contact@namiq.io · namiq.io

NamiQ

SECURITY ADVISORY

Idunno AntiCsrf exploited

CVE-2026-131914 · on CISA's exploited list since 24 November 2026

Analysis Report

Anyone whose ASP.NET application uses Idunno.AntiCsrf must act by 27 November 2026 — the CISA deadline. The record names no fixed version and CISA advises discontinuing the product; the first step is …

PREPARED BY	NamiQ	DATE	25 November 2026
FACTS AS OF	25 November 2026	VERSION	1.0
SEVERITY	**Critical: CVSS 3.1 9.8**	STATUS	On CISA's exploited list; due 27 November 2026

NamiQ © 2026 NamiQ. All rights reserved. This document is the property of NamiQ. It may be shared unaltered, with attribution.
Companion article: namiq.io/insights/idunno-anticsrf-cve-2026-131914

The bottom line

> *Anyone whose ASP.NET application uses Idunno.AntiCsrf must act by 27 November 2026 — the CISA deadline. The record names no fixed version and CISA advises discontinuing the product; the first step is to find whether the library is present in your applications.*

At a glance

Fact	Detail
Product	Idunno AntiCsrf (CVE-2026-131914)
Severity	Critical: 9.8 (CVSS 3.1)
CISA	On its exploited list since 24 November 2026. Deadline for US federal agencies: 27 November 2026.
Chance of exploitation (EPSS)	0.04, the estimated chance in the next 30 days, as at 25 November 2026
Ransomware use	Not recorded by CISA. That is not the same as none.
Fixed in	No source we read names a fixed version
Who looks first	Any team running an ASP.NET web application that includes Idunno.AntiCsrf at version 0.9.1 Beta or earlier. The record does not name specific ASP.NET versions or hosting configurations beyond that.

Does this apply to you?

Answer in order. Stop at the first no.

Question	If no	If yes
Do you run any ASP.NET web application?	Nothing to do for this CVE.	Check whether Idunno.AntiCsrf is a dependency.
Does any of those applications include the Idunno.AntiCsrf library?	Nothing to do for this CVE.	All listed versions are affected; plan to remove the library.
Is the application reachable from outside your internal network?	Still affected; treat as medium priority and act before end of week.	Treat as high priority; act within the CISA window.

Has the library been removed or replacement CSRF protection deployed?	Complete removal before the CISA deadline of 27 November 2026.	Confirm the deployment and keep a record of what replaced it.

What it is

The library's double-submit cookie check deserialises data it receives from a client request without first verifying it. An unauthenticated attacker can send a crafted request that causes the server to instantiate a malicious object and run arbitrary code — no login is needed.

Why it matters

A CVSS score of 9.8 reflects that no authentication is required and the attack comes over the network. The flaw sits inside the CSRF protection layer — code that runs on every inbound form request — so there is no secondary condition an attacker needs to meet beyond being able to send a request. The record lists no unaffected version, and CISA says to discontinue use, together suggesting no patch is coming. Our reading is that this is not a patching task but a removal task: the library should be replaced before any further requests are processed.

Things people will get wrong

- Updating Idunno.AntiCsrf to the latest release will fix this. No source names a fixed or unaffected version. The record covers all versions up to and including 0.9.1 Beta, and CISA advises discontinuing use — not patching.
- Only internet-facing applications are at risk. The record requires no authentication and no specific network position. An attacker with access to the internal network can also reach an internal application.
- Our CSRF protection is working, so we are protected. The flaw is in the CSRF library itself. The vulnerability is triggered through the validation step, not bypassed by it.

How we would triage it

This is our judgement.

Step 1: Find whether Idunno.AntiCsrf is a dependency

- Search your ASP.NET projects' package references and NuGet package lists for Idunno.AntiCsrf.
- Check both production and staging environments — a library present in one is often present in the other.

Step 2: Confirm the installed version

- The record names all versions up to and including 0.9.1 Beta as affected. No unaffected version is listed, so any installed version should be treated as affected.

Step 3: Remove or replace the library

- CISA advises discontinuing use. Remove the library from your application's dependencies.
- Work with your development team to put a different CSRF protection in place before redeploying without it.

Step 4: Review logs for signs of exploitation

- Look for unexpected serialisation errors or unusual process activity in the web server logs during any period the application was reachable.
- Check for files in the application directory not present in your last deployment, which could indicate code was written to disk.

To close it, read the version from the product itself, not from an old inventory, and save the output with the date. If you cannot update today, write down who owns it, the new date, and what limits exposure until then. CISA's wording adds that you should stop using the product if no mitigation is available.

Send this to your IT provider

Copy it, change the names and dates, and send it. It is a draft for you to check, not a template tested with any provider.

> *We are writing about CVE-2026-131914, a critical vulnerability (CVSS 9.8) in the Idunno.AntiCsrf library used in ASP.NET applications. The flaw allows an unauthenticated remote attacker to execute arbitrary code via object injection. CISA has listed it with a remediation deadline of 27 November 2026. No fixed version is recorded; the guidance is to discontinue use of the library entirely. Please check whether any of our ASP.NET applications include this library, confirm the version if found, and advise us on removal and replacement. We need written confirmation of the library's status across all environments and your planned remediation steps by 27 November 2026.*

Signs it was used

An update closes the hole. It does not tell you whether someone used it first. CISA's required action points to its forensic triage requirements (BOD 26-04). Look for:

- Unexpected deserialisation errors in the web application event log during any window the application was externally reachable.
- New processes spawned by the web server process, or outbound network connections from it to unexpected addresses.
- Files in the web application directory created or modified after the last known deployment.

The weakness behind it

The record classes this as CWE-502 (Deserialization of Untrusted Data).

Likely ATT&CK mapping

MITRE ATT&CK is a public catalogue of the steps attackers take, from getting in to causing damage. No source has mapped this flaw to it yet, so the steps below are our prediction from what the record says the flaw allows. They are not MITRE's, the vendor's or CISA's, and an attacker may do other things.

Steps an attacker would likely take with this flaw

The step, in plain words	Stage of an attack	Why we think so
Exploit Public-Facing Application (T1190): Breaking in through a service that faces the internet, such as a web server, VPN or login page.	Initial Access: getting in	The flaw is in a web library reachable on any inbound request with no authentication required; an attacker over the network would likely exploit it as the initial entry point.
Command and Scripting Interpreter (T1059): Running commands or scripts on a machine, such as a shell or PowerShell.	Execution: running the attacker's code	Deserialization object injection leading to arbitrary code execution would typically involve running commands or scripts on the server once the crafted payload is deserialised.

How to use this: ask your IT provider which control or log would show each step, for example whether sign-in attempts from the internet are recorded. If nobody can answer, that is the gap to close first.

MITRE ATT&CK® is a registered trademark of The MITRE Corporation. Technique names are MITRE's.

The sources, in their words

CISA's catalogue entry:

> *Idunno AntiCsrf contains a deserialization of untrusted data vulnerability that allows an unauthenticated remote attacker to execute arbitrary code via object injection. The impacted product(s) could be end-of-life (EoL) and/or end-of-service (EoS). Users are advised to discontinue use and/or transition to a supported version.*

The CVE record, published by Mandiant on 19 November 2026:

> *Deserialization of untrusted data in double-submit cookie vs. form field validation in Idunno.AntiCsrf versions through 0.9.1 Beta on ASP.NET allows remote unauthenticated attackers to execute code via object injection*

Versions of Idunno AntiCsrf the record lists as affected:

- all versions up to and including 0.9.1.0
- all versions up to and including 0.9.1 Beta

CISA's required action:

> *Apply mitigations in accordance with vendor instructions.*

Check your own project

If you build software, you can compare what a project declares with this CVE from a terminal. The tool reads requirements files, package.json, Maven, Gradle, Cargo, CMake and Dockerfile files, runs nothing, and only the CVE id leaves your machine. Naming the CVEs needs no key.

```
lattice-scan project . --cve CVE-2026-131914
```

Each row says in-range, outside-range, undetermined or lead, with the file and line. That is evidence about what a manifest declares, not about what runs, and a person decides. The verdicts, the CSV columns and a pipeline recipe are in the guide.

Check the servers you run

The same id can be checked against a host. Run this on the machine itself, not on the laptop that deploys to it: it reads installed packages (dpkg, rpm or apk), snaps, the hardware (maker, model, firmware), the running container images with their operating system and packages, and the kernel version, compares them on the spot, and only the CVE ids leave the machine; the inventory stays on

it, in ~/.lattice-scan. Naming the CVEs needs no key. Do it on each server after a deployment, and again when a new entry lands.

```
lattice-scan --cve CVE-2026-131914
```

A match is a candidate for a person to confirm, not a finding. A distribution can fix a CVE under an unchanged version number, so a match is a candidate; on Debian-family hosts the report also shows whether the package's own changelog names the CVE. By default it opens the images your containers are running and reads their operating system and package list; a deep scan (--deep, a few minutes) opens every local image and the Python, Node, Java and Ruby packages inside them; for what an image declares, use the project check on its Dockerfile.

Guide: namiq.io/lattice/guides/choose-your-channel. MCP server (io.namiq/lattice in the MCP registry): namiq.io/lattice. Package: pypi.org/project/lattice-scan, or namiq.io/apt.

What we do not know yet

- No source names a fixed or unaffected version of Idunno.AntiCsrf.
- Whether the product is formally end-of-life and whether the maintainer has issued any public statement.
- Whether a workaround exists that reduces exposure without removing the library entirely; no vendor page could be read.
- Which specific ASP.NET hosting models or configuration details affect exploitability beyond what the record states.
- We have not tested this against any environment. Versions and scores are as the sources state them and as at 25 November 2026; the vendor's advisory wins where it differs.
- The sections headed "The bottom line", "Does this apply to you?", "What it is", "Why it matters", "Exposure by situation", "Things people will get wrong", "Likely ATT&CK mapping", "How we would triage it" and "Send this to your IT provider" are our interpretation of the sources, not a statement by CISA or the vendor.

Appendix 1. Severity decoded

CVE-2026-131914: score 9.3, `CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/E:A`
As the CVE record gives it. The record's own score can differ from the base score on the NVD record.

Metric	Value	What it means here
Attack vector	Network	Reachable over a network, with no need to be on the same segment.
Attack complexity	Low	No special conditions beyond the flaw itself.
Attack requirements	None	Works against a default deployment.
Privileges required	None	No account is needed.
User interaction	None	No one has to do anything.
Vulnerable system: confidentiality	High	Total loss of confidentiality.
Vulnerable system: integrity	High	Total loss of integrity.
Vulnerable system: availability	High	Total loss of availability.
Subsequent systems: confidentiality	None	No data exposed.
Subsequent systems: integrity	None	Nothing can be altered.
Subsequent systems: availability	None	No effect on availability.

Appendix 2. Timeline

Date	Event
19 November 2026	CVE record published by Mandiant.
24 November 2026	CISA added it to its list of exploited vulnerabilities.
27 November 2026	CISA's deadline for US federal agencies.

Appendix 3. Exposure by situation

These priorities are our judgement from the attacker prerequisites in the sources. They are not vendor ratings.

Your situation	Priority	Timeframe

ASP.NET application using Idunno.AntiCsrf, reachable from the internet	High	Within the CISA window — by 27 November 2026
ASP.NET application using the library, internal network only	High	This week
ASP.NET applications in estate but library presence not yet confirmed	Medium	Confirm today; act if found
No ASP.NET applications in the estate	None	No action for this CVE

Appendix 4. Definition of done

- Every Idunno AntiCsrf you run is listed, with its version.
- Each is outside the affected list, or the vendor's mitigation is in place.
- The version output or a screenshot is saved with the date.
- Your projects' declared dependencies were checked against this CVE, and every row has a written reason.
- Logs for the weeks before the update were searched, and anything odd was written down.
- Anything not done has an owner and a date.

Sources

- CISA Known Exploited Vulnerabilities catalogue: https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- CVE record for CVE-2026-131914: https://www.cve.org/CVERecord?id=CVE-2026-131914
- FIRST: Exploit Prediction Scoring System: https://www.first.org/epss/
- Reference in the CVE record: github.com: https://github.com/mandiant/Vulnerability-Disclosures/blob/master/2026/MNDT-2026-0048.md
- Reference in the CVE record: codeplexarchive.org: https://codeplexarchive.org/project/AntiCSRF