USERAUTHGUARD

BLOG PRODUCT

How to Block AI on Student Chromebooks: The API-Level Approach That Actually Works

Stef Verleysen | January 20, 2026

> Tab-based filters like Securely and GoGuardian can't stop AI access on Chromebooks. Here's how to block it at the API level for complete protection, plus a free extension that does it automatically.

AI tools like ChatGPT, Claude, and Gemini have become the new frontier of academic dishonesty in K-12 schools. Students use them to generate essays, solve homework problems, and bypass critical thinking entirely. The challenge for IT directors is that traditional content filters were not designed to block AI access effectively.

Most schools rely on **tab-based content filters** like Securely and GoGuardian to manage student internet access. These tools monitor browser tabs and attempt to block specific URLs. But AI services are not static websites, they are dynamic applications with multiple access points, subdomains, and API endpoints. A student who knows how to open a new tab or use a different subdomain can bypass tab-based blocking in seconds.

This guide explains why tab-based AI blocking fails, how **API-level blocking** works, and how to implement it in your district using a free Chrome extension that requires zero configuration.

TLDR:

- Tab-based filters (Securely, GoGuardian) only monitor visible browser tabs and miss AI API calls, subdomains, and embedded access points
- API-level blocking uses Chrome's declarativeNetRequest to block 83 AI sites (ChatGPT, Claude, Gemini, Quillbot, Chegg, etc.) at the network layer before requests leave the device
- UserAuthGuard Free extension implements this approach with zero data collection, works offline, and requires no configuration, just force-install via Google Admin

- Includes technical implementation details for IT directors who want to understand or customize the blocking rules

Why Tab-Based AI Blocking Fails

Content filters like Securely and GoGuardian were built to monitor and control what students see in their browser tabs. They work by injecting JavaScript into web pages, analyzing visible content, and blocking or flagging inappropriate material. This approach works well for traditional websites with predictable URL structures.

AI services break this model in several ways:

1. Multiple subdomains and access points
ChatGPT alone can be accessed via `chat.openai.com`, `chatgpt.com`, and `openai.com`. Block one, and students simply use another. Claude operates on `claude.ai` and `anthropic.com`. Gemini uses `gemini.google.com` and `bard.google.com`. Tab-based filters must maintain an ever-growing list of domains, and they are always one step behind.

2. API calls bypass tab monitoring
AI tools do not require a visible browser tab to function. A student can embed an AI chatbot in a Google Doc sidebar, use a browser extension that calls the OpenAI API directly, or access AI through a third-party wrapper site. Tab-based filters only see the visible tab, they have no visibility into background API requests.

3. Incognito mode and guest sessions
Many tab-based filters rely on browser extensions that do not run in incognito mode or guest sessions by default. A student who opens an incognito window can access AI tools without any filtering. While some filters can be configured to run in incognito mode, this is not the default behavior and requires additional setup.

4. Mobile and app-based access
Tab-based filters are browser-specific. If a student uses the ChatGPT mobile app, a standalone AI app, or accesses AI through a non-browser interface, tab-based filters provide no protection. While this guide focuses on Chromebook management, the broader point is that tab-based filtering is inherently limited to what happens inside a monitored browser tab.

How API-Level Blocking Works

API-level blocking takes a fundamentally different approach. Instead of monitoring what students see in their browser tabs, it blocks network requests to AI services **before they leave the**

device. This is done using Chrome's `declarativeNetRequest` API, which allows extensions to define network-level blocking rules that apply to all requests, visible tabs, background requests, iframes, API calls, and more.

Here is how it works:

1. Define blocking rules at the domain level
Instead of trying to block specific pages or subdomains, API-level blocking targets entire domains. A single rule that blocks `*.openai.com` will block `chat.openai.com`, `api.openai.com`, `platform.openai.com`, and any future subdomains OpenAI creates. This makes the blocking future-proof and eliminates the whack-a-mole problem.

2. Block at the network layer
The `declarativeNetRequest` API operates at the network layer, before any HTTP request is sent. This means it blocks API calls, background requests, embedded iframes, and any other network activity targeting the blocked domain. A student cannot bypass this by opening a new tab, using an incognito window, or embedding AI in a sidebar, the network request is blocked before it leaves the Chromebook.

3. No server communication required
Because the blocking happens locally on the device, there is no need for the extension to communicate with a remote server, send student data to a third party, or rely on an internet connection to function. The blocking rules are embedded in the extension itself and enforced by Chrome's built-in network stack. This makes the solution faster, more private, and more reliable than server-side filtering.

4. Works across all browsing contexts
Unlike tab-based filters, API-level blocking applies to all browsing contexts: regular tabs, incognito tabs, guest sessions, iframes, service workers, and background requests. If the Chromebook tries to connect to a blocked domain, the request is denied, no exceptions.

What Gets Blocked: The Complete List

The UserAuthGuard Free extension blocks **83 sites** across three categories:

AI Chatbots (24 sites)
ChatGPT, Claude, Gemini, Perplexity, Character.AI, Poe, Microsoft Copilot, Meta AI, Cohere, DeepSeek, Mistral, Groq, and more. This includes all major consumer-facing AI chatbots that students use for homework assistance.

AI Writing Tools (23 sites)
Quillbot, Grammarly, Copy.ai, Jasper, Writesonic, Wordtune, Rytr, HyperWrite, Jenni.ai, Sudowrite,

and more. These tools are specifically designed to generate or rewrite essays, making them popular for academic dishonesty.

Homework & Cheating Sites (36 sites)
Chegg, Course Hero, Brainly, Photomath, Mathway, Quizlet, SparkNotes, and more. These sites have been problematic for years, and blocking them alongside AI tools creates a comprehensive anti-cheating policy.

The full list is maintained in the extension's source code and can be customized by IT directors who want to add or remove specific sites.

Implementation: Force-Install the Extension

The UserAuthGuard Free extension is designed for zero-configuration deployment. IT directors force-install it via the Google Admin console, and it immediately begins blocking AI access on all managed Chromebooks. No student interaction is required, and no settings need to be configured.

Step 1: Add the extension to your Admin console
Navigate to **Devices > Chrome > Apps & extensions > Users & browsers** in the Google Admin console. Search for "UserAuthGuard Free" in the Chrome Web Store, or use the extension ID to add it directly. (Note: The extension is currently in beta and not yet published to the Chrome Web Store. Contact support@userauthguard.com for early access.)

Step 2: Force-install for all student OUs
Select the organizational units (OUs) that contain student devices. Set the installation policy to **Force install**. This ensures the extension is installed automatically on all student Chromebooks and cannot be disabled or removed by students.

Step 3: Verify deployment
Log in to a student Chromebook and attempt to access `chat.openai.com`, `claude.ai`, or `quillbot.com`. You should see a "This site can't be reached" error, indicating that the network request was blocked. If the site loads, verify that the extension is installed and enabled in `chrome://extensions`.

Step 4: Communicate the policy
Inform students, teachers, and parents that AI tools are blocked on school devices as part of your academic integrity policy. Provide guidance on acceptable use of technology for learning and clarify which tools are permitted (e.g., Google Docs, Khan Academy, district-approved educational software).

Technical Implementation Details

For IT directors who want to understand or customize the blocking rules, here is how the extension works under the hood:

manifest.json

The extension uses Manifest V3, which is required for all new Chrome extensions. The `declarativeNetRequest` permission allows the extension to define network-level blocking rules. The `host_permissions` field specifies which domains the extension can block (in this case, all domains via `*://*/*`).

```
{
  "manifest_version": 3,
  "name": "UserAuthGuard Free",
  "version": "1.0",
  "permissions": ["declarativeNetRequest"],
  "host_permissions": ["*://*/*"],
  "declarative_net_request": {
    "rule_resources": [{
      "id": "ruleset_1",
      "enabled": true,
      "path": "rules.json"
    }]
  }
}
```

rules.json

The blocking rules are defined in a separate JSON file. Each rule specifies a domain pattern to block and the action to take (in this case, block the request). The `urlFilter` field uses wildcard matching to block all subdomains of a given domain.

```
[
  {
    "id": 1,
    "priority": 1,
    "action": { "type": "block" },
    "condition": {
      "urlFilter": "*://chat.openai.com/*",
      "resourceTypes": ["main_frame", "sub_frame", "xmlhttprequest"]
    }
  },
  {
    "id": 2,
    "priority": 1,
    "action": { "type": "block" },
    "condition": {
      "urlFilter": "*://claude.ai/*",
      "resourceTypes": ["main_frame", "sub_frame", "xmlhttprequest"]
    }
  }
]
```

The `resourceTypes` field specifies which types of requests to block. `main_frame` blocks top-level page loads, `sub_frame` blocks iframes, and `xmlhttprequest` blocks API calls. This ensures comprehensive coverage across all request types.

Customization

IT directors who want to add or remove sites can modify the `rules.json` file and repackage the extension. Each rule must have a unique `id`, and the `urlFilter` field supports wildcard matching for flexible domain blocking. The extension can block up to 30,000 rules, far more than the 83 currently defined.

Privacy and Data Collection

The UserAuthGuard Free extension collects **zero data**. It does not track student browsing, send usage analytics, or communicate with any external servers. All blocking happens locally on the device using Chrome's built-in network stack. The extension's privacy policy is available at **userauthguard.com/extension**.

This is a critical distinction from many commercial content filters, which route student traffic through proxy servers, log browsing history, and sell anonymized data to third parties. API-level blocking eliminates these privacy concerns entirely.

Comparison: Tab-Based vs API-Level Blocking

FEATURE	TAB-BASED (SECURELY, GOGUARDIAN)	API-LEVEL (USERAUTHGUARD FREE)
Blocks visible tabs	Yes	Yes
Blocks background API calls	No	Yes
Blocks iframes and embeds	⚠ Sometimes	Yes
Works in incognito mode	⚠ Requires config	Yes
Works offline	No	Yes
Data collection	⚠ Varies by vendor	None
Cost	$3-8/student/year	Free

Limitations and Considerations

No blocking solution is perfect. Here are the limitations of API-level blocking that IT directors should understand:

1. Does not block personal devices
This solution only works on managed Chromebooks. Students who use personal devices, home computers, or smartphones can still access AI tools outside of school. This is a policy and education challenge, not a technical one.

2. VPNs and proxies can bypass blocking
If a student installs a VPN extension or uses a proxy service, they can route their traffic around the blocking rules. To prevent this, IT directors should block VPN and proxy extensions via the Google Admin console and monitor for unauthorized extension installs.

3. New AI services require rule updates
As new AI tools emerge, the blocking rules must be updated to include them. The UserAuthGuard Free extension is updated regularly to include new AI services, but there will always be a lag between a new tool's release and its addition to the blocklist.

4. May block legitimate educational AI use

Some teachers use AI tools for lesson planning, differentiation, or as a teaching aid. Blocking AI district-wide may interfere with these legitimate use cases. IT directors should work with curriculum leaders to define acceptable AI use policies and create exceptions for teacher devices or specific educational contexts.

Beyond Blocking: Teaching Digital Citizenship

Blocking AI access is a technical control, not an educational solution. Students need to understand why using AI for homework is problematic, not just that it is blocked. Effective digital citizenship education includes:

- Explaining the difference between using AI as a learning tool (e.g., asking clarifying questions) versus using it to bypass learning (e.g., generating entire essays)
- Teaching students to evaluate AI-generated content for accuracy, bias, and appropriateness
- Providing opportunities for students to use AI in supervised, educationally appropriate contexts
- Reinforcing the value of original thinking, problem-solving, and intellectual honesty

Technology controls like API-level blocking buy time for these conversations to happen. They are not a substitute for teaching students to make ethical decisions about technology use.

Getting Started

The UserAuthGuard Free extension is currently in beta and available for early access. To request access, contact **support@userauthguard.com** with your district name and student enrollment count. The extension will be published to the Chrome Web Store in Q2 2026.

For IT directors who want to implement a comprehensive Chromebook management solution that includes device tracking, repair workflows, and accountability systems in addition to content filtering, explore **UserAuthGuard's full platform**. The free extension is a standalone tool and does not require a paid UserAuthGuard subscription.

Conclusion

AI access on student Chromebooks is a solvable problem, but it requires moving beyond tab-based content filters. API-level blocking provides comprehensive, privacy-respecting protection that works across all browsing contexts and requires zero configuration. The UserAuthGuard Free extension makes this approach accessible to any K-12 district, regardless of budget or technical expertise.

As AI tools continue to evolve, the challenge for IT directors will be balancing access control with educational opportunity. Blocking AI entirely is not a long-term strategy, students will need to learn how to use AI responsibly as part of their digital literacy education. But in the short term, API-level blocking gives schools the control they need to enforce academic integrity policies while that education happens.

Want to see UserAuthGuard in action?

Free: every device, user and OU in your Workspace, plus every assignment gap. No device cap, no credit card. Pay only to enforce.

userauthguard.com/signup | **Book a Demo**