E Com Security Solutions

Application Security Assessment & Penetration Test - Procedure Manual

Planning and Preparation

In order to make the penetration test done preparation needs is a mandate and is therefore followed in the following ways:

- A kickoff meeting will be called between the Client organization and the penetration testers.
- The kickoff meeting shall discuss matter concerning the scope and objective of the penetration test as well as the parties involved.
- The scoping of the penetration test is done by identifying the machines, systems and network, operational requirements and the staff involved.
- The Application Security Assessment Questionnaire filled by the client organization captures all the requirements required to determine the scope of a Penetration test and the timing and duration of the penetration tests are performed.
- This is vital, as it will ensure that while penetration tests are being conducted; normal business and everyday operations of the organization will not be disrupted.
- SOW and Penetration Test Sign off
- Penetration test process will be initiated once the Client Organization approves our SOW before a penetration test is carried out.
- Confidentiality of Information and Data:
- Any information or data obtained during the penetration tests will be treated as confidential and will be returned or destroyed accordingly after the tests are performed. Prior to any penetration test engagements the client Organization approval and sign off is a mandate.

Application Penetration Test Methodology

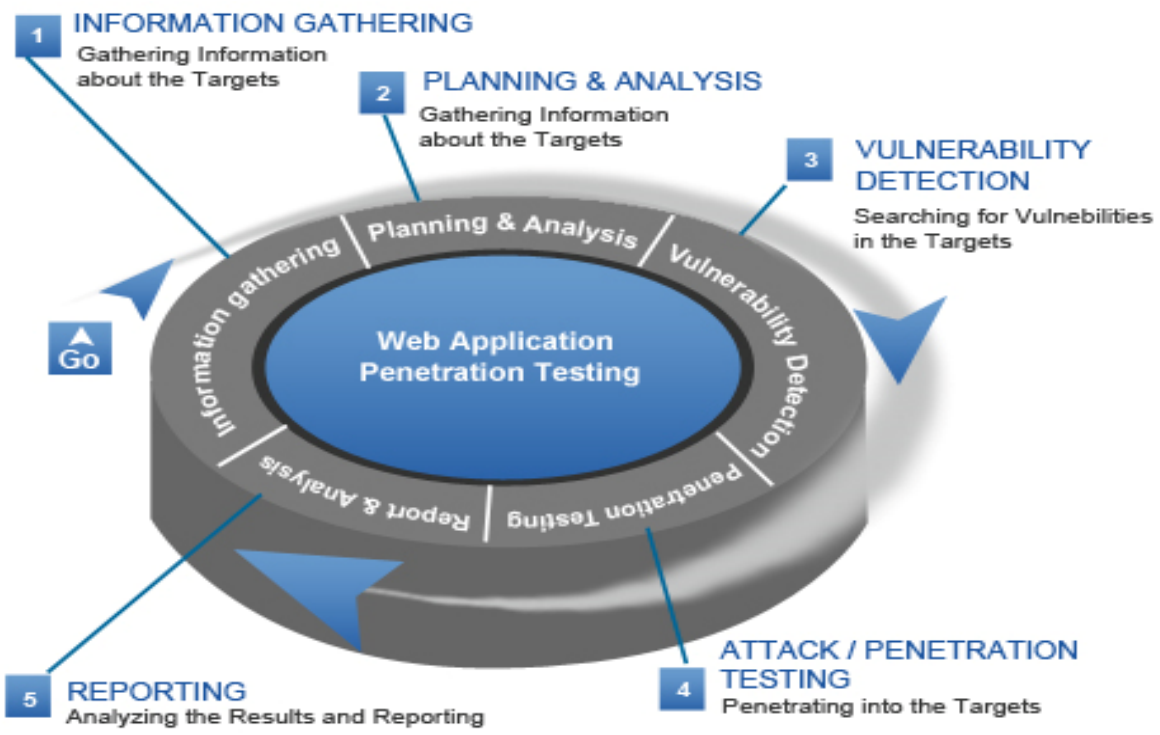
Web application penetration testing is carried out in various phases to ensure clear planning and delivery model. When performing external or internal penetration tests, E Com Security Solutions employs a standard 5-step methodology. This methodology allows for a systematic testing process that ensures all appropriate tests have been applied to the proper devices.

E Com Security Solutions Penetration testing methodology is based on industry best practices such as the OSSTMM (Open Source Security Testing Methodology Manual) and OWASP (Open Web Application Security Project). This ensures that you receive quality and repeatable results, and minimizes the risk to your systems under test.

Our team uses an arsenal of penetration testing tools similar to those used by attackers on the internet - in conjunction with best of breed open source penetration tools. Indeed, keeping up to date with the latest security vulnerabilities, trends and hacking techniques is our business.

We produce a comprehensive report covering the approach taken, the techniques applied, and the vulnerabilities identified and make procedural and strategic recommendations to ensure your systems are secure against future attack. E Com Security Solutions Penetration tests are performed by meeting the requirements and regulations such as PCI DSS, SOX, HIPAA and industry standards such as ISO 17799 and ISO 27001.

Typical phases of the application Penetration Test are depicted in the figure below:



Phase 1 & 2– Information Gathering, Planning and Analysis

- Determine the network components running web application
- Determine the technology and apply suitable tests
- Understand security objectives of the website
- Navigate through application and carry out threat modeling to identify possible threats and vulnerabilities
- Confirm that web applications running are suitable for testing.

The information is gathered in the following areas:

- Logistics (predominantly network and infrastructure related) – This phase of the Information Gathering process consists of browsing and capturing resources related to the application being tested.
- Operating System Fingerprinting – Operating system of the target is identified and confirmed
- Identification of actively listening ports – Ports are scanned and open TCP/UDP ports are identified
- Identifying the relevant services bound to the actively listening ports – The services and their versions bound to the actively listening ports are identified
- Web Server Fingerprinting – The Web Server software running is identified
- Application Discovery – More information about the application is gathered by analyzing the HTTP Headers, Error Pages analysis, Resources enumeration, HTML source sifting etc.
  Application discovery is an activity oriented to the identification of the web applications hosted on a web server/application server. This analysis is important because often there is not a direct link connecting the main application backend. Discovery analysis can be useful to reveal details such as web applications used for administrative purposes. In addition, it can reveal old versions of files or artifacts such as undeleted, obsolete scripts, crafted during the test/development phase or as the result of maintenance.
- Gleaning More Information – More information about the web site under test is harvested through search engines like Google
- Web service Information (if applicable) – Web Services are identified if there are any Identify application entry points - Enumerating the application and its attack surface is a key precursor before any attack should commence. This section will help you identify and map out every area within the application that should be investigated once your enumeration and mapping phase has been completed.
- Analysis of Error Codes - During a penetration test, web applications may divulge information that is not intended to be seen by an end user. Information such as error codes can inform the tester about technologies and products being used by the application. In many cases, error codes can be easily invoked without the need for specialist skills or tools, due to bad exception handling design and coding.

Clearly, focusing only on the web application will not be an exhaustive test. It cannot be as comprehensive as the information possibly gathered by performing a broader infrastructure analysis.

Phase 3 – Vulnerability Detection

- Identification of vulnerabilities related to the under lying architecture
- Identification of OS, web server and database server along with their versions
- Identification of known threats associated with the application versions
- Identification of open ports and services running on those ports
- Identification of security risks/threats associated with the open port and the service running on it
- Ensure that the web server does not support the ability to manipulate resources from the Internet (e.g., PUT and DELETE)
- Validate user authentication processes, password reset mechanisms and session management schemes
- Verify applications are properly configured to prevent unnecessary data from being revealed
- Identify strengths and weaknesses of web applications in terms of overall security
- Prioritize exposures that present greatest risk

Phase 4 – Attack(s)/Privilege Escalation

- Identification of all the web pages with input controls
- Perusal of client side code for disclosure of potentially compromising information
- Tests (both manual and automated) are carried out to identify vulnerabilities like HTTP Response Splitting, SQL Injection, Cross Site Scripting (XSS), Directory Browsing/Traversal, AJAX Testing and Parameter Tampering
- Checks for the presence of vulnerabilities related to the latest breed of technologies
- Checks for presence of encryption. Ensure that usernames and passwords are sent over an encrypted channel.
- Detect code information leaks through invalid input responses and unintentional d disclosure (error messages, etc)
- Checks to see if it is possible to access pages or functions that re quire logon but can be bypassed
- Determine the application logic to maintain the authentication sessions - number of (consecutive) failure logins allowed
- Ensure that where the application requires the user to perform actions in a specific sequence, the sequence is enforced
- Ensure that users are only asked to submit authentication credentials on pages that are served with SSL (if applicable)
- Ensure that no backup files of source code are accessible on the publicly accessible part of the application
- Ensure that common configuration issues such as directory listings and sample files have been addressed
- Check for existence of common directories within the application root
- Ensure the applications will not process operating system commands from the user

Phase 5 – False Positive Analysis

A False Positive is when you think you have a specific vulnerability in your program but in fact you don't. During vulnerability assessment, an application is scanned or service/daemon and attempt to find a vulnerability in it. Sometimes the signatures (the 'check logic') make mistakes and report a vulnerability that may not exist. During this phase, false positive analysis is performed to make sure that they don't exist.

Phase 6 – Post Assessment

This phase is performed to finalize the findings, gather information about the specific finding along with collection of evidences, and listing all the vulnerable points to a specific vulnerable finding.

Phase 5 – Reports & Deliverables

- All the findings are compiled in an easy to understand format and rated based on their criticality level
- Risk, Technical Impact and Business impact shall be determined based on the vulnerabilities detected
- Implications of the findings will be explained in detail
- All the findings are supported with the relevant screenshots and Video Evidences (wherever possible)

Web Application Penetration Test Checklist

To ensure quality and consistency of results, we follow a standard testing framework. The framework is designed to be comprehensive in nature to ensure all potential weaknesses in the application are properly assessed, but it is also flexible in testing methods and the specific tests that are performed to allow consultants to apply their expertise to the application being assessed. A summary of testing framework is below.

All applicable tests would be executed against the application and detailed results with sufficient evidences would be provided. E Com Security Solutions plans to execute tests listed are the table below was part of web application penetration testing for the application provided.

Web Application General Checks	
AUTHENTICATION	
DESIGN FLAWS	
1	Authentication by pass Tests
2	Check for default accounts
3	Transmission vulnerabilities
4	Distribution
5	Validation
6	Impersonation check
7	Reflection checks
IMPLEMENTATION FLAWS	
8	Open Login Mechanisms
9	Multistage Login Mechanisms
10	Storage
MANAGEMENT FLAWS	
11	Credential management
12	Privacy violation
13	Key exchange
14	Login Management
ACCESS CONTROL	
15	Privilege Management
16	Cross Site Request Forgery
17	Secret Parameters
18	Forced Browsing

19	Path Traversal
SESSION HANDLING MECHANISM	
SESSION TOKEN GENERATION	
20	session prediction
21	Session fixation
SESSION TOKEN MAPPING	
22	Session hijack
23	Token mapping
APPLICATION LOGIC	
24	Work flow patterns
25	User privilege and bypass
26	User access rights
27	Boundary value tests
APPLICATION FUNCTIONALITY	
28	Cache poisoning
29	Cross user defacement
30	Object hijack
CRPTOGRAPHIC EMPLOYMENT	
31	Cryptanalysis
32	Communication Security
33	Storage Security
WEB SERVER	
SERVER CONFIGURATION	
34	Environment related vulnerabilities
35	Credential management
36	Back-up Files and references
37	Web Server Components
38	Common Paths
39	Application Admin Interfaces
40	HTTP methods
41	Proxying
SERVER SOFTWARE	
42	Virtual Hosting
43	Canonicalization

44	Path Traversal
APPLICATION ERROR HANDLING	
45	Uncaught Exception
46	Unchecked Error condition
47	Data in HTML
48	Data Protection – SSL
49	Digital Certificate Validity and algorithms
50	Mis configuration
51	Information Leakage
52	Exception handling
INPUT DATA SANITIZATION	
53	Script Injection
54	SQL Injection
55	OS Command Injection
56	LDAP Injection
57	XSS Cross Site Scripting
58	Injection flaws
59	Cross Site Request Forgery
60	Debug mode
61	Thread Safety
62	Hidden Form Field Manipulation
63	Cookie Security
64	Dangers of HTML Comments
65	Malicious File inclusion
66	Insecure Direct Object Reference
67	Failure to restrict URL access
68	Insecure cryptographic usage
69	Insecure communications
70	Command Injection
71	HTTP Response Splitting
72	Arbitrary file creation
73	Arbitrary file deletion

74	Code execution
75	Cookie manipulation
76	CRLF Injection
77	Cross frame scripting
78	Directory traversal
79	File inclusion
80	File tampering
81	Full path disclosure
82	URL Redirection
83	Blind SQL / X Path Injection
INFORMATION EXPOSURE	
84	Trojan shell script
85	Cross site scripting in path
86	Directory permissions
87	HTTP Verb tampering
88	Directory indexing
89	Resource location
90	Path disclosure
91	Source code disclosure
92	Google hacking database
93	Path Traversal
Business Logic tests in web application	
94	Identify logical threats
95	Understand the functionality
96	Perform logical tests
97	Check work flow patterns
98	Verify user privilege
99	Verify flaws in user access rights
100	Verify bypass flaws

Reporting & Deliverables

Comprehensive web application penetration test report consisting of Executive Summary and detailed vulnerability analysis, recommendations with prioritized action plan

A. Summary

This section describes the objectives and scope of the work done as well as the methodology used for each phase of the penetration test performed.

B. General conclusions and recommendations

This section describes overall conclusions for each penetration test phase performed detailing the critical aspects of the customer's security infrastructure that should be modified or fixed in order to enhance the security of the components within the defined scope.

C. List of Vulnerabilities

For each of the vulnerabilities found, a vulnerability record will be presented using the following format:

Code	Description
Name	Name to identify the vulnerability
Risk level	An estimated risk level from 1 (Informational) to 5 (Critical), based on the particular characteristics of the organization's information infrastructure and business model.
Status	Actual state of the vulnerability, fixed (**F**), partially fixed (**P**), or unfixed (**O**).
Class	The exploitation of the vulnerability implies problems with the category reference: ENC Encryption Used AMG Account Management ATH Authentication And Authorization SES Session Management WEB Specific Web Server Vulnerabilities DSE Data Sanitization And Error Handling ID Information Disclosure APP Application Logic
Impact	A brief description of the impact of the exploitation of the vulnerability
Vulnerable systems	A list of vulnerable systems
Resources	Resources needed to exploit the vulnerability: **Common user (USR)**: This implies that the vulnerability's exploitation is trivial, or that there exists and exploitation code in the public domain. **Programmer (PGM):** Requires writing the exploitation code, but does not imply any profound knowledge of information security. **Hacker (HACK):** Requires a profound knowledge in the field of information security.
Description	A brief description of the problem
Recommendation	A detailed description to fix the issue along with any additional references.

E. Evidences

The Video and Image evidences will be provided along with the project report, which show cases the live demonstration of exploiting the vulnerability. The video consists of the audio, which will detail about the vulnerability and the exploitable sections within the application. This will help the developer understand about the vulnerability and the recommendation provided in the report to fix the vulnerable finding faster and effectively.