Saumyaa Prajapat
21BCI0265

Task3:

1. CWE: CWE-284: Improper Access Control

OSWAP CATEGORY: A01:2021- Broken Access Control

DESCRIPTION: The product does not restrict or incorrectly restricts access to a resource from an unauthorized actor.

BUSINESS IMPACT: Access control involves the use of several protection mechanisms such as:

- Authentication (proving the identity of an agent)
- Authorization (ensures that a given agent can access the resource)
- Accountability (tracking of activities that were performed)

When a mechanism is not enforced or fails, an attacker can compromise product security by gaining privileges, reading sensitive information, executing commands, evading detection, and more.
Two separate behaviours can cause access control weaknesses:

- Specification: incorrect privileges, permissions, ownership, etc. are explicitly specified for either the user or the resource (for example, setting a password file to be world-writable, or giving administrator capabilities to a guest user). This action could be performed by the program or the administrator.
- Enforcement: the mechanism contains errors that prevent it from properly enforcing the specified access control requirements (e.g., allowing the user to specify their own privileges, or allowing a syntactically-incorrect ACL to produce insecure settings). This problem occurs within the program itself, in that it does not actually enforce the intended security policy that the administrator specifies.

2. CWE: CWE-261: Weak Encoding for password

OSWAP CATEGORY: A02:2021- Cryptographic Failures

DESCRIPTION: Obscuring a password with a trivial encoding does not protect the password.

BUSINESS IMPACT: Password management issues occur when a password is stored in plaintext in an application's properties or configuration file. A programmer can attempt to remedy the password management problem by obscuring the password with an encoding function, such as base 64 encoding, but this effort does not adequately protect the password.

3. CWE: CWE-917: Improper Neutralization od Special Elements used in an Expression Language Statement ('Expression Language Injection')

OSWAP CATEGORY: A03:2021- Injection

DESCRIPTION: The product constructs all or part of an expression language (EL) statement in a framework such as a Java Server Page (JSP) using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended EL statement before it is executed.

BUSINESS IMPACT: Frameworks such as Java Server Page (JSP) allow a developer to insert executable expressions within otherwise-static content. When the developer is not aware of the executable nature of these expressions and/or does not disable them, then if an attacker can inject expressions, this could lead to code execution or other unexpected behaviors.

4. CWE: CWE-501: Trust Boundary Violation

OSWAP CATEGORY: A04:2021- Insecure Design

DESCRIPTION: The product mixes trusted and untrusted data in the same data structure or structured message.

BUSINESS IMPACT: A trust boundary can be thought of as line drawn through a program. On one side of the line, data is untrusted. On the other side of the line, data is assumed to be trustworthy. The purpose of validation logic is to allow data to safely cross the trust boundary - to move from untrusted to trusted. A trust boundary violation occurs when a program blurs the line between what is trusted and what is untrusted. By combining trusted and untrusted data in the same data structure, it becomes easier for programmers to mistakenly trust unvalidated data.

5. CWE: CWE-1004: Sensitive Cookie Without 'HttpOnly' Flag

OSWAP CATEGORY: A05:2021- Security Misconfiguration

DESCRIPTION: The product uses a cookie to store sensitive information, but the cookie is not marked with the HttpOnly flag.

BUSINESS IMPACT: The HttpOnly flag directs compatible browsers to prevent client-side script from accessing cookies. Including the HttpOnly flag in the Set-Cookie HTTP response header helps mitigate the risk associated with Cross-Site Scripting (XSS) where an attacker's script code might attempt to read the contents of a cookie and exfiltrate information obtained. When set, browsers that support the flag will not reveal the contents of the cookie to a third party via client-side script executed via XSS.