



How to Revoke Every Joomla API Token After a Hack

A Joomla password reset never touches API tokens. How to find and revoke every one after a hack, by hand on one site or in one click with mySites.guru.

Phil E. Taylor | 29 September 2026

You clean up a Joomla hack the right way. Patch the extension that let the attacker in, delete the webshells, force every password to reset, force every session out. A week later the site is hacked again, through a route that makes no sense, because the hole is patched and every password is new.

A Joomla API Token is not a password, so a password reset never touches it. If the attacker created one before you cleaned up, it authenticates against `/api/` exactly as it did the day it was issued, and it keeps working until someone specifically revokes it. Joomla has no page listing every token on a site and no button that clears them all, so on a site with more than a handful of users, revoking one specifically almost never happens.

Revoke Every Joomla API Token From mySites.guru in One Click

mySites.guru's [Revoke Every API Token After a Hack](#) tool does in one click what Joomla makes you do one profile at a time. It reads every issued token across every user, the `joomla_token.*` rows in `#__user_profiles`, shows you who holds one, and removes all of them in a single action. Three options sit alongside it:

Revoke every token

Deletes every issued Joomla API Token on the site, across every user, in one action.

Rotate the update key

For sites on Joomla 5.4+ with Automated Core Updates on: writes a new `update_token` and re-registers the site with `autoupdate.joomla.org`, so a stolen key stops working.

Switch the API off

Disables the webservices and api-authentication plugins in the same pass, so there is nothing left to authenticate against.

Log everyone out

Ends every #__session row and every remember-me row in #__user_keys, on top of the tokens. Off by default, because it also ends your own session if you reached the site through mySites.guru.

It is a manual tool you run when you need it, not a background check: find it from the tool finder (press Cmd-K, or Ctrl-K on Windows) or from the link on a hacked site's card, run it once, and every token on that site is gone. It clears a Super User's token in the same click as an editor's, because a hack does not care which of them the attacker actually used.

One count is worth knowing before you run it. The token total is read straight from `#__user_profiles`, so a token belonging to a user account that has since been deleted still counts and still gets removed, but it cannot be named in the holder list, because that list has to join against `#__users` to show a username. If the two numbers disagree by one or two, that is why, not a bug.

What Is a Joomla API Token?

A Joomla API Token is a per-user credential for Joomla's REST API, which Joomla's own screens call Web Services. It is created from that user's own profile, on a **Joomla API Token** tab that only appears when the **User – Joomla API Token** plugin is enabled and the user's group holds the Web Services Login permission. A request authenticates with an **Authorization: Bearer** header, or **X-Joomla-Token** on servers that strip the Authorization header before PHP sees it.

Nicholas Dionysopoulos built token authentication in [pull request #27021](#) in March 2020, and it shipped live with [Joomla 4.0.0 on 17 August 2021](#).

Joomla never stores the raw token. What sits in `#__user_profiles` is a seed, and a request is verified with HMAC-SHA256 against the site's own `secret` value in `configuration.php`. That detail is the whole reason a password reset changes nothing: the token is not derived from the user's password at all. It is a second, independent credential, checked against a site-wide secret rather than anything about the user holding it.

Why Cleanup Usually Skips This Step

Every hack-cleanup checklist says roughly the same thing: change every password, rotate every credential, force every session out. [Joomla's own Security Checklist](#) says it, and so does our own guide to [finding rogue admin accounts](#). None of that is wrong, but "rotate every credential" rarely gets broken down far enough to name a credential that lives outside the password system entirely, which is exactly the shape of gap that leaves a token still working.

The reason is not carelessness. Joomla gives you nowhere to look. There is no screen that lists who holds a token, so checking for one is not a step anyone can act on without opening every profile in turn, and on a site with real editorial staff that is dozens of profiles for a check most people have never heard needs doing at all.

The checklist stops at the technical fix in every case we have seen, too. Whether anyone outside your own team now has to be told what that token could reach is a separate question, and it is the one covered last in this post, not because it matters least.

How Do You Revoke a Joomla API Token by Hand?

Without mySites.guru, revoking a token means working through the user list one profile at a time:

1. In the administrator, go to **Users**, then **Manage**, and open the profile you want to check.

2. Open the **Joomla API Token** tab. It only appears for a user whose group holds Web Services Login, so most ordinary accounts will not show it at all.
3. Set **Enabled** to **No** to remove the token completely, or set **Reset** to **Yes** to issue a fresh token and invalidate the old one in the same save. Click **Save & Close**.
4. Repeat for every Super User, and for any other group your site has given Web Services Login to. There is no list of who currently holds a token, so working through the whole user list is the only way to be sure you have not missed one.
5. Confirm from outside with the old token:

```
curl -H "Authorization: Bearer OLDTOKEN" https://example.com/api/index.php
```

A revoked token answers 401. On a site with five editors this is ten minutes. On a client site with forty accounts built up over a decade, it is the step that gets deferred, and then forgotten.

Rotating the site secret is a last resort, not a shortcut

Every token verifies against `configuration.php`'s `secret` value, so changing it invalidates every token on the site in one move, without touching a single profile. It is real, and it is not designed as a revocation feature. Joomla core uses the same secret to sign sessions and to encrypt some stored values, but core is not the only thing that might be using it: any third-party extension is free to read `secret` out of `configuration.php` for its own encryption, licensing checks or anything else a developer decided to key off it, and there is no registry of which ones do. You cannot know what changing it will break on a given site without auditing every installed extension first, so it can just as easily make things worse as make them better. Reach for it only on a site you are already rebuilding from scratch, never as a routine way to clear tokens.

Is There a Known Vulnerability in the Joomla API Worth Checking For?

Yes, if the site has not been updated recently. [CyStack's research, published in April 2026](#), found two flaws in Joomla's core REST API: an SQL injection in the articles endpoint's ordering parameters ([CVE-2026-21630](#)) and a missing access check on the `com_config` endpoints ([CVE-2026-23899](#)) that let an authenticated caller read the site's `secret`, the same value every token is checked against. Both affected Joomla 4.0.0 to 5.4.3 and 6.0.0 to 6.0.3, and both were fixed on 31 March 2026, part of the record we keep on [the Joomla API's security history](#).

If a site was on an affected version when it was hacked, treat every token issued since as suspect. A leaked secret can be used to forge a valid token without ever touching a user's profile, which is one more reason a revoke has to cover every token rather than just the ones that look unfamiliar.



Should You Also Switch the Joomla API Off?

If nothing on the site needs it, yes, and it is worth doing alongside a token revoke rather than instead of it. Revoking tokens clears the credential. [Switching the API off](#) removes the whole surface those credentials authenticate against, including Basic authentication if it is enabled and any third-party webservises plugin. mySites.guru's revoke tool has a switch for exactly this, right next to the token revoke, so both happen in the same pass.

What Automated Core Updates Change About This

If the site runs Joomla 5.4 or later with [Automated Core Updates](#) switched on, there is a second credential worth handling at the same time: the `update_token` that authenticates Joomla's own update server calling back into

`/api/index.php/v1/joomlaupdate/`. It is a different mechanism to a user's API token,

registered with `autoupdate.joomla.org` rather than issued from a profile, but it lives on the same API surface, and mySites.guru's revoke tool can rotate it and re-register the site in the same action that clears the tokens. Joomla 4 and Joomla 5.0 to 5.3 have no automated updates, so this step does not apply to them.

Does Revoking Tokens Stop the Site From Being Hacked Again?

No, and nothing here replaces working out how the attacker got in. Removing tokens closes one specific door, a credential a password reset never touches. It does not remove a rogue admin account, a webshell, or the vulnerable extension that let someone in to begin with. Pair it with [finding rogue admin accounts across every Joomla site you manage](#) and, for the file side of the cleanup, our [guide to cleaning a hacked Joomla or WordPress site](#). If you would rather hand the whole thing to someone, [fix.mysites.guru](#) is a single fixed fee of £120 per incident, usually same day, and non-subscribers get a free month included.

The Step After the Technical Cleanup: Do You Need to Report a Breach?

Every checklist covered so far, ours included until this post, stops at the technical fix: patch the hole, remove the malware, revoke the credentials. None of it asks the other question a hack can raise, which is whether someone outside your team now has to be told.

If there is a reasonable chance an attacker used a live API token to read personal data, usernames, email addresses, order records, anything that identifies a real person, that is very likely a personal data breach under UK GDPR, or the equivalent under the EU GDPR for an EU-established controller, or your own jurisdiction's data protection law elsewhere. The law does not care how the data was read. An API token pulling the users endpoint for three weeks and a defaced homepage are both a breach if personal data was exposed, and only the second one looks like an incident.

Two duties then run on their own clock, alongside the technical cleanup rather than after it:

- Tell your supervisory authority. In the UK that is the Information Commissioner's Office, and the rule is 72 hours from when you become aware of the breach, not from when the token was first created, unless the breach is unlikely to put anyone's rights and freedoms at risk.
- Tell the people affected, without undue delay, if the breach is likely to result in a high risk to them, for example because what leaked could be used for fraud or identity theft.

Working out whether either duty applies takes evidence the cleanup already produces: when the token was issued, what endpoints it could reach, and how long it went unnoticed. That is one more reason to revoke it and write down what you found in the same sitting, rather than treating the technical fix as the end of the incident. This is not legal advice, and the thresholds differ by jurisdiction, so involve whoever handles data protection for the business, or a solicitor, rather than deciding it alone from a blog post.

Where mySites.guru Fits

Token revocation is a five-minute job on one site and a real, recurring cost across a portfolio, especially since it is the step every generic checklist leaves out. A mySites.guru subscription covers the rest of what a hack response needs from one dashboard:

- Rogue Super Admin accounts are flagged on every connected Joomla site automatically, with a one-click delete.
- Installed extensions are matched against our Joomla vulnerability database, so the extension that let the attacker in the first time gets flagged before it happens again.
- Backups run before every mass change, so a revoke, a Joomla update, or an extension update on a whole portfolio is never a one-way action.
- Sites still on an end-of-life Joomla version, where the API's known flaws have no fix available, are flagged so you know where the risk actually sits.

See the plans, or run a free audit on one site to see what it finds.

Further Reading

- [Pull request #27021, joomla-cms](#) - the change that added Joomla API Token authentication.
- [CyStack: Security Research on the Joomla REST API](#) - the SQL injection and access-control flaws fixed in Joomla 5.4.4 and 6.0.4.
- [Joomla Security Checklist: You have been hacked or defaced](#) - Joomla's own cleanup checklist.
- [Joomla 4.0 and Joomla 3.10 release announcement](#) - the release that shipped the API and token authentication live.

Frequently Asked Questions

Does a Joomla password reset revoke API tokens?

No. A Joomla API Token is verified with HMAC-SHA256 against the site's secret in configuration.php, not against the user's password hash, so changing a password has no effect on it. It keeps working until someone specifically disables or resets it on that user's profile.

How do I see who holds a Joomla API token?

Joomla core has no screen for this. mySites.guru's Joomla API (Web Services) Investigate page lists the users holding an enabled token, by id, username and name, without ever reading or sending the token itself. Without that, the only way is opening every user's profile and checking the Joomla API Token tab by hand.

Can I revoke every Joomla API token at once without mySites.guru?

Not through any button in the administrator. There is a blunt site-wide side effect: changing the secret value in configuration.php invalidates every token at once, because every token is verified against it. It also ends every session and can break anything else that relies on the secret staying stable, so treat it as an emergency measure, not a routine one.

Does revoking API tokens break Joomla's automated core updates?

Not on its own. Automated Core Updates on Joomla 5.4 and later authenticate with a separate update_token, not a user's API token. mySites.guru's revoke tool has its own option to rotate that key and re-register the site, alongside the token revoke, for sites that use the feature.

Does this apply to Joomla 3?

No. The API (Web Services) layer, and API Tokens with it, arrived in Joomla 4.0.0 in August 2021. Joomla 3 has nothing to revoke here.

Is there a WordPress equivalent of this tool?

Yes. WordPress's equivalent credential is the Application Password, and mySites.guru has a matching post-hack tool that revokes every one on a connected WordPress site in one click. See revoke every WordPress Application Password after a hack.

A legitimate integration lost its token when I revoked everything. What now?

Reissue a token on purpose from a dedicated account's profile rather than reusing the deleted one, and only for the specific integration that needs it. A token created for one purpose years ago and never revisited is exactly the kind of standing credential this whole exercise exists to clear out.

Do I need to report a Joomla API token hack to a data protection authority?

If there is a reasonable chance the token was used to read personal data, such as user names, email addresses or order records, it may be a notifiable personal data breach under UK GDPR or the equivalent law in your jurisdiction, regardless of whether the token or a webshell was how the data was read. In the UK, the ICO's rule is 72 hours from when you become aware, and affected individuals must be told without undue delay if the risk to them is high. This is not legal advice; get a definite answer from whoever handles data protection for the business.

AI MODIFIED

Written and edited by a human, with AI assistance. [Our approach to AI](#)

Get Your Free Site Audit

See how your WordPress and Joomla sites measure up.
No credit card required.

<https://manage.mysites.guru/en/register>

Get in touch

Phil E. Taylor
phil@phil-taylor.com

mySites.guru

© 2026 Blue Flame Digital Solutions Limited. All rights reserved.

mysites.guru