

MyESI Automated Compliance & Vulnerability Assessment

Project: myesi-user-service

Generated: 2025-11-12 10:35 UTC **User:** 1

1. Executive Summary

- Compliance Score: **60.0%**
- Average Risk Score: **9.50**
- Total Vulnerabilities: **8**

Standards Compliance:

Standard	Status
ISO_27001	Partially
NIST_SP_800_53	Needs Review
OWASP	Needs Review

2. Compliance Controls Overview

critical — (General) — Score: **50.0%**

- 1 Plan fix in next release cycle. Limit exposure through configuration hardening.
- 2 Validate compliance and re-run assessment.

moderate — (General) — Score: **50.0%**

- 1 Plan fix in next release cycle. Limit exposure through configuration hardening.
- 2 Validate compliance and re-run assessment.

3. Vulnerability Findings

#	Component	Version	Vuln ID	Severity	Fix / Recommendation
1	bcrypt	4.3.0	GHSA-5wg4-74h6-q47v	Moderate	Update to: 5.0.0
2	loguru	0.7.3	MAL-2025-25559	Critical	### Summary This control requires attention in General. ### Remediation Steps - Immediately patch or isolate affected components. Deploy emergency fixes and block exploit paths. - Validate compliance and re-run assessment.
3	mypy-extensions	1.1.0	MAL-2024-2685	Critical	### Summary This control requires attention in General. ### Remediation Steps - Immediately patch or isolate affected components. Deploy emergency fixes and block exploit paths. - Validate compliance and re-run assessment.
4	pydantic	2.12.2	MAL-2025-4867	Critical	### Summary This control requires attention in General. ### Remediation Steps - Immediately patch or isolate affected components. Deploy emergency fixes and block exploit paths. - Validate compliance and re-run assessment.
5	pyjwt	2.10.1	MAL-2025-48036	Critical	### Summary This control requires attention in General. ### Remediation Steps - Immediately patch or isolate affected components. Deploy emergency fixes and block exploit paths. - Validate compliance and re-run assessment.
6	python-dotenv	1.1.1	MAL-2025-48037	Critical	### Summary This control requires attention in General. ### Remediation Steps - Immediately patch or isolate affected components. Deploy emergency fixes and block exploit paths. - Validate compliance and re-run assessment.
7	typing-extensions	4.15.0	MAL-2025-47895	Critical	### Summary This control requires attention in General. ### Remediation Steps - Immediately patch or isolate affected components. Deploy emergency fixes and block exploit paths. - Validate compliance and re-run assessment.

8	uvicorn	0.37.0	MAL-2025-4901	Critical	<p>### Summary This control requires attention in General. ###</p> <p>Remediation Steps - Immediately patch or isolate affected components. Deploy emergency fixes and block exploit paths. - Validate compliance and re-run assessment.</p>
---	----------------	--------	---------------	-----------------	--

4. Code Findings (Static Analysis)

1. python.fastapi.web.fastapi-cookie-samesite-none.fastapi-cookie-samesite-none (Info / HIGH)

/app/tmp/repos/semgrep-20251112094210/app/api/v1/users.py

Detected a cookie options with the `SameSite` flag set to "None". This is a potential security risk that arises from the way web browsers manage cookies. In a typical web application, cookies are used to store and transmit session-related data between a client and a server. To enhance security, cookies can be marked with the "SameSite" attribute, which restricts their usage based on the origin of the page that set them. This attribute can have three values: "Strict," "Lax," or "None". Make sure that the choice of the `None` value is intentional and that you understand the potential security implications. In FastAPI apps, the `set_cookie` function's argument `samesite` is set to 'Lax' by default. While 'Strict' is the most secure option, 'Lax' is a good compromise between security and usability and this default value is secure for most applications. Do not set `samesite` to 'None' to turn off this security feature.

samesite="none",

https://owasp.org/Top10/A01_2021-Broken_Access_Control

<https://web.dev/articles/samesite-cookies-explained>

<https://www.starlette.io/responses/>

Suggested Fix:

Summary This control requires attention in security. Remediation Steps Investigate severity and verify exploitability before mitigation planning. Validate compliance and re-run assessment.

2. python.djangoproject.web.djangoproject-cookie-samesite-none.djangoproject-cookie-samesite-none (Info / HIGH)

/app/tmp/repos/semgrep-20251112094210/app/api/v1/users.py

Detected a cookie options with the `SameSite` flag set to "None". This is a potential security risk that arises from the way web browsers manage cookies. In a typical web application, cookies are used to store and transmit session-related data between a client and a server. To enhance security, cookies can be marked with the "SameSite" attribute, which restricts their usage based on the origin of the page that set them. This attribute can have three values: "Strict," "Lax," or "None". Make sure the `SameSite` attribute of the important cookies (e.g., session cookie) is set to a reasonable value. When `SameSite` is set to "Strict", no 3rd party cookie will be sent with outgoing requests, this is the most secure and private setting but harder to deploy with good usability. Setting it to "Lax" is the minimum requirement.

samesite="none",

https://owasp.org/Top10/A01_2021-Broken_Access_Control

<https://web.dev/articles/samesite-cookies-explained>

Suggested Fix:

Summary This control requires attention in security. Remediation Steps Investigate severity and verify exploitability before mitigation planning. Validate compliance and re-run assessment.

3. dockerfile.security.missing-user.missing-user (Error / MEDIUM)

/app/tmp/repos/semgrep-20251112094210/Dockerfile

By not specifying a USER, a program in the container may run as 'root'. This is a security hazard. If an attacker can control a process running as root, they may have control over the container. Ensure that the last USER in a Dockerfile is a USER other than 'root'.

CMD unicorn app.main:app --host 0.0.0.0 --port 8001 --workers 1 --reload

https://owasp.org/Top10/A04_2021-Insecure_Design

Suggested Fix:

Summary This control requires attention in security. Remediation Steps Investigate severity and verify exploitability before mitigation planning. Validate compliance and re-run assessment.