Software Vulnerabilities Classification

Vulnerabilities vs bugs

CWE

Fortify Taxonomy: the 7 kingdoms

- Input Validation and Representation (buffer overflows, format strings, code injection)
- API Abuse (improper use of chroot)
- Security Features (improper use of access control, cryptography, privilege management, leaking privileged information)
- Time and State (ToCToU, authentication runs concurrently with check of authentication result)
- Error Handling (missing error handling may let the program continue causing unpredicted behavior, or generate an exception that leaks sensitive information)
- Code Quality (not respecting coding guidelines may lead to unexpected behaviors: arithmetic operation on boolean in C)
- Encapsulation (not protecting code in a browser from other code running in the same browser: no sandboxing -> cross-session contamination, store data on localStorage instead of sessionStorage)

OWASP Top 10

Ranking of each vulnerability class is based on
- Number of CWEs mapped to the class
- Max incidence rate of the vulnerabilities in the class
- Average incidence rate
- Average weighted exploit (severity of the vulnerabilities)
- Average weighted impact (impact of the vulnerabilities)
- Max coverage (applications tested using automatic tools to search for those vulnerabilities)
- Average coverage
- Total occurrences
- Total CVEs

2017
1. Injection
2. Broken Authentication
3. Sensitive Data Exposure
4. XML External Entities XXE
5. Broken Access Control
6. Security Misconfiguration
7. Cross-Site Scripting XSS
8. Insecure Deserialization
9. Using Components with known Vulnerabilities
10. Insufficient Logging and Monitoring

2021

1. Broken Access Control (was 5) [Security Features Kingdom+Encapsulation Kingdom]
   - Missing or breakable Access Control (Privilege API available for unauthenticated users, violation of least-privilege principle deny-by-default)

2. Cryptographic Failures (includes 3) [Security Features Kingdom]
   - Cryptography wrongly used/not used when necessary (sensitive data stored or transmitted in plaintext, unsalted password db, old or weak cryptographic algorithms, missing security certificate validation, improper key management (storage or refreshing), wrong use of criptographic APIs)

3. Injection (was 1+7) [Input Validation and Representation Kingdom]
   - Attacker can sends hostile data to an interpreter, without validation or sanitization (filtering) (sql, php, xss)
   - XSS: js code inside html to be executed by the browser js engine bypassing the Same-Origin-Policies.
     - SOP: except for img and script elements, the js code downloaded from one origin (scheme+host+port)
       - cannot access documents (DOM tree) downloaded from other sources
       - cannot access cookies set by other origins
       - cannot request operations (get,post,delete) on other origins
     - Types of XSS
       - Reflected XSS
         1. Website reflects user inputs without validation and sanitization
         2. Attacker crafts url to the real website, including malicious input
         3. Victim clicks on url, the real website generates the response page including the malicious input (script tag pointing to the attacker website)
         4. Since the GET request comes from a script tag in the real website, it can be performed even if the origin is different
         5. The malicious script can access documents (DOM tree) and cookies of the real website, and can request operations (GET or POST) to communicate those information to the attacker website -> stolen data can be used to access victim account
       - Stored XSS: even more powerful because since the malicious input is stored in the db, all the regular requests to the application will compromise the users!
       - DOM XSS: script is not inserted by the server while generating the page, but is inserted by the browser dynamically, after the page have already been served

4. Insecure Design (new)
   - Security was not properly considered during the development process (absent/incomplete threat model leads to neglection of some possible attacks, wrong choice of security controls)

5. Security Misconfiguration (was 6) [Security Features Kingdom]
   - Security features not properly configured (development settings not changed for production release: default accounts, directory listing not disabled on the webserver, improperly configured permissions)

6. Vulnerable and Outdated Components (was 4+9) [Code Quality Kingdom]
   - (Typically happens when third-party components used in web application run with the same privileges of the main application)
   - XML External Entities XXE: attacker can send hostile XML data to an XXE-vulnerable XML interpreter (automatically tries to interpret entity statement, which could let the attacker access files which would normally be unaccessible)

7. Identification/Authentication Failures (was 2) [Security Features Kingdom]
   - (Permitting (stolen) credential stuffing, allowing weak passwords, improper invalidation of sessionID after the session termination, improper rotation of sessionIDs (reuse recently used sessionID), exposure of session IDs in url, browser history, browser log, unverified user ID received from the user, weak credential recovery mechanisms)

8. Software/Data Integrity Failures (includes 8) [Input Validation and Representation Kingdom]
   - (Updates download without integrity verification, downloading plugins/libraries from untrusted repositories/CDNs, insecure deserialization)
   - Insecure deserialization: serialized data accepted from untrusted source and not deserialized securely

9. Security Logging and Monitoring Failures (was 10) [Errors Kingdom]
   - Let attackers perform attacks without being detected (Absence of logging security-relevant events (errors, anomalous traffic..), unclear log messages, logs not constantly monitored, logs only stored locally, scans and pentesting do not trigger alerts)

10. Server Side Request Forgery SSRF (new) [Input Validation and Representation Kingdom]
    - Web application fetches remote resource without validating the user-supplied malicious URL (can stole server information file:///etc/passwd, can make the server perform unintended operations on resources protected by firewalls, vpns, acls)