MyESI Automated Compliance & Vulnerability Assessment

Project: backend

Generated: 2025-11-16 05:24 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 Vulnerability Assessment: Identify and confirm the specific vulnerabilities present in your environment.

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

3 Review Access Controls: Implement strict access control policies to limit user privileges and ensure only authorized personnel have access to sensitive data. Regularly review and update user permissions based on the principle of least privilege.

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 unauthorized access to sensitive logs and enabling attackers to exploit logged information for further attacks. ### Remediation Steps 1. **Update loguru Library**: Immediately update the loguru library to the latest version where the vulnerability has been patched. 2. **Example Command**: If using pip, execute the following command: ```bash pip install --upgrade loguru ``` 3. **Review Logging Practices**: Implement strict logging practices by ensuring sensitive information is not logged. Review and modify your logging configuration to exclude sensitive data and enforce access controls on log files.
2	**pydantic**	2.12.2	MAL-2025-4867	**Critical**	### Summary The vulnerability MAL-2025-4867 in the Pydantic library poses a critical risk due to potential remote code execution or data leakage. Unpatched instances may allow attackers to exploit affected applications, compromising confidentiality and integrity. ### Remediation Steps 1. **Upgrade Pydantic**: Immediately upgrade to the latest version of Pydantic that addresses this vulnerability. 2. **Example Command**: Use the following command to upgrade Pydantic via pip: ```bash pip install --upgrade pydantic ``` 3. **Review Dependencies**: Conduct a thorough review of your project's dependencies to ensure no indirect dependencies are using vulnerable versions of Pydantic. Consider implementing a policy to regularly check for and update critical libraries.

3	**pyjwt**	2.10.1	MAL-2025-48036	**Critical**	### Summary The vulnerability identified as MAL-2025-48036 in the pyjwt library poses a critical risk, potentially allowing attackers to exploit weak token validation mechanisms. This could lead to unauthorized access and data breaches. ### Remediation Steps 1. **Upgrade pyjwt**: Immediately update the pyjwt library to the latest version where the vulnerability has been patched. 2. **Example Command**: Run the following command to upgrade: ```bash pip install --upgrade pyjwt ``` 3. **Review Token Validation**: Ensure that your application implements strong token validation practices, such as verifying the token signature and expiration time, and using a secure secret for signing tokens.
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 and configurations. This can lead to data breaches and compromise application security. ### Remediation Steps 1. **Upgrade the Library** Update `python-dotenv` to the latest stable version where the vulnerability has been addressed. 2. **Example Command** ```bash pip install --upgrade python-dotenv ``` 3. **Configuration Guidance** Review and restrict access to environment variable files (e.g., `.env`) to ensure they are not publicly accessible. Implement appropriate file permissions and use secure storage solutions when managing sensitive configurations.

5	**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 attackers to exploit weaknesses in type hinting and type safety, leading to unauthorized access or data leakage. ### Remediation Steps 1. **Upgrade the `typing-extensions` Library** Ensure that you are using the latest version of the `typing-extensions` library to mitigate the vulnerabilities associated with earlier releases. 2. **Example Command** Use the following command to upgrade the library via pip: ```bash pip install --upgrade typing-extensions ``` 3. **Policy Guidance** Implement a regular update policy for all dependencies in your projects. Schedule periodic reviews to check for and update any outdated libraries, especially those with known vulnerabilities.
6	**uvicorn**	0.37.0	MAL-2025-4901	**Critical**	### Summary The vulnerability MAL-2025-4901 in the Uvicorn web server poses a critical risk, potentially allowing remote code execution or denial of service, compromising the integrity and availability of applications using this server. ### Remediation Steps 1. **Upgrade Uvicorn** Update Uvicorn to the latest stable version that addresses this vulnerability. 2. **Example Command** ```bash pip install --upgrade uvicorn ``` 3. **Configuration Guidance** Review and apply security best practices in your Uvicorn configuration, such as limiting access to trusted IPs, enabling HTTPS, and using proper authentication mechanisms.

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