CodeAssistant 报告

（由自动审查与测试生成系统输出）

Contents

1 代码审查（Review）

1.1 概览

- 问题总数：28
- 高/中/低：28 / 0 / 0
- 工具数：1
- DS 规则命中总数：0

1.2 严重性分布

- high：28

1.3 工具分布

- pip-audit：28

1.4 Top 20 问题

1. **[high] pip-audit VULN**
   说明：None None -> PYSEC-2021-421 Babel.Locale in Babel before 2.9.1 allows attackers to load arbitrary locale .dat files (containing serialized Python objects) via directory traversal, leading to code execution.

2. **[high] pip-audit VULN**
   说明：None None -> PYSEC-2022-42986 Certifi is a curated collection of Root Certificates for validating the trustworthiness of SSL certificates while verifying the identity of TLS hosts. Certifi 2022.12.07 removes root certificates from ”TrustCor” from the root store. These are in

the process of being removed from Mozilla's trust store. TrustCor's root certificates are being removed pursuant to an investigation prompted by media reporting that TrustCor's ownership also operated a business that produced spyware. Conclusions of Mozilla's investigation can be found in the linked google group discussion.

3. **[high] pip-audit VULN**
   说明：None None -> PYSEC-2023-135 Certifi 2023.07.22 removes root certificates from "e-Tugra" from the root store. These are in the process of being removed from Mozilla's trust store. e-Tugra's root certificates are being removed pursuant to an investigation prompted by reporting of security issues in their systems.

4. **[high] pip-audit VULN**
   说明：None None -> PYSEC-2024-60 A vulnerability was identified in the kjd/idna library, specifically within the 'idna.encode()' function, affecting version 3.6. The issue arises from the function's handling of crafted input strings, which can lead to quadratic complexity and consequently, a denial of service condition. This vulnerability is triggered by a crafted input that causes the 'idna.encode()' function to process the input with considerable computational load, significantly increasing the processing time in a quadratic manner relative to the input size.

5. **[high] pip-audit VULN**
   说明：None None -> PYSEC-2021-66 This affects the package jinja2 from 0.0.0 and before 2.11.3. The ReDoS vulnerability is mainly due to the '_punctuation_re regex' operator and its use of multiple wildcards. The last wildcard is the most exploitable as it searches for trailing punctuation. This issue can be mitigated by Markdown to format user content instead of the urlize filter, or by implementing request timeouts and limiting process memory.

6. **[high] pip-audit VULN**
   说明：None None -> PYSEC-2019-217 In Pallets Jinja before 2.10.1, str.format_map allows a sandbox escape.

7. **[high] pip-audit VULN**
   说明：None None -> CVE-2024-22195 The 'xmlattr' filter in affected versions of Jinja accepts keys containing spaces. XML/HTML attributes cannot contain spaces, as each would then be interpreted as a separate attribute. If an application accepts keys (as opposed to only values) as user input, and renders these in pages that other users see as well, an attacker could use this to inject other attributes and perform XSS. Note that accepting keys as user input is not common or a particularly intended use case of the 'xmlattr' filter, and an application doing so should already be verifying what keys are provided regardless of this fix.

8. **[high] pip-audit VULN**
   说明：None None -> CVE-2024-34064 The 'xmlattr' filter in affected versions of Jinja accepts keys containing non-attribute characters. XML/HTML attributes cannot contain spaces, '/', '>', or '=', as each would then be interpreted as starting a separate attribute. If an application accepts keys (as opposed to only values) as user input, and renders these in pages that other users see as well, an attacker could use this to inject other attributes and perform XSS. The fix for the previous GHSA-h5c8-rqwp-cp95 CVE-2024-22195 only addressed spaces but not other characters. Accepting keys as user input is now explicitly considered an unintended use case of the 'xmlattr' filter, and code that does so without otherwise validating the input should be

flagged as insecure, regardless of Jinja version. Accepting _values_ as user input continues to be safe.

9. **[high] pip-audit VULN**
   说明：None None -> CVE-2024-56326 An oversight in how the Jinja sandboxed environment detects calls to ‘str.format‘ allows an attacker that controls the content of a template to execute arbitrary Python code. To exploit the vulnerability, an attacker needs to control the content of a template. Whether that is the case depends on the type of application using Jinja. This vulnerability impacts users of applications which execute untrusted templates. Jinja’s sandbox does catch calls to ‘str.format‘ and ensures they don’t escape the sandbox. However, it’s possible to store a reference to a malicious string’s ‘format‘ method, then pass that to a filter that calls it. No such filters are built-in to Jinja, but could be present through custom filters in an application. After the fix, such indirect calls are also handled by the sandbox.

10. **[high] pip-audit VULN**
    说明：None None -> CVE-2025-27516 An oversight in how the Jinja sandboxed environment interacts with the ‘|attr‘ filter allows an attacker that controls the content of a template to execute arbitrary Python code. To exploit the vulnerability, an attacker needs to control the content of a template. Whether that is the case depends on the type of application using Jinja. This vulnerability impacts users of applications which execute untrusted templates. Jinja’s sandbox does catch calls to ‘str.format‘ and ensures they don’t escape the sandbox. However, it’s possible to use the ‘|attr‘ filter to get a reference to a string’s plain format method, bypassing the sandbox. After the fix, the ‘|attr‘ filter no longer bypasses the environment’s attribute lookup.

11. **[high] pip-audit VULN**
    说明：None None -> PYSEC-2021-140 An infinite loop in SMLLexer in Pygments versions 1.5 to 2.7.3 may lead to denial of service when performing syntax highlighting of a Standard ML (SML) source file, as demonstrated by input that only contains the ”exception” keyword.

12. **[high] pip-audit VULN**
    说明：None None -> PYSEC-2021-141 In pygments 1.1+, fixed in 2.7.4, the lexers used to parse programming languages rely heavily on regular expressions. Some of the regular expressions have exponential or cubic worst-case complexity and are vulnerable to ReDoS. By crafting malicious input, an attacker can cause a denial of service.

13. **[high] pip-audit VULN**
    说明：None None -> PYSEC-2023-117 A ReDoS issue was discovered in pygments/lexers/smithy.py in pygments through 2.15.0 via SmithyLexer.

14. **[high] pip-audit VULN**
    说明：None None -> PYSEC-2018-28 The Requests package before 2.20.0 for Python sends an HTTP Authorization header to an http URI upon receiving a same-hostname https-to-http redirect, which makes it easier for remote attackers to discover credentials by sniffing the network.

15. **[high] pip-audit VULN**
    说明：None None -> PYSEC-2023-74 Requests is a HTTP library. Since Requests 2.3.0, Requests has been leaking Proxy-Authorization headers to destination servers when redirected to an HTTPS endpoint. This is a product of how we use ‘rebuild_proxies‘ to reattach the

‘Proxy-Authorization‘ header to requests. For HTTP connections sent through the tunnel, the proxy will identify the header in the request itself and remove it prior to forwarding to the destination server. However when sent over HTTPS, the ‘Proxy-Authorization‘ header must be sent in the CONNECT request as the proxy has no visibility into the tunneled request. This results in Requests forwarding proxy credentials to the destination server unintentionally, allowing a malicious actor to potentially exfiltrate sensitive information. This issue has been patched in version 2.31.0.

16. **[high] pip-audit VULN**
    说明：None None -> CVE-2024-35195 When making requests through a Requests ‘Session‘, if the first request is made with ‘verify=False‘ to disable cert verification, all subsequent requests to the same origin will continue to ignore cert verification regardless of changes to the value of ‘verify‘. This behavior will continue for the lifecycle of the connection in the connection pool. ### Remediation Any of these options can be used to remediate the current issue, we highly recommend upgrading as the preferred mitigation. * Upgrade to ‘requests>=2.32.0‘. * For ‘requests<2.32.0‘, avoid setting ‘verify=False‘ for the first request to a host while using a Requests Session. * For ‘requests<2.32.0‘, call ‘close()‘ on ‘Session‘ objects to clear existing connections if ‘verify=False‘ is used. ### Related Links * https://github.com/psf/requests/pull/6655

17. **[high] pip-audit VULN**
    说明：None None -> CVE-2024-47081 ### Impact Due to a URL parsing issue, Requests releases prior to 2.32.4 may leak .netrc credentials to third parties for specific maliciously-crafted URLs. ### Workarounds For older versions of Requests, use of the .netrc file can be disabled with ‘trust_env=False‘ on your Requests Session ([docs](https://requests.readthedocs.io/en/latest/api/#re
    ### References https://github.com/psf/requests/pull/6965 https://seclists.org/fulldisclosure/2025/Jun/2

18. **[high] pip-audit VULN**
    说明：None None -> PYSEC-2021-108 An issue was discovered in urllib3 before 1.26.5. When provided with a URL containing many @ characters in the authority component, the authority regular expression exhibits catastrophic backtracking, causing a denial of service if a URL were passed as a parameter or redirected to via an HTTP redirect.

19. **[high] pip-audit VULN**
    说明：None None -> PYSEC-2019-133 The urllib3 library before 1.24.2 for Python mishandles certain cases where the desired set of CA certificates is different from the OS store of CA certificates, which results in SSL connections succeeding in situations where a verification failure is the correct outcome. This is related to use of the ssl_context, ca_certs, or ca_certs_dir argument.

20. **[high] pip-audit VULN**
    说明：None None -> PYSEC-2019-132 In the urllib3 library through 1.24.1 for Python, CRLF injection is possible if the attacker controls the request parameter.

1.5 复杂度摘要

- 来源：Radon Cyclomatic Complexity（CC）
- 说明：等级通常为 A（简单）到 F（复杂），括号内为复杂度分数

文件	类型	符号（函数/方法）	等级	分数
D:/code_assistant/Git_repo/realpython__python-guide/test_issues.py	F	process_list	A	3
D:/code_assistant/Git_repo/realpython__python-guide/test_issues.py	F	append_to_list	A	1
D:/code_assistant/Git_repo/realpython__python-guide/test_issues.py	F	add_to_dict	A	1
D:/code_assistant/Git_repo/realpython__python-guide/test_issues.py	F	increment_counter	A	1
D:/code_assistant/Git_repo/realpython__python-guide/test_issues.py	F	read_file_bad	A	1
D:/code_assistant/Git_repo/realpython__python-guide/test_issues.py	F	read_file_good	A	1
D:/code_assistant/Git_repo/realpython__python-guide/docs/_themes/flask_theme_support.py	C	FlaskyStyle	A	1

2 测试生成（TestGen）

2.1 指标

- 写入测试文件数：0
- 覆盖函数数：0
- 输出目录：D:/code_assistant/Git_repo/realpython__python-guide/reports/realpython__python-guide/generated_tests

2.2 覆盖率报告

Name	Stmts	Miss	Cover	Missing
N/A	0	0	0%	