Penetration testing and OWASP Top 10 report

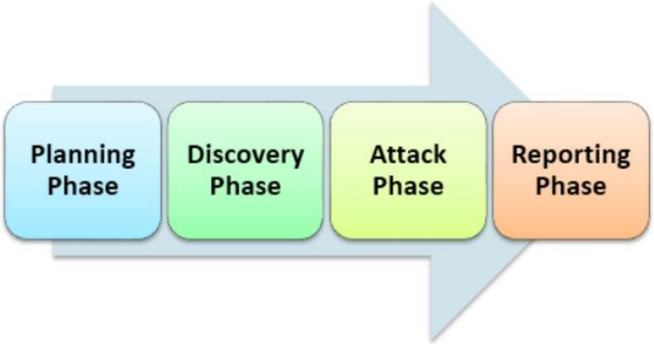
-Payment concentrator system and Literary Association-

Penetration test is a type of Security Testing used to uncover vulnerabilities, threats and risks that an attacker could exploit in software applications, networks or web applications.
It consists of two parts: Vulnerability Assessment and Pen Testing.

Purpose is to identify and test all possible vulnerabilities that are present in the software application. Vulnerability is the risk that an attacker can disrupt or gain authorized access to the system or any data contained within it.

In this case, the **white-box** penetration testing was taken, because the tester was provided with complete information abouth the system to be tested.

How to do Penetration Testing

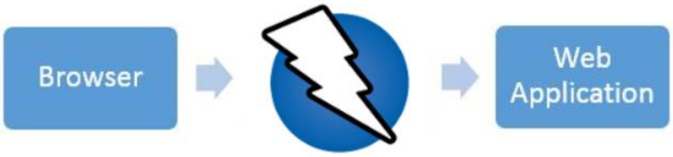


1) **Planning phase** – determining the scope and strategy
2) **Discovery phase** - collecting as much information as possible (data, like usernames and passwords) and checking for possible vulnerabilities of the system.
3) **Attack phase** – carrying out attacks according to data and vulnerabilities that have been found.
4) **Reporting phase** – providing report about risks of vulnerabilities found and their impact on business.

Tools used for Penetration Testing

Tools that have been used for providing penetration testing reports:

- **OWASP ZAP** (link: https://www.zaproxy.org/) – penetration testing tool, known as "man-in-the-midde" proxy. It stands between the browser and the web application, intercepts and inspects messages sent between browser and web application, modifying the content needed and then forward packets on to the destination.



- **Burp Suite** (link: https://portswigger.net/burp) – set of manual tools for exploring web security. It enables tester to proxy traffic, edit and repeat requests, decode data, perform some of known cyber attacks and more.

Within the Discovery phase technique called **spidering** was used**.** It deals with crawling through the website and records all the files, links and methods it can get. It gives us the idea of how web app is structured, how it works and how to break through it.

After that Automation and Manual tests were taken with both pen testing tools.
According to reports generated from OWASP ZAP on the basis of automation testing for Literary Association and Payment Concentrator application, two risks stood out:

- *Cross Site Scripting Weakness* (Reflected in JSON Response),
- *Weak Authentication Method,*

where solutions for those problems were mentioned.

Burp Suite was used to intercept requests and responses between browser and web application, trying to change request payloads in order to send malicious data to the server.
Attacks that were conducted:

- *SQL Injection*
- *Simulation of brute-force attack on application login*

Tools that are used within Burpe Suit:

- Scanner – performs automated scans (crawling for content and auditing for vulnerabilities)
- Intruder – it enables tester to change request payloads, put possible values and gain some useful information about the system and possible weak points:
  - enumerating identifiers,
  - harvesting useful data,
  - fuzzing for vulnerabilities

Owasp depdendency check report

As a part of testing web application and finding all vulnerabilities which attackers could take advantage of, for all parts of system **Owasp Dependency checker** was used to detect publicly disclosed vulnerabilities contained within a project's dependencies.

Reports are provided.

OWASP TOP 10 Risks

1. **Injection** - occurs when untrusted data is sent to an interpreter as a part of command or query.
   The most known Injection attack is **SQL Injection**.

   Security controls applied:
- *Frontend validation*
- *Backend validation*

2. **Cross-site scripting** – occurs when application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data.

   Security controls applied:
   - *Data validation*
   - *ReactJS is quite sage by design since string variables in views are escaped automatically and with JSX you pass a function a the event handler, rather than a string that can containt malicious code*.

3. **Broken authentication** – assuming other users' identities by compromising passwords, keys, session tokens etc.

   Security controls applied:
   - *Password complexity (at least one number, capitalized letter, special characters and minimum 8 characters)*
   - *Verification token registration*
   - *Using Bcrypt encoder when storing passwords in database*

4. **Broken access control** – exploit flaws in restrictions on what authenticated users are allowed to do.

   Security controls applied:
   - *Implement authorization by assigning adequate roles to each registered user*
   - *Use React Gourded routes*

5. **Sensitive data exposure** – occurs when attacker steal or modify weakly protected and exposed sensitive data (financial, credentials..).

   Security controls applied:
   - *Using Bcrypt encoder when storing passwords in database*
   - *Not exposing sensitive data on user interface like PAN or passwords*

6. **Insufficent Logging or Monitoring** – leads to slower detection of threat or attacker presence inside system.

   Security controls applied:
   - *Implement Logging service for storing logs of all system applications in logs database providing information who, where, when and what has happened*