MyESI Automated Compliance & Vulnerability Assessment

Project: myesi-risk-service

Generated: 2025-11-12 15:32 UTC **User:** 1

1. Executive Summary

• Compliance Score: **60.0%**

• Average Risk Score: **10.00**

• Total Vulnerabilities: **4**

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 Conduct a Security Assessment Evaluate your current security posture to identify and understand the scope of the vulnerability.

2 Apply Security Patches Ensure that all software and systems are up to date with the latest security patches. For example, use the command: bash sudo apt-get update && sudo apt-get upgrade

3 Implement Access Controls Review and tighten access controls and permissions to limit exposure. Establish a policy that enforces the principle of least privilege across all systems.

3. Vulnerability Findings

#	Component	Version	Vuln ID	Severity	Fix / Recommendation
1	**pydantic**	2.12.3	MAL-2025-4867	**Critical**	### Summary The vulnerability MAL-2025-4867 in the Pydantic library poses a critical risk, potentially allowing attackers to exploit data validation flaws, leading to unauthorized access or data manipulation. Immediate action is required to mitigate the risk. ### Remediation Steps 1. **Upgrade Pydantic**: Update to the latest version of Pydantic where the vulnerability has been patched. 2. **Command to Upgrade**: Run the following command in your terminal: \`\`\`bash pip install --upgrade pydantic \`\`\` 3. **Review and Update Code**: Ensure that all instances of Pydantic usage in your codebase are reviewed for proper validation and error handling, and update any deprecated methods or practices as per the latest documentation.
2	**python-dotenv**	1.2.1	MAL-2025-48037	**Critical**	### Summary The vulnerability identified as MAL-2025-48037 in the \`python-dotenv\` package poses a critical risk, potentially allowing unauthorized access to sensitive environment variables. This can lead to data breaches and compromise of application integrity. ### Remediation Steps 1. **Upgrade the Package** Update the \`python-dotenv\` package to the latest version that addresses this vulnerability. 2. **Example Command** Use the following command to upgrade: \`\`\`bash pip install --upgrade python-dotenv \`\`\` 3. **Review Configuration and Policies** Ensure that environment variables are not stored in version control and implement strict access controls on configuration files. Additionally, review your application’s dependency management policies to include regular vulnerability scans.

3	**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 risk, potentially allowing unauthorized access or code execution in affected systems. Immediate action is required to mitigate risks associated with this vulnerability. ### Remediation Steps 1. **Upgrade the `typing-extensions` library** to the latest version that addresses this vulnerability. 2. **Run the following command** to update the library via pip: ```bash pip install --upgrade typing-extensions ``` 3. **Review and update your dependency management policies** to include regular checks for vulnerabilities in third-party libraries, ensuring that all libraries are up-to-date and secure.
4	**uvicorn**	0.38.0	MAL-2025-4901	**Critical**	### Summary The vulnerability identified as MAL-2025-4901 affects the Uvicorn ASGI server, posing a critical risk that could allow unauthorized access or remote code execution. Immediate action is required to mitigate potential exploitation. ### Remediation Steps 1. **Upgrade Uvicorn**: Update Uvicorn to the latest version that addresses this vulnerability. 2. **Command**: Run the following command to upgrade: ```bash pip install --upgrade uvicorn ``` 3. **Configuration Review**: Ensure that Uvicorn is configured to run behind a reverse proxy (e.g., Nginx) and restrict access to trusted IPs only. Review your firewall rules to block unauthorized access.

4. Code Findings (Static Analysis)

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

/app/tmp/repos/semgrep-20251112111552/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", "8004", "--reload"]
```

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

Suggested Fix:

Summary The vulnerability indicates that a Dockerfile does not specify a user to run the container, which can lead to potential security risks. Running containers as the root user can expose the host system to various attacks, including privilege escalation and unauthorized access. Remediation Steps Specify a Non-Root User: Modify the Dockerfile to use a non-root user for running the application inside the container. Example Command: Add the following lines in your Dockerfile: dockerfile RUN addgroup --system appgroup && adduser --system --ingroup appgroup appuser USER appuser Policy Guidance: Ensure that your organization’s container security policy mandates the use of non-root users in all Dockerfiles to minimize security risks associated with running containers as root.

Generated by MyESI on 2025-11-12 15:33 UTC — User: 1