MENLO SECURITY

2026

# State of Browser Security Threat Report

## Evasive Threats, Zero-Day Lures, and the New Browser-First Kill Chain


REPORT

Table of **Contents**

# 2026 State of Browser Security Threat Report

## Evasive Threats, Zero-Day Lures, and the New Browser-First Kill Chain

PAGE

# The Bottom Line

## What This Report Establishes:

The browser is where work happens. It's also where attacks start — and where your security stack has its biggest blind spot.

In 2026, your endpoint and network tools are failing to keep up. Nation-state actors, ransomware groups, and for-profit criminal enterprises have shifted their primary strategy to browser-first initial access — and the tools built for network-layer attacks weren't designed to see it. Attacks arrive as a fake CAPTCHA, a routine verification screen, a PDF that isn't a PDF. They bypass MFA, harvest session tokens, and sell access to ransomware groups before a single alert fires.

The kill chain has compressed from days to minutes. And the same techniques that fool your users work just as effectively against the AI agents already operating in your environment — at machine speed, and without effective governance.

## Three Numbers That Define the Business Risk

| 1 in 3 | 1 in 5 | 90 |
|---|---|---|
| Highly Evasive Threats are launched from sites already categorized as 'safe' | Phishing links actively clicked by users go completely undetected by legacy URL filtering | Zero-day vulnerabilities actively exploited in 2025 —none caught before first victim |

*(Sources: Menlo Threat Research; Google/BleepingComputer 2025)*

# The Strategic Mandate

Your existing security investments — SASE, SSE, EDR, DLP — are performing exactly as designed. The architectural limitation is that none of them were designed to operate at the browser session layer. That's not a criticism of those investments. It's a map of where your blind spot is.

The mandate is not to replace what's working. It's to close the layer that was never covered: the session layer, where encrypted traffic executes, credentials are entered, sensitive data moves between your people and the services they use every day — and where every major attack vector in this report originates.

## Three Things You Should Do in the Next 90 Days:

*(The attack techniques, threat groups, and real-world cases in this report point to the same three gaps in enterprise security postures. The rest of this report gives you the evidence. Here's where to start acting on it.)*

**1.** Most organizations find they have zero visibility into browser session activity. Run the five self-assessment questions in Chapter 2 against your current security stack. If you can't answer yes to all five, you have a potential browser-security gap. Map which questions you answered 'no' to against the six attack techniques in Chapter 3. That's your exposure list.

**2.** Pull your proxy and endpoint logs from the last 90 days and look for something you probably won't find: browser session activity. Search for executables delivered through the browser, newly registered domains your users visited, and any download that didn't match the file type the page claimed to serve. Don't assume absence of alerts means absence of activity. In the healthcare case study documented in this report, zero tools fired — and the attack was real. Absence of evidence is not evidence of absence at the browser layer.

**3.** Map which of your security controls would fire if an attack was hosted on Google Drive, Dropbox, or SharePoint. 35% of the evasive threats Menlo blocked this quarter originated from sites your reputation filters had already approved as safe. If your detection depends on a domain being flagged as malicious, it will miss anything hosted on infrastructure you've already decided to trust — which is exactly where attackers are operating.

MENLO BY THE NUMBERS · 90-DAY IMPACT: JAN–MAR 2026

# Threats Neutralized by Menlo in the Last 90 Days

Your existing security stack has a layer it wasn't built to see. Network tools inspect the connection. Endpoint tools watch the device. Neither one governs what happens inside the browser session — where credentials are entered, files are downloaded, and attackers have learned to operate invisibly.

The figures below represent what was waiting in that layer — Menlo platform telemetry across enterprise customer environments from January 1 through March 31, 2026, drawn from active deployments, not simulated environments, and reflecting threats that bypassed customers' existing security stacks before reaching the Menlo Cloud.

| | |
|---|---|
| **4,937** | **Zero-day Attacks Blocked**<br>Stopped up to 6 days before reputation filters knew they existed. |
| **52,185** | **Threats Originating from Sites Classified as 'Safe'**<br>Hosted on domains your security stack was configured to trust — Google Drive, Dropbox, SharePoint, and others like them. |
| **115,842** | **Evasive Phishing Attacks Identified Across Active Campaigns**<br>Each purpose-built to bypass detection, using techniques like CAPTCHA abuse, TDS redirection, HTML smuggling, and brand impersonation. |
| **340,871** | **Exploitable Files Disarmed**<br>Of which 110,357 were identified from password protected files |

*(Source: Menlo Threat Research)*

## Top Threat Events: Q1 2026

### January 2026 — Sneaky 2FA (Session Hijacking)

**Target:** Large California Resort & Casino

**Attack type:** Phishing-as-a-Service (PhaaS) credential theft with 2FA bypass

**Technique:** Sophisticated PhaaS operation utilizing brand impersonation and session cookie harvesting to bypass multi-factor authentication. The attacker intercepts the authenticated session token — not just the credential — making MFA a non-factor in access control.

**Why traditional tools missed it:** The phishing page rendered correctly over HTTPS from a newly registered domain. No static payload. No known-malicious URL. Session token harvesting occurred inside the browser session, below the visibility of network inspection.

### February 2026 — RMM Adobe Lure (Malware Delivery)

**Target:** One of the Nation's large healthcare organizations

**Attack type:** Social engineering with trojanized RMM tool delivery

**Technique:** A user clicked an email-delivered link to a fake Adobe secure document portal hosted on a compromised WordPress site. The 'PDF download' was a trojanized Remote Management executable (Premier_Electronic_Solutions.exe).

Blocked by: Menlo HEAT Shield AI detected that the page was delivering an executable disguised as a PDF, identifying the impersonation of the Adobe portal before download executed.

**VirusTotal at time of click:** Zero vendors flagged the domain as malicious.

### March/April 2026 — ClickFix Server-Side Polymorphism (Malware Delivery)

**Target:** Enterprises across financial services, healthcare, and government

**Attack type:** Social engineering with clipboard-injected payload delivery

**Technique:** Users landed on pages mimicking legitimate browser verification flows — fake CAPTCHAs, Cloudflare human verification screens, browser update prompts. Malicious JavaScript silently injected a PowerShell command or script into the user's clipboard. The page then instructed the user to paste and execute the content directly on their device. The payload was polymorphic server-side — meaning each delivery instance was uniquely generated to evade signature-based detection.

**Blocked by:** HEAT Shield AI identified the malicious intent of the verification page in real time, recognizing the clipboard injection and user-executed payload at time of click.

CHAPTER 1

# The New Battlefield

## Browser-First Initial Access

### The Browser Has Become the Enterprise Operating System

Email, SaaS applications, collaboration tools, AI assistants, financial systems, credential management — the majority of productive enterprise work now happens inside a browser session. That concentration of activity made the browser the most consequential attack surface in the enterprise. Attackers followed the work.

The result is a fundamental shift in initial access strategy. Nation-state actors, organized ransomware groups, and for-profit criminal enterprises have largely abandoned brute-force network breaches in favor of browser-first compromise. The browser is where credentials live, where sessions exist, where data moves, and where the gap in your security stack is widest.

- **Over the last 90 days**, Menlo blocked an average of 116 zero-day threats per customer tenant, with over 35% of these originating from sites classified as 'safe'.
- **1 out of 5 phishing links** actively engaged by users went completely undetected by legacy URL filtering. The attack is happening; the tool doesn't know.
  *(Source: Menlo Threat Research)*

### The Psychology Behind Browser-First Attacks

Modern browser-based attacks don't just exploit technical vulnerabilities. They exploit cognitive ones. Techniques like ClickFix weaponize familiar browser elements — CAPTCHAs, error messages, Cloudflare verification screens — to bypass user vigilance by making the first interaction feel routine. That small, familiar action primes the user for the next instruction, which is the actual payload.

The sequence: a user encounters what appears to be a routine verification step (a CAPTCHA, a browser error, a document loading prompt). The minor interaction (like clicking a checkbox or pressing a key combination) feels low-stakes and familiar. That action primes the user psychologically for the next instruction, which is the actual payload delivery: pasting a PowerShell command, clicking a download, or entering credentials on a spoofed page.

The user becomes the attack's execution mechanism. And that's the point: when the victim runs the command themselves, every technical control that monitors for 'malicious behavior' sees a legitimate user performing a legitimate action.

## Case Study in Focus — Why Patching Isn't Enough

In March 2025, Kaspersky disclosed active exploitation of CVE-2025-2783, a high-severity sandbox escape in Chrome's Mojo IPC framework (CVSS: 8.3). The campaign — dubbed Operation ForumTroll — deployed the Trinper backdoor via a one-click phishing email. Clicking the link triggered immediate infection — no secondary download, no additional user action required. Targets included universities, financial institutions, and government agencies across multiple countries.

### The Patch Gap — Why CVE-2025-2783 Mattered Beyond Its Technical Severity

**Day 0:** Exploit active in the wild. Zero vendors aware.

**Day 6:** Google releases Chrome patch for Chromium engine.

**Days 6+:** Enterprise browsers (including Replacement Browsers like Island) required additional time to ingest the Chromium patch, test compatibility, and push the update to endpoints.

Total enterprise exposure window: 6 days minimum, potentially weeks depending on patch deployment velocity.

This is the structural problem with any local-browser security model: the patch gap is an architectural feature, not a process failure. Organizations running Menlo render this exposure window irrelevant — the exploit detonates in the cloud, not on the endpoint.

## The 2025 Chrome Zero-Day Landscape: Not an Anomaly

**CVE-2025-2783** was one of eight actively exploited Chrome zero-days patched in 2025. Beyond the zero-days: over 80 high/critical vulnerabilities were confirmed in Chromium-based browsers in 2025 alone. The 'patch and pray' model is structurally broken — not because teams are slow, but because the exposure window between discovery and enterprise deployment is an architectural inevitability for any browser that executes code locally.

The patch gap is one structural problem with any browser that executes code locally. The rise of AI agents inside those same browsers is another — and unlike the patch gap, it doesn't wait for a vulnerability to be discovered.

# Your Browsers Already Have AI Agents. Your Security Stack Doesn't Know.

Your employees are already using AI agents every day, inside the browsers they already have. Chrome ships with Gemini built in. Edge ships with Copilot. Every major browser now has some form of agentic capability turned on by default — summarizing pages, executing tasks, interacting with web content on behalf of the user. And beyond the browser itself, tools like Claude and Gemini have become part of how people actually work: researching, drafting, analyzing, and acting across web applications throughout the day.

The security question isn't whether your workforce is using these tools. They are. The question is whether they're doing it inside a security architecture that can see what's happening.

> Recent research — including Anthropic's Claude 'computer use' demonstration — has confirmed that fully autonomous attack flows are now possible with minimal human intervention. An AI agent can be directed to navigate web applications, extract data, authenticate to services, and execute transactions at machine speed. The same capability that makes these tools productive makes them dangerous.

## Why AI Agents Are Uniquely Vulnerable

- **Prompt injection:** Malicious instructions embedded in web page content, documents, or API responses can redirect an AI agent's behavior without the user's knowledge. An attacker controls what the agent does by controlling what the page says.
- **Lack human skepticism:** When an AI agent navigates to a weaponized page, there is no user to notice something looks wrong. The agent proceeds.
- **Machine-speed exfiltration:** An agent that can be manipulated can move data volumes that would be operationally impossible for a human attacker — in minutes, not hours.
- **Default-on exposure:** These capabilities aren't something employees have to opt into. They're already present in Chrome, Edge, and the AI tools people use every day.

### Menlo's Position on Agentic AI in the Browser:

Your employees should be able to use Gemini, Claude, Copilot, and the AI tools that make them more productive. Blocking those tools isn't a sustainable security posture — it's a friction point that drives shadow usage and erodes trust between security teams and the workforce.

Instead: Focus on securing AI tool usage at the session layer — governing what data flows into those tools, what actions agents can take, and what content reaches the user — without getting in the way of the work. As Chrome and Edge continue to expand their native agentic capabilities, your security posture must be at the browser session layer, not the network layer. Menlo Agent Runtime Security (MARS) governs both human and agentic browser sessions through a single control plane, giving employees the tools they want and security teams the visibility they need.

The question for every security team finishing this chapter isn't whether these attacks are happening. It's whether your current stack would have caught any of them.

CHAPTER 2

# Why Your Security Stack Falls Short

## The Layer It Was Never Built to Protect

## The Architectural Flaw: Inspecting the Wrong Layer

The security tools you've invested in — SASE, SSE, SWG, EDR, CASB — are good at what they do. The problem isn't their effectiveness at the layers they protect. The problem is that none of them protect the browser session layer, and that's exactly where modern attacks now live. This "Inspecting the Wrong Layer" flaw exposes the enterprise to three primary risks:

- **Network Blindness:** Legacy SWGs rely on reputation and signatures, yet 75% of phishing links are now hosted on "trusted" domains (like Google, Microsoft, etc) that cannot be blocked. These tools scan packets, not the dynamic, rendered interactions where highly evasive threats and AI-driven social engineering now live.
  *(Source: Menlo Threat Research)*
- **Endpoint Latency:** EDR is a reactive, post-execution technology. In modern browser-based attacks—including zero-day sandbox escapes like CVE-2025-2783—the credentials or session tokens are exfiltrated before a "bad" file ever hits the disk for the EDR to analyze.
- **The Ungoverned Perimeter:** Attackers are now bypassing the network perimeter entirely by delivering threats via collaboration tools, shared documents, and AI tools that sit completely outside the visibility of your email gateway and SWG. The 2025 Verizon DBIR found that 15% of employees access generative AI platforms from corporate endpoints at least twice a week, with 72% signing in using personal email addresses — creating data flows that no enterprise DLP tool can see. Without session-layer controls, you are relying on human vigilance to stop attacks arriving through channels your security stack was never built to inspect.

### Self-Assessment: Run This Test Against Your Own Stack

**Answer These Five Questions Honestly:**

1. Can your current tools block a zero-day exploit before the patch is deployed?
2. Can your current tools detect a fraudulent credential form on an unknown domain?
3. Can your current tools inspect content assembled inside the browser's JavaScript engine rather than delivered as a file?
4. Can your current tools identify a web page impersonating a legitimate brand or service — based on its behavior, not its domain reputation?
5. Can your current tools govern the behavior of an AI agent browsing on behalf of a user?

If the answer to any of these is no — or 'I don't know' — that's the gap. The six attack techniques in Chapter 3 each map directly to one or more of these questions. Use them as your evaluation test cases.

# Reputation-Based Detection Will Always Fall Short

The dominant detection model for browser threats — URL categorization, domain reputation, block/allow lists — is structurally broken against modern attack infrastructure. The reason is simple: attackers have moved their operations to infrastructure that cannot be blocked.

35% of HEAT attacks this quarter originated from websites already categorized as safe — domains your reputation filters had already approved. And beyond that, campaigns are increasingly hosted on platforms like Google Drive, Dropbox, BitBucket, Azure Blob Storage, Discord, and SharePoint. You cannot block google.com or microsoft.com. Attackers know this and host their payloads there deliberately — a technique known as 'Living off Trusted Sites.

*(Source: Menlo Threat Research)*

## Why Reputation Lists Can't Keep Up: Three Structural Problems

### 1. The Speed Problem

Reputation lists are built on historical data. Modern threat infrastructure is designed to outpace that data. PhaaS (Phishing-as-a-Service) operations — organized criminal enterprises with dedicated infrastructure teams — rotate attack pages across new domains and IP addresses faster than automated crawlers can classify them. By the time a domain appears on a threat feed, the campaign has already moved.

Menlo Threat Research internal analysis and independent research confirms that reputation providers typically take 6 or more days to classify a new malicious domain — the same window documented in the CVE-2025-2783 patch gap. Attackers operate in that window deliberately.

*(Sources: SecureList by Kaspersky; Menlo Threat Research)*

### 2. The Anti-Analysis Problem

Sophisticated attack infrastructure actively defeats automated site analysis. Fake CAPTCHAs — designed to look identical to legitimate Cloudflare verification screens — and similar bot-detection mechanisms are deployed by attackers specifically to block the automated crawlers that reputation services rely on to classify sites. An automated crawler hits the fake CAPTCHA and gets nothing. The human victim — or the AI agent — gets the phishing page.

TDS infrastructure makes this filtering systematic. Attackers configure TDS to serve entirely different content depending on who's asking: a security scanner gets a blank page or a redirect to a legitimate site, a targeted victim gets the payload. The scale of TDS abuse has grown sharply — more than 120 distinct campaigns abused the Keitaro TDS platform alone in a four-month window between October 2025 and January 2026, spanning 13,500 domains. The economics are straightforward: every visitor a reputation service can't analyze is a potential victim that never gets warned.
*(Source: The Hacker News, April 2026)*

### 3. The Encrypted Session Problem

Modern SWGs perform TLS inspection to see inside HTTPS traffic. But even with TLS inspection enabled, the SWG operates at the network layer — it can see the traffic, but it cannot see what executes inside the browser's rendering engine. Dynamic script injection, DOM manipulation, and in-session payload assembly happen inside the browser process, below the visibility of any network inspection tool.

HEAT (Highly Evasive and Adaptive Threats) techniques are specifically designed around this limitation: payloads are assembled dynamically within browser memory for example, with no static artifact for network tools to detect. The attack exists, in its final form, only inside the browser session.

## Why LLM-Based Threat Detection Is a Breakthrough

The answer to attacks that evade static analysis is not better static analysis. It's intent-based detection: understanding what a page is attempting to do, not just what it looks like or where it comes from.

Large Language Models understand context and intent within web content in a way that rule-based systems cannot. A reputation filter asks: 'Is this domain on a known-bad list?' An LLM-native detection system asks: 'What is this page trying to accomplish? Does its behavior match its stated purpose? Is the credential form connected to the domain it claims to represent?'

This matters because modern attacks are specifically engineered to pass the first question while failing the second. AI-generated phishing pages look legitimate. Fake Adobe portals render correctly. ClickFix pages look like standard browser verification. The tell is always in the intent — and intent is what LLM-based detection is built to read.

### In Practice: How Intent-Based Detection Caught What Everything Else Missed

In February 2026, Menlo Security detected an attack targeting a user at one of the nation's largest healthcare organizations. The user clicked an email link to what appeared to be an Adobe secure document portal. The page rendered correctly. The domain was clean — zero detections across all vendors on VirusTotal at time of click. Every reputation-based tool in the stack saw nothing wrong.

What the page was actually doing was different from what it claimed to be doing. It was requesting the download of an executable file from a site with no legitimate connection to Adobe. Because Menlo operates at the browser session layer — analyzing what a page actually does, not just where it comes from — the gap between the page's stated purpose and its actual behavior was identifiable in real time. The download was blocked before it executed.

Security teams that have integrated LLM-based detection directly into the browser session layer — rather than relying on post-delivery scanning — are closing the window that reputation-based tools leave open. For zero-day lures and AI-generated impersonation pages, that window is the entire attack.

CHAPTER 3

# The Weapon

## Six Attack Techniques Engineered to Bypass Your Security Stack

## From Lure to Breach: The Modern Attack Chain

The browser is where the chain starts. Understanding that changes everything about where you intervene.

Modern ransomware groups don't breach networks. They buy access to them — access that was harvested from a browser session, packaged by an infostealer, and sold through an underground market before your security team knew anything happened. By the time ransomware encryption triggers on your endpoints, the initial compromise is four steps in the past and hours or days in the rearview.

The chain looks like this:

**Step 1 — The lure lands in the browser.** A phishing link, a fake CAPTCHA, a compromised site pushing a fake browser update. The user interacts. The chain begins — entirely inside a browser session, below the visibility of every tool in your stack.

**Step 2 — The infostealer harvests the session.** It doesn't just steal passwords. It harvests browser-stored session cookies — authenticated tokens that represent live, MFA-cleared sessions to your SaaS applications, VPNs, and cloud infrastructure bypassing every authentication control you have. Stolen credentials are now the second most common initial access vector in Mandiant's investigations, rising from 10% to 16% year over year, driven directly by infostealer activity. (Source: Mandiant M-Trends 2025)

**Step 3 — The access gets sold.** Stolen data is packaged into stealer logs and sold through underground markets within hours. Initial Access Brokers resell verified enterprise access to ransomware affiliates. 54% of ransomware victims had their credentials in stealer marketplaces before the attack hit. (Source: Verizon DBIR 2025)

**Step 4 — Ransomware deploys.** The affiliate logs in as a legitimate user, moves laterally, and deploys the payload. Ransomware appeared in 44% of all confirmed breaches in 2025 — up from 32% the prior year. Time from credential sale to encryption: sometimes under 48 hours. (Source: Verizon DBIR 2025)

**Step 5 — Your EDR fires.** This is the first moment most security stacks see the attack. Four steps in. The damage is done.

The implication for browser security is direct: the infostealer reaches the endpoint through the browser. Stop it there, and Steps 2 through 5 never happen. But ransomware is only one possible ending. What happens to stolen access when a ransomware group isn't the buyer is, in many ways, a larger problem.

### The New Kill Chain — Beyond Just Ransomware

The browser has become the primary extraction point for infostealers: tools capable of harvesting session tokens, credentials, payment data, and cryptocurrency wallets before an EDR ever flags a suspicious process.

Once inside, attackers leverage this access to deploy Remote Access Trojans (RATs) or weaponize legitimate Remote Management and Monitoring (RMM) tools for stealthy persistence. This access is rarely single-use — it is immediately monetized:

- Compromised endpoints conscripted into botnets for distributed attacks
- Silent crypto mining via hijacked compute resources
- Session tokens sold to Initial Access Brokers (IABs) for downstream ransomware deployment
- dentity packages — credentials, session cookies, personal data — sold or used for account takeover

In this ecosystem, the browser isn't just a vulnerability. It's the supply line for the entire threat landscape.

# Named Threat Groups: Who Is Running This Chain

## Qilin Ransomware (RaaS)

**Qilin (RaaS)** is among the most active ransomware groups operating today — 113 victims in February 2026 alone, the highest single-month volume of any active group globally, and 24% of all reported U.S. state, local, and government incidents in Q2 2025.

Their browser attack chain follows the pipeline exactly: phishing delivers credential theft, browser-stored passwords are harvested from Chrome's credential store via LSASS memory dumping, and stolen sessions enable lateral movement through legitimate tools (ScreenConnect, AnyDesk, RDP) that generate no behavioral alerts.

In 2026, Qilin introduced BYOVD techniques capable of disabling more than 300 EDR drivers — including a zero-day in a gaming anti-cheat driver — meaning that by the time encryption begins, your endpoint protection may already be offline.

*(Sources: CIS MS-ISAC Q2 2025; Cisco Talos; The Hacker News, 2026)*

## Interlock Ransomware

Interlock's distinguishing characteristic is that their entire initial access strategy is built around the browser. Entry point: compromised legitimate websites serving fake Chrome or Edge update installers. The "update" executes a PowerShell backdoor.

In 2025, they added FileFix — abusing Windows File Explorer's address bar to deliver payloads through a trusted-looking interface — and were directly linked to ClickFix infrastructure targeting healthcare organizations.

Interlock's dwell times are measured in months before encryption triggers, which is why browser-layer detection matters: the chain can be broken at Step 1 long before the ransomware group decides to act. Mandiant has independently tracked Interlock's custom backdoor tooling under the designation WINDYTWIST.SEA.

*(Source: Mandiant; CIS MS-ISAC, 2025)*

# Attack Vector Deep-Dive: Six Techniques Your Stack Doesn't Catch

The following six techniques represent the highest-impact, most evasive attack patterns observed across Menlo customer environments in 2025–2026. Each was specifically engineered to bypass legacy detection tools — reputation filters, signature-based scanners, network inspection, and endpoint controls. For each technique, use the 'Why Existing Tools Miss It' column as an evaluation test case against your own security stack.

## 1. ClickFix — The #1 Initial Access Method of 2025

| Associated Threat Groups | Targeted Industries | Why Existing Tools Miss It |
|---|---|---|
| Interlock, Qilin affiliates, Storm-0426, Latrodectus, Lumma Stealer campaigns, MuddyWater (attributed) | Financial services, government, healthcare, transportation, education — effectively all sectors | No malicious binary on disk. User executes the payload themselves via legitimate Windows processes (PowerShell, mshta.exe). EDR sees normal user behavior. URL reputation tools see a legitimate-looking page. |

ClickFix is the dominant initial access technique of 2025, identified by Microsoft as responsible for 47% of all initial compromises observed across their threat intelligence — more than traditional phishing — and by ESET as having surged 517% in detections in the first half of 2025 alone.

The technique is deceptively simple: a user lands on what appears to be a routine verification page — a CAPTCHA, a browser error, a Cloudflare check — and is prompted to complete a short sequence of keystrokes. What they don't know is that malicious JavaScript has already injected a PowerShell command or mshta.exe call into their clipboard. When they follow the instructions, they execute it themselves. No malicious binary lands on disk. No suspicious process spawns from an untrusted parent. From the EDR's perspective, a user opened PowerShell normally.

The reason ClickFix is so effective is precisely why it's so hard to stop with conventional tools: the user is the execution mechanism, and the entire chain is built from legitimate system actions. When the same page renders inside Menlo's isolation layer, the clipboard injection happens in a cloud container — not the user's real session — and the command, if executed, detonates where it has no target.

*(Source: Microsoft Digital Defense Report 2025; ESET H1 2025 Threat Report)*

## 2. FileFix — Interlock's 2025 Evolution of ClickFix

| Associated Threat Groups | Targeted Industries | Why Existing Tools Miss It |
|---|---|---|
| Interlock | Healthcare, city government, regional enterprises across North America and Europe | Abuses Windows File Explorer's address bar — a trusted, familiar interface that security awareness training has not addressed. Payload execution path is identical to a legitimate user action. |

Where ClickFix uses the Windows Run dialog (Win+R), FileFix abuses Windows File Explorer's address bar. The technique achieves the same outcome — user-executed malicious command — through a different, less-trained interface. Users who have learned to be suspicious of unexpected Run dialogs may not question a prompt appearing within File Explorer's familiar context.

FileFix represents the evolutionary pressure that defenders need to anticipate: every time a specific technique becomes recognizable, the variant emerges. The common thread across ClickFix, FileFix, and their successors is not the specific interface abused — it's the underlying strategy of using the user as the execution mechanism.

*(Source: BleepingComputer, July 2025)*

## 3. HTML Smuggling

| Associated Threat Groups | Targeted Industries | Why Existing Tools Miss It |
|---|---|---|
| Widely adopted across ransomware and nation-state actors; used in campaigns attributed to Qakbot, Nokoyawa, and multiple Eastern European RaaS groups | Financial services, professional services, critical infrastructure | Payload never traverses the network in its malicious form. Assembly happens inside the browser's JavaScript engine — entirely below network inspection visibility. |

HTML smuggling has become a favored delivery technique across both ransomware groups and nation-state actors. A malicious HTML file — delivered as an email attachment or via link — contains an encoded payload embedded in JavaScript. The browser renders the page, the JavaScript assembles the payload locally in memory, and a binary is downloaded to the endpoint. The network sees a legitimate HTML fetch. Nothing malicious ever crosses the wire. By the time a post-download scanner looks at the file, it's already assembled on the endpoint.

*(Source: Mandiant M-Trends 2025)*

## 4. Adversary-in-the-Middle (AiTM) Phishing — Identity as the Target

| Associated Threat Groups | Targeted Industries | Why Existing Tools Miss It |
|---|---|---|
| UNC3944 (Scattered Spider), Storm-1167, PhaaS platforms including Greatness and EvilProxy; Sneaky 2FA campaign | Financial services, technology, SaaS-heavy enterprises; any organization relying on Microsoft 365 | MFA is bypassed by design — the attacker intercepts the authenticated session token, not the password. Traditional credential monitoring watches for password reuse, not token theft. The session appears fully legitimate to every downstream tool. |

In a traditional phishing attack, the attacker captures a username and password. MFA stops reuse: even with valid credentials, the attacker can't authenticate without the second factor. AiTM defeats this by sitting between the user and the legitimate service, intercepting the authenticated session token after MFA is completed. The attacker receives not just the credential but the live session — fully authenticated, MFA bypassed.

This is an identity compromise event, not just a credential theft event. Resetting a password does not invalidate a stolen session token. Credential monitoring that watches for password reuse does not detect session token theft. The compromise exists at the identity layer — the authenticated session — not the password layer.

PhaaS platforms like Greatness, EvilProxy, and others provide AiTM capability as a service: pre-built phishing infrastructure, brand impersonation templates, session harvesting, and live relay to the legitimate target service. The Sneaky 2FA campaign (January 2026) used this model specifically to target Microsoft 365 accounts.

*(Source: Menlo Threat Research)*

### Why This Matters for Zero Trust:

Zero trust architecture extends to applications, not just networks. The JLR cyberattack (facilitated by an SAP vulnerability) and subsequent £485M loss — representing a 0.1% drop in UK GDP — demonstrated that trusting an authenticated session without continuous verification of session behavior is a critical architectural gap. Qilin affiliates have exploited this same gap, using SAP NetWeaver (CVE-2025-31324) as an entry point. Zero trust must include session-layer behavioral verification, not just initial authentication.

(Sources: Cyber Monitoring Centre, October 2025; Bank of England Monetary Policy Report, November 2025)

## 5. RMM Tool Abuse — Legitimate Tools as Backdoors

| Associated Threat Groups | Targeted Industries | Why Existing Tools Miss It |
|---|---|---|
| Qilin affiliates, Interlock, Storm-0426 | Healthcare, financial services, managed service providers and their downstream customers | RMM tools are legitimate, signed, and expected on enterprise networks. Traffic blends seamlessly with normal IT operations. Behavioral detection has no baseline anomaly to trigger on when a known tool is used by an unknown actor. |

Remote Management and Monitoring (RMM) tools — ScreenConnect, AnyDesk, TeamViewer, and others — are legitimate, trusted enterprise IT tools. Attackers abuse them because they appear legitimate to every security tool in the stack, blend into normal IT traffic, and provide full remote control of compromised endpoints.

The attack on one of the largest Healthcare Systems in North America (February 2026) used exactly this technique: a trojanized Remote Management executable disguised as a PDF download via a fake Adobe portal. Had the user executed the file, the attacker would have achieved persistent remote access behind the cover of a legitimate IT tool.

Qilin affiliates use legitimate RMM tools including ScreenConnect and AnyDesk for lateral movement after initial access. Interlock's NodeSnakeRAT provides a custom alternative for environments where commercial RMM tools are blocked.

*(Source: Menlo Threat Research Case Study; Microsoft Security Blog)*

## 6. Clickjacking and Malvertising

| Associated Threat Groups | Targeted Industries | Why Existing Tools Miss It |
|---|---|---|
| Vextrio TDS network; Lumma Stealer distribution campaigns; multiple IAB-affiliated advertising fraud operations | Broad consumer and enterprise targeting via legitimate ad networks; financial services and retail disproportionately impacted by credential harvesting outcomes | Malicious ads served through legitimate advertising networks on legitimate websites. URL reputation tools cannot block the legitimate sites serving the ads. |

Clickjacking overlays an invisible or transparent element over a legitimate-looking page, causing a user's click to trigger an action they didn't intend — authorizing a download, granting permissions, or submitting a form. The interaction appears entirely normal from the user's perspective. EDR and network tools see a legitimate user performing a legitimate click.

Malvertising distributes attack infrastructure through legitimate advertising networks, reaching users on legitimate websites through ad placements. A Microsoft security blog analysis documented a May 2025 malvertising campaign using free movie streaming sites: users pressing 'play' were redirected to ClickFix landing pages delivering Lumma Stealer. Single-day traffic to scam pages: tens of thousands to hundreds of thousands of unique visitors.

*(Source: Menlo Threat Research; Microsoft Security Blog August 2025; HP Wolf Security Q3 2025)*

CHAPTER 4

# The Path Forward

## Your 6-Step Browser Security Playbook

## 2026 Predictions: Where This Goes Next

### Agentic AI Is Already Here — and Already a Security Problem

our employees aren't waiting for a policy decision on AI browsers. The agents are already running — inside the browsers they already have. Gemini is built into Chrome. Copilot is built into Edge. Claude and ChatGPT run inside browser tabs across your workforce every day. These aren't tools employees had to seek out or install. They came with the browser, turned on by default.

The security question isn't whether your workforce is using AI agents. They are. The question is whether those agents are operating inside a security architecture that can see what they're doing — what data they're accessing, what instructions they're following, what actions they're taking on behalf of users who may not even realize an agent is running.

Blocking AI browsers such as Atlas or Comet misses the point entirely. The exposure is already inside Chrome and Edge. The 2026 strategic position isn't about which AI browsers to allow. It's about whether the browser session layer is governed, visible, and controlled — regardless of which AI is running inside it..

### AI-Accelerated Attack Velocity

The compression of the kill chain from days to minutes will continue. As adversarial AI matures, expect: fully automated spear-phishing campaigns generated and launched with zero human input; AI-driven target profiling that tailors lures to individual users based on scraped public data; and machine-speed AiTM operations that harvest and monetize session tokens before any detection system fires.

### Credential Theft as Identity Compromise — at Scale

The transition from password theft to session token theft means that traditional identity security metrics (password complexity, MFA coverage) are becoming inadequate proxies for actual identity security posture. 2026 will see increased regulatory and board-level focus on session-layer identity controls — and organizations without browser-layer visibility will struggle to demonstrate compliance.

# The 6-Step Browser Security Playbook

The following six steps represent the practical implementation path from current-state browser exposure to a controlled, visible, and resilient browser security posture. Each step maps to specific capabilities and produces specific business outcomes.

## Step 1: Eliminate the Attack Surface via Cloud Isolation

Move all web content execution — DOM rendering, JavaScript, CSS — to a remote cloud container. Web content executes in the Menlo Cloud before it reaches the user's browser or AI agent's session.

Why it matters: This is the structural answer to the CVE patch gap. When a Chrome zero-day is discovered, it detonates in the cloud — not on your endpoint. The 6-day exposure window between vulnerability discovery and enterprise patch deployment becomes irrelevant. The exploit has no target.

Practitioner note: This applies equally to human browser sessions and agentic AI sessions. The same isolation architecture that protects employees from zero-day exploits governs the behavior of AI agents operating in your environment.

## Step 2: Counter Adversarial AI with Intent-Based Detection

Deploy Computer Vision and LLM-native detection that analyzes web page intent — not just content signatures or domain reputation. This means understanding what a page is attempting to accomplish, not just where it comes from.

HEAT Shield AI powered by Gemini LLM applies this at the session layer: analyzing page rendering behavior, user interaction requests, and form structure to identify social engineering lures that have never been seen before. It leverages threat intelligence from Google Threat Intelligence and Mandiant to correlate local browser events with global campaign data and TTPs.

Large Healthcare System Case Study: Zero VirusTotal detections at time of click. Blocked by HEAT Shield AI because the page's intent — delivering an executable disguised as a PDF — was identifiable regardless of domain reputation.

## Step 3: Disrupt Credential Theft at the Input Layer

Implement read-only protection modes for suspicious or unknown pages. When a user navigates to a page that exhibits credential-harvesting indicators, the browser technically prevents credential input — typing is blocked, form submission is disabled.

This neutralizes AiTM attacks at the point of execution: even if the user reaches the phishing page, their credentials cannot be entered. Session token harvesting requires a live authenticated session to intercept — blocking credential input prevents the initial authentication that the AiTM attack depends on.

Identity implication: Zero trust must extend to the browser session layer. Not trusting the application is as important as not trusting the network — as the SAP/JLR incident and Qilin's SAP NetWeaver exploitation both demonstrated.

## Step 4: Sanitize All File Downloads with CDR

Implement Content Disarm and Reconstruction (CDR) for all downloads processed through the browser. CDR inspects files in the cloud, strips active content (macros, embedded scripts, JavaScript), and delivers a safe, sanitized version to the endpoint.

What CDR catches: weaponized Office documents, PDFs with embedded scripts, HTML files containing smuggled payloads, archive files with DLL sideloading components.

Practitioner note: CDR operates on the file before it reaches the endpoint, not after. This matters because HTML smuggling and DLL sideloading operate below the visibility of post-download scanning.

## Step 5: Enforce Last-Mile Data Loss Prevention at the Session Layer

Deploy Browser DLP controls inside the browser session to intercept data exfiltration at the point of interaction — before sensitive content leaves the organization's control.

What Browser-layer DLP catches that traditional DLP cannot: sensitive content typed or pasted into AI tools (ChatGPT, Claude, Gemini, Copilot); data entered into web forms on unmanaged services; copy-paste of regulated data between a managed application and an external site; shadow AI usage that never creates a file or traverses the email gateway.

Data loss is almost never malicious. It's almost always someone trying to get their job done — pasting content into a collaboration tool, uploading a document to an AI assistant, sharing something without realizing it contains sensitive information. The intent is good. The exposure is real. Browser-layer DLP detects and masks sensitive content in real time, preserving the workflow while protecting the data.

## Step 6: Enforce Isolation-First Secure Application Access

Adopt an isolation-first approach to private application access, enforcing least-privilege through browser session controls rather than VPN or agent-based access.

This approach allows secure access to private applications from unmanaged devices without requiring traditional VPN infrastructure — significantly reducing the attack surface associated with VPN credential theft and edge device exploitation (both primary Qilin initial access vectors).

Zero trust implication: Zero trust must extend to application trust as well as network trust. The Oracle Business Server exploit and the SAP NetWeaver zero-day (CVE-2025-31324, CVSS 10.0) that Qilin exploited both demonstrate that trusting authenticated access to enterprise applications without session-layer verification is a critical architectural gap. Isolation-first access means that even a compromised application session is contained — the endpoint is not exposed.