File name	Procedure for Handling Dependency Check Vulnerabilities
Author	Milica Blagojevic
Confidentiality	Internal
Last save date	Thursday, July-11-2024 at 4:11:00 PM

Table of Contents

1 Introduction

This document provides guidelines and procedures for identifying, prioritizing, suppressing, and resolving vulnerabilities in dependencies detected by the **OWASP Dependency-Check Maven plugin**.

Regular scanning of project dependencies is crucial to maintaining a secure and resilient software environment. Dependency vulnerability checks should be conducted on a **weekly basis** to detect newly introduced vulnerabilities promptly, and it is **mandatory** to perform a comprehensive scan before any new release to ensure that no high or critical vulnerabilities are present.

2 Regular Dependency Scanning Policy

To maintain a secure and resilient software environment, it is essential that all project dependencies are regularly scanned for known vulnerabilities. This ensures that any new vulnerabilities discovered in third-party libraries are identified and addressed promptly.

2.1 Scanning Frequency

2.1.1 Weekly Scans

A full dependency scan must be performed **at least once a week**. This ensures that any new vulnerabilities introduced into our dependencies or newly discovered issues in existing libraries are identified in a timely manner.

2.1.2 Mandatory Pre-Release Scan

Before any release, it is **mandatory** to perform a comprehensive dependency scan to ensure that no high or critical vulnerabilities are present. The build should be blocked if such vulnerabilities are detected unless they have been appropriately mitigated or suppressed.

3 Severity Levels and Deadlines

When vulnerabilities are identified in a project's dependencies, they are categorized by severity. The following outlines the typical resolution deadlines based on severity:

3.1 Severity Levels and Corresponding Deadlines

3.1.1 CRITICAL (CVSS Score: 9.0 - 10.0)

Resolution Deadline: Immediate to within 7 days.
Description: Vulnerabilities that allow attackers to gain control, execute remote code, or cause severe data loss.

3.1.2 HIGH (CVSS Score: 7.0 - 8.9)

Resolution Deadline: Within 30 days.
Description: Vulnerabilities that can lead to significant security breaches, such as privilege escalation or sensitive data exposure.

3.1.3 MEDIUM (CVSS Score: 4.0 - 7.0)

Resolution Deadline: Within 60 to 90 days.
Description: Vulnerabilities with moderate impact, requiring specific conditions to exploit.

3.1.4 LOW (CVSS Score: 0.1 - 3.9)

Resolution Deadline: Addressed during normal patch cycles (90-180 days).
Description: Low-risk vulnerabilities with limited or minimal impact.

4 OWASP Dependency-Check Maven Plugin Configuration

Currently this is our OWASP Dependency-Check Maven plugin configuration:

```
<plugin>
    <groupId>org.owasp</groupId>
    <artifactId>dependency-check-maven</artifactId>
    <version>10.0.4</version>
    <configuration>
        <autoUpdate>false</autoUpdate>
        <assemblyAnalyzerEnabled>false</assemblyAnalyzerEnabled>
        <suppressionFile>suppressions.xml</suppressionFile>
        <failBuildOnCVSS>7.0</failBuildOnCVSS>
    </configuration>
    <executions>
        <execution>
            <goals>
                <goal>check</goal>
            </goals>
        </execution>
    </executions>
</plugin>
```

Figure 1. OWASP Maven plugin configuration

Key Configurations:

- **Fail Build on CVSS Threshold:** Set the failBuildOnCVSS to 7.0 to automatically fail the build if any high or critical vulnerabilities are found.
- **Suppression File:** Reference a suppression file (suppressions.xml) to suppress specific vulnerabilities, such as false positives or known but accepted risks.
- **Assembly Analyzer**: Set <assemblyAnalyzerEnabled>false</assemblyAnalyzerEnabled> to disable scanning of assemblies such as .zip or .tar files, which can reduce unnecessary analysis and improve build performance.

- **Auto Update**: Set <autoUpdate>false</autoUpdate> to disable the automatic updates of the vulnerability database, which can speed up builds. You will need to manually update the database periodically to ensure it's up-to-date.

5 Handling Vulnerabilities

5.1 Step 1: Identify Vulnerabilities

After running the Maven build with the OWASP Dependency-Check plugin, review the report to identify vulnerabilities in your dependencies. Vulnerabilities will be categorized by severity.

```
[ERROR] Failed to execute goal org.owasp:dependency-check-maven:10.0.4:check (default) on project ytm.db:
[ERROR]
[ERROR] One or more dependencies were identified with vulnerabilities that have a CVSS score greater than or equal to '7.0':
[ERROR]
[ERROR] commons-compress-1.19.jar: CVE-2021-35517(7.5), CVE-2021-35516(7.5), CVE-2021-35515(7.5), CVE-2021-36090(7.5)
[ERROR]
[ERROR] See the dependency-check report for more details.
[ERROR]
[ERROR]
[ERROR] -> [Help 1]
[ERROR]
```

Figure 2. Example of Maven build failure

5.2 Step 2: Prioritize by Severity

Use the severity thresholds mentioned above to prioritize vulnerability remediation efforts:

- **Critical vulnerabilities**: Address immediately within 7 days.
- **High vulnerabilities**: Address within 30 days.
- **Medium vulnerabilities**: Plan to address within 60-90 days.
- **Low vulnerabilities**: Address during regular patch cycles.

5.3 Step 3: Suppress Specific Vulnerabilities

In cases where a vulnerability is deemed acceptable (e.g., it's a false positive or mitigated through other means), you can suppress it using a suppression file.

5.3.1 Adding Suppressions

- All suppressions should be added to the `suppression.xml` file, which is referenced in the Maven configuration. This ensures that the build process excludes the suppressed vulnerabilities in future scans.
- Suppression details, such as CVE IDs or GAV coordinates (GroupId, ArtifactId, Version), can be extracted from the **generated report**. This report provides the necessary information to correctly format the suppression entries.

Figure 3. Generate suppression XML for vulnerability from generated report

5.4 Step 4: Re-scan and Verify

Once you have remediated vulnerabilities or added suppressions, re-run the build to ensure that the suppressed or resolved vulnerabilities no longer block the build.