

django-base-site

github.com/epicserve/django-base-site

Scanned 24 September 2026 at 09:52 · Report generated 29 October 2026 · Python · 452 stars

OVERALL ASSESSMENT

High

Combined score 192 · 128 findings after filtering

Summary

This repository shows two distinct levels of security issues that need immediate attention before it can be used in production. The most critical finding is a high vulnerability risk score of 160, which indicates the code contains dangerous patterns like running command outputs directly into shells and missing CSRF protection, making it highly exploitable by attackers. While there are also medium-level concerns about potential malicious intent, these appear to stem from common insecure coding practices rather than intentional attacks. The developer should prioritize fixing the high-risk issues first, such as securing shell commands and enabling HTML autoescaping, before considering the repository safe for use.

Written by a language model running locally on the scanning machine, from the findings in this report. The risk levels above and below were decided by the scoring model, not by this summary.

Risk, by type

MalDet scores two different questions separately, because they call for different responses: code can be badly written without being hostile, and hostile code is not always badly written.

VULNERABILITY

High (160)

Insecure coding an attacker could exploit. Usually mistakes, and usually fixable in place.

MALICIOUS PATTERN

Medium (32)

Signs the code may be trying to do harm: hidden payloads, credential access, persistence, exfiltration.

What each detector found

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
semgrep	Insecure coding patterns across many languages	3	20	55	78

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
dep_checker	Vulnerable, unpinned or typosquatted dependencies	0	18	11	29
bandit	Insecure coding patterns in Python source	0	8	9	17
yara	Signatures of malicious behaviour in source files	3	0	0	3
guarddog	Malicious package and install-time behaviour	0	1	0	1

106 of 128 findings were rated more likely genuine than noise by the classifier.

Evidence

Identical findings are grouped, so a single rule firing many times takes one row. Ordered worst first. "Seen" is how many times that finding occurred.

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	A `run:` step pipes the output of `curl` or `wget` directly into a shell interpreter. This is the "curl bash" install pattern — if the remote server is compromised or the URL is hijacked, an attacker can execute arbitrary code in your CI runner. Consider downloading the file first, verifying its c	3	/.github/workflows/ci.yml:141
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	1	/.github/workflows/ci.yml
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	2	/config/bash/django_bash_completion
MEDIUM	semgrep	'mark_safe()' is used to mark a string as "safe" for HTML output. This disables escaping and could therefore subject the content to XSS attacks. Use 'django.utils.html.format_html()' to build HTML for rendering instead.	4	/apps/base/templatetags/vite.py:59

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	This Dependabot configuration does not set a cooldown period. Newly published packages can be malicious or unstable. Add a `cooldown` block with `default-days: 7` to each `package-ecosystem` entry under `updates` to wait 7 days before proposing updates to newly published package versions. Reference:	4	<code>/.github/dependabot.yml:9</code>
MEDIUM	semgrep	Detected Flask route directly returning a formatted string. This is subject to cross-site scripting if user input can reach the string. Consider using the template engine instead and rendering pages with <code>'render_template()'</code> .	1	<code>/apps/base/ninja_pagination.py:75</code>
MEDIUM	bandit	Potential XSS on <code>mark_safe</code> function.	4	<code>/apps/base/templatetags/accounts.py:21</code>
MEDIUM	bandit	Use of <code>mark_safe()</code> may expose cross-site scripting vulnerabilities and should be reviewed.	4	<code>/apps/base/templatetags/vite.py:59</code>
MEDIUM	semgrep	Detected a template block where autoescaping is explicitly disabled with <code>'{% autoescape off %}'</code> . This allows rendering of raw HTML in this segment. Turn autoescaping on to prevent cross-site scripting (XSS). If you must do this, consider instead, using <code>`mark_safe`</code> in Python code.	4	<code>/apps/accounts/templates/account/email/account_already_exists_message.html:18</code>
MEDIUM	guarddog	Detects access to sensitive files (credentials, configs, keys) [threat-filesystem-read]	1	<code>/config/settings/_base.py:26</code>
MEDIUM	semgrep	Detected usage of <code>@csrf_exempt</code> , which indicates that there is no CSRF token set for this route. This could lead to an attacker manipulating the user's account and exfiltration of private data. Instead, create a function without this decorator.	1	<code>/apps/billing/webhooks.py:144</code>
MEDIUM	dep_checker	Unpinned dependency: 'django' has no version lock — supply chain risk	1	<code>pyproject.toml</code>

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	dep_checker	Unpinned dependency: 'celery' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'boto3' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'fido2' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'hiredis' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'pillow' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'django-storages' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'django-ninja' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'django-hijack' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'django-maintenance-mode' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'epicenv' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'psycpg2-binary' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'python-json-logger' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'django-allauth' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'django-alive' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'whitenoise' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'django-ses' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'gunicorn' has no version lock — supply chain risk	1	pyproject.to ml

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	GitHub Actions step uses a mutable tag or branch reference. Tags and branch names can be silently repointed by the action owner, enabling supply-chain attacks — as seen in the trivy-action and kics-github-action compromises. Pin the reference to a full 40-character commit SHA instead, e.g. `uses: ac	6	/.github/workflows/ci.yml:22
LOW	semgrep	Translated strings will not be escaped when rendered in a template. This leads to a vulnerability where translators could include malicious script tags in their translations. Consider using `force_escape` to explicitly escape a translated text.	55	/apps/accounts/templates/account/email/password_changed_message.html:6
LOW	bandit	Possible hardcoded password: '/accounts/password/reset/key/{key}'	1	/config/settings/_base.py:381
LOW	bandit	Consider possible security implications associated with the subprocess module.	2	/.agents/skills/weekly-maintenance/scripts/merge_dependabot.py:30
LOW	bandit	Starting a process with a partial executable path	2	/.agents/skills/weekly-maintenance/scripts/merge_dependabot.py:61
LOW	bandit	subprocess call - check for execution of untrusted input.	2	/.agents/skills/weekly-maintenance/scripts/dependabot_report.py:37
LOW	dep_checker	Loose version constraint: 'redis>=7.1,<9.0' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: '@heroicons/vue^2.2.0' — attacker could push malicious update	1	package.json

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: '@tailwindcss/vite^4.3.3' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@vitejs/plugin-vue^6.0.9' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'reka-ui^2.10.4' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'stripe>=14,<16' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'tailwindcss^4.3.3' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'vite^8.3.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'vue-advanced-cropper^2.8.9' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'vue-router^5.3.1' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'vue^3.5.42' — attacker could push malicious update	1	package.json
LOW	bandit	Use of assert detected. The enclosed code will be removed when compiling to optimised byte code.	2	/apps/organizations/ middleware.py:17

How to read this report

- **This is a static scan.** The code was read, never executed. Behaviour that only appears at runtime is outside what any of these detectors can see.
- **Findings are not all equally real.** A classifier rates each finding on how likely it is to be genuine rather than noise, and that rating weights the scores. Measured against a hand-reviewed sample it reached precision 0.52 and recall 0.62 — useful for ranking, not a verdict on any single finding.
- **A high score is not proof of malice.** Security tools, penetration-testing frameworks and malware research repositories contain the same patterns as real malware by design. The two risk types above exist to keep that distinction visible.

- **Absence of findings is not proof of safety.** A detector that is not installed, or a language none of them parse, produces silence rather than a warning.
- **This scan is a snapshot.** The repository may have changed since 24 September 2026.