MyESI Automated Compliance & Vulnerability Assessment

Project: backend

Generated: 2025-11-16 05:23 UTC **User:** 1

1. Executive Summary

• Compliance Score: **60.0%**

• Average Risk Score: **10.00**

• Total Vulnerabilities: **6**

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 comprehensive vulnerability assessment to identify and classify all existing vulnerabilities within your infrastructure.
2. Use tools like Nessus or OpenVAS to scan your systems for known vulnerabilities and generate reports.
3. Implement a patch management policy that ensures timely updates and patches for all software and systems to mitigate risks associated with vulnerabilities.

3. Vulnerability Findings

#	Component	Version	Vuln ID	Severity	Fix / Recommendation
1	**loguru**	0.7.3	MAL-2025-25559	**Critical**	### Summary The vulnerability identified as MAL-2025-25559 in the loguru library poses a critical risk, potentially allowing an attacker to exploit logging functionalities to execute arbitrary code or leak sensitive information. Immediate action is required to mitigate this risk. ### Remediation Steps 1. **Upgrade the loguru library** Update to the latest stable version of loguru where this vulnerability has been patched. 2. **Example command to upgrade** ```bash pip install --upgrade loguru ``` 3. **Review and update logging configurations** Ensure that logging configurations do not expose sensitive data and implement proper sanitization of log messages to prevent injection attacks.
2	**pydantic**	2.12.2	MAL-2025-4867	**Critical**	### Summary The vulnerability identified as MAL-2025-4867 in the `pydantic` library poses a critical risk, potentially allowing an attacker to exploit improper data validation leading to data corruption or unauthorized access. Immediate action is required to mitigate this risk. ### Remediation Steps 1. **Upgrade the Library**: Immediately update the `pydantic` library to the latest version where the vulnerability has been patched. 2. **Example Command**: Use the following command to upgrade: ```bash pip install --upgrade pydantic ``` 3. **Review Code Dependencies**: Audit your codebase for usage of the `pydantic` library and ensure that any related configurations or validations are aligned with the latest security practices. Update any outdated patterns that may expose vulnerabilities.

3	**pyjwt**	2.10.1	MAL-2025-48036	Critical	### Summary The MAL-2025-48036 vulnerability in the pyjwt library poses a critical risk, potentially allowing attackers to exploit weaknesses in JWT (JSON Web Tokens) handling. This could lead to unauthorized access or data breaches. ### Remediation Steps 1. **Update the pyjwt Library** Ensure that you are using the latest version of pyjwt, which contains patches for known vulnerabilities. 2. **Example Command** ```bash pip install --upgrade pyjwt ``` 3. **Configuration Guidance** Review your JWT implementation and ensure that you are using strong algorithms (e.g., RS256) for signing tokens. Avoid using none or weak algorithms, and implement proper validation checks for token expiration and claims.
4	**python-dotenv**	1.1.1	MAL-2025-48037	Critical	### Summary The vulnerability identified as MAL-2025-48037 in the `python-dotenv` library poses a critical risk, potentially allowing unauthorized access to sensitive environment variables. This could lead to data breaches or system compromises if exploited. ### Remediation Steps 1. **Upgrade `python-dotenv` to the latest version** Ensure you are using a secure version of the library that addresses the vulnerability. 2. **Upgrade Command** ```bash pip install --upgrade python-dotenv ``` 3. **Review and Update Usage** Audit your application code to ensure that environment variables are being loaded securely and that sensitive information is not exposed in logs or error messages. Consider implementing stricter access controls and monitoring for any unauthorized access attempts.

5	**typing-extensions**	4.15.0	MAL-2025-47895	Critical	### Summary The vulnerability identified as MAL-2025-47895 in the `typing-extensions` package poses a critical risk that could lead to unauthorized access or execution of arbitrary code. Immediate action is required to mitigate potential exploitation. ### Remediation Steps 1. **Upgrade the Package** Update the `typing-extensions` package to the latest version that addresses the vulnerability. 2. **Example Command** Run the following command to upgrade the package using pip: ```bash pip install --upgrade typing-extensions ``` 3. **Policy Guidance** Implement a policy to regularly check for and apply updates to all third-party packages, ensuring that your environment remains secure against known vulnerabilities. Consider using automated tools for dependency management and vulnerability scanning.
6	**uvicorn**	0.37.0	MAL-2025-4901	Critical	### Summary MAL-2025-4901 is a critical vulnerability associated with the `uvicorn` web server, which could allow attackers to execute arbitrary code or cause denial-of-service conditions. Immediate action is required to mitigate risks to your application and infrastructure. ### Remediation Steps 1. **Upgrade Uvicorn**: Update `uvicorn` to the latest version where the vulnerability has been patched. 2. **Command to Upgrade**: Run the following command to upgrade `uvicorn`: ```bash pip install --upgrade uvicorn ``` 3. **Configure Security Settings**: Review and enforce security settings in your application. Ensure that your `uvicorn` server is not exposed to untrusted networks, and consider using a reverse proxy (like Nginx) to handle incoming requests securely.

4. Code Findings (Static Analysis)

No code findings detected in the last scan.

Generated by MyESI on 2025-11-16 05:24 UTC — User: 1