OWASP analys

Top 10 OWASP Open Web Application Security Project 2017

”76% av alla websiter på internet har säkerhetshål…”

Cyber Security Consulting Sverige AB hjälper Er att säkerhetsanalysera Era webapplikationer…

OWASP topp 10 säkerhetsanalys

Säkerhetsanalys: OWASP Topp 10

Visste du att cirka 3 av 4 webbapplikationer/siter har säkerhetshål (* källa: Symantec Inc samt IBM)? Detta beror framför allt på att programvaror och hårdvaror inte uppdateras ordentligt eller att det finns säkerhetsproblem i bakomliggande kod och applikation.

Hur går en OWASP analys till?

För att minimera risk och få kontroll över eventuella säkerhetsproblem i webbapplikationer har OWASP (Open Web Applikation Security Project) kartlagt de 10 vanligaste säkerhetsriskerna/attackvektorerna med webbapplikationer. Dessa risker är namngivna A1-A10, där exempelvis risk A1 är ”Injection attack”. En sådan attack som har som mål att stjäla dåligt skyddad data ifrån i applikationen.Vid en OWASP säkerhetsanalys kontrollerar våra säkerhetskonsulter Er webbapplikation(er) enligt denna topp 10 lista. Webapplikationen kontrolleras punkt för punkt för att se om applikationen visar tecken att vara utsatt för någon av dessa attacktyper. Testet utförs i 4 steg:

- Uppstartsmöte samt diskussioner om applikationens funktioner
- Säkerhetsscanning av applikationen för att hitta eventuella säkerhetsproblem
- Manuell analys samt manuell testning av applikationen
- Slutrapportering av säkerhetsanalysen

Genom att kontinuerligt testa sina webbapplikationer enligt OWASP topp 10 så kan man minska sin säkerhetsrisk för datastöld och dataintrång med upp till 50% (* Källa: owasp.org)

Tidsåtgång

Det tar cirka 2-4 veckor att utföra en komplett OWASP analys beroende på applikationens storlek och komplexitet. Vi utför även enklare sårbarhetsanalyser av webapplikationer, och dessa tar ca 1 vecka att utföra, men ger då inte en lika detaljerad rapport som en komplett OWASP analys.

OWASP topp 10 kontrollpunkter:

A1:2017-Injection
Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

A2:2017-Broken Authentication
Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.

A3:2017-Sensitive Data Exposure
Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.

A4:2017-XML External Entities (XXE)
Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.

A5:2017-Broken Access Control
Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

A6:2017-Security Misconfiguration
Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.

A7:2017-Cross-Site Scripting (XSS)
XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

A8:2017-Insecure Deserialization
Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

A9:2017-Using Components with Known Vulnerabilities
Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

A10:2017-Insufficient Logging&Monitoring
Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

Vill du veta mer? Ta kontakt med oss för en gratis konsultationstimme.

Telefon: 0735-95 11 12

Epost: offert@cybersecurityconsulting.se

Cyber Security Consulting