What is penetration testing?

It is a testing method that helps find vulnerabilities of a network, web application or computer system. This testing method helps in identifying whether the existing defensive measures that are incorporated in the system are enough to prevent any security breaches.

Advantages of Pen Testing:

a) **Adherence to Compliance Requirements:** Helps to meet regulatory requirements such as PCI DSS, EU GDPR, and ISO.
b) **Identify and Remediate Vulnerability:** Helps identify vulnerabilities that can exploit the security systems and find a solution.
c) **Ensure Business continuity:** By running pen tests, organizations can reduce the risk of attacks.
d) **Enhance Customer Trust:** Pen testing can minimize the risk of attacks and assures clients that their data is secure.

Goals of Pen Testing:

1. Check if web application can identify spam attacks on contact forms used on the website.
2. Proxy server – Check if network traffic is monitored by proxy appliances. Proxy servers make it difficult for hackers to get internal details of the network thus protecting the system from external attacks.
3. Spam email filters – Verify if incoming and outgoing email traffic is filtered and unsolicited emails are blocked. Many email clients come with in-build spam filters which needs to be configured as per needs. These configuration rules can be applied on email headers, subject, or body. Can be identified by the keywords used in the subject line as well as the body.
4. Firewall – Make sure entire network or computers are protected with Firewall.
5. Verify that all usernames and passwords are encrypted and transferred over secured connection like https.
6. Verify information stored in website cookies. It should not be in readable format.
7. Verify if there is no open port in network.
8. Verify all HTTP methods. PUT and Delete methods should not be enabled on web server.
9. Password monitoring (Using password with different combinations)
10. Application behaviour after multiple logins and more activity of the user. (can be locked or access can be retrieved).
11. Generation of Error messages in terms of invalid login details
12. Scan all incoming files uploaded by the users.
13. Sensitive data should be hidden (Like invisible password)
14. Blocking Login page after multiple failed attempts.
15. Verification of reset password with double authentication (Email and Mobile)
16. Relog in after the password is reset and logging out from different devices.
17. Session ending after inactivity of user (Time limit of 15-30 mins).
18. Maintaining of logs (User activity like login and logout).
19. No redirection to third party websites
20. Warning the user when public networks are used.

21. GPS spoofing, Email ID spoofing, IP address spoofing, Caller ID spoofing, Referrer spoofing, ARP spoofing etc., should be verified.
22. Monitor Trojan attacks by scanning incoming network traffic.

Goal 1: Check if web application can identify spam attacks on contact forms.

Use Case: Spam Attack Detection on Contact Forms

Test Case 1: Spam Attack Simulation

Test Step: Submit contact forms with a high volume of requests in a short period.

Expected Result: The web application should identify and mitigate the spam attack, preventing successful form submissions.

Goal 2: Check if network traffic is monitored by proxy appliances.

Use Case: Proxy Server Traffic Monitoring

Test Case 1: Proxy Bypass Attempt

Test Step: Attempt to bypass the proxy server and access a restricted resource.

Expected Result: Proxy server should detect and block attempts to bypass, ensuring all traffic goes through the proxy.

Goal 3: Verify if incoming and outgoing email traffic is filtered.

Use Case: Spam Email Filters Verification

Test Case 1: Email Spoofing Attempt

Test Step: Send emails with spoofed headers, subjects, or bodies.

Expected Result: Spam filters should identify and block emails with suspicious content.

Goal 4: Make sure entire network or computers are protected with Firewall.

Use Case: Firewall Protection Verification

Test Case 1: Port Scanning

Test Step: Conduct a port scan to identify open ports in the network.

Expected Result: Firewall should block unauthorized access through open ports and prevent information of services on ports being sent.

Goal 5: Verify that all usernames and passwords are encrypted and transferred securely.

Use Case: Secure User Authentication

Test Case 1: Login Credential Encryption

Test Step: Capture network traffic during login to verify that usernames and passwords are encrypted.

Expected Result: User credentials should be transmitted securely over HTTPS and/or other secure protocols.

Goal 6: Verify information stored in website cookies is not in readable format.

Use Case: Secure Cookie Storage

Test Case 1: Cookie Inspection

Test Step: Inspect cookies stored by the website to ensure they are not in a readable format.

Expected Result: Cookies should be encrypted or hashed for enhanced security.

Goal 7: Verify if there is no open port in the network.

Use Case: Closed Port Verification

Test Case 1: Port Scanning

Test Step: Conduct a port scan to confirm that there are no open ports in the network.

Expected Result: All ports should be closed, except for essential services.

Goal 8: Verify all HTTP methods; PUT and Delete methods should not be enabled on the web server.

Use Case: HTTP Method Verification

Test Case 1: PUT and Delete Method Test

Test Step: Attempt to use PUT and Delete methods on the web server.

Expected Result: PUT and Delete methods should be disabled, allowing only safe methods.

Goal 9: Password monitoring (Using password with different combinations)

Use Case: Password Strength Testing

Test Case 1: Weak Password Attempt

Test Step: Attempt to use weak passwords, common passwords, and dictionary words.

Expected Result: The system should enforce password complexity and reject weak password attempts.

Goal 10: Application behaviour after multiple logins and more user activity

Use Case: User Activity Monitoring

Test Case 1: Multiple Login Attempts

Test Step: Attempt multiple login sessions within a short timeframe.

Expected Result: System behaviour should be monitored, and any suspicious activity should be flagged or restricted.

Goal 11: Generation of error messages in terms of invalid login details

Use Case: Invalid Login Error Handling

Test Case 1: Invalid Login Attempt

Test Step: Attempt to log in with incorrect credentials.

Expected Result: The system should provide generic error messages without revealing specific details about the invalid input.

Goal 12: Scan all incoming files uploaded by users.

Use Case: File Upload Security

Test Case 1: Malicious File Upload

Test Step: Attempt to upload files with malicious content or executable scripts.

Expected Result: The system should reject files with potentially harmful content.

Goal 13: Sensitive data should be hidden (Like invisible password)

Use Case: Hidden Sensitive Data

Test Case 1: Inspect Hidden Elements

Test Step: Inspect webpage elements to check for hidden sensitive information.

Expected Result: Sensitive data should not be visible in the webpage source code or rendered content.

Goal 14: Blocking login page after multiple failed attempts.

Use Case: Account Lockout Policy

Test Case 1: Multiple Failed Login Attempts

Test Step: Attempt multiple consecutive failed login attempts.

Expected Result: The system should temporarily lock the account after reaching the specified threshold.

Goal 15: Verification of reset password with double authentication (Email and Mobile)

Use Case: Two-Factor Authentication (2FA) Verification

Test Case 1: Reset Password with 2FA.

Test Step: Initiate the password reset process and verify using both email and mobile authentication.

Expected Result: Password reset should require verification from both email and mobile channels.

Goal 16: Relogging after the password is reset and logging out from different devices.

Use Case: Session Management

Test Case 1: Logout from Different Devices

Test Step: Log in from multiple devices, then log out from one device and attempt to re-login.

Expected Result: The system should manage sessions correctly, allowing login from one device while maintaining security.

Goal 17: Session ending after inactivity of the user (Time limit of 15-30 mins)

Use Case: Session Timeout

Test Case 1: User Inactivity Timeout

Test Step: Log in and remain inactive for the specified time limit.

Expected Result: The system should automatically log out the user after the defined inactivity period.

Goal 18: Maintaining logs (User activity like login and logout)

Use Case: Logging Verification

Test Case 1: User Activity Logging

Test Step: Perform various user activities, such as login and logout.

Expected Result: System logs should accurately capture user activities, including login and logout events.

Goal 19: No redirection to third-party websites

Use Case: URL Redirection

Test Case 1: Redirection Testing

Test Step: Attempt to manipulate URLs for redirection to third-party websites.

Expected Result: The system should prevent unauthorized URL redirection.

Goal 20: Warning the user when public networks are used.

Use Case: Public Network Warning

Test Case 1: Accessing System on Public Network

Test Step: Access the system from a public network.

Expected Result: The system should display a warning to the user regarding potential security risks when using public networks.

Goal 21: Verify GPS spoofing, Email ID spoofing, IP address spoofing, Caller ID spoofing, Referrer spoofing, ARP spoofing, etc.

Use Case: Spoofing Verification

Test Case 1: Spoofing Attempts

Test Step: Attempt various spoofing techniques, including GPS, Email ID, IP, Caller ID, Referrer, and ARP spoofing.

Expected Result: The system should detect and prevent spoofing attempts.

Goal 22: Monitor Trojan attacks by scanning incoming network traffic.

Use Case: Trojan Detection

Test Case 1: Trojan Traffic Simulation

Test Step: Simulate Trojan-like traffic patterns in the network.

Expected Result: Intrusion detection systems should detect and alert on simulated Trojan attacks.

Tools available for Pen Testing:

1. Fiddler- Category: Proxy server application
2. Nmap- Category: Port scanner
3. Wireshark- Category: Web vulnerability scanner
4. Metasploit- Category: Vulnerability exploitation framework
5. Nikto- Category: Web vulnerability scanner
6. John the Ripper- Category: Password cracking
7. Burp Suite- Category: Net Scanner
8. OpenVAS- Category: Vulnerability scanner
9. Aircrack-ng- Category: Password cracking
10. Kismet- Category: Packet sniffer