



Enterprise Browser Security Buyer's Guide

**Selecting the right solution for your
organization's security requirements**

In This Guide

Browser Security Is Enterprise Security

It's easy to ignore browsers. They've been in widespread use by both consumers and enterprises for nearly three decades, though they remain primarily a consumer software product. The slow-and-steady assimilation of enterprise software functionality by browsers has been the security equivalent of "boiling a frog," with the result that browsers are at the nexus of all of the most pressing threats that organizations are confronted with, including credential-based threats such as phishing and session cookie/token theft, endpoint security concerns like malware and ransomware, and data leakage and loss — as well as general policy enforcement and governance.

To secure the browser is to secure the organization, and owing to both its outsized role in the enterprise and its dual persona as a professional and personal tool, leveraging this ubiquitous application to deliver security capabilities can pay enormous dividends. This guide provides an overview of available solutions and considerations to help you select the best browser security implementation for your organization.

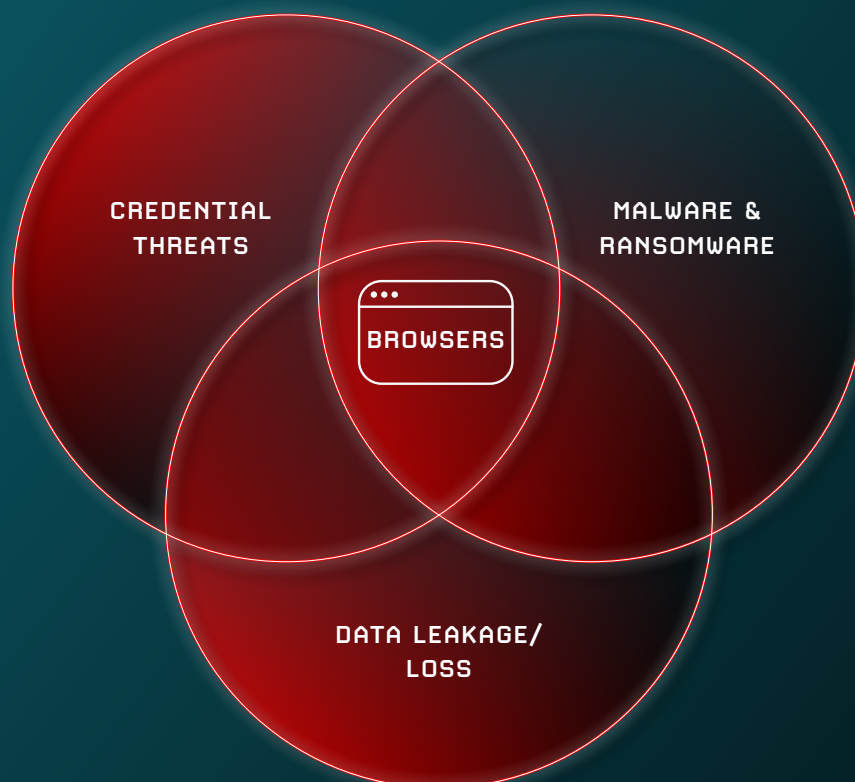


Figure 1. Browsers are at the intersection of the most pervasive threats confronting organizations

Table of Contents

An Overview of Browser Security Solutions	4
Browser Functionality	4
Legacy Solutions	5
Modern Solutions	7
Summary Coverage of Browser Subfunctions and Security Capabilities	9
Deployment Considerations	10
User and Device Types	10
OS Support	11
Enterprise Integration and Operations	11
Data Residency/Sovereignty	12
User Experience (UX) and Privacy	12
Core Security Capabilities	13
Attack Surface Reduction	13
User Identity/Credential Protection	13
Malicious Web Content	13
Data Leakage/Loss Prevention (DLP)	14
Governance	15
Additional Capabilities	16
On-premises Web App (Remote) Access	16
SaaS App Access Control	16
Buyer's Checklist/Vendor Questionnaire	17
Example Architecture/Deployment Diagram	21
Corporate-owned/Managed Device Deployment	21
Employee-owned (BYO)/Unmanaged Device Deployment	22

An Overview of Browser Security Solutions

Browser Functionality

There are two major categories of browser functionality that are relevant for any browser solution:

- Retrieving and rendering code, usually a combination of HTML, CSS, and JavaScript
- Enabling users to interact with the content produced by the rendered code

These functions involve different subfunctions, each of which requires different security capabilities.

BROWSER SUBFUNCTION	RELEVANT SECURITY CAPABILITIES
Networking	URL filtering
Rendering/content presentation	Data masking
Document Object Model (DOM) manipulation	User action control (e.g., copy/paste controls, preventing input to form fields)
Programmatic access to browser (i.e., public browser extension APIs)	Extension inventory, control of extension permissions, and API access
Execution	Exploit prevention
Data management (memory, browser persistent storage)	Advanced attack detection and response, storage encryption
Host OS integration	Posture checking, anti-tampering

Legacy Solutions

Web security has been an issue since internet access became commonplace in businesses, and web security solutions have been in widespread use for many years.

PROXIES

Proxies are the oldest form of web security solution and were originally deployed to facilitate internet access from machines without requiring direct access or a large pool of publicly available IP addresses. Today, proxies serve two roles:

- Secure web gateways (SWGs) provide authenticated user access (for logging and auditing), content filtering, and limited malware detection
- Cloud access security brokers (CASBs) provide more advanced access control and logging/auditing

PROS

- Mature technology with comprehensive URL categorization and web application databases

CONS

- Requires traffic steering/redirection, which can be difficult to implement on remote/unmanaged systems and can introduce latency that impairs user experience
- Requires TLS interception (aka break-and-inspect) to perform deep packet inspection (DPI) and detect malware; this approach is not compatible with sites and apps that use HTTP Public Key Pinning (HPKP, aka cert pinning)
- Limited sandboxing capabilities for dynamic analysis due to compute and network performance overhead

REMOTE BROWSER ISOLATION (RBI)

RBI works by moving browser functionality off of the user's endpoint to remote infrastructure to maintain security while reducing the performance impact that less powerful endpoints incur from virtualization. There are several competing implementations of RBI:

"Pixel-pushing"

A video-based approach similar in function to remote desktop systems, in which the pixels of the browser window are captured and streamed to the end user's screen (sometimes using a proprietary client).

PROS

- Works with existing browsers
- The browser session is completely isolated from the end user's machine

CONS

- Requires costly and complex on-premises server infrastructure or costly cloud subscriptions
- Network latency can affect the responsiveness of the session and compromise user experience

Document Object Model (DOM) Reconstruction

DOM reconstruction is intended to overcome the performance issues associated with pixel pushing. It strips the active/dynamic elements out of web content before forwarding it to the native browser on the local endpoint. The underlying HTML, CSS, and JavaScript are rewritten or removed in an attempt to prevent the delivery of potentially malicious content.

PROS

- Works with existing browsers
- Eliminates page elements (e.g., JavaScript) that are most commonly used to exploit or otherwise compromise browsers

CONS

- Incompatible with many modern webpages/web-based applications due to removal/neutralization of necessary code
- Network latency can affect the responsiveness of the session and compromise user experience

Field Insights and Market Feedback: BROWSER ISOLATION

Early adopters of browser isolation technology have reported that there was significant cost and operational overhead associated with server infrastructure. Migration to cloud RBI reduced the on-premises server burden, but often increased costs. They also reported the need to bypass RBI for large numbers of sites/web apps due to compatibility and user workflow issues, stating that the protection RBI offered continually decreased over time.

Network Vector Rendering (NVR)

NVR is an attempt to address the issues related to performance and fidelity in the other two approaches. It intercepts a remote browser's rasterization/rendering commands, tokenizes and compresses them, then transmits them to a WebAssembly (Wasm)-based library that unpacks, decrypts, and replays them in a local browser.

PROS

- Works with existing browsers
- Better webpage/web application fidelity than DOM reconstruction
- Less sensitive to network latency than pixel-pushing

CONS

- Incompatible with many modern webpages/web-based applications due to removal/neutralization of necessary code

Modern Solutions

ENTERPRISE BROWSERS

Like many modern consumer browsers, virtually all enterprise browsers are derived from the open-source Chromium project. These dedicated browsers are effectively interface wrappers for the core browser functionality. This enables support for current web standards without the need to build and maintain a browser from the ground up. It also provides the ability to modify higher-level functions of the browser, as well as removing unnecessary consumer-focused features and adding enterprise-focused features (e.g., data masking).

PROS

- Web compatibility and functionality equivalent to mainstream commercial browsers
- Easier enforcement of some enterprise policies like content filtering and data loss prevention (DLP) compared to consumer browsers

CONS

- Users can install other browsers and (re)introduce risks that the enterprise browser was intended to mitigate
- Inherits all vulnerabilities from Chromium
- Limited ability to protect against browser exploits or other advanced attacks
- Forces users to migrate from existing browsers to a new, unfamiliar browser that may lack features introduced by mainstream browser vendors

Field Insights and Market Feedback: ENTERPRISE BROWSERS

Early adopters of enterprise browsers have reported that users resisted migrating away from their existing browsers due to concerns about usability. They also reported that their “patch gap” — the time between when a security vulnerability was publicly disclosed and when a patch became available — widened compared to standard commercial browsers because it took longer for upstream security fixes to be integrated into and deployed to enterprise browser codebases, increasing their exposure to exploitation.

BROWSER SECURITY EXTENSIONS

All modern browsers allow their functionality to be augmented through the use of extensions. Browser security extensions use the same extension APIs as conventional extensions, but leverage them to provide organizations with additional controls and functionality (vs. letting end users customize browser behavior).

PROS

- Does not require migrating to a new, dedicated browser

CONS

- Functionality may be inconsistent across different browsers due to differences in the extension APIs exposed by different browser vendors; vendors change and/or restrict API capabilities over time and constraints eliminate capabilities like exploit prevention
- Limited network visibility
- Limited ability to protect sensitive data (e.g., session cookies)
- No OS integration for device posture checking and anti-tampering

BROWSER RUNTIME MODULES

Runtime modules of various types have always played a vital role in security. Just as modern endpoint security tools hook the operating system kernel, browser runtime modules hook the “kernel of the browser”: the JavaScript Engine.

PROS

- The same sensor can work with virtually any browser without modification (i.e., there is a single code base for all sensor functionality)
- Can be pre-installed or dynamically injected into sessions “just in time”
- Provides more security functionality by operating at the core of the browser (compared to enterprise browsers and security extensions, both of which use — and are constrained by — higher-level APIs)

CONS

- Capabilities may vary widely between browser runtime modules because they are not defined/standardized by browser vendor APIs, requiring more due diligence while selecting

Field Insights and Market Feedback: BROWSER SECURITY EXTENSIONS

Early adopters of security extensions have reported that changes to the browser extension APIs have forced security extension vendors to change or remove important functionality. They have also reported that getting all of the advertised functionality requires more complex distribution/installation processes using private stores since some of the required capabilities/permissions are restricted or forbidden in browser vendors' public web stores.

Field Insights and Market Feedback: BROWSER RUNTIME MODULES

Early adopters were initially concerned that browser runtime modules would impact site/app compatibility, browser performance, or both, but ultimately found that those issues did not materialize in their deployments. They also reported very high levels of user satisfaction when migrating from solutions such as RBI. And, they found that browser runtime modules helped them detect and prevent a large number of attacks or threats that had bypassed other security tools.

Summary Coverage of Browser Subfunctions and Security Capabilities

The following table maps the browser subfunctions and relevant security capabilities supported by each of the enterprise browser security solutions described in the previous section.

BROWSER SUBFUNCTION	RELEVANT SECURITY CAPABILITIES	PROXIES	RBI	EXTENSIONS	ENTERPRISE BROWSERS	ENTERPRISE BROWSERS
Networking	URL filtering	✓		✓	✓	✓
Rendering/ content presentation Investigation Analysis	Data masking Investigation and Correlation			✓	✓	✓
Document Object Model (DOM) manipulation	User action control (e.g., copy/paste controls, preventing input to form fields)		✓	✓	✓	✓
Programmatic access to browser (i.e., public browser extension APIs)	Extension inventory, control of extension permissions, and API access			✓	✓	✓
Execution	Exploit prevention		✓			✓
Data management (memory, browser persistent storage)	Advanced attack detection and response, storage encryption					✓
Host OS integration	Posture checking, anti-tampering				✓	✓

Deployment Considerations

Enterprise browser security is a new category. This section identifies features and capabilities that a solution must or should include to ensure a successful deployment.

User and Device Types

The modern enterprise is made up of a mix of full-time employees and contractors/third parties using a combination of corporate (managed) and personal (unmanaged) devices. The various combinations create complexity for deploying and managing an enterprise browser security solution.

USER TYPE	CORPORATE (MANAGED) DEVICE	PERSONAL (UNMANAGED) DEVICE
Full-time employee (FTE)	PRO(S): Provides organizations with the highest level of control CON(S): Can reduce flexibility for employee work time, location, and device	PRO(S): Offers more flexibility for employee work time, location, and styles CON(S): Significantly decreases the organization's control due to ownership and privacy concerns
Contractor/third-party	PRO(S): Provides organizations with the highest level of control CON(S): Adds costs and/or administrative overhead associated with endpoint lifecycle management	PRO(S): Eliminates costs and administrative overhead associated with endpoint lifecycle management CON(S): Severely limits the organization's ability to enforce policies/security controls, collect security-relevant telemetry, and maintain compliance

An enterprise browser security solution **MUST**:

- Support all possible user and device combinations from the matrix above
- Enable an organization to create a boundary to protect its resources, whether those resources are full-fledged managed endpoints or isolated “workspaces” on unmanaged endpoints
- Provide protection for all browsers that may be installed on a system
- Respect user privacy

OS Support

Just as the workforce is made up of different types of users, modern endpoint fleets are composed of heterogeneous operating systems. An enterprise browser security solution **MUST** support all major operating systems: Windows, MacOS, and Linux. It **MUST** also support the two major mobile operating systems: Android and iOS.

INSTALLATION METHODS

Perhaps the most significant part of OS support is the availability of different installation methods so that IT and security teams have the flexibility to deploy the solution in a way that achieves the highest coverage. An enterprise browser security solution **MUST** support:

- **Mobile device management (MDM) or unified endpoint management (UEM):** For organizations with modern endpoint management tools
- **Standard OS installation packages:** For organizations with traditional endpoint management tools, script-based software installation, or whose admins need the ability to more extensively customize installations
- **Non-persistent installation/OS-agnostic injection:** For scenarios where an organization needs to ensure that a browser is protected and/or controlled while accessing its resources but does not have the ability to install software directly

Enterprise Integration and Operations

ENTERPRISE DIRECTORIES/IDENTITY PROVIDERS (IDPS)

User identities are a frequent target of attackers and are an essential component for protecting enterprise resources. An enterprise browser security solution **MUST** integrate with existing enterprise directories and/or IdPs to support access controls and other policies, as well as identification and correlation of user activity.

SECURITY INFORMATION AND EVENT MANAGEMENT (SIEM)/EXTENDED DETECTION AND RESPONSE (XDR)

Browsers are an important source of telemetry because the events they generate are very high-fidelity. Collecting this telemetry can significantly improve threat detection, shorten investigation/triage, and improve incident response. An enterprise browser security solution **MUST** export events in common log formats — e.g., syslog, Common Event Format (CEF), and JavaScript Object Notation (JSON) — that can be ingested, normalized, and used for enriching other data sources in popular log aggregation and analysis platforms.

Data Residency/Sovereignty

Many national governments are creating regulations covering data generated by or relating to their citizens. An enterprise browser security solution **SHOULD NOT** require transmission of user data to vendor and/or third-party data centers for policy evaluation or enforcement.

User Experience (UX) and Privacy

One of the main factors influencing users' attempts to bypass security controls is the amount of friction those controls create in existing workflows. Preserving the native user experience can disincentivize the desire to bypass security controls. An enterprise browser security solution **SHOULD** work transparently with any user's preferred browser(s) and clearly indicate which activity is monitored/controlled and which is not.

Core Security Capabilities

Attack Surface Reduction

Despite focused and prolonged hardening efforts, browsers have been the single largest source of zero-day vulnerabilities and exploits for the past four years, and the time it takes for companies to patch can range from weeks to months. An enterprise browser security solution **MUST** provide prevention against zero-day and N-day exploits to protect organizations during the period when a vulnerability has been disclosed and an exploit is in the wild, but a patch is not available or deployed (aka the patch gap).

User Identity/Credential Protection

PHISHING

Phishing is a constant threat to organizations of all sizes, but emails belonging to phishing campaigns are increasingly difficult to identify, and phishing sites are becoming better obfuscated. The existing anti-phishing ecosystem is usually only effective during the last third of a typical campaign. Additionally, phishing attacks that leverage reverse proxies (e.g., Evilginx) have gained popularity due to the difficulty of detecting them and because they enable attackers to intercept users' session cookies/tokens and bypass multifactor authentication (MFA). An enterprise browser security solution **MUST** be able to identify and block well-known phishing sites, as well as being able to identify and block previously unknown/uncategorized phishing sites in real time.

SESSION COOKIE/TOKEN THEFT PROTECTION

MFA has helped significantly reduce account compromise. As a result, adversaries have begun targeting the long-lived secondary credentials (tokens) created after MFA has been completed that are kept in browser storage. An enterprise browser security solution **MUST** be able to encrypt session cookies to prevent abuse/reuse and mitigate the risk of theft.

CREDENTIAL MONITORING

Just as monitoring consumer identities can help prevent compromise or fraud, monitoring enterprise identities can provide an early warning for at-risk users. An enterprise browser security solution **SHOULD** alert administrators when a user attempts to reuse passwords between accounts and sites. Additionally, it **SHOULD** monitor for and alert on the compromised credentials found on the internet.

Malicious Web Content

MALWARE

JAVASCRIPT

JavaScript is a popular choice for malware because the language is very powerful, and scripts written in it are highly portable, trivial to obfuscate, and easy to deliver. Moreover, it is especially useful for defense evasion because — unlike other varieties of malware — JavaScript execution can stay confined within the browser process, rather than running in its own process and/or spawning child processes. Common JavaScript malware families include Balada Injector, SocGhosh, and WoofLocker. An enterprise browser's security solution **MUST** provide detection of and protection against JavaScript-based malware.

DRIVE-BY COMPROMISE (AKA DRIVE-BY DOWNLOADS) AND HTML SMUGGLING

Drive-by downloads target users via their browsers during normal web browsing (often without their knowledge or direct interaction) using a technique known as HTML smuggling. This technique abuses standard features of HTML 5 and/or JavaScript to transmit binaries (or other file types) as text, enabling them to evade scanning and detection, and reassembling them into their original format within the browser. This technique is highly effective because it does not depend on existing vulnerabilities to deliver a malicious payload. An enterprise browser security solution **MUST** be able to identify and prevent drive-by downloads and HTML smuggling.

EXTERNAL MALWARE ANALYSIS SANDBOX INTEGRATION

Users may download malicious files from the internet in a wide variety of different formats. To prevent the introduction of malware from downloaded files that are handled by applications other than the browser, an enterprise browser security solution **SHOULD** support integration with external malware analysis platforms (e.g., Cuckoo Sandbox, hybrid-analysis.com, etc.).

MALICIOUS SITES

Although adversaries have largely pivoted to compromising legitimate sites instead of running dedicated malicious sites, many malicious sites remain active, usually as a result of “domain squatting,” domains using alternate character sets to appear legitimate, or domains based on common typos for legitimate sites. An enterprise browser security solution **MUST** be able to identify and block both overtly malicious sites and the malicious elements that may be present on a compromised — but otherwise trustworthy or high-reputation — site (e.g, popular websites that have been compromised via WordPress or other content management system vulnerabilities).

MALICIOUS EXTENSIONS

Malicious extensions are an excellent way for adversaries to attract targets, since all an attacker needs to do is offer an extension with some functionality a user needs or values. They are also difficult for users to identify because they usually come from curated sources (e.g., extension stores operated by browser vendors), and there are very few obvious indicators for users to observe. Additionally, benign extensions can (and have) become malicious as a result of updates. An enterprise browser security solution **MUST** be able to identify and block malicious extensions, including tracking changes to extensions over time.

Data Leakage/Loss Prevention (DLP)

Browsers are the primary tools for employees to interact with enterprise applications and data. Browsers are also the primary interface for most generative AI platforms. Enterprise browser security solutions can provide a vital extension to existing DLP solutions for existing applications and can help organizations safely adopt public generative AI tools.

USER ACTION CONTROL

An enterprise browser security solution **MUST** provide the ability to control copy/paste, screen capture, print, file download (including blocking and/or encryption of downloads), right-click, and use of developer tools. It **SHOULD** provide granular enforcement of these policies using attributes like user identity, device type/posture, and source/destination (e.g., for copy/paste operations).

STRUCTURED/UNSTRUCTURED DATA SCANNING AND MASKING

An enterprise browser security solution **MUST** predefine patterns for common personally identifiable information (PII) data types (e.g., credit card numbers, dates of birth, Social Security numbers, etc.), as well as the ability to create custom patterns for sensitive data. It **MUST** provide the ability to dynamically mask data in rendered content, as well as the ability to scan file downloads/uploads for sensitive data and block and/or encrypt (as appropriate). The solution **SHOULD** support optical character recognition (OCR) to identify sensitive data in images.

Governance

BROWSER CONFIGURATION

IT and security administrators need to uniformly apply corporate policies, regardless of which browsers are in use and without the need to maintain multiple, redundant configurations using each browser's proprietary management interface and settings. An enterprise browser security solution **MUST** provide a unified interface for managing settings across a heterogeneous browser population.

CONTENT FILTERING

IT and security administrators need to restrict corporate web browsing to appropriate/authorized sites. An enterprise browser security solution **MUST** provide the ability to enforce acceptable use policies (AUPs) and/or block undesirable content directly in the browser without the need for inspection, evaluation, or analysis on/by external systems.

EXTENSION MANAGEMENT

Browser extensions offer users a way to install potentially unsanctioned software even if they do not have local administrative or regular software installation privileges. Extensions may request unnecessarily loose permissions and may have access to any of the resources and data that a user accesses through the browser, creating a data leakage/loss vector. An enterprise browser security solution **SHOULD** provide reporting on installed extensions, requested permissions, risk scoring, store ratings, and number of installations, as well as blocking and/or removal of installed extensions based on an administrator-defined policy.

Additional Capabilities

On-premises Web App (Remote) Access

Vulnerabilities in VPN infrastructure have been pervasive across multiple vendors and are some of the most regularly exploited, due to the extensive access remote access systems provide to enterprise networks. An enterprise browser security solution **SHOULD** provide least-privilege remote access to on-premises web applications, as well as resources accessed using Remote Desktop Protocol (RDP) or Secure Shell (SSH) without an additional VPN client, requiring the exposure of additional internet-facing services, or peering with third-party networks.

SaaS App Access Control

The built-in authentication and authorization for enterprise SaaS platforms is often limited: It relies primarily on user identity and may incorporate attributes such as browser type (user agent) and location but does not evaluate other information. An enterprise browser security solution **SHOULD** supplement SaaS access control with additional attributes like device type (corporate/managed vs. personal/unmanaged), posture, etc., that can be used in authorization decisions. Additionally, it **SHOULD** perform access control without the need to install/configure a separate traffic steering client or by requiring a “vanity domain.”

Buyer's Checklist/Vendor Questionnaire

Use the following checklist to identify requirements and evaluate enterprise browser security solutions.

CATEGORY/REQUIREMENT	MANDATORY	OPTIONAL	VENDOR-SUPPORTED	ADDITIONAL NOTES
DEPLOYMENT				
Device support: Corporate/managed	☐	☐	☐	
Device support: Personal/unmanaged	☐	☐	☐	
Employee type: Full-time (FTE)	☐	☐	☐	
Employee type: Contractor	☐	☐	☐	
OS support: Windows	☐	☐	☐	
Installation: Windows — Group Policy Objects (GPO), ADMX templates	☐	☐	☐	
Installation: Windows — MDM, UEM	☐	☐	☐	
OS support: macOS	☐	☐	☐	
Installation: macOS — .dmg, .pkg	☐	☐	☐	
Installation: macOS — MDM, UEM	☐	☐	☐	
OS support: Linux	☐	☐	☐	

CATEGORY/REQUIREMENT	MANDATORY	OPTIONAL	VENDOR-SUPPORTED	ADDITIONAL NOTES
OS installation: Linux — compressed archive (e.g., .tar.gz)	☐	☐	☐	
OS installation: Linux — package managers (e.g., .deb, .rpm)	☐	☐	☐	
OS installation: Linux — shell script	☐	☐	☐	
OS support: Android	☐	☐	☐	
OS installation: Android — MDM, UEM	☐	☐	☐	
OS support: iOS	☐	☐	☐	
OS installation: iOS/Android App — MDM, UEM	☐	☐	☐	
OS installation: iOS/iPadOS — user self-service	☐	☐	☐	
Installation: ephemeral/per-session	☐	☐	☐	
Enterprise directory integration (e.g., Active Directory, LDAP)	☐	☐	☐	
IdP integration (e.g., Azure AD, Okta, Ping Identity)	☐	☐	☐	
SIEM/XDR integration: CEF export	☐	☐	☐	
SIEM/XDR integration: JSON serialization	☐	☐	☐	
SIEM/XDR integration: Syslog export	☐	☐	☐	

CATEGORY/REQUIREMENT	MANDATORY	OPTIONAL	VENDOR-SUPPORTED	ADDITIONAL NOTES
Data residency/sovereignty: Local policy evaluation/processing (i.e., no data processing in vendor or third-party clouds required for policy enforcement)	☐	☐	☐	
Data privacy: Clear separation between work and personal browsing activity	☐	☐	☐	
UX browser support: Google Chrome, desktop	☐	☐	☐	
UX browser support: Microsoft Edge, desktop	☐	☐	☐	
UX browser support: Apple Safari	☐	☐	☐	
UX browser support: Mozilla Firefox	☐	☐	☐	
CORE SECURITY CAPABILITIES				
Attack surface reduction (ASR): Prevention of zero-day and unpatched N-day browser exploits	☐	☐	☐	
Identity Threat Detection and Response (ITDR): Real-time detection and input blocking for known and unknown phishing sites	☐	☐	☐	
ITDR: Session cookie encryption to prevent MFA bypass	☐	☐	☐	
ITDR: Password reuse notification	☐	☐	☐	
ITDR: Compromised credential monitoring and alerting	☐	☐	☐	
DLP: End-user actions (copy, paste, etc.) and interaction with sensitive data (e.g., PII)	☐	☐	☐	

CATEGORY/REQUIREMENT	MANDATORY	OPTIONAL	VENDOR-SUPPORTED	ADDITIONAL NOTES
DLP: Data scanning/masking — pre-defined patterns for sensitive data (e.g., PII)	☐	☐	☐	
DLP: Data scanning/masking — customer-defined patterns for sensitive data	☐	☐	☐	
DLP: Data scanning/masking — OCR to detect sensitive data in images	☐	☐	☐	
GOVERNANCE				
Browser configuration: Unified control of settings for browsers from different vendors	☐	☐	☐	
Content filtering: Browser-local AUP enforcement and objectionable/unwanted content blocking	☐	☐	☐	
Extension management: Inventory, reporting, risk scoring, and policy-based disabling/removal of extensions	☐	☐	☐	
ADDITIONAL CAPABILITIES				
On-premises web app access: Clientless remote access without internet-facing infrastructure or third-party network peering	☐	☐	☐	
SaaS app access control: Supplemental authorization attributes (e.g., device ownership, posture, etc.)	☐	☐	☐	
SaaS app access control: Access policy enforcement without traffic steering (proxy) clients and/or vanity domains	☐	☐	☐	

Example Architecture/ Deployment Diagram

Corporate-owned/Managed Device Deployment

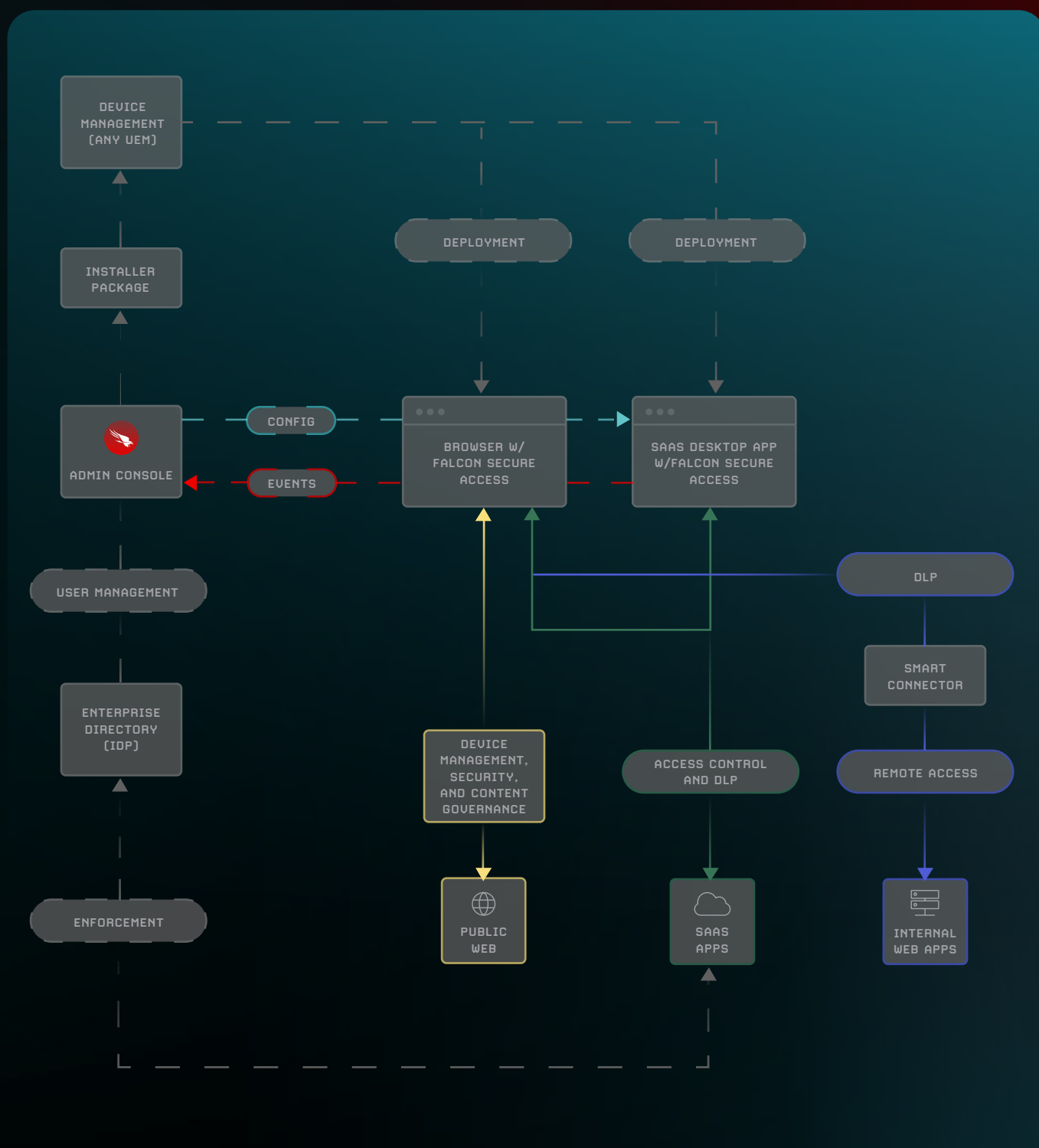


Figure 2. Architecture of deploying and managing Falcon Secure Access on corporate-owned devices across browsers and SaaS desktop apps

Employee-owned (BYO)/Unmanaged Device Deployment

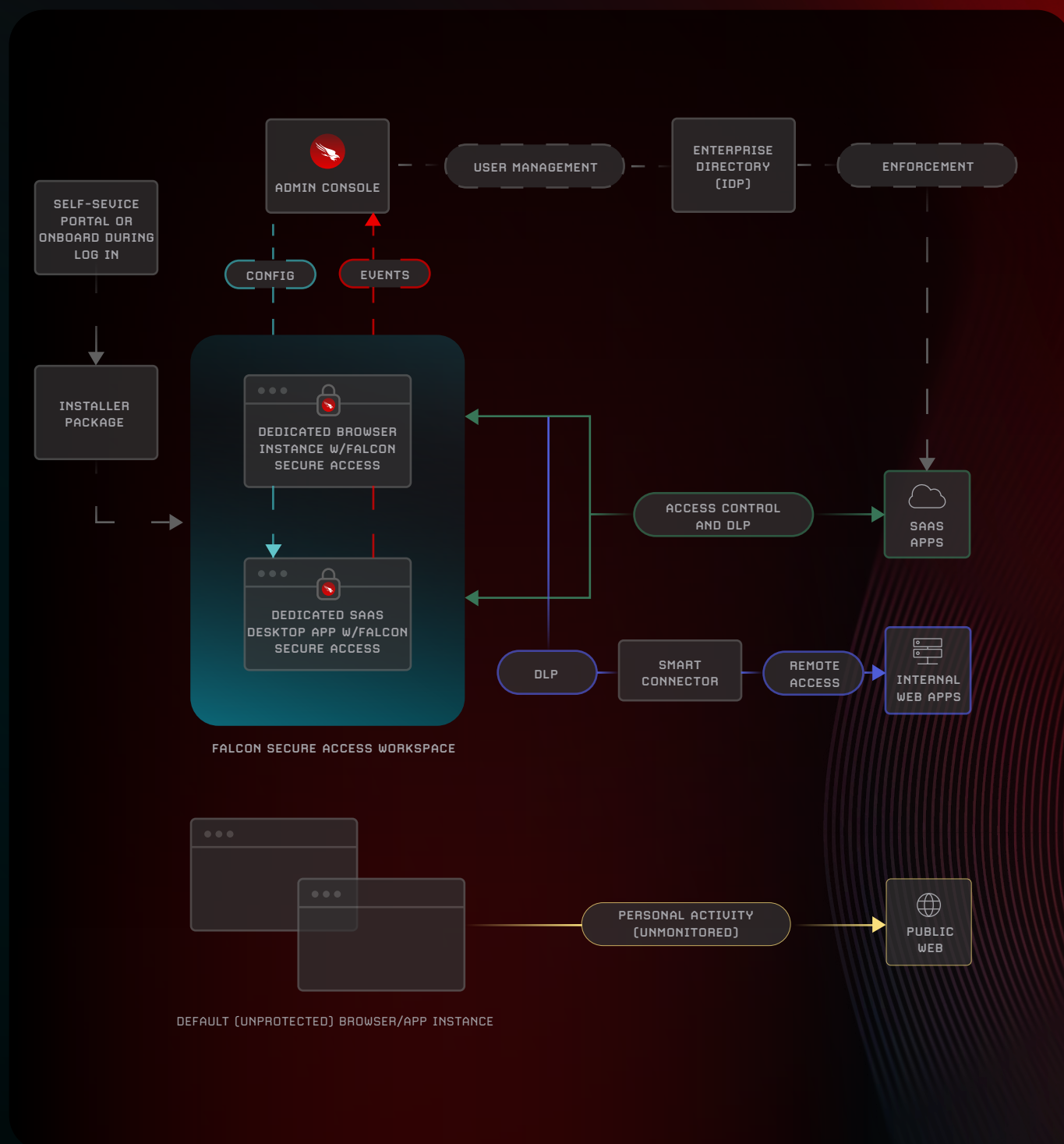


Figure 3. Architecture of deploying and managing Falcon Secure Access on unmanaged devices across browsers and SaaS desktop apps

[Learn more](#) →

About CrowdStrike

[CrowdStrike](#) (Nasdaq: CRWD), a global cybersecurity leader, has redefined modern security with the world's most advanced cloud-native platform for protecting critical areas of enterprise risk — endpoints and cloud workloads, identity and data.

Powered by the CrowdStrike Security Cloud and world-class AI, the CrowdStrike Falcon® platform leverages real-time indicators of attack, threat intelligence, evolving adversary tradecraft and enriched telemetry from across the enterprise to deliver hyper-accurate detections, automated protection and remediation, elite threat hunting and prioritized observability of vulnerabilities.

Purpose-built in the cloud with a single lightweight-agent architecture, the Falcon platform delivers rapid and scalable deployment, superior protection and performance, reduced complexity and immediate time-to-value.

CrowdStrike: We stop breaches.

Learn more: www.crowdstrike.com

Follow us: [Blog](#) | [X](#) | [LinkedIn](#) | [Facebook](#) | [Instagram](#) | [YouTube](#)

Start a free trial today: www.crowdstrike.com/free-trial-guide

© 2026 CrowdStrike, Inc. All rights reserved.