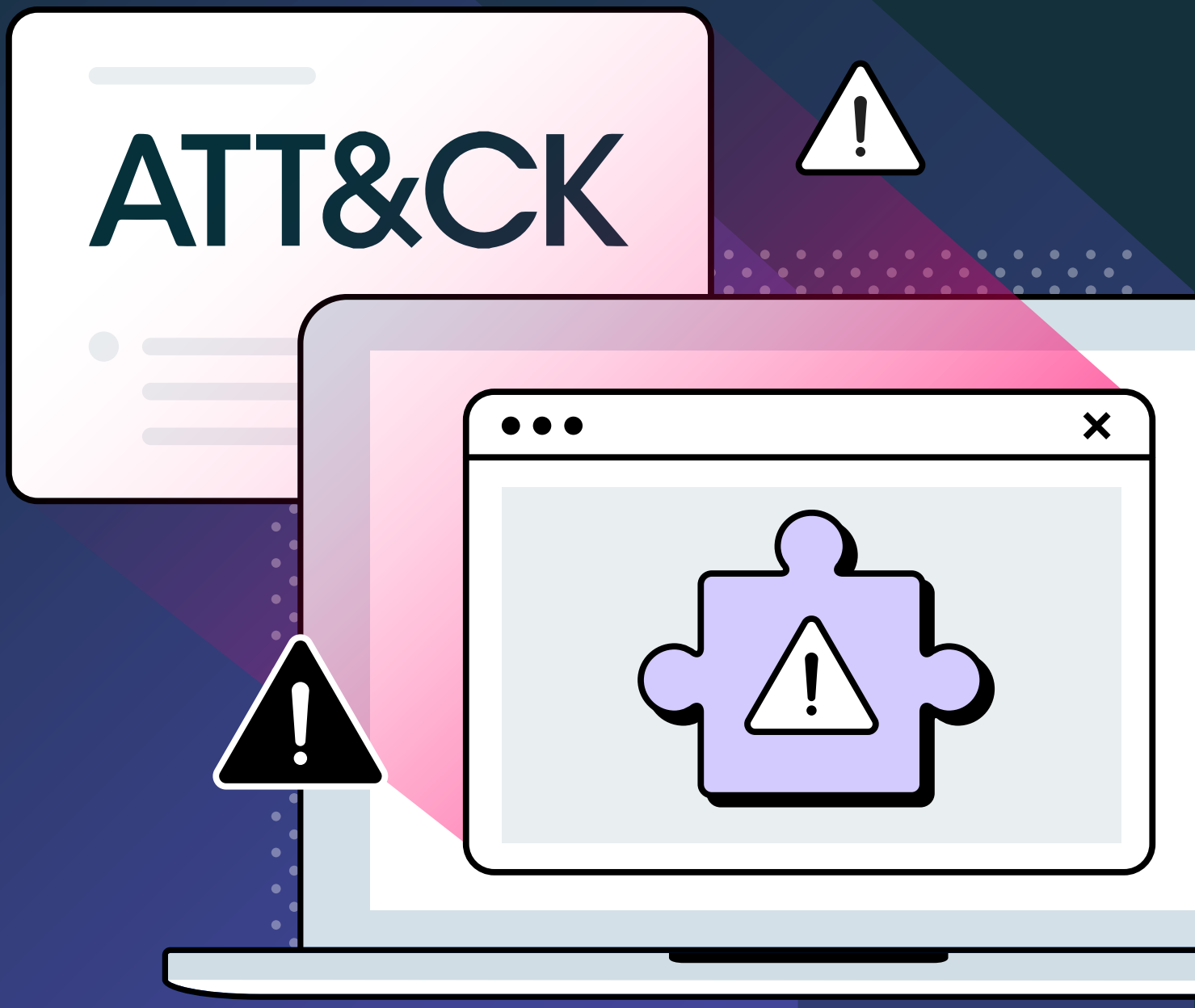
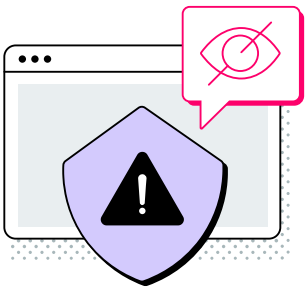
LayerX

Mapping Browser Extension Risks to the MITRE ATT&CK Framework

Practical Guidance on Applying the MITRE ATT&CK Framework to Identity and Data Risks by Malicious Browser Extensions





The Overlooked Browsing Security Risk

Browser extensions have become a ubiquitous part of the browsing experience, and many users often use such extensions to fix their spelling, find discount coupons, pin notes, and other productivity uses. However, most users don't realize that browser extensions are routinely granted extensive access permissions that can lead to severe data exposure should those permissions fall into the wrong hands.

Common access permissions requested by extensions include access to sensitive user data such as cookies, identities, browsing data, text input, and more, which can lead to data exposure on the local endpoint and credential theft of user identities.

This is particularly a risk to organizations since many organizations do not control what browser extensions users install on their endpoints, and credential theft of a corporate account can lead to exposure and data breach at the organizational level.

Mapping Permissions to Data and Identity Risks

Browser extensions' permissions are governed by the APIs provided by the browser providers such as Google, Microsoft, or Mozilla.

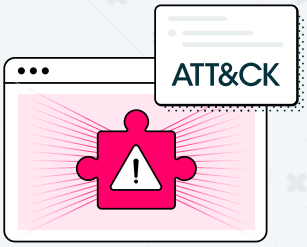
These APIs are publicly available, and extensions authors can use them for the functionality provided by the extension.

Key permissions that extensions can access through such APIs include:

- **Cookies:**
  access to read/write/modify the user's cookies, which can be used for website authentication. It appears that in this incident, cookies were the primary objective of the compromised browser extensions
- **Identities:**
  access to the user's identity and profile
- **Browsing history:**
  view the user's browsing history and see where they've been
- **Browsing data:**
  see the URL the user is browsing to and see all browsing meta-data
- **Passwords:**
  view plaintext passwords as they are being submitted to websites as part of web requests, before the web session encrypts them
- **Web page content:**
  visibility into all web page data, across all open tabs, so it can potentially copy data from internal system otherwise note accessible online
- **Text input:**
  track every keystroke on a web page, just like a keylogger
- **Audio/video capture:**
  access the computer's microphone and/or camera

While most browser extensions don't have access to all of these permissions, many extensions do have access to some (or many) of these permissions.

The problem is that should these permissions become compromised, either through the installation of a malicious browser extension, or the compromise legitimate one, their exploitation can lead to credential theft and/or data exposure.



Applying the MITRE ATT&CK Framework to Browser Extension Risks

The MITRE ATT&CK Framework is a globally recognized resource designed to enhance cybersecurity by providing a comprehensive knowledge base of adversary tactics, techniques, and procedures (TTPs). It enables security professionals to better understand, detect, and mitigate threats by mapping real-world attacker behaviors to structured models.

By leveraging MITRE ATT&CK, organizations can identify gaps in their defenses, enhance threat intelligence capabilities, and align their security strategies to anticipate and counter evolving threats.

Below is a list of MITRE ATT&CK Techniques relevant to attacks by malicious browser extensions, and how they are exploited:

MITRE ATT&CK ID	Technique Name	Technique Description	How Browser Extensions Can Exploit This Technique
T1176	**Browser Extensions**	Adversaries target or abuse browser extensions to manipulate user interactions, steal sensitive data, or perform malicious activities.	Attackers can target browser extensions through a number of means: • Create their own malicious browser extensions • Impersonate legitimate browser extensions with fake, malicious versions • Purchase existing legitimate browser extensions and add malicious code to them • Compromise legitimate browser extensions and inject malicious code into them The rows below further explain specific techniques that can be used by individual extension permissions or APIs to compromise user data.

MITRE ATT&CK ID	Technique Name	Technique Description	How Browser Extensions Can Exploit This Technique
T1539	**Steal Web Session Cookie**	Adversaries steal session cookies to hijack an active user session, bypassing authentication mechanisms and gaining unauthorized access to web applications.	The `cookies` API allows extensions to read, modify, and delete cookies. A malicious extension could steal or delete session cookies, effectively hijacking or terminating the user's session, and/or create fake cookies to impersonate legitimate users.
T1185	**Browser Session Hijacking**	Adversaries use malicious software to intercept and manipulate data within a web browser, enabling activities such as stealing credentials or altering transactions in real time.	The `webRequest` API allows extensions to observe and intercept network requests. Malicious extensions could intercept session cookies or modify request headers to impersonate users or disrupt active sessions. The `cookies` API allows extensions to read, modify, and delete cookies. A malicious extension could steal or delete session cookies, effectively hijacking or terminating the user's session, and/or create fake cookies to impersonate legitimate users. The `tabs` API allows extensions to manage browser tabs, including creating, updating, and removing them. It could be used to force the user to navigate to malicious websites or phishing pages, or close legitimate session-related tabs to disrupt the user's workflow or session continuity. Content scripts could be used to scrape session tokens stored in the DOM (e.g., in cookies, localStorage, or sessionStorage), and could inject malicious scripts to modify or monitor user input and behavior within active sessions.

MITRE ATT&CK ID	Technique Name	Technique Description	How Browser Extensions Can Exploit This Technique
T1528	**Steal Application Access Token**	Adversaries compromise application tokens (e.g., OAuth tokens) to gain unauthorized access to cloud services or APIs without needing user credentials.	The `webRequest` API enables extensions to monitor and intercept network requests and responses, including headers and query parameters. Malicious extensions could intercept access tokens if tokens are included in URLs, headers, or payloads of HTTP requests. The `declarativeNetRequest` API allows extensions to define declarative rules to modify or block network requests. This could be misconfigured to capture tokens in request headers or URLs, especially during OAuth workflows. The `cookies` API allows extensions to read, modify, and delete cookies, and can access cookies that store session or authentication tokens. Finally, content scripts that run in the context of web pages interact with the DOM could be used to scrape tokens stored in cookies, localStorage, or as hidden form fields on web pages.

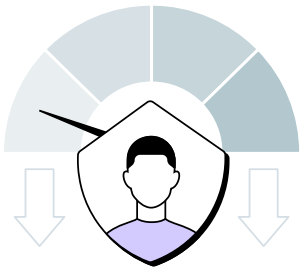
MITRE ATT&CK ID	Technique Name	Technique Description	How Browser Extensions Can Exploit This Technique
T1649	**Steal or Forge Authentication Certificates**	Adversaries exploit Kerberos authentication by stealing or forging tickets (e.g., Golden Tickets) to gain persistent access to resources in Active Directory environments.	The `webRequest` API could be used to intercept HTTPS requests to gather sensitive data such as certificate information if users are tricked into using insecure connections (e.g., MITM attacks). It could also tamper with headers to inject malicious certificates into communication. The `cookies` API could be used to steal cookies that store certificate-related data or tokens used for client-side authentication. Content scripts can be used to extract certificate-related information displayed on web pages or stored in web application data (e.g., localStorage, sessionStorage). This could be used to inject scripts into pages to manipulate authentication workflows or forge certificate-like data. The `scripting` API allows injection of JavaScript into web pages. Injected scripts could be used to forge or manipulate certificate-like data used in web applications (e.g., spoofing client-side validation of certificates). The `declarativeNetRequest` API enables extensions to define declarative rules to block, redirect, or modify network requests. Malicious rules could block valid certificate requests or redirect users to phishing sites that use fake certificates. The `enterprise.platformKeys` API allows enterprise extensions to use client certificates for authentication. In an enterprise context, a compromised extension with access to this API could misuse stored certificates for unauthorized authentication.

MITRE ATT&CK ID	Technique Name	Technique Description	How Browser Extensions Can Exploit This Technique
T1555	**Credentials from Password Stores**	Adversaries extract credentials stored in password managers or credential stores on a system, such as browser password vaults or system keychains.	The `webRequest` allows extensions to observe and intercept network requests and responses. If credentials are transmitted over insecure HTTP (rather than HTTPS), malicious extensions could intercept sensitive information in request headers, URL parameters, or form submissions. This API be exploited to track session cookies or hijack login sessions if they are not protected by secure mechanisms. The `cookies` API allows extensions to read, modify, and delete cookies. It could be used to steal session cookies, enabling unauthorized access to user accounts. The `scripting` API can be used to capture login forms, keystrokes, or scrape credentials from fields on the page, or to manipulate web pages to exfiltrate stored credentials, such as interacting with password fields or session tokens. The `tabs` API can be used to track forms on active pages to capture login details. Content scripts could be used to scrape username and password fields, or capture login credentials entered by the user, or used to extract credentials stored in the DOM or within JavaScript variables.
T1115	**Clipboard Data**	Adversaries capture sensitive information, such as passwords or tokens, by monitoring or manipulating clipboard data copied by the user.	Browser extensions can read and write to the clipboard using `clipboardRead` and `clipboardWrite` APIs, respectively.

MITRE ATT&CK ID	Technique Name	Technique Description	How Browser Extensions Can Exploit This Technique
T1217	**Browser Information Discovery**	Adversaries access data saved by the browser (such as bookmarks, browser history, or accounts) to gather intelligence about frequently visited sites, potential targets, or sensitive resources.	The `identity` API provides access to user account information when interacting with Google services, including OAuth authentication and access to user profile data. The `history` API allows extensions to interact with the browser's history, including retrieving, searching, and deleting visited URLs. The `bookmarks` API allows extensions to view, create, organize, and manage bookmarks in the user's browser. These APIs can be used to track user browsing patterns, discover internal web resources, and identify potential targets for data theft.
T1056	**Input Capture**	Adversaries log user inputs such as keystrokes or mouse movements to steal credentials, gain access to sensitive systems, or monitor user behavior.	The `input.me` API allows extensions to create custom Input Method Editors (IMEs) for text input in Chrome OS. This enables recording and manipulating user input through the IME interface. In addition, content scripts can interact with web pages to capture user input by attaching event listeners (e.g., `keydown`, `keyup`, `input`). This enables monitoring of user interactions like keystrokes, form inputs, and clicks. Finally, the `webNavigation` and `webRequest` APIs an observe and potentially modify web traffic, including form submissions (though not user keystrokes directly).

MITRE ATT&CK ID	Technique Name	Technique Description	How Browser Extensions Can Exploit This Technique
T1113	**Screen Capture**	Adversaries take screenshots of the user's desktop or application windows to gather sensitive information displayed on the screen.	The `tabCapture` API allows extensions to capture the visible content of a browser tab, including video and audio. This enables screen recording of a specific active tab. The `desktopCapture` API allows extensions to capture the screen, an application window, or a browser tab. It is primarily designed for scenarios like video conferencing or screen sharing. The `tab.captureVisibleTab` API can capture a snapshot of the currently visible content in a tab as an image, but cannot record video or capture tabs in the background. In addition, content scripts can be used to capture the user's screen by using JavaScript APIs like HTMLCanvasElement to capture screenshots of specific web elements or pages.
T1123	**Audio Capture**	Adversaries record audio from a device's microphone to capture sensitive conversations or environmental sound data.	The `getUserMedia` API provides access to the user's microphone through the MediaDevices.getUserMedia() Web API. It allows extensions to capture and record audio. The `tabCapture` API enables capturing audio (and video) from a specific browser tab, although user interaction is required to start capturing, and this API is limited to capturing audio from the browser tab, not from the user's microphone. The `desktopCapture` API allows extensions to capture audio from the desktop, including system audio or application-specific audio, but extensions cannot capture microphone audio unless explicitly selected by the user.

MITRE ATT&CK ID	Technique Name	Technique Description	How Browser Extensions Can Exploit This Technique
T1125	**Video Capture**	Adversaries leverage a device's webcam to record video, capturing sensitive visuals or identifying individuals in the environment.	The `getUserMedia` API provides access to the user's camera through the MediaDevices.getUserMedia() Web API. It allows extensions to extensions to record video directly from the camera. However, it requires active user consent and cannot be used passively or in background tabs. The `desktopCapture` API allows extensions to capture video from the desktop, application windows, or browser tabs. It can include video streams if a user explicitly selects the window displaying the camera feed. However, it does not provide direct access to the camera feed itself. The `tabCapture` API enables capturing video (and audio) from a browser tab. While it cannot access the camera directly, it can record content displayed in the browser, including video streams from the user's camera embedded in a tab. It cannot capture the camera feed unless it is displayed in the tab being recorded.



A Strategic Framework for CISOs to Mitigate Browser Extension Risk

While many users and organizations are not aware of the potential risks associated with browser extensions, there are a number of key actions they can take to protect themselves:

1

Audit all extensions

Many organizations don't have a full picture of all extensions that are installed in their environment. Many organizations allow their users to use whichever browsers (or browsers) they wish to use and install whatever extensions they want. However, without a full picture of all extensions on all browsers of all users, it is impossible to understand your organization's threat surface. This is why a full audit of all browser extensions is a foundational requirement for protecting against malicious extensions.

2

Categorize extensions

Some categories of browser extensions seem to be more susceptible to exploitation than others. Part of this is the popularity of certain types of extensions that makes them appealing to attack because of their broad user base (such as various productivity extensions), and part of it is because of the permissions granted to such extensions, that hackers may wish to exploit (such as access to network and browsing data given to VPN extensions, for example). This is why categorizing extensions is a useful practice in assessing the browser extension security posture.

3

Enumerate extension permissions

While understanding which extensions are installed in corporate environments is one side of the coin, the other side of the coin is understanding what those extensions can do. This is done by enumerating their precise access permissions and listing all the information they can potentially access.

4

Assess extension risk

Once they understand what permissions they have installed on corporate endpoints and the information that these extensions can touch (via their permissions), organizations need to assess the risk posed by each individual extension. A holistic risk assessment should encompass both the permission scope of the extension (i.e., what it can do), as well as external parameters such as its reputation, popularity, publisher, installation method, and more (i.e., how much we trust it). These parameters should be combined into a unified risk score to help organizations assess the risk posed by each extension, and whether it is safe for that extension to be installed.

5

Apply adaptive, risk-based enforcement

Finally, taking into consideration all the information they have at hand, organizations should apply adaptive, risk-based enforcement policies tailored to their uses, needs and risk profile. They can define policies to block extensions that have certain permissions (e.g., access to cookies), or define more complex rules tailored to their specific use case (e.g., block AI and VPN extensions with a 'High' risk score).

While browser extensions offer many productivity benefits, they also expand organizations' threat surface and their risk of exposure. Recent attack campaigns targeting browser extensions with malicious code should be a wakeup call for organizations to define how they protect against malicious and compromised browser extensions.

About LayerX
One Browser Extension to Rule Them All

Comprehensive Audit

Discover all extensions on all browsers for all users, with full visibility and control

Rich Risk Classification

Assess the risk profile of each extension using internal and external risk factors

Adaptive Enforcement

Go beyond manual blocklists to automatically disable or block extensions based on their risk

0% User Friction

Easy deployment with no impact on the user browsing experience or existing workflows

LayerX browser security platform provides full protection against malicious browser extensions. LayerX's secure browser extension can integrate with any browser, and as such has full visibility into all other installed extensions.

LayerX's extension continuously monitors the existing and newly installed extensions, evaluating permissions, installation method, web store parameters, and external risk parameters.

LayerX identifies risky browser extensions using a comprehensive risk-scoring approach that combines internal and external risk factors. LayerX examines the access permissions requested by each extension and whether it has access to sensitive information such as passwords, cookies, user input, and more. At the same time, LayerX analyzes the extension's reputation based on external factors such as user rating, number of downloads, age, and more. These parameters are combined to create a unified score reflecting each extension's risk.

With its granular policy engine, LayerX enables its users to trigger notifications, alerts, or even complete disablement of an extension, when any risk indicator or combination of these are detected. LayerX extension runs at a higher permission level than ordinary extensions, and cannot be tampered with or uninstalled by users.

To learn more about how LayerX can help you manage and secure your browser extensions, go to www.layerxsecurity.com and schedule a demo today!