EasyChair Preprint

№ 2100

Vulnerability Assessment Penetration Testing (VAPT) for Web Applications

Md Shahidullah

December 5, 2019

Vulnerability Assessment Penetration Testing for Web Application

Md Shahidullah

Bangladesh University of Professional

ABSTRACT:

Vulnerability assessment and penetration testing-(VAPT) provides a critical observation of organization OS-operating systems, web servers, DB-database servers, access points, and loopholes or back doors. Attackers use these vulnerabilities to exploit the victim's system. It is better to find out these vulnerabilities in advance before the attacker does. In this paper, we proved Vulnerability Assessment and Penetration Testing (VAPT) as a Cyber defense technology, how we can provide active cyber defense using Vulnerability Assessment and Penetration Testing. It gives a more detailed view of threats, loopholes, bugs, back doors so that the information security specialist fixes all these vulnerabilities and back doors will help to provide more security and better protection from malicious attacks. Vulnerabilities can be found in two ways, internal testing, and external testing. We described the complete life cycle of Vulnerability Assessment and Penetration Testing on systems or networks and proactive action taken to resolve that vulnerability and stop the possible attacks. We have described the complete process of how to use Vulnerability Assessment and Penetration Testing as a powerful Cyber Defense Technology. Meaningful information or data are transferred over the web, cyberattacks are increasing every day with the increased use of Web applications. Globally, statistics show that more than 70 per- cent of the applications either have vulnerabilities that could potentially be exploited by a hacker, or worse, or they have already been exploited. So it needs to be secure. The best way to secure our web application of the website is to hack ourselves or by conducting penetration testing. In this research paper, penetration analysis of web security issues of the website is presented, using Kali Linux, OWASP ZAP, Burp Suite, etc. VAPT ensures that organization applications, web servers, database servers brought back to the initial state. Top 13 attacks list published by OWASP (open web application security project).

Keywords— VAPT, Penetration Testing, SQL injection, information security, ethical hacking, cyber security, Injection, Broken Authentication, Sensitive data exposure, XML External Entities (XXE), Broken Access control, Security misconfigurations, Cross-Site Scripting (XSS), Insecure Deserialization, Using Components with known vulnerabilities, Insufficient logging and monitoring.

INTRODUCTION:

Vulnerability assessment can help identify the loopholes in a system while penetration testing is a proof-of-concept approach to actually explore and exploit vulnerability. To find the security loopholes in a Web application searching and finding, with the objective that none of the loopholes are missed. It primarily adopts a scanning approach which is done both manually and performed by certain tools. The outcome of a process is a report showing all vulnerabilities, which are categorized based on their severity.

OBJECTIVES OF VAPT:

- ✓ To find the security weakness.
- ✓ Helps in fixing Web Application Vulnerabilities.
- ✓ We can perform manual attack.
- ✓ It gives a great deal of accuracy to the results
- ✓ Provides evidence, enabling the replication of problems.
- ✓ Helps in Gathering Information of Hosting Servers, Users, sub domains etc.
- ✓ Mimics the behavior of real life hackers
- ✓ No false positives.

FUNCTIONALITIES:

Secure Coding Practices Checklist:

- ✓ Input Validation
- ✓ Output Encoding
- ✓ Authentication and Password Management
- ✓ Session Management
- ✓ Access Control
- ✓ Cryptographic Practices
- ✓ Error Handling and Logging
- ✓ Communication Security
- ✓ System Configuration
- ✓ Database Security
- ✓ File Management
- ✓ Memory Management
- ✓ General Coding Practices

Testing Checklist:

Information Gathering:

- ✓ Conduct Search Engine Discovery and Reconnaissance for Information Leakage
- ✓ Fingerprint Web Server
- ✓ Review Webserver Metafiles for Information Leakage
- ✓ Enumerate Applications on Webserver
- ✓ Review Webpage Comments and Metadata for Information Leakage
- ✓ Identify application entry points
- ✓ Map execution paths through application
- ✓ Fingerprint Web Application Framework
- ✓ Fingerprint Web Application
- ✓ Map Application Architecture

Configuration and Deploy Management Testing:

- ✓ Test Network/Infrastructure Configuration
- ✓ Test Application Platform Configuration
- ✓ Test File Extensions Handling for Sensitive Information
- ✓ Backup and Unreferenced Files for Sensitive Information
- ✓ Enumerate Infrastructure and Application Admin Interfaces
- ✓ Test HTTP Methods
- ✓ Test HTTP Strict Transport Security
- ✓ Test RIA cross domain policy

Identity Management Testing:

- ✓ Test Role Definitions
- ✓ Test User Registration Process
- ✓ Test Account Provisioning Process
- ✓ Testing for Account Enumeration and Guessable User Account
- ✓ Testing for Weak or unenforced username policy
- ✓ Test Permissions of Guest/Training Accounts
- ✓ Test Account Suspension/Resumption Process

Authentication Testing:

- ✓ Testing for Credentials Transported over an Encrypted Channel
- ✓ Testing for default credentials
- ✓ Testing for Weak lock out mechanism
- ✓ Testing for bypassing authentication schema
- ✓ Test remember password functionality
- ✓ Testing for Browser cache weakness
- ✓ Testing for Weak password policy
- ✓ Testing for Weak security question/answer
- ✓ Testing for weak password change or reset functionalities
- ✓ Testing for Weaker authentication in alternative channel

Authorization Testing:

- ✓ Testing Directory traversal/file include
- ✓ Testing for bypassing authorization schema
- ✓ Testing for Privilege Escalation
- ✓ Testing for Insecure Direct Object References

Session Management Testing:

- ✓ Testing for Bypassing Session Management Schema
- ✓ Testing for Cookies attributes
- ✓ Testing for Session Fixation
- ✓ Testing for Exposed Session Variables
- ✓ Testing for Cross Site Request Forgery
- ✓ Testing for logout functionality
- ✓ Test Session Timeout
- ✓ Testing for Session puzzling

Data Validation Testing:

- ✓ Testing for Reflected Cross Site Scripting
- ✓ Testing for Stored Cross Site Scripting
- ✓ Testing for HTTP Verb Tampering
- ✓ Testing for HTTP Parameter pollution
- ✓ Testing for SQL Injection

- ✓ Testing for LDAP Injection
- ✓ Testing for ORM Injection
- ✓ Testing for XML Injection
- ✓ Testing for SSI Injection
- ✓ Testing for XPath Injection
- ✓ IMAP/SMTP Injection
- ✓ Testing for Code Injection
- ✓ Testing for Local File Inclusion
- ✓ Testing for Remote File Inclusion
- ✓ Testing for Command Injection
- ✓ Testing for Buffer overflow
- ✓ Testing for Heap overflow
- ✓ Testing for Stack overflow
- ✓ Testing for Format string
- ✓ Testing for incubated vulnerabilities
- ✓ Testing for HTTP Splitting/Smuggling

Error Handling:

- ✓ Analysis of Error Codes
- ✓ Analysis of Stack Traces

Cryptography:

- ✓ Testing for Weak SSL/TSL Ciphers, Insufficient Transport Layer Protection
- ✓ Testing for Padding Oracle
- ✓ Testing for Sensitive information sent via unencrypted channels

Business Logic Testing:

- ✓ Test Business Logic Data Validation
- ✓ Test Ability to Forge Requests
- ✓ Test Integrity Checks
- ✓ Test for Process Timing
- ✓ Test Number of Times a Function Can be Used Limits
- ✓ Testing for the Circumvention of Work Flows

- ✓ Test Defenses against Application misuse
- ✓ Test Upload of Unexpected File Types
- ✓ Test Upload of Malicious Files

Client Side Testing:

- ✓ Testing for DOM based Cross Site Scripting
- ✓ Testing for JavaScript Execution
- ✓ Testing for HTML Injection
- ✓ Testing for Client Side URL Redirect
- ✓ Testing for CSS Injection
- ✓ Testing for Client Side Resource Manipulation
- ✓ Test Cross Origin Resource Sharing
- ✓ Testing for Cross Site Flashing
- ✓ Testing for Clickjacking
- ✓ Testing WebSockets
- ✓ Test Web Messaging
- ✓ Test Local Storage

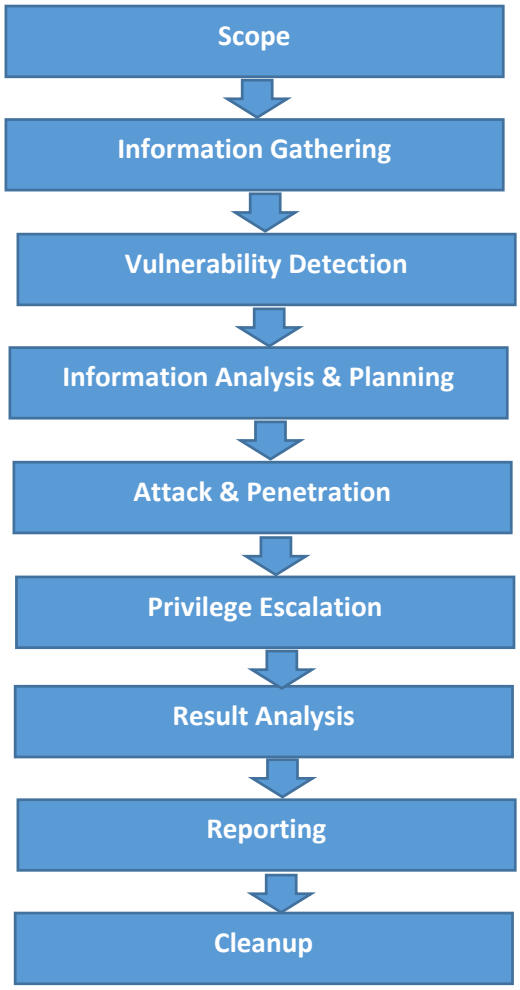
OWASP Top 10 Security Risks:

To bring awareness to what threatens the integrity of websites, we are continuing a series of posts on the **OWASP top 10 security risks.**

The OWASP Top 10 list consists of the 10 most seen application vulnerabilities:

1. Injection
2. Broken Authentication
3. Sensitive data exposure
4. XML External Entities (XXE)
5. Broken Access control
6. Security misconfigurations
7. Cross-Site Scripting (XSS)
8. Insecure Deserialization
9. Using Components with known vulnerabilities
10. Insufficient logging and monitoring

WORK FLOW OF PENTEST:



CONCLUSION:

In Kali Linux there are so many utilities and one utility might be better to perform a particular kind of job than the other utility. As a penetration tester or ethical hacker we must know which utility is good to perform the task better. We found that we cannot discover every type of vulnerability by using the single technique, so the pentester must think out of the box and do the penetration testing that will be beneficial for the client and if the cost of the penetration testing is more than the profit of the organization than there is no mean to conduct the penetration testing in such a case the pen tester must give the suitable alternate advice to the client.

REFERENCES:

https://www.owasp.org/index.php/OWASP_Secure_Coding_Practices_Checklist

https://www.owasp.org/index.php/Testing_Checklist

https://www.owasp.org/index.php/Web_Application_Security_Testing_Cheat_Sheet

https://github.com/OWASP/CheatSheetSeries

https://github.com/OWASP/CheatSheetSeries/issues/13

https://github.com/OWASP/CheatSheetSeries/tree/master/cheatsheets_excluded

https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets_excluded/PL_SQL_Security_Cheat_Sheet.md

https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets_excluded/Secure_SDLC_Cheat_Sheet.md

https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets_excluded/Security_Testing_Cheat_Sheet.md

https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets_excluded/Web_Application_Security_Testing_Cheat_Sheet.md

https://www.owasp.org/index.php/OWASP_Top_Ten_Cheat_Sheet

https://www.owasp.org/index.php/Secure_Coding_Cheat_Sheet

https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet

https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents

https://www.owasp.org/index.php/OWASP_Testing_Project

Mr. Nitumani Sarmah, Mahammad Hachan, Assistant Professor, Department of Computer Science & Electronics,

University of Science & Technology, Meghalaya, Department of Computer Science & Electronics,

University of Science & Technology, Meghalaya.

Jai Narayan Goela, BM Mehtreb School of Computer and Information Sciences, University of Hyderabad,

Hyderabad 500046, India Center for Information Assurance and Management,

Institute for Development and Research in Banking Technology, Hyderabad 500057, India.

Mohd. Muneer Khan M. Tech (Cyber Security), People's University BHOPAL | INDIA | ASIA