List of CWE which are applicable to Shell Script

CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection'): This weakness involves improper handling of user-controlled data, which allows an attacker to execute arbitrary commands on the system.

CWE-73: External Control of File Name or Path: This weakness occurs when a script uses externally influenced input to construct file or directory paths, potentially allowing an attacker to access unintended files.

CWE-88: Argument Injection or Modification: This weakness involves manipulating command-line arguments in a way that could lead to unauthorized access or data manipulation.

CWE-20: Improper Input Validation: If a shell script doesn't validate user input properly, it could be susceptible to various attacks, including code injection and command execution.

CWE-94: Improper Control of Generation of Code ('Code Injection'): This weakness involves injecting malicious code into a script's execution flow, potentially leading to unauthorized code execution.

CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal'): If a script does not properly restrict file paths, attackers might traverse directories and access sensitive files.

CWE-134: Uncontrolled Format String: Using user-controlled data in format strings without proper validation can lead to memory corruption vulnerabilities.

CWE-209: Information Exposure Through an Error Message: Detailed error messages from a shell script can provide valuable information to attackers, aiding in further attacks.

CWE-269: Improper Privilege Management: Inadequate privilege management could lead to unauthorized access and manipulation of sensitive resources.

CWE-352: Cross-Site Request Forgery (CSRF): In a web context, shell scripts might be vulnerable to CSRF attacks, leading to unauthorized actions on behalf of a user.

OWASP top 10

Injection: Shell scripts can be susceptible to various injection attacks, including command injection and SQL injection, if they don't properly validate and sanitize user input.

Broken Authentication: If shell scripts are used in authentication mechanisms or session handling, vulnerabilities in these scripts could lead to unauthorized access.

Sensitive Data Exposure: If shell scripts handle sensitive data and do not properly protect it, attackers might be able to gain unauthorized access to this data.

XML External Entity (XXE): If shell scripts process XML data, they might be vulnerable to XXE attacks, allowing attackers to read sensitive files or perform Denial of Service (DoS) attacks.

Broken Access Control: Improper access controls in shell scripts could allow unauthorized users to perform actions they shouldn't be allowed to.

Security Misconfiguration: Improperly configured shell scripts might expose sensitive information or allow unauthorized access.

Cross-Site Scripting (XSS): If shell scripts interact with web components, they could be vulnerable to XSS attacks, which might allow attackers to execute malicious scripts in the context of other users.

Insecure Deserialization: If shell scripts process serialized data, they might be vulnerable to attacks that exploit insecure deserialization, leading to remote code execution.

Using Components with Known Vulnerabilities (OWASP Top 10 #9): If shell scripts incorporate external components or libraries, vulnerabilities in those components could pose a risk.

Insufficient Logging & Monitoring (OWASP Top 10 #10): Lack of proper logging and monitoring in shell scripts might hinder the detection of attacks or unauthorized activities.