MyESI Automated Compliance & Vulnerability Assessment

Project: myesi-user-service

Generated: 2025-11-12 10:15 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. Perform a Vulnerability Assessment Identify the scope and nature of the vulnerability in your environment.
2. Apply Security Patches Ensure all software, systems, and applications are updated with the latest security patches. Example command for Debian-based systems: bash sudo apt-get update && sudo apt-get upgrade
3. Review Access Controls Verify and restrict user permissions to the least privilege necessary. Ensure that only authorized individuals have access to sensitive resources. Configuration Guidance: Implement role-based access control (RBAC) and regularly audit user access rights.

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

1. Conduct a Comprehensive Vulnerability Assessment - Identify all potential vulnerabilities in systems and applications.
2. Patch Relevant Software - Ensure all software, including operating systems and applications, are up to date. For example, use a command like: bash sudo apt update && sudo apt upgrade
3. Implement Access Controls - Review user permissions and limit access to sensitive data. Consider using role-based access control (RBAC) as part of your configuration guidelines.

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	```markdown ### Summary The vulnerability identified as MAL-2025-4867 in the `pydantic` library poses a critical risk, potentially allowing attackers to compromise application integrity or expose sensitive data. Immediate action is required to mitigate potential exploitation. ### Remediation Steps 1. **Upgrade `pydantic` to the latest version.** - Ensure you are using the most recent stable release to incorporate security patches. 2. **For pip users, run the following command:** ```bash pip install --upgrade pydantic ``` 3. **Modify your dependency management system configuration if necessary.** - Update the version requirement in your `requirements.txt` or `setup.py` to reflect the latest version: ``` pydantic>=YOUR_NEW_VERSION ``` - Regularly review and monitor dependencies for updates to avoid similar vulnerabilities in the future. ```
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 The vulnerability identified as MAL-2025-47895 in the `typing-extensions` library poses a critical security risk, potentially allowing for unauthorized access or exploitation in applications that rely on this library. It is imperative to address this vulnerability to safeguard system integrity and security. ### Remediation Steps 1. **Update the `typing-extensions` Library** Ensure that you are using the most recent version of the `typing-extensions` library, as updates typically contain critical security fixes. 2. **Upgrade Command** Run the following command to upgrade the library: ```bash pip install --upgrade typing-extensions ``` 3. **Validate Dependency Configuration** Review your project dependencies and ensure that `typing-extensions` is specified correctly in your `requirements.txt` or equivalent configuration file. Avoid using old or vulnerable versions by implementing a dependency management tool like `pip-tools` or `Poetry` to lock versions securely.
8	**uvicorn**	0.37.0	MAL-2025-4901	**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. 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.django.web.django-cookie-samesite-none.django-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 uvicorn 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.