Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

Vulnerability Assessment and Penetration Testing

Gaurav Bhatia
Vivekanand Education Society Polytechnic,
Chembur, Mumbai

Om Bhatia
Vivekanand Education Society Polytechnic,
Chembur, Mumbai

Aryan Bhandare
Vivekanand Education Society Polytechnic,
Chembur, Mumbai

Vishnu Bagde
Vivekanand Education Society Polytechnic,
Chembur, Mumbai

Prof. Alka Prayagkar
Vivekanand Education Society Polytechnic,
Chembur, Mumbai

***Abstract*— This paper describes about the technical approach for manual web-app penetration testing for maintaining the security of the web applications . We will also look for OWASP top 10 vulnerabilities in detail and its exploitation. It also contains some courses that anyone can do for learning Penetration Testing and Vulnerability Assessment. The main purpose for writing this paper is to educate the people regarding vulnerabilities and cyber threats .**

Index Terms*— *Vulnerability, Penetration Testing, Security, Data, Protection

I. INTRODUCTION

Information is Wealth. Each and every bit of informa- tion has a cost in this digital world. All that information is stored in the form of Data in Internet. There are two types of data,Public and Private. The public data are resources that are available publicly on the Internet. Ex: data that results from a Google search query. The private data are the resources that are bagged behind a wall of authentication. Ex: Your email data. Emails are protected by wall of authentication which requires your user name and password to authenticate successfully. But what if someone can read your emails by acquiring your credentials from you without your knowledge? There comes the need for Web Application Security. Everything is web based now. Most of the Software has their own web app version too. But all the Web Applications are prone to Hacking. This is why, Web Application Penetration emerge as need of the hour. Website need a defense in depth approach to mitigate against the security flaws. It's essential to Penetration test every web application before it goes online and gets hacked by a Black Hat cyber warrior out there. Hackers constantly hunt for web app vulnerabilities. The best way to mitigate against the hacker attacks is to learn their methodologies. Here, we discuss about the most mandatory penetration tests that has to be done before the application is released and Techniques explaining how to perform those tests.

II. WHAT IS PENETRATION TESTING?

Penetration testing which is additionally called pen test- ing or ethical hacking, is that the practice of testing a computing system, network or web application to hunt out security vulnerabilities that an attacker could exploit. Penetration testing are mostly automated with software applications or performed manually. The methodinvolves gathering information about the target before the test, identifying possible entry points, attempting to interrupt in – either virtually or for real and reporting back the findings. The main purpose of ethical hacking or penetration testing is to identify security weaknesses. in systems or networks Penetration testing is also to test an organization's secu- rity policy, its adherence to compliance requirements, its employees' security awareness and thus the organization's ability to identify and answer security incidents. Also, the security issues that was identified or exploited during penetration testing process is provided to the organization's IT and network system managers, enabling them to make strategic decisions and prioritize remediation efforts. Penetration tests also sometimes called white hat attacks because during a pen test, the great guys attempt to interrupt in.

III. TYPES OF PENETRATION TESTING

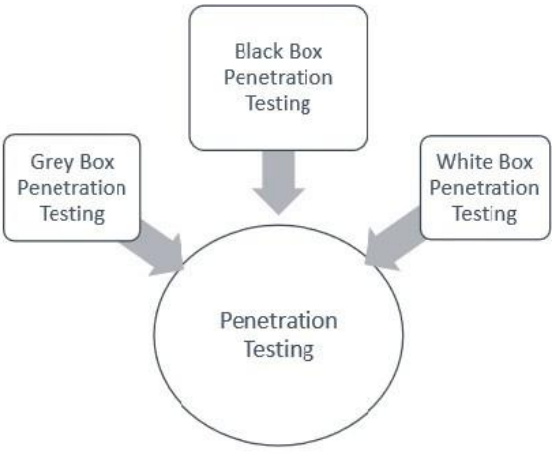


Fig. 1.Types of Penetration Testing

- **Grey Box Penetration Testing:** In this sort of test- ing, a tester usually provides partial or limited infor- mation

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

about the interior details of the program of a system. It are often considered as an attack by an external hacker who had gained illegal access to an company's network and infrastructure documents.

- **Black Box Penetration Testing:** In this type of penetration testing, tester has no idea about the systems that he's testing. He is interested to collect informa- tion about the target network or system. For exam- ple, in this type of testing, a tester only have the idea of what should be the expected outcome rather than knowing how the outcomes arrives.

- **White Box Penetration Testing:** It is a comprehen- sive testing, as tester has been given whole range of data about the systems and/or network like Schema, Source Code , OS details, IP address, etc. it's nor- mally considered as a simulation of an attack by an indoor source. it's also referred to as structural, glass box, clear box, and open box testing.

IV. WHAT IS VULNERABILITY ASSESSMENT?

A vulnerability assessment is a detailed review of security weaknesses in an data system . It checks if the system is vulnerable to any known vulnerabilities, allocate severity levels to those vulnerabilities, and suggest some solution or a patch , if and whenever needed.

V. TYPES OF VULNERABILITY ASSESSMENT

- **Host assessment:** The assessment of critical servers, which can be vulnerable to attacks if it's not tested properly or not generated from a tested machine.

- **Network and wireless assessment:** The assessment of actions and operations to stop unauthorized ac- cess to non-public or public networks and network- accessible resources.

- **Database assessment:** The assessment of databases or big data system for vulnerabilities and misconfigurations, identifying insecure development and testing environments, and categorizing sensitive data across an organization's infrastructure.

- **Application scans:** Hunting of security vulnerabilities in web-based applications and their source code by performing automated scans on the front-end or by static/dynamic analysis of source code.

VI. 4 STEPS OF VULNERABILITY ASSESSMENT

Step 1: Initial assessment

The goal here is to know the importance of devices on your network and therefore the risk related to each. Risk are often determined using several factors, including but not limited to:

i) Whether a given device is accessible to the net(whether via internal or external IP addresses)

ii) Whether the device is made accessible publicly (such as a kiosk machine)

iii) Whether a device's users have moderate-level or high- up permissions (such as administrators)

iv) The device's role in business processes

The determined risk prioritize the rest of the assessment and establish the vulnerability assessment scans in proper order. It also can be used as input for a business impact analysis that's a part of an enterprise risk management initiative.

Step 2: Define a system baseline

For each given device to be assessed for vulnerabilities, it's necessary to know whether its configuration meets ba- sic security best practices. a number of the configuration factors that need to be an area of a baseline include:

- Operating system (OS), version, and repair pack or build, if applicable
- Approved software
- Installed services and required ports
- Any necessary open ports
- Any special security configuration, if applicable

Approach each device as if you were a malicious actor; once you perform a scan within subsequent step, you'd wish to ascertain what an inside or external threat actor can access, and be able to compare that against known vulnerabilities and insecure configurations so you'll interpret the results of the scan properly. additionally to the configuration factors, gathering up any additional detail known about the system (such as log data pushed into a SIEM solution), and any already-known vulnerabilities for the precise OS and version, any installed applications or any enabled services, are getting to be useful.

Step 3: Perform a vulnerability scan

There are a couple of options available when it involves vulnerability scans. All provides a touch of various context to the results. Generally, vulnerability scans are per- formed either via unauthenticated or authenticated means. In an unauthenticated scan, a system is checked by the net- work point of view by trying to find open ports and testing for the utilization of exploits and attacks. An authenticated scan will perform a credentialed scan of the OS and applications trying to find misconfigurations and missing patches which will be taken advantage of by threat actors, like weak passwords, application vulnerabilities and malware. Part of the vulnerability assessment is only done from the attitude of getting an honest security posture. But, organizations in regulated industries or those subject to specific compliance laws got to consider scanning to supply that security-specific mandates are met. for instance , businesses accepting credit cards got to confirm that they meet requirements found in section 11.2 of the Payment Card Industry Data Security Standard (PCI DSS). Likewise, those businesses subject to regulations like insurance Portability and Accountability Act (HIPAA), the overall Data Protection Regulation (GDPR).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

Step 4: Vulnerability assessment and reporting

Reporting a vulnerability is critical because it indicates the output of the scan, the risk and importance of the de- vices and systems scanned, and the future steps that should be taken to patch the vulnerability. In vulnerability assess- ment, it's important that reporting must be actionable.

Reporting should include appropriate details that can be used to respond to found vulnerabilities, including:

- Vulnerability discovered
- Common Vulnerabilities and Exposure (CVE) reference and score should be specified clearly and vulnerabilities with a medium or high CVE score should be addressed immediately
- A list of systems and devices found vulnerable
- Detailed steps to solve the vulnerability, which can include patching and/or reconfiguration of operating systems or applications
- Mitigation steps (like adding automatic OS updates in place) to keep the same type of issue from happening again

Reporting provides an organization with a detailed understanding of their current security loop holes and what work is necessary to both fix the potential threat and to mitigate the same source of vulnerabilities in the future.

VI. OWASP TOP 10 VULNERABILITY AND ITS PREVENTION

1. INJECTION:

Injection vulnerabilities usually occurs when a query or command is used to insert malicious data into the inter- preter via SQL, OS, NoSQL, or LDAP injection. The hos- tile data injected through this attack vector tricks the in- terpreter to make the application do something it had been not designed for, like generating unintended commands or accessing data without proper authentication.

(e.g. SQL, LDAP, command line)

String query = "SELECT * FROM accounts WHERE

custID='" + request.getParameter("id") + "'";id = " '; drop table accounts – "

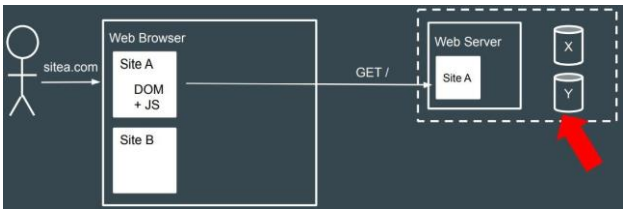


Fig. 2. Injection

Prevention:

SQL statements combine code and data
=> Separate code and data
- Parameterise your queries
- Validate which data can be entered
- Escape special characters

2. BROKEN AUTHENTICATION:

When an application execute functions incorrectly, related with session management or user authentication, intruders can be able to compromise passwords, security keys, or session tokens and permanently or temporarily assume the identities and permissions of other users. This vulnerability poses a grave threat to the safety of the applications and therefore the resources it accesses and may also severely compromise other assets connected to an equivalent network. The following points show us that if the application is vulnerable to broken authentication

- Weak session management
- Credential stuffing
- Brute force
- Forgotten password
- No multi-factor authentication
- Sessions don't expire

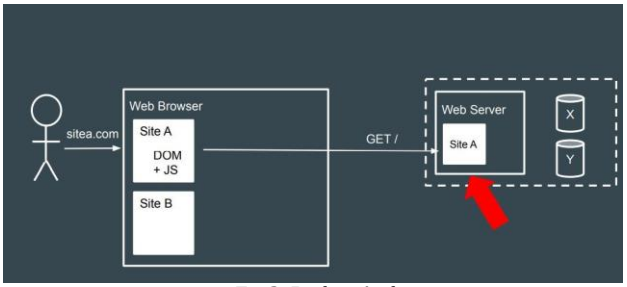


Fig.3. Broken Authentication

.

Prevention:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

- Use good authentication libraries
- Use MFA
- Enforce strong passwords
- Detect and prevent brute force or stuffing attacks

3. SENSITIVE DATA EXPOSURE:

Lack of data protection measures like encryption of data in transit or at rest, attackers can get access to your sensitive data like credentials, credit card or social security numbers, and medical information. Unencrypted data is a main target for damaging exploits associated with identity theft, fraud, and industrial espionage, to call just a couple of security vulnerability examples. Data protection is very critical for web applications that involve financial transactions, healthcare records, and personally identifiable information (PII). The following points show us that if the application is vulnerable to sensitive data exposure.

- Clear-text data transfer
- Unencrypted storage
- Weak crypto or keys
- Certificates not validated
- Exposing PII or Credit Cards

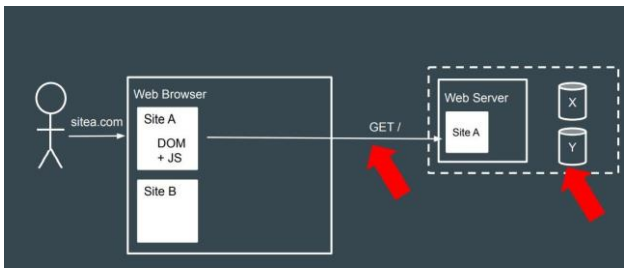


Fig.4. Sensitive Data Exposure

Prevention:

- Don't store data unless you need to.
- Encrypt at rest and in transit.
- Use strong crypto.

4. XML EXTERNAL ENTITIES (XXE) :

For web applications that parse XML input, a poorly configured XML parser are often tricked to send sensitive data to an unauthorized external entity, i.e., a storage unit like a tough drive. XXE attacks are conducted by hackers to watch critical information, internal files disclosure and file shares, internal ports scan, remotely execution of code, and mount denial of service (DoS) attacks.

e.g <?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
<!ELEMENT foo ANY >
<!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<foo>&xxe;</foo>

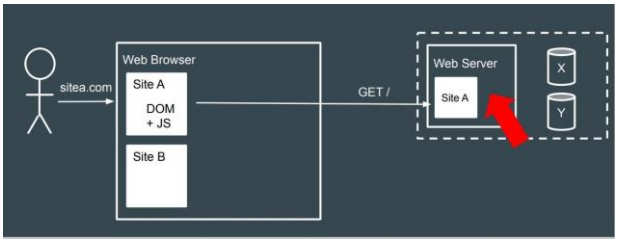


Fig.5. XML External Entities

Prevention:

- Avoid XML
- Use modern libraries, and configure them well!
- Validate XML

5. BROKEN ACCESS CONTROLS :

Website security access controls should limit visitor access to only those pages or sections needed by that sort of user. for instance , administrators of an ecommerce site got to be ready to add new links or add promotions. These functions shouldn't be accessible to other sorts of visitors.

Developers must be encouraged to internalize "security first" discipline to avoid pitfalls like content management systems (CMS) that generate all-access permission by default—up to and including admin-level access. Broken access control can give website visitors access to admin panels, servers, databases, and other business-critical applica- tions. In fact, this OWASP Top Ten threat could even be used to redirect browsers to other targeted URLs. The fol- lowing points show us that if the application is vulnerable to broken access control.

- Access hidden pages

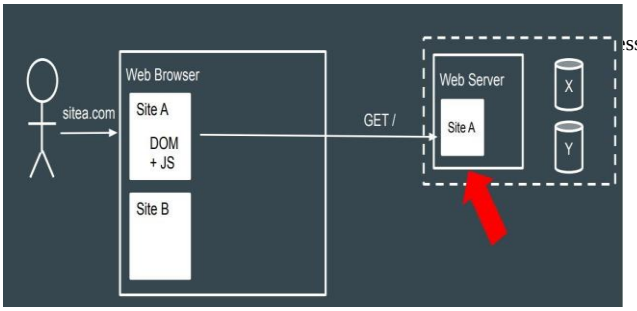
http://site.com/admin/user-management

- Elevate to an administrative account
- View other people's data

http://site.com/user?id=7

- Modifying cookies or JWT tokens Prevention:
- Use proven code or libraries
- Deny access by default
- Log failures and alert

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021



ss Control

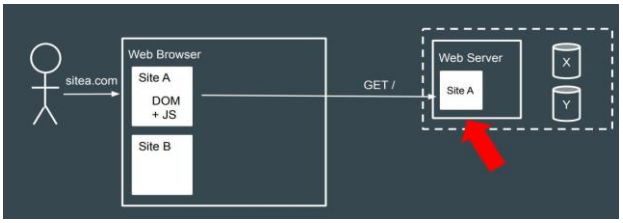
- Use proven code or libraries
- Deny access by default
- Log failures and alert
- Rate limit access to resources

6. SECURITY MISCONFIGURATION :

According to the study, approx 95% of cloud breaches are the results of human errors. Security setting misconfigurations are one among the prime drivers of that statistic, with OWASP noting that, of the highest ten, this vulnerability is that the commonest . There are many sorts of misconfiguration that expose the corporate to cybersecurity risk, including:

- Security features not configured properly
- Unnecessary features enabled
- Default accounts not removed
- Error messages expose sensitive information

Fig.7. Security Misconfigurations



Prevention:

- Have a repeatable build process or “gold master”
- Disable all unused services
- Use tools to review settings

7. CROSS SITE SCRIPTING (XSS) :

Cross-site scripting is vulnerability that spreads widely and can affects 53% of all web applications. It takes place when malicious client-side JavaScript or HTML scripts are injected in an internet page and then uses the online application as an attack vector to hijack sessions of users, damage victims website, or to redirect the victim to sites under the attacker’s control.HTML mixes content, presentation and code into one string (HTML+CSS+JS).If an attacker can alter the DOM, they can do anything that the user can do. It can be found using automated tools.

For e.g.

- <img src="javascript:alert(1)">
- <BODY ONLOAD=alert(’XSS’)>
- <script>alert(’XSS’)</script>

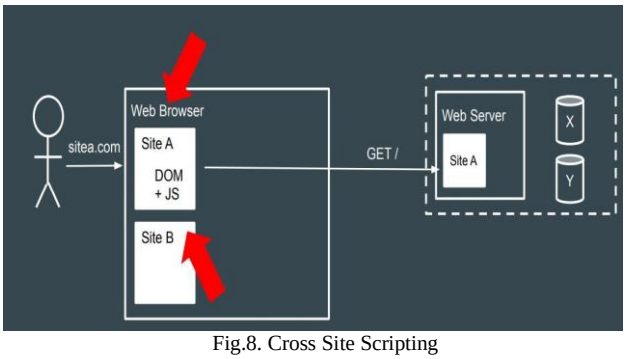


Fig.8. Cross Site Scripting

Prevention:

- Encode all user-supplied data to render it safeKirk <script> => Kirk <script>
- Use appropriate encoding for the context
- Use templating frameworks that assemble HTML safely
- Use Content Security Policy

8. INSECURE DESERIALIZATION :

Insecure deserialization offers hackers an attack vector that’s most typically used for remote code execution but also can be used to perform injection attacks, replay attacks, and attacks utilizing privilege escalation. An example of an insecure deserialization is the serialized information about the logged-in user is stored inside the super cookie . An attacker could deserialize the cookie, modify it to allow himself/herself as an admin role and than serialize it again.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

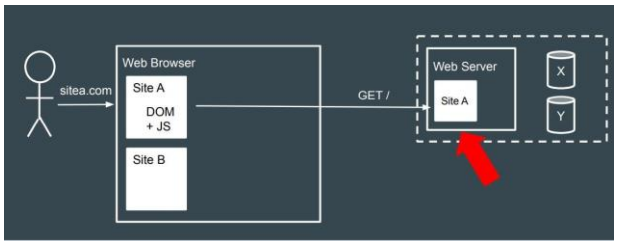


Fig.9. Insecure Deserialization

Prevention:

- Avoid serialising and deserialising objects
- Use signatures to detect tampering
- Configure your library safely
- Check out the OWASP Deserialisation Cheat Sheet

9. USING COMPONENTS WITH KNOWN VULNERABILITIES:

Modern distributed web applications often incorporate open-source components like libraries and frameworks. Any component with a known vulnerability becomes a weak link which will impact the safety of the whole application.Although the utilization of open-source components with known vulnerabilities is ranked low in terms of security problem severity, it's #1 when ranking the OWASP Top 10 by how often a vulnerability was the actual reason for an actual data breach.

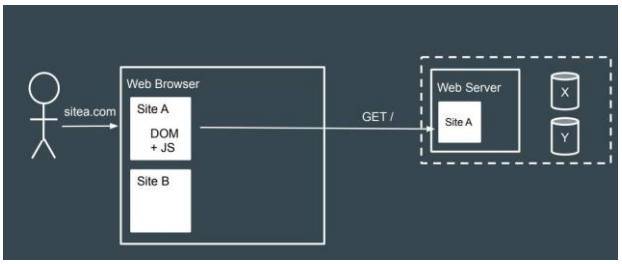


Fig.10. Using Components with Known Vulnerabilities

Prevention:

- Reduce dependencies
- Patch management
- Scan for out-of-date components
- Budget for ongoing maintenance for all software projects

10. INSUFFICIENT LOGGING AND MONITORING :

According to study, time from attack to detection can take approx 200 days, and often longer. Due to this cyber thieves get enough of time to perform changes within servers, corrupt databases, steal confidential information, and plant malicious code and backdoor. Logs are also important for:

- Detecting incidents
- Understanding what happened
- Proving who did something

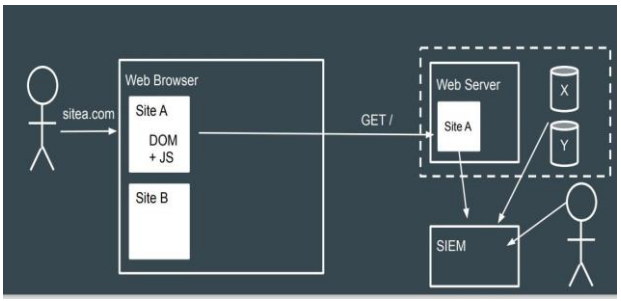


Fig.11. Insufficient Logging and Monitoring

VII. PLATFORMS FOR LEARNING VAPT

1. HackersEra University : This application is made by Vikas Chaudhary and provides various courses related to vulnerability and penetration testing.
2. Kongsec.io : This website is of aditya shinde which provides the course of hunting for critical vulnerabilities in web applications

VIII. CONCLUSION

Main conclusion on VAPT is that to make the softwares and networks safe and bug free. Also to reduce the cyber crime by educating or awaring the people on how to the hacker and how to remain safe from them. This paper also provides basic knowledge of bugs and how to prevent it from them.

IX. REFERENCES

[1] [Nagendran K, Adithyan A, Chethana R, Camillus P, Bala Sri Varshini K B ,"Web Application Penetration Testing" ISSN: 2278-3075, Volume-8 Issue-10.]
[2] [K. Nirmal, B. Janet And R. Kumar, "Web Application Vulnerabilities - The Hacker's Treasure," 2018 International Conference On Inventive Research In Computing Applications (Icirca), Coimbatore, India, 2018, Pp. 58-62.]
[3] [The Web Application Hacker's Handbook by Dafydd Stuttard , Marcus Pinto]
[4] [Hunter 2.0 course by Vikas Chaudhry]

[5] [WAPTT - web application penetration testing tool,
[6] Zoran Duric, DOI:10.4316/AECE.2014.01015,
[7] ResearchGate]