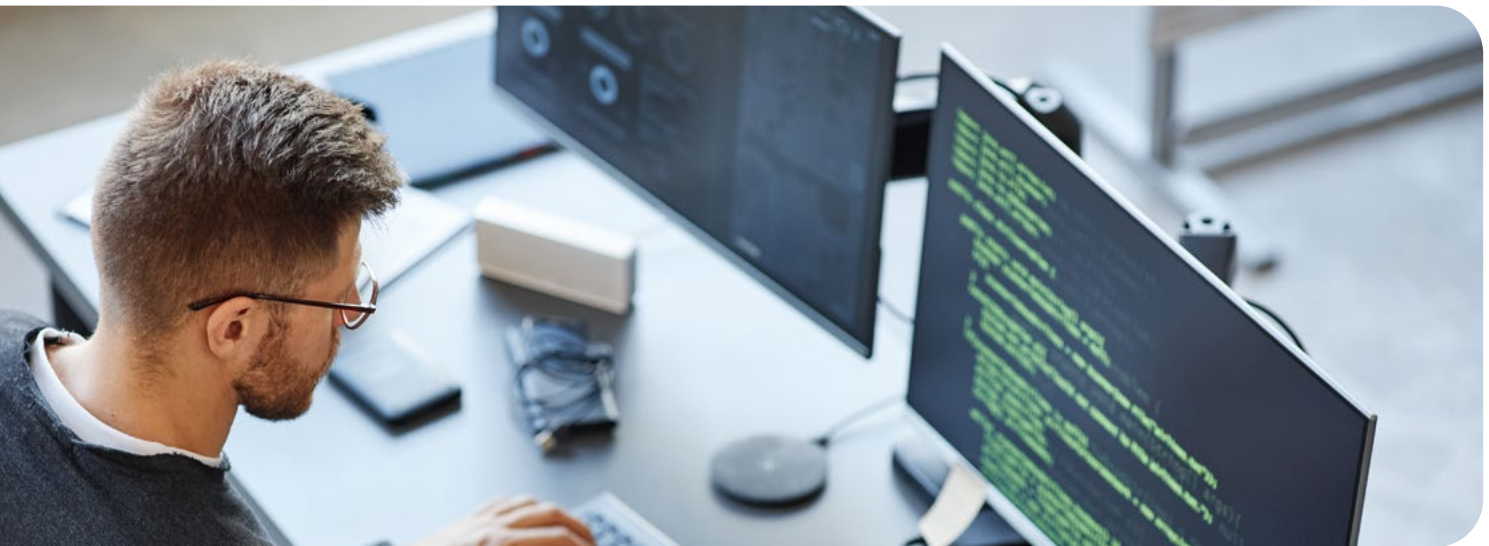
Figure 8. Coverage erosion in practice. Each exemption removes traffic from inspection scope. Bar widths are illustrative.

Browser layer enforcement doesn't require decryption. Instead, policy fires at the point of user action, inside the browser, before data leaves the application context. The user experience changes too. A network layer block introduces latency and usually breaks the session with a block page. A browser layer block prevents risky action inline, which means the paste doesn't complete, and the form doesn't submit. No redirect, no disruption. This dramatically improves the user experience: it's easy to communicate the block reason, and the user has the ability to ask for a one-time bypass.

Your CISO's main issue will be coverage integrity. If pinned applications, high bandwidth traffic, and exception-heavy workflows are routinely excluded from inspection, your organization won't know where and how browser-mediated data loss or threat delivery is controlled.

There's a budget argument to be made as well. Browser layer enforcement doesn't just improve coverage but can reduce or replace compensating tools for this gap. Decryption infrastructure, certificate management, pinning exceptions, and bandwidth bypasses can all carry costs and leave gaps.

For covered browser workflows, session-layer enforcement can reduce dependence on network decryption and its associated operational burden. That benefit does not necessarily eliminate the need for SSE coverage outside the browser.



3. What We Tested and What We Found

Section 2 was the diagnosis.

This next part is about what happened when we finally put the product to the test.

Every scenario below ties back to one of the six gaps we just walked through. We ran each one in a controlled lab against a live Prisma Browser deployment. We'll tell you exactly where the product holds up, where outcomes succeed based on what you already bring to the table, and what's explicitly out of scope.

Before we get into the details, **here's an overall view of each gap that we tested and how, along with a summary of our findings.**

GAP (FROM SECTION 2)	WHAT WE TESTED	WHAT WE FOUND
AI tools and data leaving (2.1, 2.3)	PII, and financials pasted, uploaded, and submitted to public AI tools, personal cloud, and webmail, with a sanctioned tool as the control.	The tested unencoded PII and financial data transfers were blocked across the evaluated channels and destinations. The sanctioned workflow remained usable. Base64-encoded test content was not blocked. Any compensating policy needs separate validation for the relevant workflow.
Agentic AI (2.2)	The product doesn't yet distinguish agentic vs. user actions. This is on the roadmap and wasn't tested. Adding a human-in-the-loop step is currently available for any workflow.	Human-in-the-loop approval works, but you can't differentiate agentic vs. user workflows yet. This is currently being developed and tested.
Investigating the block (2.3)	Whether a blocked event carries enough information for the SOC to work through it without a SIEM in place.	The browser's own record carries who, device, destination, pattern, and action.
Browser attack surface (2.4)	Malicious URLs, malicious downloads, and C2 callbacks.	Caught at delivery, before anything wrote to disk. EDR functions as a backstop.
Extension supply chain (2.5)	An off-allowlist install, a broadly permissioned allowed extension reading protected content, and a sanctioned extension as the control.	Off-allowlist installs blocked, with risk-scored policy behind it.

The architectural implications of browser-layer enforcement and decryption dependency are discussed in Sections 2.6 and 4 as this is not something that can effectively be evaluated and tested.

3.1 AI and Data Security Controls

This section covers the AI tool blind spot from Section 2.1, and the application-layer exfiltration issue we went over in Section 2.3.

While they look like separate problems on a risk register (and while in some ways, they are) they also fail at the exact same moment: when a user pastes, uploads, or submits data through the browser.

That's why we tested them together, instead of running the same scenario twice with a different destination. This is relevant for any CISO trying to better enable AI use, without losing control over enterprise data.

We did this by emulating behavior we see in every single environment. Sensitive data moves through the browser into tools the business uses, but security doesn't fully govern. Customer data and internal financials were submitted to public AI assistants and open-source endpoints, via chat, forms, and file uploads. The same data was shared to personal Google Drive, Dropbox, and external webmail. We also tested a sanctioned enterprise AI tool as our control, to make sure enforcement was policy-driven and not just blocking anything that looked like AI.

LAB FINDING: PII and financial data blocked inline at the point of action. Sanctioned business apps stayed usable. One encoding bypass found.

Result: Blocked inline at the point of action, across every channel and destination tested

- Confirmed that sensitive data was blocked from leaving to unsanctioned destinations. PII and financial data were blocked on copy-paste, file upload, and form submission to unsanctioned tools such as ChatGPT, Google Sheets, and Gmail.
- Data types we ran: SSN, home address, other PII, and credit card and bank account numbers.
- Sanctioned enterprise AI tool stayed usable. No false block on approved workflows.
- A pop-up blocked sensitive data from being pasted. File submissions failed with a pop-up notification.
- Control boundary: Base64-encoded file content bypassed DLP inspection. The encoded payload was not normalized before submission, so the engine couldn't classify it. The same evasion boundary surfaced in threat-prevention testing (Section 3.4).



You'll notice we're not including a block rate, which is on purpose. We could technically run 100 pieces of financial data through it, in order to report that it caught all 100. But that number doesn't actually tell you very much. And what matters isn't really an overall percentage, or if Palo Alto Networks' DLP can catch a stolen credit card number. The big takeaway is that the block fires inline, right at the paste or the upload, and that it held across every data type and destination we threw at it.

Prisma Browser includes more than 1,000 prebuilt data classifiers, so you spend less time creating classifiers from scratch and more time protecting important data. But getting value from DLP still requires a clear strategy and a point of view. Define which data needs protection, how it should be handled, and where enforcement should apply.

Because enforcement happens inline, you also have flexibility in how users experience it. This can be a surprisingly tricky element of the browser experience. A silent block might look like an error and result in unnecessary Help Desk tickets; meanwhile a generic "You can't do that!" message can create friction and hurt adoption. Prisma Browser lets you tailor enforcement to your workforce by providing context, educating users, and coaching them toward the right behavior.

We did find one limitation. Our team submitted Base64-encoded sensitive content without triggering the configured DLP block. This is a boundary of the evaluated configuration, and it needs to be included in the deployment's test plan. Blocking a destination or transfer action may provide a compensating control, but it can also restrict legitimate work. This DLP bypass is not specific to Prisma Browser, but requires a plan to mitigate, such as blocking non-standard encodings in your environment.

WHAT THE POLICY BUILD ACTUALLY INVOLVED

With Prisma Browser, you can block PII and financial data from being submitted to AI tools, cloud storage, and webmail, while still keeping those tools available to users.

Although Prisma Browser and Enterprise DLP are part of the same Palo Alto Networks platform, the browser can distinguish between cloud and local DLP profiles. Based on your requirements, you may need to rebuild a data profile itself so the browser can properly reference it. What should have been a ten minute session to operationalize a policy became a multi-hour project that required Palo Alto Networks' DLP expertise. The capability is real and the integration works. But in practice, it leaves a bit of simplicity to be desired for a practitioner. DLP rules can be complex in nature and require effort and time to design. Budget for the policy build to be real work, not a checkbox.



3.2 Agentic AI Defense

Security leaders have been asking about agentic AI, but product reality isn't as far along as market messaging.



Palo Alto Networks recently announced agentic AI defense capabilities at the 2026 RSA conference, including the ability to distinguish agent activity from human activity inside the browser. As of October 2026, Prisma Browser has delivered on its product roadmap and can distinguish between agentic and user activity inside the browser. However, since this is a recent feature, AHEAD was unable to validate it during our evaluation.

The distinction matters. If an agent moves data, submits a form, changes a record, or triggers a workflow, the SOC needs more information than the name of the authenticated user. This additional context can clarify whether a user or agent took the action, what policy evaluated the action, and whether that action should have required approval before it completed.

What AHEAD could validate was human-in-the-loop approval. Prisma Browser can pause any browser workflow and require explicit user approval before allowing it to continue. This can be useful control logic for sensitive destinations, risky actions, or workflows that need an additional checkpoint before data can move. However, it's important to be precise about what we tested. These approvals apply to browser activity based on policy conditions.

As it stands, we can validate that Prisma Browser has meaningful workflow interruption and approval capabilities. And its ability to now attribute agent-specific actions is an important feature that demonstrates the ability to securely enable agentic workflows in the browser.

What your CISO needs to understand is that the browser is ultimately still the right enforcement plane for tracking agentic activity as it moves through authenticated web sessions. The architecture all makes sense.



HUMAN-IN-THE-LOOP APPROVAL

We tested Prisma Browser's approval process, by attempting to move sensitive data from a sanctioned workflow into an unsanctioned destination. Our test used sample PII and financial data, submitted through file uploads, paste, and form entry. The policy condition was the action itself, not agent identity. Whenever we attempted to move sensitive data to a restricted destination, it triggered a user prompt: a justification requirement, permission request, or MFA challenge, depending on the workflow and how the policy was configured.

LAB FINDING: *Human-in-the-loop workflows work as expected*

Result: Approval Fired

- Approval prompt triggered correctly: 100% of applicable scenarios
- Workflow resumption after approval: PII and financial data were uploaded to a gated destination, in this case Google Drive.
- Workflow termination after denial: The upload was blocked, preventing data upload.
- Policy conditions that triggered approval: Uploading PII to GenAI tools and file-sharing applications required the user to submit a permission request.
- Admin alert generated: Yes

WHAT THIS CONTROL COVERS

Today, this layer lets you interrupt browser workflows, and require human approval before certain actions continue. This is critical when sensitive data is moving to a risky destination.

Prisma Browser can now also distinguish between agentic and user activity inside the browser, though as this is a recent feature, AHEAD could not validate during our evaluation.



3.3 Forensics and the Limits of the Control

Section 3.1 answered whether the block fires. This section answers what happens after it does. Can the SOC investigate it properly? Does the event carry enough context? And where does the browser control stop?

We're still looking at application-layer exfiltration from Section 2.3, but our focus shifts from enforcement to evidence and control boundaries.



We conducted our own adversarial testing of the data loss prevention and the in-browser controls ourselves. We found them genuinely robust.

Grant Sewell, Chief Security Officer, AHEAD

CAN YOUR SOC INVESTIGATE THE BLOCK?

Any block your SOC can't properly investigate isn't a real control. For the blocked events from Section 3.1, we checked whether Prisma Browser captured the details that matter:

- User Identity
- Device Information
- URL
- Destination App
- Detected Classifiers
- Matched Data Profiles
- Action Taken

We didn't have a SIEM connected for this round, so downstream correlation was out of scope. What we evaluated instead was whether the browser's event record gives the SOC enough information to work through the event in the first place.

LAB FINDING: *The necessary fields exist, along with native integrations to 3rd party*

Result: Successful

- Fields present in the browser's own event record per blocked action: User Identity, Device Information, URL, Destination App, Detected Classifiers, Matched Data Profiles
- Print to PDF of restricted content: Successfully blocked, with the ability to allow after a user prompt.
- Native export to SIEM available? Yes, Strata Logging Service can send logs from Prisma Browser events to a SIEM

3.4 Threat Prevention

Section 3.3 examined data leaving through the browser. Here, we tested whether browser controls could stop incoming threats at navigation and download. Endpoint security remains essential, including its ability to prevent execution and investigate suspicious activity across the device.

MALICIOUS URL AND FILE DOWNLOAD PREVENTION

We hit a set of known malicious URLs from active threat feeds, including credential phishing pages, and malware hosting sites. We also tried to download malicious tools from the browser, along with hosting a malicious file on a web server to test identification.

LAB FINDING: *Malicious URLs and downloads blocked at the point of delivery*

Result: Blocked at delivery, before disk write

- All malicious URLs and downloads were stopped, including live malicious downloads from self-hosted web servers
- Credential phishing pages were blocked in real time
- The one technique that got through was Base64-encoding the file before download. The encoded content was not inspected, and the file reached disk. This is the same evasion boundary we saw in DLP testing (Section 3.1)

COMMAND AND CONTROL (C2) CALLBACK DETECTION

We tested whether the browser blocks navigation to known command-and-control infrastructure from inside an authenticated session. We validated this against sample destinations categorized as C2, using feeds and tooling already on hand. For this round, we didn't generate live C2 traffic, simulate active beaconing, or exercise DNS- or framework-based C2 channels.

LAB FINDING: *Known C2 destinations blocked, advanced beaconing was not tested*

Result: Known C2 endpoints blocked

- We confirmed that navigation to known C2 destinations was blocked, and testing was deliberately limited in scope. The finding confirmed similar protections to those of an SSE/SASE platform.

3.5 Extension Governance

As mentioned, extensions are an under-addressed attack surface. They sit inside authenticated sessions with permissions to read webpage content, grab the clipboard, and act on SaaS as the user. Shadow AI has made this worse. Users who can't access or struggle with sanctioned AI tools will install extensions that replicate the tool's abilities. Functionally, it's the same thing as putting an unreviewed AI agent inside every authenticated session. Verizon's 2026 DBIR puts a finer point on it: more than 15 percent of users have unauthorized AI browser extensions installed, with these same extensions holding onto context from every single page they see, including internal sites.

That's a data exfiltration path running completely independently of your DLP, with the extension reading every page before your controls can intervene.

We tested three things here. First, whether the browser blocks an extension that isn't on the allowlist from being installed. Second, and even more interesting, we wanted to see whether an installed extension could successfully ask for broad permissions and read content on pages that DLP is protecting. It's easy to check if an installed extension is a security blind spot. Just install a legitimate extension from your allowlist, and agree when it makes the typical request to "read and change all your data on all websites." Then simply check whether it can see sensitive page content the browser is otherwise guarding. Last but not least, we used an enterprise-sanctioned extension as a sanity check to make sure normal tooling wasn't getting caught in the net.

LAB FINDING: Extension governance held across all three tests

Result: Off-allowlist install blocked; approved extension blocked from reading protected content

- Could a broadly permissioned extension read DLP-protected page content: Blocked.
- With Restrict Extension Host Permissions enabled, an approved extension holding broad read-and-change host permissions could not read the protected content on the page, including SSNs and financial data.
- Sanctioned extension worked normally, no false interference: Yes



Be precise about the read path, especially because it's the piece most people skip. By default, an extension with broad host permissions can read the content of a page. Your paste and upload controls can govern what the user does with the data, but not what an extension reading the same content on the same page can do. Restrict Extension Host Permissions is the best way to close that path, and worked cleanly in our testing, with approved extensions still able to run normally. The only catch is that it's a setting, not your default. It ships, and then you scope it to the domains that matter. Turn it on for sensitive applications, and the extension read path will be covered. Leave it off, though, and even approved extensions can be used to exploit the controls you took so much time and care to build.



It's worth noting that the policy model isn't limited to static allow and deny lists. Every extension in the directory carries a risk score built from its permissions, publisher reputation, and review credibility. Policy keys off that score directly, which means you don't have to personally evaluate every extension request in the queue. Instead, you can set up a risk threshold and let the scoring do the triage. Same caveat as DLP, though. Risk scoring reduces the problem, but the control is only as good as the allowlist behind it.



3.6 End User Adoption and Onboarding

Many enterprise browser deployments fail at the adoption stage. Limited use means limited control, and that only gets even worse as you start carving out exceptions. We pushed hard on this part of the evaluation because it determines whether everything we've discussed up to this point actually works in the real world.

CHROMIUM FOUNDATION: ZERO LEARNING CURVE

Anyone who's ever shipped a browser change or any high-impact user change to a workforce knows exactly where these projects die: in week two of the pilot. That's about when the help desk queue fills up, or an executive asks for that first exception and sets a precedent. This all comes down to one decision: asking users to learn a different tool, or in this case, a browser.

Prisma Browser is built on Chromium, the same engine under Chrome. Shortcuts, dev tools, tab behavior, and even the way extensions load are all identical. During our tests, we saw zero friction that came from a lack of user familiarity. This removes a significant barrier in user adoption before deployment even begins.

CUSTOMER DEPLOYMENT EXPERIENCE: SRE

User resistance is a fair concern. It deserves a pilot with real workflows before it becomes a reason to rule out browser security. The questions are practical: can people keep working, and can IT support the deployment without adding more complexity?

We put those questions to Luís Gaspar, Director of Technology Services at Madeira's regional education department, SRE. His organization reports deploying Prisma Browser protection to 20,000 users in one day.

"Prisma Browser can be deployed incredibly quickly," Gaspar said.

The preparation matters. His team ran a small pilot, used its existing device management platform, and synchronized accounts, licenses, and devices. Deployment then ran automatically in the background. In our interview, he described that process as straightforward for his team and transparent to users. That is a deployment approach worth testing with the management tools you already have.

The day-to-day results were just as important. Their previous remote access solution had generated recurring connectivity and synchronization issues. After the move, Gaspar described a sharp reduction in support requests, with most remaining issues handled within his team.

"We have cut the time spent managing SASE security by 60% using Prisma Browser."

That operational improvement came with control over the content users could access, including away from the organization's network. Administrators could adjust policies and individual profiles remotely. SRE also reports roughly 50,000 incidents blocked daily, predominantly phishing attempts.

For a CISO, this is a reason to test the adoption concern directly. Put the browser into real workflows and measure the support burden alongside security enforcement. Ask users where they hit friction. Track the exceptions. Those results should drive the rollout decision. Assuming people will reject the change leaves a useful security control unevaluated.

BOOKMARK, PASSWORD, AND PROFILE MIGRATION

Separately, in AHEAD's lab, we ran a full migration from an active Chrome profile which included bookmarks across multiple folders, saved passwords, cookies, history, open tab groups, and a set of installed extensions.

LAB FINDING: *Migrating from Chrome to Prisma Browser was nearly seamless*

Result: Clean Migration

- Total migration time from first launch to fully operational state: 2 minutes or less
- Bookmarks migrated successfully: Yes; folder structure preserved: Yes
- Saved passwords migrated: Yes, saved passwords within Chrome were imported into the built-in Prisma Browser Enterprise Password Manager
- Cookies and active sessions preserved: Cookies can be imported, but active sessions are not preserved, meaning a user must reload/reauthenticate their sessions.
- Extensions migrated or prompted for install: Migrated, but they prompt for install/permission access

Browser-session enforcement can reduce decryption related friction for covered workflows, but the migration still has to be engineered. Internal applications need testing, and the new policy model will require a shift in team mindset from traditional network rules. The browser needs to be implemented like infrastructure, rather than deployed like another endpoint tool. The first launch wizard walks the user through import and automatically detects which browsers are installed locally. For a Chrome standardized environment, migration is a single screen. For broad rollouts, you should preconfigure the wizard via Strata Cloud Manager. Silent deployment via GPO, MDM, or SCCM is fully supported if you don't want any user interaction at install.

4. Can the Browser Replace Your SSE Stack?

Every CISO will want to know the answer once they understand where enforcement has moved. It's the one question in this paper we can't answer from the lab, so we'll answer it from the market: **Enterprise browsers don't replace SSE platforms. Instead, pairing the enterprise browser as part of the SSE platform delivers the most complete approach.** Since we can't test this, let's look at the vendors for validation.



Palo Alto Networks led the charge by introducing Prisma Browser as part of the Prisma SASE platform back in the middle of 2024. Zscaler announced its enterprise browser in April 2026 inside its existing Zero Trust Exchange platform. Both arrived at the browser from the network. Island took the opposite approach, launching the Island Enterprise

Platform in March 2026 to extend its browser into desktop applications and the network layer. Its desktop client is designed to bring thick client and native applications into view, including those the browser wasn't positioned to see. As of early 2026, Prisma Browser has also introduced data protection for desktop clients.

That convergence says something about where the browser now sits. No one asks whether EDR replaces the firewall, or if email security replaces identity. Each control operates at a different layer, and together they form a foundational part of the stack. The browser is in the same place now. What stays outside the browser sorts into two categories.

First, applications that run as their own process. A thick client that launches outside the browser and connects directly to its back end is invisible to the browser session, even if the vendor also ships a web front end. Reaching that traffic requires a local client to redirect it for inspection.

Second, sessions the user never starts. Server to server traffic, API calls between services, IoT and OT traffic, local agentic traffic, and inbound administrative access can all run with no browser involved at all. Your agentic roadmap will run straight into this one.

None of this reduces browser coverage, but it does set boundaries. SSE was built for a real problem. Applications and protocols outside a browser's reach still need inspection. That's why Prisma Browser ships inside a larger SSE platform, and why the rest of the market is converging on a similar architecture. Desktop application traffic and browser session traffic need to meet the same inspection standard, but most organizations are still running them as two disconnected enforcement models.

5. Conclusion

Six pressure points drove this evaluation, with all six converging in the same place. The browser session layer is where work happens, where data moves, and where agents act. And it's the one layer your existing security stack wasn't built to govern. That's not a gap you fix with another agent or proxy. The browser is an essential control plane that your architecture is missing. AI agents have fundamentally shifted the security landscape while providing significant potential for user productivity. The browser is uniquely positioned to safely enable AI adoption across the enterprise. Now couple the browser with an SSE platform, and you can deliver a holistic security and control solution across the endpoint, browser surface and the network.

The SOC stack won't change. Rather, the browser enables a layer of visibility it never had before: what happens inside the authenticated session at the moment of action. That telemetry can be exported to the SIEM alongside the rest of your security data, so teams can incorporate it into existing workflows without standing up a separate investigation model. The browser complements EDR. It does not replace it. Run the two together, and you get the most complete picture of endpoint and browser activity available, instead of two tools fighting over the same data.

Identity matters here too. Because enforcement is tied to identity, policy follows the user into whatever SaaS or AI tool they open next. A group change or session revocation in your identity provider can track them across the browser as well, instead of getting stuck at the network edge.



WHY A DEDICATED ENTERPRISE BROWSER, NOT A HARDENED CONSUMER ONE

Managing Chrome or Edge does not automatically close the browser security gaps in your environment. You can control extensions, keep browsers updated, and restrict access to websites without having content-aware controls over what employees paste or upload. Chrome Enterprise Premium and Edge with Purview add those protections, but coverage depends on what you deploy and where it works. Microsoft's browser-DLP protection for unsanctioned AI apps, for example, requires Intune-managed Windows devices. That particular capability does not cover someone using a Mac laptop or personal laptop outside device management. The question is whether your deployed controls protect the work people actually do, on the devices they actually use. We favor Prisma Browser where bringing these controls into one browser policy framework can simplify security operations. It can inspect AI prompts, keep protected downloads restricted to the browser, and require additional authentication before a file transfer. Its Account Protection capabilities also extend controls to applications outside corporate single sign-on. Those are practical options for letting people work while limiting what they can do with sensitive data. For teams maintaining separate browser and data-protection controls, this approach can reduce the effort of keeping policies aligned. Prisma Browser reduces operational complexity by providing all of these capabilities in one solution, one management plane.

We set out to test whether Prisma Browser could close the gaps we identified, and it delivered on every outcome that mattered to us:

- DLP fired at the point of user action
- AI tool governance became enforceable within the session
- Extension control improved
- Threat prevention moved closer to the point of delivery
- Browser-layer enforcement showed a credible path away from decryption-heavy coverage models

Agentic protections are still maturing, and some capabilities still live only on the roadmap. It's worth reiterating that enforcement is only as good as the policy behind it. But all the same, our conclusion holds. Prisma Browser is the right control at the right layer for the gaps we set out to evaluate.

Policy quality still drives enforcement quality. If data classifications aren't defined, start there. Our evaluation proves the technology can do the work. The proof is in the metrics, with AI tool usage shifting from "we don't know" to a real-time answer. DLP coverage extends to the places where data moves. A successful rollout should reach the full target population rather than stall at partial coverage.

For CISOs whose strategic priorities include safe AI enablement, application layer data protection, and reducing the operational complexity of their enforcement infrastructure, the browser is the primary ungoverned surface where all these priorities converge. It's also the control plane that enables the rest of your architecture to finally do what you bought it to do.

5.1 Deployment Considerations

WORK TO DO BEFORE DEPLOYMENT

- Audit installed extensions across the target population. Build the allowlist from that audit, not from a pre-existing template.
- Run a compatibility check on internal web apps during the pilot. Flag exceptions before broad rollout.
- Confirm SOC capacity for the new telemetry. Define the alert triage workflow during the pilot stage.

Our recommended deployment sequence:

- 1 Define policy
- 2 Build an extension allowlist
- 3 Run a pilot with 25 to 50 users from a population with high DLP activity
- 4 Validate compatibility
- 5 Lock in telemetry and SOC workflows
- 6 Broad rollout with change management communications



FOUR NUMBERS FOR THE BOARD, ONE PER OUTCOME

Reported after rollout. Each aggregates the operational work underneath.

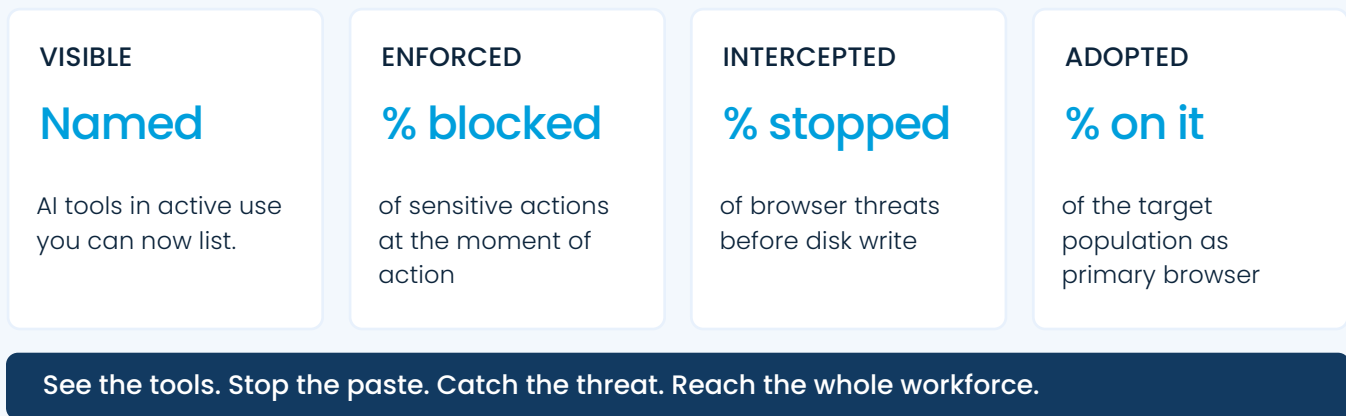


Figure 9. Four numbers to report to the board after rollout, one per outcome.

WHAT YOU MEASURE 30 AND 90 DAYS AFTER ROLLOUT

- Can you now answer which AI tools are in active use across the workforce?
- Are production enforcement rates consistent with your pilot findings?
- What percentage of tested browser threats is stopped at navigation or download, before disk write?
- What percentage of the target population is using Prisma Browser as their primary browser?

Those four numbers will tell you if the deployment is working as intended.

If the gaps in this paper sound like your own environment, set up a working session with our experts to determine your best next steps.

About the Authors

Matt Elliott | Specialist Solutions Engineer, Security

Matt Elliott is a Specialist Solutions Engineer in AHEAD's security practice, helping enterprises design, implement, and continuously improve resilient network security architectures. He specializes in NGFWs and SASE solutions, drawing on an early foundation in IT that informs how he aligns security architecture with the systems and data it's built to protect. With over 11 years in cybersecurity, Matt has guided organizations through every stage of their security journey, from initial design to long-term optimization.

Grant Sewell | Chief Security Officer, AHEAD

Grant Sewell is Chief Security Officer at AHEAD, where he has built a modern security program grounded in the same principles AHEAD brings to its clients. He oversees AHEAD's internal security and technology operations while partnering with clients to navigate their own security transformations. Drawing on more than 20 years of experience in cybersecurity strategy, risk, and privacy, he is a frequent keynote speaker and recognized contributor to the security community.

Appendix

Test Environment

COMPONENT	VERSION
Prisma Browser	149.18.4.115
Strata Cloud Manager	2026.r2.2
Test OS	macOS 26.5.1
Source browser for migration test	Chrome
Testing period	June 10–30, 2026
Testers	Doug Lorenzen, Chris Linke, Joachim Johannesen

Sources

Gartner. Innovation Insight: Secure Enterprise Browsers. Evgeny Mirolyubov, Max Taggett, and Dave Messett. April 1, 2025.

Gartner. Cybersecurity Must Block AI Browsers for Now. Dennis Xu, Evgeny Mirolyubov, and Dave Messett. December 1, 2025. Gartner document ID 7211030.

Verizon. 2026 Data Breach Investigations Report. Verizon Business, May 2026.

Verizon. 2025 Data Breach Investigations Report. Verizon Business, April 2025.

Palo Alto Networks Unit 42. 2026 Global Incident Response Report. February 2026.

IBM Security. Cost of a Data Breach Report 2025. Research conducted by Ponemon Institute. July 2025.

LayerX. Enterprise Browser Extension Security Report 2026. April 2026.

Island. New Island Enterprise Platform Secures and Enables Work Across Enterprise and Consumer Browsers, Desktop Apps, and Networks. March 17, 2026.

Zscaler. Introducing the Next Phase of the Zero Trust Browser. Vishal Gupta. April 29, 2026.

Cyberhaven. Final analysis: Chrome extension security incident. January 14, 2025.

Cyberhaven. Cyberhaven’s Chrome extension security incident and what we’re doing about it. December 27, 2024; updated January 2, 2025.

Palo Alto Networks. Prisma Browser product documentation and User Guide. docs.paloaltonetworks.com/prisma-access-browser.

Gaspar, Luís. Interview conducted by Matt Elliott, AHEAD. September 15, 2026.



Combining cloud-native capabilities in software and data engineering with an unparalleled track record of modernizing infrastructure, we're uniquely positioned to help accelerate the promise of digital transformation.

Visit us at ahead.com.

Global Hub Offices

CHICAGO

444 W. Lake Street
Suite 3000
Chicago, IL 60606
United States

NEW YORK

500 5th Avenue
17th Floor
New York, NY 10010
United States

UNITED KINGDOM

310 Wharfedale Rd
Winnersh, Wokingham
RG41 5TP
United Kingdom

NETHERLANDS

Kerkenbos 10-123
6546 BJ Nijmegen
The Netherlands

INDIA

L-21, One Horizon Center
Golf Course Road, DLF Phase V
Sector 43, Gurugram 122002
India