Hacking a WebSite

ALIN ZAMFIROIU

Top Ten OWASP

- **A1 -** Injection
- **A2 -** Cross-Site Scripting (XSS)
- **A3 -** Broken Authentication and Session Management
- **A4 -** Insecure Direct Object References
- **A5 -** Cross-Site Request Forgery (CSRF)
- **A6 -** Security Misconfiguration
- **A7 -** Insecure Cryptographic Storage
- **A8 -** Failure to Restrict URL Access
- **A9 -** Insufficient Transport Layer Protection
- **A10-** Unvalidated Redirects and Forwards

A1 - Injection

- Mistakes related to injection, such as SQL or LDAP injection, occurs when data are not reliable are sent to an interpreter as part of a command or query.

- With hostile data, an attacker can execute commands to cheat the interpreter for the unauthorized data access.

A2 - Cross-Site Scripting (XSS)

- XSS problems occurs when the application takes data that can not be trusted and send them to a browser without valid and sanitizedthem properly.

- XSS allows attackers to execute scripts in the victim's browser, which can deterioration of web pages or to redirect users to malicious Web sites.

A3 - Broken Authentication and Session Management

- Application functions that are related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, session tokens, or exploit other implementation mistakes and thus to secure the identity of other users.

A4 - Insecure Direct Object References

- A direct reference to an object occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key.

- Without an access control check or without any other form of protection, attackers can manipulate these references to access unauthorized data.

A5 - Cross-Site Request Forgery (CSRF)

- A CSRF attack forces a victim's browser to send an HTTP request logged counterfeit, including the victim's session cookie authentication and other information automatically included to a vulnerable web application.

- This allows the attacker to force the victim's browser to generate requests the vulnerable application it believes are legitimate request from victim.

A6 - Security Misconfiguration

- Good security practices require the existence of a secure configuration defined and deployed applications, architectures, web servers, databases and platforms.

- All these settings must be defined, implemented and maintained, because many of them come with secure default configurations. This involves keeping up-to-date for all applications and code libraries used by them.

A7 - Insecure Cryptographic Storage

- Many web applications do not properly protect sensitive data such as credit cards, ID's, authentication credentials, using encryption or hashing good mechanisms.

- Attackers may steal or modify such weakly protected data so as to determine identity theft, credit card fraud, or other criminal acts.

A8 - Failure to Restrict URL Access

- Most web applications check URL access rights before play protected links and buttons.

- However, applications have to perform the same type of checks each time these pages are accessed, or attackers will be able to forge URLs to access these pages.

A9 - Insufficient Transport Layer Protection

- Applications frequently fail to authenticate, encrypt or protect the confidentiality and integrity of sensitive network traificului.

- You could not protect and encrypt traffic using weak algorithms, use expired certificates are valid or not, or do not use them properly.

A10- Unvalidated Redirects and Forwards

- Web applications frequently redirect the users to other pages mode and sites and using data not reliable to determine the landing page.

- Without concrete validation, attackers can redirect victims to phishing or malware pages, or use the redirects to access unauthorized pages.

Pentesting

- Penetration tests are performed using manual or automated tools to detect potential points of exposure.
- Information about any vulnerability successfully exploited are presented to the owner of that system.

Benefits of pentesting

- manage vulnerabilities;
- avoid the cost of network downtime;
- minimize client-side attacks;
- evaluate security investment.

Tools for pentesting

- Nmap
- Metasploit penetration testing software
- John the Ripper
- THC Hydra
- OWASP Zed
- Wireshark
- Aircraft-ng
- Cain and Abel
- Nikto website vulnerability scanner

Hacking a WebSite