chrome enterprise

Permission Risk whitepaper

Understand the risks of permissions for Chrome extensions

Enterprise & Education

chrome enterprise

Table of contents

Purpose of this guide

This document is for Windows IT administrators managing the Chrome Browser on Windows, Mac, or Linux computers. There are trade-offs administrators need to consider when deciding which Chrome extensions to allow and block in their enterprise.

This guide is meant for security-conscious administrators evaluating or deploying the Chrome Browser to their organization. This document lists the permissions different extensions require to run, and what to watch out for. It's meant as a companion guide to the longer and more-comprehensive Managing Extensions in Your Enterprise guide. If you have questions, comments or concerns, please contact your Chrome Enterprise Browser Specialist.

What's covered	How to evaluate the security risk of the different types of permissions that Chrome extensions require to run.
Primary audience	IT administrators and Chrome Browser administrators
IT environment	Windows, Mac, or Linux
Takeaways	Know how to evaluate the risks that extension permissions pose and the next steps you should next take in managing extensions in your enterprise.

Last updated: *July 25, 2019 for Chrome 75*

Introduction

Chrome Enterprise provides many different options of managing Chrome Browser for an administrator. With more than 300 policies and deployment packages, there are many options for managing Chrome within an enterprise. There's a different part of Chrome that needs further visibility: Chrome extensions. With thousands of third-party extensions that can be installed to customize or augment the browser experience, it's important to understand extensions and what they can access.

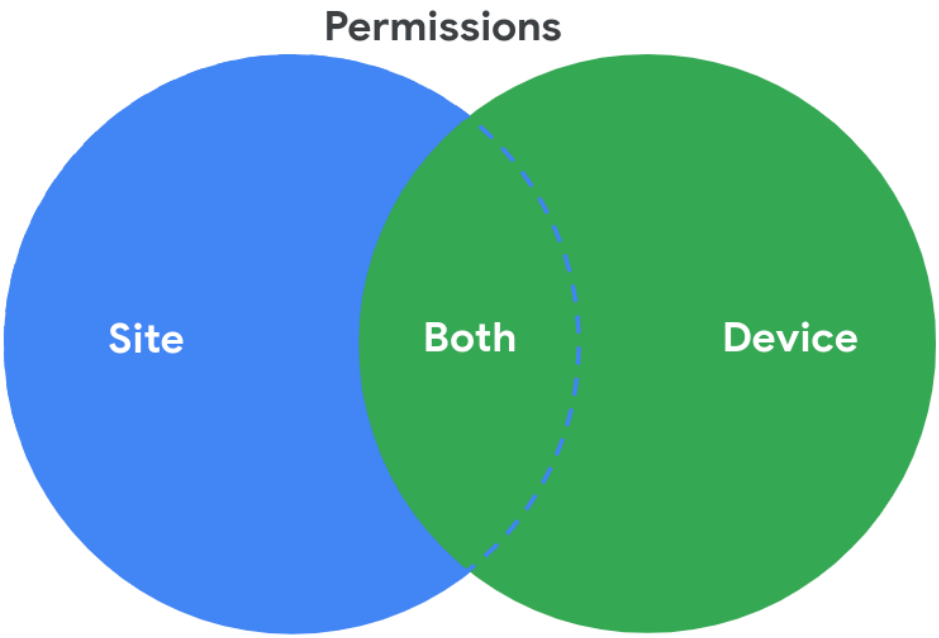
This whitepaper discusses:

- The rights and permissions an extension requires to run
- How these permissions are declared in the development process
- Common permissions and the risk they pose to enterprises

What are permissions?

Some extensions can require rights to make changes on a machine or a web page to run properly. These rights are called permissions. There are two main categories, but many extensions have both:

- **Site permissions** require the extension to list sites it may view or modify.
    - Examples: An extension wants permission to modify a webpage, access cookies, or modify tabs.
- **Device permissions** are the rights needed by an extension on the machine where it's running.
    - Examples: An extension wants to access the USB port / storage / viewing screen, or talk to native programs on the computer.



How are permissions declared?

To limit the risk to users, extensions declare which sites they're going to interact with as well as what risky actions they may take. Developers list most of the rights their extensions require in the extension's manifest file. Some can be present elsewhere but most are present in the permission section. Here's an example of a manifest file:

```
},
"permissions": [
    "activeTab",
    "tabs",
    "storage",
    "chrome://favicon/",
    "webRequest",
    "webRequestBlocking",
    "http://*/*",
    "https://*/*",
    "background",
    "topSites",
    "https://www.googleapis.com/*"
],
```

Some extensions are required to notify the user during installation through the Permissions Warnings, and some do not. For more information about each permission and the Chrome API that they can get access to, see Declare Permissions.

Compare common permissions and their risk levels

The process for vetting extensions based on their permissions can feel overwhelming. To help you get started, here is a list of some common permissions broken down by the level of risk. Note that this is not an exhaustive list. Each enterprise will have different considerations on what is deemed a security risk and what is not. Also keep in mind that each extension will have its own requirements for more (or less) rights within a required permission. So when vetting a core extension needed by users that has a permission that you deem a security risk, consider speaking with your vendor. They should be able to provide more visibility into what rights they are actually using.

Understand extension permission risk

These permissions require rights on devices or sites that could pose a security risk. Note that risk is subjective to each enterprise or IT admin. The information in the tables below is a starting point, and it is up to you to evaluate this information according to the needs and security protocols of your organization.

Highest risk permission

Risk	Permission	Description
Highest	host permissions	Host permissions allow developers to get access based on a specified list of hosts. Host permissions can be any match pattern [1] - ":///*" means "everything", "://.google.com/*" means "all google sites", etc. The risk level of these permissions is dependent on the site it's requesting access to. If it's for your corporate site, that poses a higher risk than an extension requesting access to another site such as example.com, which has no user-specific data.

High risk permissions

Risk	Permission	Description
High	<all_urls>	The extension wants to interact with the code running on pages which matches any URL that starts with a permitted scheme (http:, https:, file:, ftp:, or chrome-extension). Can be a broad permission with possible risks.
High	app.window.fullscreen.overrideEsc	Can prevent escape button from exiting fullscreen
High	audioCapture	Capture audio from attached mic or webcam. Could be used to listen in on user
High	browsingData	Clears browsing data which could result in a forensics/logging issues
High	content_security_policy	CSP works as a block/allow listing mechanism for resources loaded or executed by your extensions. Can manipulate default CSP.
High	contentSettings	Could allow plugins to run unsandboxed
High	copresence	P2P communication
High	debugger	Provides high level super user access
High	declarativeNetRequest	The chrome.declarativeNetRequest API is used to block or redirect network requests by specifying declarative rules
High	declarativeWebRequest	chrome.declarativeWebRequest API to intercept, block, or modify requests in-flight. It is significantly faster than the chrome.webRequest API because you can register rules that are evaluated in the browser rather than the JavaScript engine, which reduces roundtrip latencies and allows higher efficiency.
High	downloads	Use the chrome.downloads API to programmatically initiate, monitor, manipulate, and search for downloads. Can be used to download scripts.
High	downloads.open	Used in conjunction with downloads. Allows an extension to open a downloaded file; as above, can be used to run a downloaded script.
High	experimental	Access to any of the experimental APIs
High	hid	Access USB devices (can act as driver)
High	history	Full history (read, add, delete)
High	nativeMessaging	Allows native (Win, Linux, Mac) programs that register with chrome to talk with extensions
High	pageCapture	Gets MHTML (archived Web page) of any page

High	privacy	Can turn off malware protections (e.g. chrome.privacy.services.safeBrowsingEnabled)
High	proxy	Manage Chrome's proxy settings. Can be used to send user's internet traffic.
High	socket	Raw TCP/UDP connection, listen on a port, etc.
High	`*://*/*`	Extension wants to interact with the code running on pages which matches any URL that uses the https: or http: scheme. Can be a broad permission with possible risks.
High	tabCapture	Get picture/video of user's current tab
High	tabs	Can access current URLs and favicons
High	unsafe-eval	Can be used to fetch and execute remote script
High	usb	Interact with any USB device (Raw Format)
High	usbDevices	Used in conjunction with USB permission to specify the types of USB devices the extension wants access to
High	videoCapture	Extension can use webcam, possibly screen contents depending on settings
High	vpnProvider	Extension. can create a VPN tunnel and send/receive packets through it, across sessions
High	web_accessible_resources	This is an array of strings specifying the paths of packaged resources that are expected to be usable in the context of a web page. This can be used to execute remote scripts.
High	webNavigation	Listen to the websites that a user visits

Medium risk permissions

These permissions require rights on devices or sites that could propose a medium security risk. Note that risk is subjective to each enterprise or admin.

Risk	Permission	Description
Medium	activeTab	Gives an extension temporary access to the currently active tab when the user invokes the extension as when clicking its browser action. For example, with user interaction, the extension can inject JavaScript. Access to the tab lasts while the user is on that page, and is revoked when the user navigates away or closes the tab.
Medium	bookmarks	Use the chrome.bookmarks API to create, organize, and otherwise manipulate bookmarks
Medium	clipboardRead	Read the machine's clipboard
Medium	clipboardWrite	Writes to the machine's clipboard
Medium	contextMenus	Integrate menus for extension throughout Chrome browser
Medium	cookies	Lookup or write cookies for any domain specified in permissions (requires host permissions to do this)
Medium	desktopCapture	Gets screenshot of entire desktop. A private profile extension with this permission could access machine's data.
Medium	downloads	Extension could: ● Download file(s) from extension's specified URL ● Get/modify download history ● Remove downloaded files ● Bypass dangerous download

Medium	fileSystem	Read files in folder that user has selected (user interaction required)
Medium	fileSystem.directory	Read any file in the directory that a user selects
Medium	fileSystem.retainEntries	Retain authorization to previously user selected files
Medium	fileSystem.write	Extension can write files in directory that the user specifies
Medium	fileSystem.writeDirectory	Write any file in the user specified directory (user interaction required to select file directory)
Medium	geolocation	Get user's physical location without prompting the user
Medium	identity	Extension request oAuth with users credentials - only shows the oauth prompt but does not automatically grant access
Medium	identity.email	Extension can get email address of signed in profile (must be used in conjunction with identity)
Medium	management	Use the chrome.management API to manage the list of extensions/apps that are installed and running. For example, the extension could: ● Uninstall itself ● Disable other extensions ● Check extensions' state
Medium	processes	Use the chrome.processes API to interact with the browser's processes
Medium	sessions	Use the chrome.sessions API to query and restore tabs and windows from a browsing session. Can keep on re-opening an ad or a bad website.
Medium	syncFileSystem	Save data to user's drive
Medium	system.storage	List storage devices currently connected and/or eject any connected storage device
Medium	topSites	Get and/or set list of top visited sites
Medium	tts	Text to speech accessibility API
Medium	webRequest	View any GET/POST request & URL (requires host permissions to do this)
Medium	webRequestBlocking	Used with webRequest. Allows blocking of a GET/POST request (requires host permissions to do this).

Low risk permissions

These permissions require rights on devices or sites that could propose a low or no security risk. Note that risk is subjective to each enterprise or admin.

Risk	**Permission**	**Description**
Low	accessibilityFeatures.modify	Extension can turn on or off accessibility APIs
Low	accessibilityFeatures.read	Extension can view current state of accessibility features
Low	alarms	Extension can set a timer to call its own functions
Low	alwaysOnTopWindows	Window stays on top
Low	app.window.alpha	Alpha transparency of application window
Low	app.window.alwaysOnTop	Application displays over other windows/applications
Low	app.window.fullscreen	App can be fullscreen without user interaction
Low	app.window.shape	Specify how App window looks
Low	background	Extension can run when chrome is running

Low	certificateProvider	Use this API to expose certificates to the platform which can use these certificates for TLS authentication(s)
Low	declarativeContent	Enables actions depending on the content of a page, without requiring permission to read the page's content
Low	documentScan	Use the chrome.documentScan API to discover and retrieve images from attached paper document scanners
Low	enterprise.deviceAttributes	Use the chrome.enterprise.deviceAttributes API to read device attributes. Note: This API is only available to extensions force-installed by enterprise policy.
Low	enterprise.hardwarePlatform	Use the chrome.enterprise.hardwarePlatform API to get the manufacturer and model of the hardware platform where the browser runs
Low	enterprise.platformKeys	Use the chrome.enterprise.platformKeys API to generate hardware-backed keys and to install certificates for these keys. The certificates will be managed by the platform and can be used for TLS authentication, network access or by other extension through chrome.platformKeys.
Low	externally_connectable	The externally_connectable manifest property declares which extensions, apps, and web pages can connect to your extension via runtime.connect and runtime.sendMessage
Low	fileBrowserHandler	Use the chrome.fileBrowserHandler API to extend the Chrome OS file browser. For example, you can use this API to enable users to upload files to your website.
Low	fileSystemProvider	Use the chrome.fileSystemProvider API to create file systems, that can be accessible from the file manager on Chrome OS.
Low	fontSettings	Use the chrome.fontSettings API to manage Chrome's font setting
Low	gcm	Extension can send or receive messages via Google Cloud Messaging service
Low	homepage_url	The URL of the homepage for this extension. The extension management page (chrome://extensions) will contain a link to this URL. This field is particularly useful if you host the extension on your own site. If you distribute your extension using the Chrome Web Store, the homepage URL defaults to the extension's own page.
Low	idle	Determine if the system is being actively used, locked, or just not being used
Low	mediaGalleries	Access media files on local system (requires consent from user)
Low	networking.config	Use the networking.config API to authenticate to captive portals
Low	notifications	Popup notifications to users on desktop
Low	overrideEscFullscreen	Prevent exiting full screen with ESC
Low	platformKeys	Use the chrome.platformKeys API to access client certificates managed by the platform. If the user or policy grants the permission, an extension can use such a certificate in its custom authentication protocol. E.g. this allows usage of platform managed certificates in third party VPNs (see chrome.vpnProvider).
Low	power	Manage Chrome OS power state (prevent sleep)
Low	printerProvider	The chrome.printerProvider API exposes events used by print manager to query printers controlled by extensions, to query their capabilities and to submit print jobs to these printers
Low	signedInDevices	Use the chrome.signedInDevices API to get a list of devices signed into chrome with the same account as the current profile
Low	storage	Store data on computer
Low	system.memory	Get physical memory information
Low	system.cpu	Information about processor capabilities
Low	system.display	Get info about monitor or change users display settings
Low	ttsEngine	Read all text spoken using synthesized speech

Low	unlimitedStorage	Removes HTML5 storage size limits
Low	wallpaper	Use the chrome.wallpaper API to change the Chrome OS wallpaper.
Low	webview	Sandboxed iframe in your extension

Next steps

Once you have gone through and determined which permissions are and aren't a risk to your enterprise, you can start managing extensions by permissions. We recommend this method (instead of allowing/blocking extensions) because it scales for large organizations and is easier to manage.

You can control what extensions your users can install by the permissions themselves. If an installed extension needs a permission that is blocked, it won't run or it will be blocked from installation. For more information, see the Managing Extensions in Your Enterprise guide.

- This guide covers the best practices for managing Chrome extensions.
- It provides steps for managing extensions using the Google Admin console, Windows Registry, and Windows Group Policies.

Additional resources

Here are more resources to help you with managing Chrome extensions in your organization:

- Managing Extensions in Your Enterprise
- App and extension policies
- Manage Chrome Browser extensions in the Admin console
- Allow or block apps and extensions
- Chrome Policy list