

WUNDERLANDMEDIA

Back Button Hijacking Audit Checklist

Check Your Site Before Google's New Spam Policy

By Kemal | wunderlandmedia.com

DEADLINE: JUNE 15, 2026

Step 1: Check for Common Culprits

Review your site for these known causes of back button hijacking. Check each one off as you verify it's not present or not causing issues.

- ☐ **Exit-intent popup plugins**
The #1 offender. Check if your popup plugin uses history.pushState() to trigger "are you sure?" overlays. Test by disabling and checking back button.
- ☐ **Ad network scripts**
Programmatic ad scripts that inject JavaScript. They sometimes push history entries to track engagement or prevent accidental navigation.
- ☐ **Remarketing tags**
Certain remarketing implementations use redirect chains that interfere with back button behavior.
- ☐ **A/B testing tools**
Some testing platforms inject history entries to track variant navigation. Not all clean up after themselves.
- ☐ **Consent management platforms**
Cookie consent overlays can interfere with navigation, especially on mobile devices.
- ☐ **Engagement widgets**
"Recommended articles" widgets that slide in from the bottom. Some use history manipulation to track scroll depth.
- ☐ **Theme-level scripts**
Check your theme's JS files for history.pushState or popstate event listeners. Some themes include built-in "engagement" features.

Watch out for mobile

These scripts often work fine on desktop but break the back button on mobile. Always test both.

Other scripts on your site to check:

Step 2: The 5-Minute Audit

Manual Test (2 minutes)

- ☐ **Open your site in Chrome**
Or any browser. Use a normal browsing session, not incognito (you want your real scripts loaded).
- ☐ **Navigate to 3-5 different pages**
Click around like a normal user. Visit the homepage, a product/service page, a blog post, and a contact page.
- ☐ **Hit the back button**
Does it take you where you expect? Or does something weird happen (loop, redirect, nothing)?
- ☐ **Repeat on mobile**
Many scripts behave differently on touch devices. Test on a real phone, not just responsive mode.

DevTools Test (3 minutes)

- ☐ **Open Chrome DevTools (F12)**
Go to the Application panel and check History entries. Are there entries you didn't create by navigating?
- ☐ **Check the Network panel**
Navigate your site. Watch for unexpected redirects (301s, 302s, or JavaScript redirects).
- ☐ **Check the Console**
Look for history.pushState or history.replaceState calls. These are the technical mechanism behind back button hijacking.

Search Console Check

- ☐ **Open Google Search Console**
Navigate to Security & Manual Actions > Manual Actions.
- ☐ **Check for warnings**
If you see a warning about back button hijacking, you're already flagged. Google is sending warning emails too.

Result
If everything checks out, you're clear. If you found something, don't panic. Move to Step 3.

Issues found (note them here):

.....

.....

.....

.....

Step 3: Fix It Before June 15☐ **Identify the source**

Disable plugins one by one in staging. Start with popup plugins, ad scripts, and overlay managers. The guilty script becomes obvious when the back button works normally again.

☐ **Check your theme**

Search your theme's JavaScript files for `history.pushState` or `popstate` event listeners.

☐ **Audit third-party scripts**

Use Chrome DevTools Network panel with "Preserve log" enabled to identify which external scripts call history APIs.

☐ **Remove or replace the culprit**

Find an alternative that doesn't hijack navigation. Modern popup plugins offer "clean" exit-intent using mouse movement tracking instead.

☐ **Test again on desktop**

Repeat the 5-minute audit from Step 2 after the fix.

☐ **Test again on mobile**

Especially important. Use a real device, not just browser responsive mode.

☐ **Test on multiple browsers**

Chrome, Firefox, Safari at minimum. Edge if your audience uses it.

☐ **Deploy to production**

Once confirmed working in staging, push the fix live.

☐ **Request reconsideration (if needed)**

If Google already flagged your site, submit a reconsideration request through Search Console after deploying the fix.

Important: You're responsible for third-party code

Even if a plugin or ad network causes the hijacking, Google holds you responsible. It's on your domain, it's your problem.

Fix notes:

Quick Reference

June 15

Enforcement deadline

2 Types

Manual action or automated demotion

2-4 Weeks

Recovery after fix

Penalty Type	What Happens	Recovery
Manual Action	Pages removed from search entirely	Fix + reconsideration request (2-4 weeks)
Automated Demotion	Rankings gradually drop site-wide	Fix + wait for recrawl (can take longer)

What to Search For	Where to Look
history.pushState()	Theme JS, plugin JS, Console
history.replaceState()	Theme JS, plugin JS, Console
popstate event listeners	Theme JS, plugin JS
Unexpected 301/302 redirects	DevTools Network panel
Manual action warnings	Google Search Console

"If your site hijacks the back button, fixing it isn't just about avoiding a penalty. It's about not being annoying. And that's a pretty low bar."

Kemal, Wunderlandmedia

Read the full breakdown

This checklist covers the essentials. For the complete guide with technical details, examples, and context:
wunderlandmedia.com/google-back-button-hijacking-spam-policy-june-2026