EC-Council Licensed Penetration Tester

Methodology: SQL Penetration Testing

Penetration Tester:			
Organization:			
Date:		**Location:**	

Test 01: List all input fields and hidden fields of post requests

Target Organization	
URL	
List of input fields and hidden fields of post requests	1. 2. 3. 4. 5. 6. 7. 8. 9. 10.
Tools/Services Used	1. 2. 3. 4. 5.

Results Analysis:

Test 02: Perform information gathering

Target Organization	
URL	
List the Information Collected such as	Database Name: Version: Users: Output Mechanism: Database Type: User Privilege Level: OS Interaction Level: Error Messages Info: Others: 1. 2. 3.
Tools/Services Used	1. 2. 3. 4. 5.

Results Analysis:

Test 03: Attempt to inject codes into the input fields to generate an error

Target Organization		
URL		
List of inject codes	1. 2. 3. 4. 5.	
Error generation by injecting codes into the input fields is successful	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 04: Try to find SQL injection vulnerabilities by interface

Target Organization		
URL		
GET Request and POST Request Parameters	1. 2. 3. 4. 5.	
Successfully modified the data in the GET Request	☐ Yes	☐ No
Response Received from the Web Server		
List SQL Injection Vulnerabilities	1. 2. 3. 4. 5.	
Tools/Services Used	1. 2. 3. 4.	

Results Analysis:

Test 05: Try to find SQL injection vulnerabilities by manipulating a parameter

Target Organization		
URL		
Values of the Parameter in the URL Field	**Original Value**	**Value Changed to**
Response Received		
List SQL Injection Vulnerabilities	1. 2. 3. 4. 5.	
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 06: Try to find SQL injection vulnerabilities using database errors and application response

Target Organization	
URL	
Error message returned from the web server	
Response returned by web server application	
List of SQL Injection Vulnerabilities	1. 2. 3. 4. 5.
Tools/Services Used	1. 2. 3. 4. 5.

Results Analysis:

Test 07: Perform fuzz testing to detect SQL injection vulnerabilities

Target Organization	
URL	
Inputs supplied to the application for testing	1. 2. 3. 4. 5.
Response Received	
List of SQL Injection Vulnerabilities	1. 2. 3. 4. 5.
Tools/Services Used	1. 2. 3. 4. 5.

Results Analysis:

Test 08: Perform function testing to detect SQL injection vulnerabilities

Target Organization		
URL		
Input Fields that can be Embedded in the SQL Query	1. 2. 3. 4. 5.	
Successfully tested Input Fields with Malicious Data to generate Errors	☐ Yes	☐ No
List of SQL Injection Vulnerabilities	1. 2. 3. 4. 5.	
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 09: Perform static/dynamic testing to detect SQL injection vulnerabilities

Target Organization	
URL	
List Static Queries present in the Source Code	1. 2. 3. 4.
List Attacking Patterns supplied as an Input to the SQL Query	1. 2. 3. 4.
List of SQL Injection Vulnerabilities	1. 2. 3. 4.
Tools/Services Used	1. 2. 3. 4.

Results Analysis:

Test 10: Perform black box pen testing

Target Organization		
URL		
Performed Black Box Testing Successfully	☐ Yes	☐ No
List of SQL Injection Vulnerabilities	1. 2. 3. 4. 5.	
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 11: Try to detect SQL injection vulnerability using automated web-app vulnerability scanners

Target Organization		
URL		
Successfully Identified the Vulnerable Source Code	☐ Yes	☐ No
List of SQL Injection Vulnerabilities	1. 2. 3. 4. 5.	
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 12: Perform a simple SQL injection attack

Target Organization		
URL		
Inputs Used	1. 2. 3. 4. 5.	
Performed a Simple SQL Injection Attack Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 13: Perform an error-based SQL injection attack

Target Organization		
URL		
Application's database error messages are disclosed to users	☐ Yes	☐ No
List of Vulnerability Exploit Query Requests Built	1. 2. 3. 4. 5.	
Performed an Error-Based SQL Injection Attack Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 14: Try to bypass website logins using SQL injection

Target Organization		
URL		
Bypassed Website logins using SQL injection Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 15: Perform SQL manipulation attacks using a WHERE clause

Target Organization		
URL		
Inputs Used	1. 2. 3. 4.	
Gained Access to a Database by inserting Exploited Query Statements with WHERE Clause Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 16: Perform UNION-based SQL injection

Target Organization		
URL		
Inputs Used	1. 2. 3. 4. 5.	
Performed UNION-based SQL Injection Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 17: Perform blind SQL injection attack

Target Organization		
URL		
Inputs Used	1. 2. 3. 4. 5.	
Performed Blind SQL Injection Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 18: Try to extract database name by blind SQL injection

Target Organization		
URL		
Performed Blind SQL Injection Successfully	☐ Yes	☐ No
Extracted Database Name		
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 19: Try to extract database users by blind SQL injection

Target Organization		
URL		
Performed Blind SQL Injection Successfully	☐ Yes	☐ No
List Database Users	1. 2. 3. 4. 5. 6. 7. 8. 9. 10.	
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 20: Try to extract column names using blind SQL injection

Target Organization		
URL		
Performed Blind SQL Injection Successfully	☐ Yes	☐ No
Extracted Table Column Names	1. 2. 3. 4. 5.	
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 21: Try to enumerate first table entry using blind SQL injection

Target Organization		
URL		
Performed Blind SQL Injection Successfully	☐ Yes	☐ No
First Table Entry		
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 22: Try to extract data from rows using blind SQL injection

Target Organization		
URL		
Performed Blind SQL Injection Successfully	☐ Yes	☐ No
Data Extracted from Table Rows	1. 2. 3. 4. 5.	
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 23: Determine privileges, DB structure, and column names

Target Organization			
URL			
Performed Blind SQL Injection Successfully	☐ Yes	☐ No	
User Level Privileges			
Admin Level Privileges			
Database Structure	**Table Names**	**Column Name Types**	**User Defined Tables**
	Others:		
Column Names	1. _____ 2. _____ 3. _____ 4. _____		
Tools/Services Used	1. _____ 2. _____ 3. _____ 4. _____		

Results Analysis:

Test 24: Try advanced enumeration techniques

Target Organization	
URL	
Advanced Enumeration Techniques Implemented	1. 2. 3.
Tables and Columns Enumerated	1. 2. 3.
Different Databases in the Server	1. 2. 3.
File Location of the Databases	1. 2. 3.
Others	1. 2. 3.
Tools/Services Used	1. 2. 3. 4. 5.

Results Analysis:

Test 25: Perform code injection attack

Target Organization		
URL		
Vulnerable Queries Used	1. 2. 3. 4.	
Performed Code Injection Attack Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 26: Perform function call injection attack

Target Organization		
URL		
Executed Vulnerable SQL Statements using Custom or Database Functions	1. 2. 3. 4.	
Performed Function Call Injection Attack Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 27: Perform buffer overflow attack

Target Organization		
URL		
Standard Database Functions that are liable to Buffer Overflow Attack	1. 2. 3. 4.	
Performed Buffer Overflow Attack Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 28: Try to grab SQL server hashes

Target Organization		
URL		
Queries Used to Grab SQL Server Hashes	1. 2. 3. 4.	
Grabbed SQL Server Hashes Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 29: Extract SQL server hashes

Target Organization		
URL		
Extracted SQL Server Hashes Successfully	☐ Yes	☐ No
List Extracted SQL Server Hashes	1. 2. 3. 4.	
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 30: Try to transfer database to a different machine

Target Organization		
URL		
Queries Used	1. 2. 3. 4.	
Transferred Database to a Different Machine Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 31: Extract OS and application passwords

Target Organization	
URL	
Target Database Running Privileges and Permissions	1. 2. 3.
Method Used to Interact with the OS	
Extracted OS and Application Passwords	1. 2. 3. 4.
Tools/Services Used	1. 2. 3. 4. 5.

Results Analysis:

Test 32: Access system files and execute commands

Target Organization	
URL	
Database Server Tested	
Functions used to Access System Files	1. 2. 3. 4.
System Commands Executed	1. 2. 3. 4.
Tools/Services Used	1. 2. 3. 4. 5.

Results Analysis:

Test 33: Try to perform network reconnaissance

Target Organization		
URL		
Commands or Utilities Executed	1. 2. 3. 4.	
Performed Network Reconnaissance Successfully	☐ Yes	☐ No
Network Information Gathered	1. 2. 3. 4. 5.	
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 34: Try IDS evasion using 'OR 1=1 equivalents

Target Organization		
URL		
Queries used to Evade the 'OR 1=1 Signature	1. 2. 3. 4.	
Evaded IDS Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 35: Try to evade IDS using Hex encoding

Target Organization		
URL		
Hex Equivalent of SQL injection Statements Used	1. 2. 3. 4.	
Evaded IDS Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 36: Try to evade IDS using Char encoding

Target Organization		
URL		
Queries with Char() function used to inject SQL Injection Statements	1. 2. 3.	
Evaded IDS Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 37: Try to evade IDS by manipulating white spaces

Target Organization		
URL		
Queries used with Manipulated White Spaces	1. 2. 3. 4.	
Evaded IDS Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 38: Try to evade IDS using In-line comments

Target Organization		
URL		
Queries Used with In-line Comments	1. 2. 3. 4.	
Evaded IDS Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis:

Test 39: Try to evade IDS using Obfuscated code

Target Organization		
URL		
Queries used with Obfuscated Code	1. 2. 3. 4.	
Bypassed Signature Detection of the IDS System Successfully	☐ Yes	☐ No
Tools/Services Used	1. 2. 3. 4. 5.	

Results Analysis: