ndaal SBOM Auditor — analysis of django_sbom.xml (SBOM #8534)

SBOM metadata

Name: django_sbom.xml * Format: cyclonedx-xml

Spec version: 1.5 * Components: 3 * Source: upload

SHA-256: e923fbc931210c27b8c545bf3d392f10c6b51c73785011ca68dc40d8680c5eaa

Ingested: 2026-11-12T12:24:32Z

Analyzer summary

syft 1.57.1 — ok — 0 finding(s)

grype 0.121.3 — ok — 35 finding(s)

trivy 0.76.2 — fail — 0 finding(s)

osv-scanner 2.7.1 — ok — 49 finding(s)

bomber 0.5.1 — ok — 49 finding(s)

bomdrift bomdrift 0.9.9 — fail — 0 finding(s)

inspektr_cli inspektr 0.1.9 — fail — 0 finding(s)

shieldbom shieldbom 0.3.0 — fail — 0 finding(s)

provenant provenant-cli 1.0.11 — ok — 0 finding(s)

Findings from grype

[critical] CVE-2024-53908 — django@5.1.2

An issue was discovered in Django 5.1 before 5.1.4, 5.0 before 5.0.10, and 4.2 before 4.2.17. Direct usage of the django.db.models.fields.json.HasKey lookup, when an Oracle database is used, is subject to SQL injection if untrusted data is used as an lhs value. (Applications that use the jsonfield.has_key lookup via __ are unaffected.)

Fixed in: 4.2.17

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-53908

[critical] GHSA-frmv-pr5f-9mcr (CVE-2025-64459) — django@5.1.2

Django vulnerable to SQL injection via _connector keyword argument in QuerySet and Q objects.

Fixed in: 5.1.14

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-64459

[high] CVE-2024-53907 — django@5.1.2

An issue was discovered in Django 5.1 before 5.1.4, 5.0 before 5.0.10, and 4.2 before 4.2.17. The strip_tags() method and striptags template filter are subject to a potential denial-of-service attack via certain inputs containing large sequences of nested incomplete HTML entities.

Fixed in: 4.2.17

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-53907

[high] CVE-2024-56374 — django@5.1.2

An issue was discovered in Django 5.1 before 5.1.5, 5.0 before 5.0.11, and 4.2 before 4.2.18. Lack of upper-bound limit enforcement in strings passed when performing IPv6 validation could lead to a

potential denial-of-service attack. The undocumented and private functions clean_ipv6_address and is_valid_ipv6_address are vulnerable, as is the django.forms.GenericIPAddressField form field. (The django.db.models.GenericIPAddressField model field is not affected.)

Fixed in: 4.2.18

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-56374

[high] CVE-2025-26699 — django@5.1.2

An issue was discovered in Django 5.1 before 5.1.7, 5.0 before 5.0.13, and 4.2 before 4.2.20. The django.utils.text.wrap() method and wordwrap template filter are subject to a potential denial-of-service attack when used with very long strings.

Fixed in: 4.2.20

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-26699

[high] CVE-2025-27556 — django@5.1.2

An issue was discovered in Django 5.1 before 5.1.8 and 5.0 before 5.0.14. The NFKC normalization is slow on Windows. As a consequence, django.contrib.auth.views.LoginView, django.contrib.auth.views.LogoutView, and django.views.i18n.set_language are subject to a potential denial-of-service attack via certain inputs with a very large number of Unicode characters.

Fixed in: 5.0.14

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-27556

[high] CVE-2025-57833 — django@5.1.2

An issue was discovered in Django 4.2 before 4.2.24, 5.1 before 5.1.12, and 5.2 before 5.2.6. FilteredRelation is subject to SQL injection in column aliases, using a suitably crafted dictionary, with dictionary expansion, as the **kwargs passed QuerySet.annotate() or QuerySet.alias().

Fixed in: 4.2.24

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-57833

[high] GHSA-6w2r-r2m5-xq5w (CVE-2025-57833) — django@5.1.2

Django is subject to SQL injection through its column aliases

Fixed in: 5.1.12

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-57833

[high] GHSA-f2ff-p2ww-7p4p (CVE-2026-71491) — sqlparse@0.5.1

sqlparse: Quadratic O(n²) DoS in group_comments

Fixed in: 0.6.0

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-71491

[high] GHSA-hpr9-3m2g-3j9p (CVE-2025-59681) — django@5.1.2

Django vulnerable to SQL injection in column aliases

Fixed in: 5.1.13

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-59681

[high] GHSA-m9g8-fxxm-xg86 (CVE-2024-53908) — django@5.1.2

Django SQL injection in HasKey(lhs, rhs) on Oracle

Fixed in: 5.1.4

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-53908

[high] GHSA-prg7-hcfm-mfcr (CVE-2026-59893) — sqlparse@0.5.1

sqlparse: Inefficient Regex Handling of Dollar-Quoted SQL Literals Leads to ReDoS (Denial of Service)

Fixed in: 0.6.0

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-59893

[high] GHSA-pwgv-4x5q-6m9f (CVE-2026-54284) — sqlparse@0.5.1

sqlparse: TokenList.__init__ materializes O(subtree) value per group, causing CPU DoS before depth/token caps trigger

Fixed in: 0.6.0

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-54284

[high] GHSA-qw25-v68c-qjf3 (CVE-2025-64458) — django@5.1.2

Django has a denial-of-service vulnerability in HttpResponseRedirect and HttpResponsePermanentRedirect on Windows

Fixed in: 5.1.14

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-64458

[high] GHSA-wvqv-fj8w-qmhm (CVE-2026-15307) — django@5.1.2

Django GeoDjango spatial lookups allow file writes and outbound requests through GDAL raster parsing

Fixed in: 5.2.17

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-15307

[medium] CVE-2025-48432 — django@5.1.2

An issue was discovered in Django 5.2 before 5.2.3, 5.1 before 5.1.11, and 4.2 before 4.2.23. Internal HTTP response logging does not escape request.path, which allows remote attackers to potentially manipulate log output via crafted URLs. This may lead to log injection or forgery when logs are viewed in terminals or processed by external systems.

Fixed in: 4.2.23

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-48432

[medium] GHSA-27jp-wm6q-gp25 — sqlparse@0.5.1

sqlparse: formatting list of tuples leads to denial of service

Fixed in: 0.5.4

[medium] GHSA-3496-9g83-7v6x (CVE-2026-59894) — sqlparse@0.5.1

sqlparse: Generated Python and PHP snippets allow SQL string breakout through unescaped backslashes

Fixed in: 0.6.0

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-59894

[medium] GHSA-7xr5-9hcq-chf9 (CVE-2025-48432) — django@5.1.2

Django Improper Output Neutralization for Logs vulnerability

Fixed in: 5.1.10

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-48432

[medium] GHSA-8498-2h75-472j (CVE-2024-53907) — django@5.1.2

Django denial-of-service in django.utils.html.strip_tags()

Fixed in: 5.1.4

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-53907

[medium] GHSA-8j24-cjrq-gr2m (CVE-2025-32873) — django@5.1.2

Django has a denial-of-service possibility in strip_tags()

Fixed in: 5.1.9

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-32873

[medium] GHSA-8qcx-xf44-272x (CVE-2026-53878) — django@5.1.2

Django: DomainNameValidator permits newline characters that may enable HTTP header injection

Fixed in: 5.2.16

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-53878

[medium] GHSA-cfqr-cjx5-5jcm (CVE-2026-84305) — sqlparse@0.5.1

sqlparse: Reindentation of tuple lists causes near-cap quadratic CPU consumption

Fixed in: 0.6.0

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-84305

[medium] GHSA-crhf-3pfg-w68w (CVE-2026-53877) — django@5.1.2

Django: GDALRaster may over-read heap memory when constructed from bytes

Fixed in: 5.2.16

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-53877

[medium] GHSA-p3fp-8748-vqfq (CVE-2025-26699) — django@5.1.2

Django Vulnerable to Allocation of Resources Without Limits or Throttling

Fixed in: 5.1.7

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-26699

[medium] GHSA-q238-5cxm-5c9h (CVE-2026-15830) — django@5.1.2

Django GeoDjango vulnerable to denial of service through deeply nested geometry collections

Fixed in: 5.2.17

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-15830

[medium] GHSA-qcgg-j2x8-h9g8 (CVE-2024-56374) — django@5.1.2

Django has a potential denial-of-service vulnerability in IPv6 validation

Fixed in: 5.1.5

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-56374

[medium] GHSA-rqw2-ghq9-44m7 (CVE-2025-13372) — django@5.1.2

Django is vulnerable to SQL injection in column aliases

Fixed in: 5.1.15

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-13372

[medium] GHSA-vrcr-9hj9-jcg6 (CVE-2025-64460) — django@5.1.2

Django is vulnerable to DoS via XML serializer text extraction

Fixed in: 5.1.15

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-64460

[medium] GHSA-wqfg-m96j-85vm (CVE-2025-27556) — django@5.1.2

Django Potential Denial of Service (DoS) on Windows

Fixed in: 5.1.8

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-27556

[low] GHSA-3h9f-r86x-qvjx (CVE-2026-48588) — django@5.1.2

Django: cache middleware may expose private responses when unrelated request cookies are present

Fixed in: 5.2.16

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-48588

[low] GHSA-8cjm-8mp7-r2xf (CVE-2026-8404) — django@5.1.2

Django: UpdateCacheMiddleware may disclose cached responses due to case-sensitive Cache-Control handling

Fixed in: 5.2.15

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-8404

[low] GHSA-923m-gv2p-w5qp (CVE-2026-48587) — django@5.1.2

Django: has_vary_header may expose cached responses when Vary values contain whitespace

Fixed in: 5.2.15

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-48587

[low] GHSA-h7pc-vwp9-298g (CVE-2026-6873) — django@5.1.2

Django: signed cookies are vulnerable to salt namespace collisions

Fixed in: 5.2.15

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-6873

[low] GHSA-q95w-c7qg-hrff (CVE-2025-59682) — django@5.1.2

Django vulnerable to partial directory traversal via archives

Fixed in: 5.1.13

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-59682

Findings from osv-scanner

[unknown] GHSA-27jp-wm6q-gp25 — sqlparse@0.5.1

sqlparse: formatting list of tuples leads to denial of service

[unknown] GHSA-3496-9g83-7v6x (CVE-2026-59894) — sqlparse@0.5.1

sqlparse: Generated Python and PHP snippets allow SQL string breakout through unescaped backslashes

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-59894

[unknown] GHSA-3h9f-r86x-qvjx (CVE-2026-48588) — django@5.1.2

Django: cache middleware may expose private responses when unrelated request cookies are present

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-48588

[unknown] GHSA-6w2r-r2m5-xq5w (CVE-2025-57833) — django@5.1.2

Django is subject to SQL injection through its column aliases

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-57833

[unknown] GHSA-7xr5-9hcq-chf9 (CVE-2025-48432) — django@5.1.2

Django Improper Output Neutralization for Logs vulnerability

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-48432

[unknown] GHSA-8498-2h75-472j (CVE-2024-53907) — django@5.1.2

Django denial-of-service in django.utils.html.strip_tags()

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-53907

[unknown] GHSA-8cjm-8mp7-r2xf (CVE-2026-8404) — django@5.1.2

Django: UpdateCacheMiddleware may disclose cached responses due to case-sensitive Cache-Control handling

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-8404

[unknown] GHSA-8j24-cjrq-gr2m (CVE-2025-32873) — django@5.1.2

Django has a denial-of-service possibility in strip_tags()

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-32873

[unknown] GHSA-8qcx-xf44-272x (CVE-2026-53878) — django@5.1.2

Django: DomainNameValidator permits newline characters that may enable HTTP header injection

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-53878

[unknown] GHSA-923m-gv2p-w5qp (CVE-2026-48587) — django@5.1.2

Django: has_vary_header may expose cached responses when Vary values contain whitespace

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-48587

[unknown] GHSA-cfqr-cjx5-5jcm (CVE-2026-84305) — sqlparse@0.5.1

sqlparse: Reindentation of tuple lists causes near-cap quadratic CPU consumption

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-84305

[unknown] GHSA-crhf-3pfg-w68w (CVE-2026-53877) — django@5.1.2

Django: GDALRaster may over-read heap memory when constructed from bytes

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-53877

[unknown] GHSA-f2ff-p2ww-7p4p (CVE-2026-71491) — sqlparse@0.5.1

sqlparse: Quadratic O(n²) DoS in group_comments

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-71491

[unknown] GHSA-frmv-pr5f-9mcr (CVE-2025-64459) — django@5.1.2

Django vulnerable to SQL injection via _connector keyword argument in QuerySet and Q objects.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-64459

[unknown] GHSA-h7pc-vwp9-298g (CVE-2026-6873) — django@5.1.2

Django: signed cookies are vulnerable to salt namespace collisions

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-6873

[unknown] GHSA-hpr9-3m2g-3j9p (CVE-2025-59681) — django@5.1.2

Django vulnerable to SQL injection in column aliases

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-59681

[unknown] GHSA-m9g8-fxxm-xg86 (CVE-2024-53908) — django@5.1.2

Django SQL injection in HasKey(lhs, rhs) on Oracle

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-53908

[unknown] GHSA-p3fp-8748-vqfq (CVE-2025-26699) — django@5.1.2

Django Vulnerable to Allocation of Resources Without Limits or Throttling

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-26699

[unknown] GHSA-prg7-hcfm-mfcr (CVE-2026-59893) — sqlparse@0.5.1

sqlparse: Inefficient Regex Handling of Dollar-Quoted SQL Literals Leads to ReDoS (Denial of Service)

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-59893

[unknown] GHSA-pwgv-4x5q-6m9f (CVE-2026-54284) — sqlparse@0.5.1

sqlparse: TokenList.__init__ materializes O(subtree) value per group, causing CPU DoS before depth/token caps trigger

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-54284

[unknown] GHSA-q238-5cxm-5c9h (CVE-2026-15830) — django@5.1.2

Django GeoDjango vulnerable to denial of service through deeply nested geometry collections

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-15830

[unknown] GHSA-q95w-c7qg-hrff (CVE-2025-59682) — django@5.1.2

Django vulnerable to partial directory traversal via archives

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-59682

[unknown] GHSA-qcgg-j2x8-h9g8 (CVE-2024-56374) — django@5.1.2

Django has a potential denial-of-service vulnerability in IPv6 validation

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-56374

[unknown] GHSA-qw25-v68c-qjf3 (CVE-2025-64458) — django@5.1.2

Django has a denial-of-service vulnerability in HttpResponseRedirect and HttpResponsePermanentRedirect on Windows

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-64458

[unknown] GHSA-rqw2-ghq9-44m7 (CVE-2025-13372) — django@5.1.2

Django is vulnerable to SQL injection in column aliases

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-13372

[unknown] GHSA-vrcr-9hj9-jcg6 (CVE-2025-64460) — django@5.1.2

Django is vulnerable to DoS via XML serializer text extraction

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-64460

[unknown] GHSA-wqfg-m96j-85vm (CVE-2025-27556) — django@5.1.2

Django Potential Denial of Service (DoS) on Windows

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-27556

[unknown] GHSA-wvqv-fj8w-qmhm (CVE-2026-15307) — django@5.1.2

Django GeoDjango spatial lookups allow file writes and outbound requests through GDAL raster parsing

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-15307

[unknown] PYSEC-2024-156 (CVE-2024-53907) — django@5.1.2

An issue was discovered in Django 5.1 before 5.1.4, 5.0 before 5.0.10, and 4.2 before 4.2.17. The strip_tags() method and striptags template filter are subject to a potential denial-of-service attack via certain inputs containing large sequences of nested incomplete HTML entities.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-53907

[unknown] PYSEC-2024-157 (CVE-2024-53908) — django@5.1.2

An issue was discovered in Django 5.1 before 5.1.4, 5.0 before 5.0.10, and 4.2 before 4.2.17. Direct usage of the django.db.models.fields.json.HasKey lookup, when an Oracle database is used, is subject to SQL injection if untrusted data is used as an lhs value. (Applications that use the jsonfield.has_key lookup via __ are unaffected.)

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-53908

[unknown] PYSEC-2025-1 (CVE-2024-56374) — django@5.1.2

An issue was discovered in Django 5.1 before 5.1.5, 5.0 before 5.0.11, and 4.2 before 4.2.18. Lack of upper-bound limit enforcement in strings passed when performing IPv6 validation could lead to a potential denial-of-service attack. The undocumented and private functions clean_ipv6_address and is_valid_ipv6_address are vulnerable, as is the django.forms.GenericIPAddressField form field. (The django.db.models.GenericIPAddressField model field is not affected.)

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-56374

[unknown] PYSEC-2025-104 (CVE-2025-13372) — django@5.1.2

An issue was discovered in 5.2 before 5.2.9, 5.1 before 5.1.15, and 4.2 before 4.2.27. `FilteredRelation` is subject to SQL injection in column aliases, using a suitably crafted dictionary, with dictionary expansion, as the `**kwargs` passed to `QuerySet.annotate()` or `QuerySet.alias()` on PostgreSQL. Earlier, unsupported Django series (such as 5.0.x, 4.1.x, and 3.2.x) were not evaluated and may also be affected. Django would like to thank Stackered for reporting this issue.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-13372

[unknown] PYSEC-2025-105 (CVE-2025-57833) — django@5.1.2

An issue was discovered in Django 4.2 before 4.2.24, 5.1 before 5.1.12, and 5.2 before 5.2.6. FilteredRelation is subject to SQL injection in column aliases, using a suitably crafted dictionary, with dictionary expansion, as the **kwargs passed QuerySet.annotate() or QuerySet.alias().

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-57833

[unknown] PYSEC-2025-106 (CVE-2025-59681) — django@5.1.2

An issue was discovered in Django 4.2 before 4.2.25, 5.1 before 5.1.13, and 5.2 before 5.2.7. QuerySet.annotate(), QuerySet.alias(), QuerySet.aggregate(), and QuerySet.extra() are subject to SQL injection in column aliases, when using a suitably crafted dictionary, with dictionary expansion, as the **kwargs passed to these methods (on MySQL and MariaDB).

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-59681

[unknown] PYSEC-2025-107 (CVE-2025-64458) — django@5.1.2

An issue was discovered in 5.1 before 5.1.14, 4.2 before 4.2.26, and 5.2 before 5.2.8. NFKC normalization in Python is slow on Windows. As a consequence, `django.http.HttpResponseRedirect`, `django.http.HttpResponsePermanentRedirect`, and the shortcut `django.shortcuts.redirect` were subject to a potential denial-of-service attack via certain inputs with a very large number of Unicode characters. Earlier, unsupported Django series (such as 5.0.x, 4.1.x, and 3.2.x) were not evaluated and may also be affected. Django would like to thank Seokchan Yoon for reporting this issue.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-64458

[unknown] PYSEC-2025-108 (CVE-2025-64459) — django@5.1.2

An issue was discovered in 5.1 before 5.1.14, 4.2 before 4.2.26, and 5.2 before 5.2.8. The methods `QuerySet.filter()`, `QuerySet.exclude()`, and `QuerySet.get()`, and the class `Q()`, are subject to SQL injection when using a suitably crafted dictionary, with dictionary expansion, as the `_connector` argument. Earlier, unsupported Django series (such as 5.0.x, 4.1.x, and 3.2.x) were not evaluated and may also be affected. Django would like to thank cyberstan for reporting this issue.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-64459

[unknown] PYSEC-2025-109 (CVE-2025-64460) — django@5.1.2

An issue was discovered in 5.2 before 5.2.9, 5.1 before 5.1.15, and 4.2 before 4.2.27. Algorithmic complexity in `django.core.serializers.xml_serializer.getInnerText()` allows a remote attacker to cause a potential denial-of-service attack triggering CPU and memory exhaustion via specially crafted XML input processed by the XML `Deserializer`. Earlier, unsupported Django series (such as 5.0.x, 4.1.x, and 3.2.x) were not evaluated and may also be affected. Django would like to thank Seokchan Yoon for reporting this issue.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-64460

[unknown] PYSEC-2025-13 (CVE-2025-26699) — django@5.1.2

An issue was discovered in Django 5.1 before 5.1.7, 5.0 before 5.0.13, and 4.2 before 4.2.20. The django.utils.text.wrap() method and wordwrap template filter are subject to a potential denial-of-service attack when used with very long strings.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-26699

[unknown] PYSEC-2025-14 (CVE-2025-27556) — django@5.1.2

An issue was discovered in Django 5.1 before 5.1.8 and 5.0 before 5.0.14. The NFKC normalization is slow on Windows. As a consequence, django.contrib.auth.views.LoginView, django.contrib.auth.views.LogoutView, and django.views.i18n.set_language are subject to a potential denial-of-service attack via certain inputs with a very large number of Unicode characters.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-27556

[unknown] PYSEC-2025-37 (CVE-2025-32873) — django@5.1.2

An issue was discovered in Django 4.2 before 4.2.21, 5.1 before 5.1.9, and 5.2 before 5.2.1. The django.utils.html.strip_tags() function is vulnerable to a potential denial-of-service (slow performance) when processing inputs containing large sequences of incomplete HTML tags. The template filter striptags is also vulnerable, because it is built on top of strip_tags().

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-32873

[unknown] PYSEC-2025-47 (CVE-2025-48432) — django@5.1.2

An issue was discovered in Django 5.2 before 5.2.2, 5.1 before 5.1.10, and 4.2 before 4.2.22. Internal HTTP response logging does not escape request.path, which allows remote attackers to potentially manipulate log output via crafted URLs. This may lead to log injection or forgery when logs are viewed in terminals or processed by external systems.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-48432

[unknown] PYSEC-2026-1296 (CVE-2025-59682) — django@5.1.2

Django vulnerable to partial directory traversal via archives

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-59682

[unknown] PYSEC-2026-3696 (CVE-2026-59894) — sqlparse@0.5.1

sqlparse: Generated Python and PHP snippets allow SQL string breakout through unescaped backslashes

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-59894

[unknown] PYSEC-2026-3697 (CVE-2026-71491) — sqlparse@0.5.1

sqlparse: Quadratic O(n²) DoS in group_comments

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-71491

[unknown] PYSEC-2026-3698 (CVE-2026-59893) — sqlparse@0.5.1

sqlparse: Inefficient Regex Handling of Dollar-Quoted SQL Literals Leads to ReDoS (Denial of Service)

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-59893

[unknown] PYSEC-2026-3699 (CVE-2026-54284) — sqlparse@0.5.1

sqlparse: TokenList.__init__ materializes O(subtree) value per group, causing CPU DoS before depth/token caps trigger

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-54284

[unknown] PYSEC-2026-3717 (CVE-2026-15830) — django@5.1.2

An issue was discovered in Django 5.2 before 5.2.17 and 6.0 before 6.0.8. GeoDjango's `django.contrib.gis.geos.GEOSGeometry` is subject to a potential denial-of-service when parsing deeply nested `GEOMETRYCOLLECTION` objects supplied as well-known text (WKT), well-known binary (WKB), or hex-encoded WKB, which triggers unbounded recursion and a segmentation fault in the underlying GEOS library. Spatial field lookups and the `django.contrib.gis.forms.GeometryField` form field are also affected. Earlier, unsupported Django series (such as 5.1.x, 5.0.x, and 4.2.x) were not evaluated and may also be affected. Django would like to thank Andrew MacPherson and kimchunbok_ for reporting this issue.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-15830

[unknown] PYSEC-2026-3923 (CVE-2026-84305) — sqlparse@0.5.1

sqlparse: Reindentation of tuple lists causes near-cap quadratic CPU consumption

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-84305

[unknown] PYSEC-2026-4035 (CVE-2026-15307) — django@5.1.2

Django GeoDjango spatial lookups allow file writes and outbound requests through GDAL raster parsing

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-15307

Findings from bomber

[critical] CVE-2025-64459 — pkg:pypi/django@5.1.2

Django vulnerable to SQL injection via _connector keyword argument in QuerySet and Q objects.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-64459

[high] CVE-2024-53908 — pkg:pypi/django@5.1.2

Django SQL injection in HasKey(lhs, rhs) on Oracle

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-53908

[high] CVE-2025-57833 — pkg:pypi/django@5.1.2

Django is subject to SQL injection through its column aliases

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-57833

[high] CVE-2025-59681 — pkg:pypi/django@5.1.2

Django vulnerable to SQL injection in column aliases

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-59681

[high] CVE-2025-64458 — pkg:pypi/django@5.1.2

Django has a denial-of-service vulnerability in HttpResponseRedirect and HttpResponsePermanentRedirect on Windows

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-64458

[high] CVE-2026-15307 — pkg:pypi/django@5.1.2

Django GeoDjango spatial lookups allow file writes and outbound requests through GDAL raster parsing

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-15307

[high] CVE-2026-54284 — pkg:pypi/sqlparse@0.5.1

sqlparse: TokenList.__init__ materializes O(subtree) value per group, causing CPU DoS before depth/token caps trigger

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-54284

[high] CVE-2026-59893 — pkg:pypi/sqlparse@0.5.1

sqlparse: Inefficient Regex Handling of Dollar-Quoted SQL Literals Leads to ReDoS (Denial of Service)

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-59893

[high] CVE-2026-71491 — pkg:pypi/sqlparse@0.5.1

sqlparse: Quadratic O(n²) DoS in group_comments

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-71491

[medium] CVE-2024-53907 — pkg:pypi/django@5.1.2

Django denial-of-service in django.utils.html.strip_tags()

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-53907

[medium] CVE-2024-56374 — pkg:pypi/django@5.1.2

Django has a potential denial-of-service vulnerability in IPv6 validation

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-56374

[medium] CVE-2025-13372 — pkg:pypi/django@5.1.2

Django is vulnerable to SQL injection in column aliases

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-13372

[medium] CVE-2025-26699 — pkg:pypi/django@5.1.2

Django Vulnerable to Allocation of Resources Without Limits or Throttling

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-26699

[medium] CVE-2025-27556 — pkg:pypi/django@5.1.2

Django Potential Denial of Service (DoS) on Windows

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-27556

[medium] CVE-2025-32873 — pkg:pypi/django@5.1.2

Django has a denial-of-service possibility in strip_tags()

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-32873

[medium] CVE-2025-48432 — pkg:pypi/django@5.1.2

Django Improper Output Neutralization for Logs vulnerability

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-48432

[medium] CVE-2025-64460 — pkg:pypi/django@5.1.2

Django is vulnerable to DoS via XML serializer text extraction

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-64460

[medium] CVE-2026-15830 — pkg:pypi/django@5.1.2

Django GeoDjango vulnerable to denial of service through deeply nested geometry collections

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-15830

[medium] CVE-2026-53877 — pkg:pypi/django@5.1.2

Django: GDALRaster may over-read heap memory when constructed from bytes

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-53877

[medium] CVE-2026-53878 — pkg:pypi/django@5.1.2

Django: DomainNameValidator permits newline characters that may enable HTTP header injection

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-53878

[medium] CVE-2026-59894 — pkg:pypi/sqlparse@0.5.1

sqlparse: Generated Python and PHP snippets allow SQL string breakout through unescaped backslashes

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-59894

[medium] CVE-2026-84305 — pkg:pypi/sqlparse@0.5.1

sqlparse: Reindentation of tuple lists causes near-cap quadratic CPU consumption

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-84305

[medium] GHSA-27jp-wm6q-gp25 — pkg:pypi/sqlparse@0.5.1

sqlparse: formatting list of tuples leads to denial of service

[low] CVE-2025-59682 — pkg:pypi/django@5.1.2

Django vulnerable to partial directory traversal via archives

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-59682

[low] CVE-2026-48587 — pkg:pypi/django@5.1.2

Django: has_vary_header may expose cached responses when Vary values contain whitespace

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-48587

[low] CVE-2026-48588 — pkg:pypi/django@5.1.2

Django: cache middleware may expose private responses when unrelated request cookies are present

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-48588

[low] CVE-2026-6873 — pkg:pypi/django@5.1.2

Django: signed cookies are vulnerable to salt namespace collisions

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-6873

[low] CVE-2026-8404 — pkg:pypi/django@5.1.2

Django: UpdateCacheMiddleware may disclose cached responses due to case-sensitive Cache-Control handling

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-8404

[unknown] CVE-2024-53907 — pkg:pypi/django@5.1.2

An issue was discovered in Django 5.1 before 5.1.4, 5.0 before 5.0.10, and 4.2 before 4.2.17. The strip_tags() method and striptags template filter are subject to a potential denial-of-service attack via certain inputs containing large sequences of nested incomplete HTML entities.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-53907

[unknown] CVE-2024-53908 — pkg:pypi/django@5.1.2

An issue was discovered in Django 5.1 before 5.1.4, 5.0 before 5.0.10, and 4.2 before 4.2.17. Direct usage of the django.db.models.fields.json.HasKey lookup, when an Oracle database is used, is subject to SQL injection if untrusted data is used as an lhs value. (Applications that use the jsonfield.has_key lookup via __ are unaffected.)

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-53908

[unknown] CVE-2024-56374 — pkg:pypi/django@5.1.2

An issue was discovered in Django 5.1 before 5.1.5, 5.0 before 5.0.11, and 4.2 before 4.2.18. Lack of upper-bound limit enforcement in strings passed when performing IPv6 validation could lead to a potential denial-of-service attack. The undocumented and private functions clean_ipv6_address and is_valid_ipv6_address are vulnerable, as is the django.forms.GenericIPAddressField form field. (The django.db.models.GenericIPAddressField model field is not affected.)

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2024-56374

[unknown] CVE-2025-13372 — pkg:pypi/django@5.1.2

An issue was discovered in 5.2 before 5.2.9, 5.1 before 5.1.15, and 4.2 before 4.2.27. `FilteredRelation` is subject to SQL injection in column aliases, using a suitably crafted dictionary, with dictionary expansion, as the `**kwargs` passed to `QuerySet.annotate()` or `QuerySet.alias()` on PostgreSQL. Earlier, unsupported Django series (such as 5.0.x, 4.1.x, and 3.2.x) were not evaluated and may also be affected. Django would like to thank Stackered for reporting this issue.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-13372

[unknown] CVE-2025-26699 — pkg:pypi/django@5.1.2

An issue was discovered in Django 5.1 before 5.1.7, 5.0 before 5.0.13, and 4.2 before 4.2.20. The django.utils.text.wrap() method and wordwrap template filter are subject to a potential denial-of-service attack when used with very long strings.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-26699

[unknown] CVE-2025-27556 — pkg:pypi/django@5.1.2

An issue was discovered in Django 5.1 before 5.1.8 and 5.0 before 5.0.14. The NFKC normalization is slow on Windows. As a consequence, django.contrib.auth.views.LoginView, django.contrib.auth.views.LogoutView, and django.views.i18n.set_language are subject to a potential denial-of-service attack via certain inputs with a very large number of Unicode characters.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-27556

[unknown] CVE-2025-32873 — pkg:pypi/django@5.1.2

An issue was discovered in Django 4.2 before 4.2.21, 5.1 before 5.1.9, and 5.2 before 5.2.1. The django.utils.html.strip_tags() function is vulnerable to a potential denial-of-service (slow performance)

when processing inputs containing large sequences of incomplete HTML tags. The template filter striptags is also vulnerable, because it is built on top of strip_tags().

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-32873

[unknown] CVE-2025-48432 — pkg:pypi/django@5.1.2

An issue was discovered in Django 5.2 before 5.2.2, 5.1 before 5.1.10, and 4.2 before 4.2.22. Internal HTTP response logging does not escape request.path, which allows remote attackers to potentially manipulate log output via crafted URLs. This may lead to log injection or forgery when logs are viewed in terminals or processed by external systems.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-48432

[unknown] CVE-2025-57833 — pkg:pypi/django@5.1.2

An issue was discovered in Django 4.2 before 4.2.24, 5.1 before 5.1.12, and 5.2 before 5.2.6. FilteredRelation is subject to SQL injection in column aliases, using a suitably crafted dictionary, with dictionary expansion, as the **kwargs passed QuerySet.annotate() or QuerySet.alias().

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-57833

[unknown] CVE-2025-59681 — pkg:pypi/django@5.1.2

An issue was discovered in Django 4.2 before 4.2.25, 5.1 before 5.1.13, and 5.2 before 5.2.7. QuerySet.annotate(), QuerySet.alias(), QuerySet.aggregate(), and QuerySet.extra() are subject to SQL injection in column aliases, when using a suitably crafted dictionary, with dictionary expansion, as the **kwargs passed to these methods (on MySQL and MariaDB).

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-59681

[unknown] CVE-2025-59682 — pkg:pypi/django@5.1.2

Django vulnerable to partial directory traversal via archives

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-59682

[unknown] CVE-2025-64458 — pkg:pypi/django@5.1.2

An issue was discovered in 5.1 before 5.1.14, 4.2 before 4.2.26, and 5.2 before 5.2.8. NFKC normalization in Python is slow on Windows. As a consequence, `django.http.HttpResponseRedirect`, `django.http.HttpResponsePermanentRedirect`, and the shortcut `django.shortcuts.redirect` were subject to a potential denial-of-service attack via certain inputs with a very large number of Unicode characters. Earlier, unsupported Django series (such as 5.0.x, 4.1.x, and 3.2.x) were not evaluated and may also be affected. Django would like to thank Seokchan Yoon for reporting this issue.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-64458

[unknown] CVE-2025-64459 — pkg:pypi/django@5.1.2

An issue was discovered in 5.1 before 5.1.14, 4.2 before 4.2.26, and 5.2 before 5.2.8. The methods `QuerySet.filter()`, `QuerySet.exclude()`, and `QuerySet.get()`, and the class `Q()`, are subject to SQL injection when using a suitably crafted dictionary, with dictionary expansion, as the `_connector` argument. Earlier, unsupported Django series (such as 5.0.x, 4.1.x, and 3.2.x) were not evaluated and may also be affected. Django would like to thank cyberstan for reporting this issue.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-64459

[unknown] CVE-2025-64460 — pkg:pypi/django@5.1.2

An issue was discovered in 5.2 before 5.2.9, 5.1 before 5.1.15, and 4.2 before 4.2.27. Algorithmic complexity in `django.core.serializers.xml_serializer.getInnerText()` allows a remote attacker to cause a potential denial-of-service attack triggering CPU and memory exhaustion via specially crafted XML input processed by the XML `Deserializer`. Earlier, unsupported Django series (such as 5.0.x, 4.1.x, and 3.2.x)

were not evaluated and may also be affected. Django would like to thank Seokchan Yoon for reporting this issue.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2025-64460

[unknown] CVE-2026-15307 — pkg:pypi/django@5.1.2

Django GeoDjango spatial lookups allow file writes and outbound requests through GDAL raster parsing

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-15307

[unknown] CVE-2026-15830 — pkg:pypi/django@5.1.2

An issue was discovered in Django 5.2 before 5.2.17 and 6.0 before 6.0.8. GeoDjango's `django.contrib.gis.geos.GEOSGeometry` is subject to a potential denial-of-service when parsing deeply nested `GEOMETRYCOLLECTION` objects supplied as well-known text (WKT), well-known binary (WKB), or hex-encoded WKB, which triggers unbounded recursion and a segmentation fault in the underlying GEOS library. Spatial field lookups and the `django.contrib.gis.forms.GeometryField` form field are also affected. Earlier, unsupported Django series (such as 5.1.x, 5.0.x, and 4.2.x) were not evaluated and may also be affected. Django would like to thank Andrew MacPherson and kimchunbok_ for reporting this issue.

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-15830

[unknown] CVE-2026-54284 — pkg:pypi/sqlparse@0.5.1

sqlparse: TokenList.__init__ materializes O(subtree) value per group, causing CPU DoS before depth/token caps trigger

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-54284

[unknown] CVE-2026-59893 — pkg:pypi/sqlparse@0.5.1

sqlparse: Inefficient Regex Handling of Dollar-Quoted SQL Literals Leads to ReDoS (Denial of Service)

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-59893

[unknown] CVE-2026-59894 — pkg:pypi/sqlparse@0.5.1

sqlparse: Generated Python and PHP snippets allow SQL string breakout through unescaped backslashes

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-59894

[unknown] CVE-2026-71491 — pkg:pypi/sqlparse@0.5.1

sqlparse: Quadratic O(n²) DoS in group_comments

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-71491

[unknown] CVE-2026-84305 — pkg:pypi/sqlparse@0.5.1

sqlparse: Reindentation of tuple lists causes near-cap quadratic CPU consumption

Source: https://vulnlookup.ndaal.eu:443/api/v1/vulnerability/CVE-2026-84305

Skipped or failed analyzers

trivy (fail): 2026-11-12T12:37:37Z FATAL Fatal error run error: sbom scan error: scan error: scan failed: failed analysis: SBOM decode error: cyclonedx-xml scanning is not yet supported

bomdrift (fail): Error: parsing JSON in: /tmp/sba-analyze-694-8534-1794487055544870034-104/input.cdx.xml Caused by: expected value at line 1 column 1

inspektr_cli (fail): Error: Failed to scan for vulnerabilities Caused by: 0: SBOM format error: Failed to decode SBOM: expected value at line 1 column 1 1: Failed to decode SBOM: expected value at line 1

column 1

shieldbom (fail): Parsed 3 components from "/tmp/sba-analyze-694-8534-1794487055544870034-104/input.cdx.xml" (CycloneDX 1.4 (XML)) Found 0 vulnerabilities Found 4 license issues DISCLAIMER: This report is provided "AS IS" without warranty of any kind. Vulnerability results are based on publicly available data sources (e.g., OSV.dev, NVD) which may be incomplete or delayed. The absence of reported vulnerabilities does not guarantee that the software is free of security issues. This tool assists with security analysis but does not constitute a complete security assessment, legal advice, or certification of regulatory compliance (including EU CRA conformity). Users are solely responsible for their own compliance determinations. Always perform additional security assessments as appropriate.