

Alert Bulletin

Bulletin N.: 84

Publication Date: 05/08/2026

Subject: Alert 2026-84 File Writing and RCE via GeoDjango (CVE-2026-15307)

Traffic Light Protocol (TLP): White

Affected product(s):

Django — High-level Python web framework

Version	Status
Django main (development branch)	Vulnerable
Django 6.1 (release candidate)	Vulnerable
Django 6.0 (up to 6.0.7)	Vulnerable
Django 5.2 (up to 5.2.16)	Vulnerable

Previous unsupported branches (5.0.x, 4.2.x, 4.1.x, 3.2.x) **were not evaluated** and may be equally affected.

Description

A high-severity vulnerability was recently disclosed in the **GeoDjango** component of the Django framework that allows an authenticated attacker with view permissions on models with spatial fields to cause **arbitrary file writing on the server** and, depending on the active raster driver, execute **remote code (RCE)**. The flaw also enables **SSRF** (Server-Side Request Forgery) attacks, allowing access to internal systems or cloud environment metadata.

Field	Detail
CVE	CVE-2026-15307
Severity	High
CWE	CWE-918 (SSRF) / server-side file writing
Component	GeoDjango — spatial lookups (filtering in admin views)
Authentication required	Yes — staff user with view permission on models with spatial fields

The issue lies in the fact that GeoDjango's **spatial lookups** optimistically parse the right-hand value of a query without properly validating or sanitizing the input. An attacker controlling that value can:

1. **File writing:** Leverage the active raster driver to write arbitrary files to the server's file system, which in certain scenarios can lead to **remote code execution**.
2. **SSRF:** Force the server to make HTTP requests to internal systems or cloud metadata endpoints (AWS IMDSv1, GCP metadata, etc.), exposing credentials or sensitive configurations.

The most likely exploitation vector is the **Django admin panel** through filtering in changelists of models containing spatial fields (RasterField, GeometryField, etc.).

The patch introduces a **backward-incompatible** change — Django now blocks dict-type values and unsafe strings in spatial lookups. Some existing queries may be affected and require code adjustments.

Mitigation

While the patch is being applied:

- **Restrict access to the admin panel** to trusted IPs or via VPN.
- **Review which models expose spatial fields** in the admin and, if possible, temporarily disable filtering in those changelists.
- **Audit staff user permissions:** remove view permissions on models with spatial fields for users who do not strictly require them.
- **Review Django's raster security guide** before applying the patch, as the change may break existing queries.

Solution

Update immediately to the fixed versions:

Affected version	Fixed version	Download link
Django 5.2.x (up to 5.2.16)	5.2.17	https://www.djangoproject.com/download/
Django 6.0.x (up to 6.0.7)	6.0.8	https://www.djangoproject.com/download/

Update via pip:

pip install "django>=5.2.17" # for 5.2 branch

pip install "django>=6.0.8" # for 6.0 branch

Fix commit (5.2 branch):

<https://github.com/django/django/commit/115ffd0463a765ab1cc93de18e94b5459b8a300e>

Additional information:

- **Security Online — CVE-2026-15307:** <https://securityonline.info/django-vulnerability-cve-2026-15307-rce/> [securityonline](#)
- **Debian Security Tracker — CVE-2026-15307:** <https://security-tracker.debian.org/tracker/CVE-2026-15307> [security-tracker.debian](#)

- **Django — Official downloads page:** <https://www.djangoproject.com/download/>
- **Django — Security advisories archive:** <https://docs.djangoproject.com/en/6.0/releases/security/>
- **GitHub fix commit (5.2.17):**
<https://github.com/django/django/commit/115ffd0463a765ab1cc93de18e94b5459b8a300e>