paper1

by yitisen yitisen

A Survey on Web Application Vulnerabilities

Matthi Naveen[*], Dr. Pragnyaban Mishra[**],

[*] Department Of CSE, KLUniversity, India
[**] Department of CSE, Associate Professor, *KLUniversity, India.*

Abstract- The world relies heavily on the Internet, and every organization and every human being uses web applications extensively for information sharing, business purposes such as online sales, money transfer, etc., and Exchange services. Nowadays, providing security for web applications is the greatest challenge in the corporate world because web applications will be the main way for their daily business and if the web application is affected, then daily business and reputation will be affected. As many organizations have been using the web application service to share or store sensitive information about their clients and assets. So, Web Applications are inclined to security attacks and new security vulnerabilities have grown in the last two decades in a web application and have become an important target for attackers. So, it is very vital to secure a web application and Major security issues will occur due to improper input validation at the client side which will take advantage by an attacker to inject malicious payloads. This paper reviews about vulnerability assessment and pretesting steps and types, website vulnerabilities like SQL Injection, Cross-Site scripting, file inclusion, cross-site request forgery, and broken authentication with types and remediations.

Index Terms- Cross-Site scripting, cross-site request forgery, file inclusion, penetrating testing, steps in VAPT, SQL Injection, vulnerability assessment, VAPT Types, Types of XSS, Types of SQL injection.

I. Introduction

Worldwide Web has advanced from a framework that delivers static pages to a stage that supports distributed and dynamic applications and web application has become the source for delivering information and service to the world. Web application advancements give a promising system of coordinating many useful segments over the web and so empower people and associations to cooperate utilizing application program interface along with enormous topographical separations. Billions of people everywhere throughout the world use web application advancements to exchange data, do money related exchanges, and have fun and do communication between them

Web application grew tremendously, and it has benefits and challenges like security associated and very important to address it. Security flaws in a web application will arise due to flaws in designing, and implementation which will be a threat to business. A web application is an important target for cyber-criminals due to their public access and increasingly used to deliver critical services to the users. Most of the websites will use a database as backend which helps to manage data easily and this data will also contain sensitive information, if the web application is compromised then this sensitive data would be infringed, resulting in severe economic damages, ethical and legal implications.

The Web platform is a dynamic architecture that involves various components and technologies such as HTTP protocol, web server, and application development technologies, web browser. For developers with insufficient security vulnerabilities, knowledge or awareness results in a high rate of websites sent on the Web are uncovered to security vulnerabilities. According to a report by the Internet Application Security Consortium, around 49% of applications being investigated contain vulnerabilities of a severe hazard level and 13% or more applications can be compromised naturally[3]. A later report uncovers that over 80% of the websites on the Web have had at least one high level of Vulnerability. Vulnerability refers to a weakness in a system's security need, design, coding or operation that could accidentally occur or intentionally violated which results in a security flaw. In the last few years, the web application security vulnerabilities have increased and new vulnerabilities are rising and the most commonly found vulnerabilities are SQL injection, cross-site scripting, command-line injection, cross-site ask forgery, and malicious file upload and execution [14].

This paper is presented as follows chapter one is an introduction , chapter two is discussed about the basics of web applications, chapter three is discussed about the VAPT steps and types, chapter four is discussed about the web application vulnerabilities introduction and discussed vulnerabilities, types and remediation and the final chapter is a conclusion.

II. Introduction to A Web Application

The Web Application Security Consortium (WASC) defines a web application as "a software application, executed by a web server, which response to dynamic (or) static web page requests over HTTP" [3]. Basically, websites will consist of style sheets, HTML pages, images, client-side scripts, etc. that will be available in a web application server.

Web applications will use the Internet Infrastructure for exchange of information and services between the organization and end-user, for a good introduction to the web application from the penetration tester's perspective, the information can be gathered from Web Consortium [3].

Now a day's web application will consist of n-tiered architecture. Where client will be a browser, an application server is where the code is deployed, and a back end will be a database in order to manage and store data and Figure 1 represents the communication flow of web application requests and response. and There may be a firewall, proxy servers, WAF's in between a web server and browser for added security.

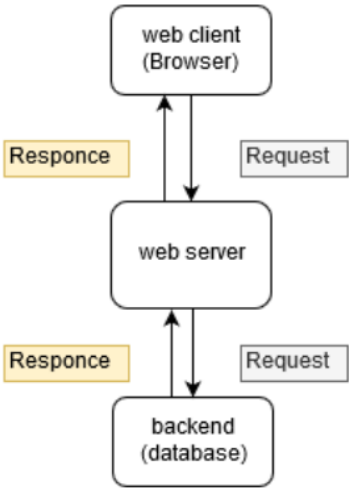


Figure 1. Web Application Environment.

III. OVERVIEW OF VAPT

A. Steps in VAPT

1) Pre-Engagement Interactions or Scoping:

In this phase, the pen testing team will discuss in detail the scope of the assessment, objectives, legal implications, goals and organizational assets available will be disused in detail. Penetration testers should collaborate with the organization to understand any threats, organizational culture, and best suited pen-testing technique, and legal issues associated will also be addressed for avoiding future inconvenience. In this process, organizational assets will have to categorize based on assets in scope (assets on which penetration testing must do) and assets out of scope (not included in penetration testing) based on all organizational assets.

2) Reconnaissance:

Reconnaissance or Open Source Intelligence gathering is one of the important steps in penetration testing. Here the tester will aim to gather all required information and the potential targets for exploit required about client organization. the pen tester will have different degrees of organizational information collected during the first phase and may also find extra critical information on their own to locate hidden vulnerabilities and entry points in the system.

In generally pen testers will use the vast range Reconnaissance techniques which include searching in different search engines, Domain name searches or WHOIS lookups, social engineering, searching for tax records, internal Footprinting for an email discussion, usernames, and social networks accounts, tailgating, external Footprinting for port scanning, reverse DNS, packet sniffing, etc.

3) Threat Modeling & Vulnerability Identification:

The Reconnaissance will act as an information-gathering phase in the threat modeling and Vulnerability Identification and this phase will also be a pre-attack phase. In this phase pen testers will think like attackers and will scan the system as deep as they can and identifies different targets and also maps organizational attack vector based on threats, here targets will be business assets like employees data, customer data, technical data, and threats will be internal threats like employees, vendors, distributors, board members, stakeholders, etc. and external threats like Ports, Network Protocols, Web Applications, Network Traffic, attackers, etc.

The internal threats will be controllable by the organization, but external threats will be out of control. In this phase, the pen testers will use different automated tools and manual testing tools to scan all the organization in-scope assets. After completing the information gained will be the different vulnerabilities in the servers, web application and all the in-scope assets.

4) Exploitation:

The pen tester team will try different exploits contained in a network, applications, and data with a list of all potential vulnerabilities and entry points gathered in Threat Modeling & Vulnerability Identification. The main aim of the pen tester is to verify how far they can get access to the system and find high-value targets without being avoid detected.

The pen tester will exploit the system based on scope defined and the penetration testers will use the standard exploits like Web Application exploits, Network exploits, Memory-based exploits, Wi-Fi exploits and will also use Physical Attacks and Social engineering attacks. In this phase, the system has to exploit after developing the threat vector and attack plan based on vulnerabilities to get access to system and sometimes the system can have the secure network which has DMZ, firewall, honey pots, and honey well so that the pen tester should use different evacuation techniques to bypass these security devices.

5) Post-Exploitation:

Upon completion of the exploitation process, the next aim is to record the methods used to get access to valuable information about the organization. The pen tester should take all the pieces of evidence required to generate the report and after collecting the evidence the team should clean up the system to revert the activities done during the exploitation phase. The cleanup activities will include Removing any executables, scripts, and temporary files, restoring system and network settings, removing any types of malware codes injected, etc. and should make sure that all installed backdoors or rootkits should be removed, and it should make sure that system configuration is restored to its original state at pre-engagement state. Any credentials that have been changed should be restored, and any additional usernames created should be removed.

6) Reporting:

This report is the best method to convey the findings of a pen test. This report will address the managers and the technical team. From the manager's perspective, they will have information like different vulnerabilities and their Business impact on the system. In the technical team perspective, they will have information like different vulnerabilities with its remediation.

The pen test report will begin with a summary that outlines all organizational business-related penetration test plans, defining outcomes by risk ranking. This section should be concise, and it could be the client's most important piece of decision-making and the business team can decide what to correct and what problems pose risk. The remaining part is a technical detail, which will be descriptive, specific and generic or ambiguous statements that help the technical team to resolve security issues.

7) Resolution & Re-Testing:

In this phase, the technical team will resolve the issues and they will get the assistant from VAPT team to solve the issues if they have any problem and Once vulnerabilities have been remediated, the client has to retest their systems to make sure that fixes were successful and whether the new vulnerability was created as a result of remediation or not. And the vulnerability assessment and penetration testing should also be conducted whenever there will be any modification in the system for finding new vulnerabilities.

B. types of VAPT

1) Black box pen testing:

Black box testing is carried out without the knowledge of evaluating infrastructure so pen tester will act as a real-world hacker. The pen tester team will scan the entire infrastructure as an outsider to find vulnerabilities. Black-box penetration testing is based on a detailed review of all available resources in the systems. A black-box penetration tester should know both automated and manual testing methodologies. Generally, this is the best approach because it helps pen testing team to think out of the box and do testing in all levels according to their realistic knowledge and ability and they will also use all the techniques and methodologies available for them by simulating the level of persistence, knowledge, and ability that a real-world hacker can do.

2) White box pen testing:

Pen testers will have full knowledge of the infrastructure and its internal design and working and known as clear box testing. The penetration tester seeks to obtain as much input to get more knowledge and better understand the system so that they can further expand their penetration tests. The main challenge with white-box testing is to analyze a large amount of data and extract data to find possible vulnerability points, making it the most time-consuming. In white-box testing, we can find logical errors, a syntax error or typographical errors, design flaws, human errors, etc. While performing the white box testing the pen testers has full access to the system for performing audits on the high-risk area.

3) Grey box pen testing:

Grey box pen testing is also known as translucent testing, more efficient but time-consuming because sometimes testers will test every single input path or field. the tester will have limited information on the internal working of the system and source code. the grey box testing will not be useful during the development phase because in the end-user perspective grey box pen testing is carried out that aims to analyze the front-end functionality and the internal workings of the system. grey box pen testing will have benefits of both white and black box pen-testing. grey box pen testing is very effective for web application testing, business domain testing and security assessment.

IV. WEB APPLICATION VULNERABILITIES

Vulnerability is a weakness in the application which can be a design flaw or implementation bug that allows an attacker to cause harm to users of an application [15]. Formally, vulnerability is defined as "The existence of a weakness, design, or implementation error that can lead to an unexpected, undesirable event compromising the security of the computer system, network, application, or protocol involved" [5].

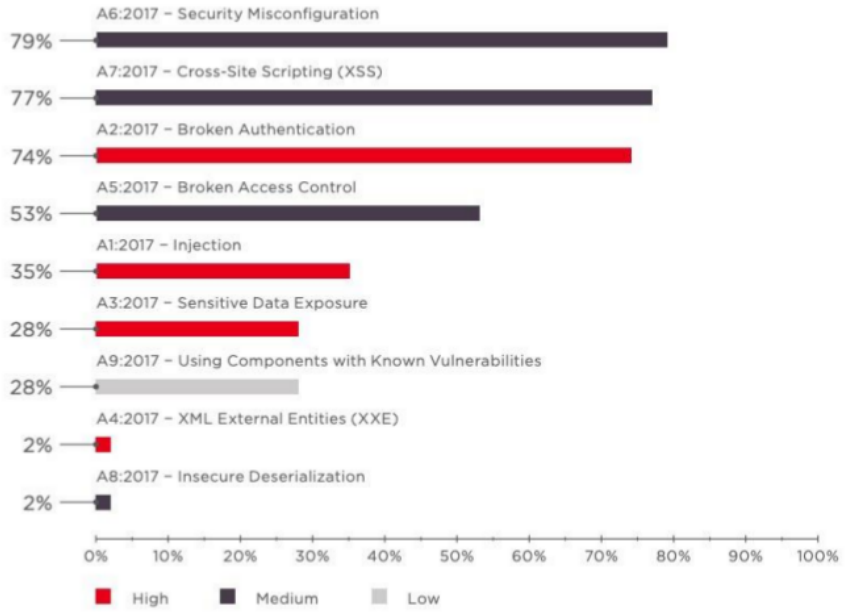
In many web applications, the Vulnerabilities will arise due to poor design strategies, configuration mismanagement, the complexity of software, accepting unsanitized input from the user, weak authentication management and features misuse. The impact of vulnerabilities depends on the severity of the vulnerability. Vulnerability assessment and pen testing should be done periodically for identifying,

classifying, remediating and mitigating vulnerabilities that arise in web applications to secure the web application. the testing guidelines for web applications will be given by the organization like OWASP, OSSTMM, ISSAF, Microsoft, etc. The commonly used standard OWASP and classification (TOP 10) of a web application vulnerability is given in the below table and OWASP will also release the testing guidelines with a checklist also.

OWASP TOP 10(2017)	
A1	Injection
A2	Broken Authentication
A3	Sensitive Data Exposure
A4	Xml External Entities (XXE)
A5	Broken Access control
A6	Security misconfiguration
A7	Cross-site Scripting
A8	Insecure Deserialization
A9	Using components with known Vulnerabilities
A10	Insufficient logging and Monitoring

Table1. OWASP (2017) Top Ten Web Application Security Risks

In 2018, around 70 types of weaknesses in web applications are found. As always, Cross-Site Scripting (XSS) vulnerabilities are present in many web applications [1]. Four out of five web applications contained configuration errors such as default settings, standard passwords, error reporting, full path disclosure, and other information leaks that might have value for potential intruders [10]. More applications are vulnerable to information exposure. Access to configuration and debug information, source code, session identifiers, and other sensitive information is possible in 79 percent of web applications [10]. This is concerning when compared to past years such as 2016 (60%) and 2017 (70%).



The percentage of an application based on vulnerabilities severity (high, medium, low). in the span of 4 years is shown in below image and the major of web applications are affected with a high level of vulnerabilities only

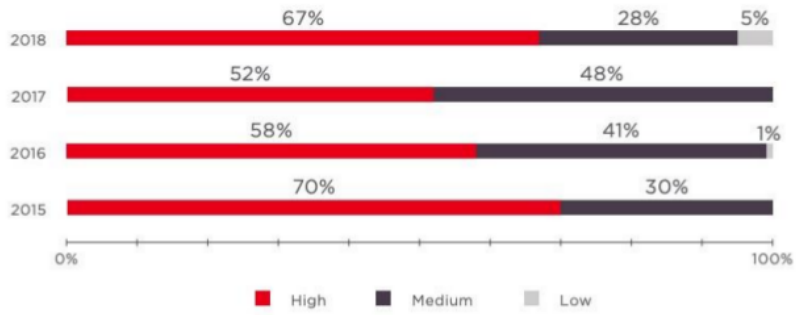


Figure 2. Percentage of Web Application Affected yearly

The different high-level vulnerabilities are disused below:

A. SQL Injection:

The attacker will use dynamic SQL statements in the SQL Injection Attack to comment on the components of the declaration or to add a statement that will always be conditionally true. The attacker uses the design faults in a web application to exploit SQL statements by injecting malicious SQL code and SQL Injection is also one of the injection attacks [7].

Usually, SQL injection happens when input is taken from a user, such as a username, password fields, id parameters, etc. The attacker will inject the SQL statements which will directly be executed by SQL parser and attacker will try to control a database server to retrieve data and can also use for bypassing authentication schema.

SQL injection vulnerability will be found in applications that are implemented using the SQL query language, using back end databases like MySQL, Oracle, SQL Server, or others. SQLI is a common attack vector by this means the malicious users can pass the malicious payloads to SQL parsers to manipulate or access confidential, sensitive information like customer information, personal data, etc. stored in a database. SQL Injection attack is the most dangerous Vulnerabilities in web applications.

An attacker must first discover vulnerable user inputs fields and parameters in the web application to do an attack and user input fields will be used to pass payload for performing SQL Injection attack. SQL injection also termed SQLI. Input content crafted by the attacker for injection is referred to as a malicious payload and it is the main component of the attack. After that malicious user tries to change the parameters with malicious payloads than database executes malicious SQL commands and gives an output with database content relevant to the payload.

SQL is a query language for managing data stored in relational databases. And it will use to access, edit, and delete data in the database. Many websites and web applications manage all the data in SQL databases. You can also use SQL commands to execute an operating system level commands in some instances. An effective SQL Injection attack can, therefore, have very severe implications.

- Attackers can use SQL Injection to identify other user's credentials in the database.
- SQL allows you to manage the information available in the database. An SQL Injection vulnerability could give full access to all information on a database server for malicious users.
- SQL also allows you to change information and add new information to a database. For instance, malicious users can use SQL Injection in a financial application to change balance, void transactions, or transfer cash to their account.
- To delete documents from a database, you can use SQL, even to drop tables as well. Even if database backups are made by the administrator, data deletion could influence the accessibility of the application until database restoration is done. Backups may not also contain the latest information.
- In some database servers, you can use the database server to access the working system. This may be accidental. In this case, malicious users will try to scan the internal network using SQL Injection.

Types of SQL injection

1) In-band SQL Injection

The attacker utilizes the same communication channel to launch their attack and collect outcomes. The simplicity and effectiveness of In-band SQLI make it one of the most popular SQLI attacks [11]. This technique has two sub-variations.

- **Error Based SQL Injection:** The malicious user tries to pass the database commenting statements in the request which will generate the database related error message in the response's then malicious user will pass the further commands based on the error message generated. By using the error messages malicious user can gain information like types of database using, version which can be used for the further attack.
- **Union Based SQL Injection:** The malicious user will try to fuse different UNION based SQL command with select statements which will be passed to the database. The database will execute the query and response based on injected payload.

2) Inferential (Blind) SQL Injection

The attacker sends payloads to the database server and observes the server's response and behavior to more information about database structure because in blind SQLI attacker cannot see or get much information in response [11]. This injection will depend on the change

and behavior in response from a web application server, to do these types of attacks consume a lot of time because the attacker has to retrieve the character by character but can be just as damaging. The following can be types of blind SQL injection:

- **Boolean based SQL Injection:** The attacker will construct Boolean based payloads and sends them to the database through SQL statements in such a way that the response from the database will change based on the result of the condition. Based on the result the response data will change. By observing the change in response the attacker can retrieve information from the database.
- **Time-based SQL Injection:** In time-based SQL injection techniques, the attacker will try to inject payloads which will make a database to give a response with time delay based on payload. Here the attacker can use SQL methods like SLEEP (), BENCHMARK (), WAITFORDELAY (), etc. which have specific special DBMS function (or) can use heavy query as payloads which will help to generate the time delays. Based on the time delay in the response, the attacker can guess information.

3) Out-of-band SQL Injection

Out - of-band SQLi will be used only if the attacker is unable to start the attack and retrieve the data using the same channel, or if a server is unstable to do perform SQL Injection. in this method, the attacker will rely on the server's ability to generate DNS or HTTP requests which help to transfer the data gained from SQL Injection [11]. The out-of-band SQL injection will only possible if and only if the relevant features are enabled and have access has database server for end-user.

The best Preventive measures for SQL Injection are:

1. Instead of placing user-provided input directly into SQL statements, we use prepared statements with parameterized queries, stored procedures can also be used which is harder to implement but a much effective way.
2. Sanitize user-provided inputs to properly escape the special characters and should verify that the input provided is in a pre-defined format or not.
3. Don't leave or transfer sensitive data in plaintext while storing in the database instead of Encrypt or hash the confidential data which will give a further level of protection if the attacker successfully enters the system.
4. Restrict the rights and privileges of the database by reducing the user's capabilities to the bare minimum. This will restrict intruders or attackers if they succeed in gaining access.
5. the common database error messages should not be displayed directly to end-user instead the custom error messages should be displayed for end-user
6. By using the WAF (Web Application Firewall) for web applications with database access which helps to protect web-based applications from malicious payload injection attempts by identifying and blocking malicious payloads based on firewall configuration rules.

B. Cross-site scripting:

Cross-site Scripting (XSS) is a client-side injection attack where malicious users try to execute the scripts in the browser by injecting malicious payload in the legitimate web application [4]. Every time the users accessing the links that are injected with the malicious script then the real attack will happen.

the web application has become a great way to execute a malicious script to the user browser. Usually, the attacker will target the web application with forums, message boards, and web pages that allow comments, search boxes, input fields will be targeted by attackers to inject malicious payload for performing cross-site scripting [2]. At first, the attacker tries to find web pages vulnerable to XSS and tries to inject the malicious payload in the vulnerable pages whenever the user tries to load that page then the malicious payload will be executed at victim browser and JavaScript will access the cookies and sends to attacker and by using these cookies the attacker can impersonate the victim by using session hijacking attack as shown in Figure 1.

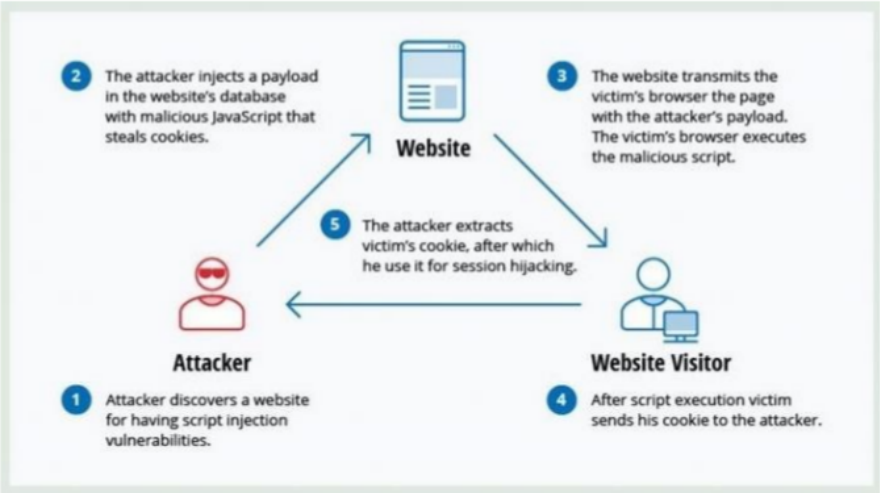


Figure 1. Demonstration of an XSS attack.

By using the XSS the attacker may damage the website instead of targeting the user and the attacker can also use injected malicious scripts to change the content of the website and may even redirect to other websites or websites with malicious content Vulnerability is regarded to have less impact than SQL injection. At first, the consequences of the ability to run JavaScript on a web page might not seem severe. Because most modern web browsers run java scripts in a controlled environment and have limited access to the systems files. But if JavaScript part of malicious content, also dangerous as Malicious JavaScript can also access all objects that the remaining web pages can access. This includes accessing the User cookies that often used to store data related to the session. If a malicious user succeeds in obtaining the cookies of a user then the malicious user will access the user accounts to do a malicious action on behalf of the user and can access the sensitive information and JavaScript can use the XMLHttpRequest object to send arbitrary HTTP requests to destinations by using HTML5 APIs in modern browsers which can be used for accessing files from user file system, geolocation, webcam, and microphone. Most of these APIs need granting permission from the user to execute in the browser, but the malicious user may use social engineering to address that restriction.

1) Reflected Cross-Site Scripting (XSS):

In Reflected cross-site scripting, the attacker will inject the malicious script in the request and send the request then the web application server will process the request and will send the response to the client with injected payload in an unsafe way which will lead to the execution of script code at client-side browser. Reflected XSS is also known as non-persistent or TYPE II XSS [8]. here script injected by a malicious user will be executed on the same place or page where injected so in generally malicious user will be delivered XSS to the victim by the other means like email messages with malicious links etc. whenever the malicious links are visited by the victim then the original attack will happen.

2) Stored Cross-Site Scripting (XSS):

In stored Cross-Site Scripting the injected payloads will be saved in the affected servers such as a database, message forum, server logs, comment field, etc. stored XSS is also known as persistent Cross-Site Scripting or type I Cross-Site Scripting [8]. In stored Cross-Site Scripting users whoever clicks on a link of the infected page will be affected and sometimes admin can also be affected. the payload injected on infected pages will be executed continuously until it has been deleted from its source.

2 DOM Based Cross-Site Scripting:

DOM-based XSS is an advanced XSS attack) [16]. It is possible if the web application's client-side scripts write data provided by the user directly to the Document Object Model (DOM)in the unsafe way and data stored by DOM is then read by the web application and outputted [9]. If the data is incorrectly handled, an attacker can inject a payload which will be stored as part of the DOM and executed when the data is read back from the DOM [9].

It has two main components they are sources and sinks; the sources are the location where the payload is being injected and the sinks are the location where the payloads are being injected. the popular sources and sinks are given in the image.

Sources	Sinks
document.URL	eval
document.documentURI	setTimeout
location	setInterval
location.href	document.write
location.Search	document.writeIn
Location. Hash	innerHTML
Window .name	outerHTML
Document.referrer	location.href

Table 2:Popular Sources And Sinks.

2 A DOM-based XSS is a client-side injection attack because the payload injected will be executed in client-side DOM and shows response immediately without the intervention of the server. which makes it difficult to detect by the security mechanisms at the server-side.

Impact of Cross-Site Scripting (XSS):

The impact of XSS on web applications will be minimal if there is no confidential information and no dynamic content change based on role.

The impact will be critical on the web applications which has sensitive data like banking transactions, emails, and health records, the impact will typically be significant and based on rights the victim has for example If the victim has admin privileges then the malicious user can take full control over the application.

○ Session Hijacking:

The most popular XSS attack vectors are stealing the victim's session cookies to hijack the victim's accounts. By using which malicious user will try to impersonate victim account and access any sensitive data or features on behalf of victims.

○ Stealing credentials:

The attacker will use HTML pages and JavaScript to steal user or victim credentials, instead of obtaining their cookies cloning the login page and then using XSS attacker steal credentials from the user. This situation will be in favor of an attacker because the attacker will ultimately acquire plaintext credentials instead of expiring ephemeral session cookies.

○ Targeting Sensitive Data:

Another strong XSS attack vector is to use it to exfiltrate sensitive data or to do unauthorized activities.

- **As Key loggers:** Using JavaScript, all keystrokes entered by a user on a vulnerable site can be logged. For this purpose, Metasploit involves an off-the-shelf payload. There are also some commercial websites offering JavaScript software that records all visitor motions, clicks, mobile gestures, or inputs in form fields that will be used for malicious reasons.
- **Port scan:** XSS is also an unexpected source for port scans to be initiated against a victim's internal network by accessing a vulnerable website.

o Web site defacement:

Changing the appearance of a website vulnerable to XSS is one of the easiest and yet most efficient ways for attackers to target companies or public organizations. Either this can take organizations to the spotlight for the improper reasons by using embarrassing pictures or hacktivism messages.

Mitigations for cross-site scripting are:

The best way to mitigate the XSS is to properly sanitize the untrusted input is taken from an input field, ids, URL parameters given by user, properly escaping the special characters in input provided and also output encoding the special characters while displaying in the HTML pages, the content security headers,httponly flag, X-XSS-Protection Response Header can also be used are the features in modern is frameworks can also be to prevent XSS

C. File inclusion:

File inclusion vulnerability will arise due to dynamic linking or execute the files or code from the web server, and a high-risk vulnerability. in file inclusion, the application will build the path based on the user-supplied input and it can be controlled by user input. by taking this as an advantage the attacker will try to give input or inject payloads that will lead to loading confidential files from the server, deface the website, remote code execution (RCE), etc. There are two types of file inclusions.

1) Local file inclusion (LFI):

Local file Inclusion is an attack targeting the input fields like id field, text boxes, text fields, URL parameters, etc that dynamically reference file and scripts from server storage and does not sanitize input fields properly, which allow an attacker to manipulate input and inject payloads with path traversal characters or to retrieve the files from the server. The local file inclusion vulnerability will further lead to directory traversal, sensitive information disclosure, and code execution or even cross-site scripting (XSS) [6].

Local file inclusion will commonly be found in PHP web applications. Based on the functionality of the application the LFI will lead to executing the input by the language parser, download the requested file or display the content of a file on the web page. The remediation of LFI is the application should accept only character and numbers for file names and should be blacklisted all the special characters, limit the access of files by the hosted application from a server through specific directories only.

2) Remote file inclusion (RFI):

In RFI malicious users will targets the web applications input fields like id parameters, text boxes, text fields, URL parameters, etc. Which dynamically reference external script that is not sanitizing the input fields properly. Using the RFI malicious users can include or load the file stored in a remote location. all kinds of web applications support the file include but most commonly it will be found in a PHP web application because, in PHP programming, we will use file includes" extensive. By using the remote file inclusion, the attacker can trick web application to load the code stored in a remote location the code will include backdoors, web shells, code execution at OS level, etc. By successfully exploiting the RFI the malicious users can gain sensitive information, take over the application or server, server hijacking, etc. The main causes of RFI are programming mistakes and misconfiguration in system settings and functionalities. The most effective way to eliminate the RFI is to completely avoid the dynamically including the files based on user inputs or keep the white list of filenames that can be included in user inputs.

D. Cross-site request forgery:

Cross-site request forgery will also be called as one-click attack or session riding, In CSRF the malicious user will try to send the forged request and makes the victim send the request that is crafted by attacker and csrf will be possible only when a user is logged in the same browser where he is submitting the request. the impact of csrf will depend on the access rights that affected user has which includes change password of victims, creating new user accounts, and can take control of application data and functionality if the user is an admin.

The CSRF will be possible in the following cases if a web application is not having any unpredictable request parameters, cookies should be used for session handling and there should be a relevant action for a request to generate the forged request, for example, password update page, new user creation pages, etc. In general, the cookies associated with the particular website will be included in the request of a browser from another domain that is stored in a browser so csrf is possible. The best mitigation for csrf is to use the random token which should be unpredictable and must be validated with every request, implementing the same-site cookie attribute to instruct the browser to include cookies to the same domain only.

E. Broken authentication

It is that allows a malicious actor to get access to the user accounts by stealing the credentials or by forging the session-related data of a website. commonly the application will implement the login functionality to give functionalities based on the user's role and will track the users based on session id. if this functionality is not implemented correctly then the malicious user will try to get access.

If the authentication mechanism has not implemented correctly then the malicious user will use the brute force technique to get credentials or if the application is using the default usernames and passwords which helps the malicious user to guess credentials very easily and he can also use Boolean based SQL injection to bypass authentication mechanisms. If the session is not managed correctly,

then the malicious user can guess the session id based on earlier sessions or session hijacking or session reusing. after successfully completing the attack by a malicious user can obtain access to user accounts which can be an admin account also.
The best methods for avoiding broken authentication-related issues is to use strong password policy, account lockout policy after invalid attempts, implementing strong captcha for avoiding brute-forcing, and implementing a strong authenticating system by hashing password rather than sending in plain text. for avoiding the session management related issues is recommend not pass the session value in URLs and should be unpredictable, the values of the cookies should properly validate with every request, session should be terminated after user logged out and new has to be generated for every new login and user session should be terminated after an inactive time.

V. CONCLUSION

This paper is disused about vulnerability assessment and pen testing, and vulnerabilities like SQL injection, file inclusion, cross-site scripting, etc. In a web application auditing both the manual pen testing and tools will also be used because by using tools, there will be so many false positives, will there and the vulnerabilities can be missed so manual intervention is also important and we conclude that the VAPT is the important way for identifying and eradication the security vulnerabilities in web application. In the future, we will be looking more closely for various issues related to the VAPT process such as identifying factors that impact adopting of VAPT process in business and problems in implementing security vulnerability remediation in a web application and continuing our research and study in this field.

ACKNOWLEDGMENT

I wish to express my sincere gratitude to the administration and Security Audit Team of Andhra Pradesh Technology Service Ltd, Department of Information Technology Govt. of Andhra Pradesh, KL University for providing an excellent environment and resource.

REFERENCES

[1] Ayeni, Bakare K., et al. “Detecting Cross-Site Scripting in Web Applications Using Fuzzy Inference System.” Journal of Computer Networks and Communications, fol. 2018, Jan. 2018, pp. 1–10., doi:10.1155/2018/8159548.

[2] "Cross-Site Scripting (XSS)." OWASP,. org/www-community/attacks/xss/.

[3] Glossary - Web Application Security Consortium, www. web app sec. org/projects/glossary/.

[4] “Injection Flaws.” OWASP, owasp. org/www-community/Injection_Flaws.

[5] Kaur, Parminder, and Navdeep Kaur. “Input Validation Vulnerabilities in Web Applications.” Journal of Software Engineering, vol. 8, no. 3, Jan. 2014, pp. 116–126., doi:10.3923/jse.2014.116.126.

[6] Muscat, Ian. “What Is Local File Inclusion (LFI)?” Acunetix, 11 Mar. 2019, www. acunetix.com/blog/articles/local-file-inclusion-lfi/.

[7] “SQL Injection.” Wikipedia, Wikimedia Foundation, 14 Jan. 2020, en. Wikipedia. org/wiki/SQL_injection.

[8] “Types of XSS.” OWASP, OWASP. org/www-community/Types_of_Cross-Site_Scripting.

[9] “Types of XSS (Cross-Site Scripting).” Acunetix, www. acunetix.com/website security/xss/.

[10] “Web Application Vulnerabilities: Statistics for 2019.” Web Application Vulnerabilities: Statistics for 2019, Positive Technologies, 13 Sept. 2019, www. ptsecurity.com/www-en/analytics/web-application-vulnerabilities-statistics-2019/.

[11] “What Is SQL Injection (SQLi) and How to Prevent It.” Acunetix, www. acunetix.com/websitesecurity/sql-injection/.

[12] “What Is SQL Injection: SQLI Attack Example & Prevention Methods: Imperva.” Learning Center, Imperva, www. imperva.com/learn/application-security/sql-injection-sqli/.

[13] “Multi-Factor Authentication.” *Wikipedia*, Wikimedia Foundation, 16 Feb. 2020, en.wikipedia.org/wiki/Multi-factor_authentication.

[14] Shar, Lwin Khin, and Hee Beng Kuan Tan. “Mining Input Sanitization Patterns for Predicting SQL Injection and Cross-Site Scripting Vulnerabilities.” *2012 34th International Conference on Software Engineering (ICSE)*, 2012, doi:10.1109/icse.2012.6227096.

[15] Blog, diadem. “Top 10 Website Vulnerabilities.”, https://diadem.in/blog/top-10-website-vulnerabilities/.

[16] “What Is DOM-Based XSS (Cross-Site Scripting)? Tutorial & Examples.” *What Is DOM-Based XSS (Cross-Site Scripting)? Tutorial & Examples*, portswigger.net/web-security/cross-site-scripting/dom-based.

paper1

ORIGINALITY REPORT

8%	7%	5%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ptsecurity.com Internet Source	2%
2	www.acunetix.com Internet Source	1%
3	scialert.net Internet Source	1%
4	www.listeningonlineingles.com Internet Source	1%
5	Rafique, Sajjad, Mamoona Humayun, Bushra Hamid, Ansar Abbas, Muhammad Akhtar, and Kamil Iqbal. "Web application security vulnerabilities detection approaches: A systematic mapping study", 2015 IEEE/ACIS 16th International Conference on Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD), 2015. Publication	1%
6	www.imperva.com Internet Source	1%

7	blog.morizyun.com Internet Source	<1%
8	mateuszroth.pl Internet Source	<1%
9	Mohamed Al-Ibrahim, Yousef Shams. "The Reality of Applying Security in Web Applications in Academia", International Journal of Advanced Computer Science and Applications, 2014 Publication	<1%
10	Sagar Ajay Rahalkar. "Chapter 10 Web Application Hacking", Springer Science and Business Media LLC, 2016 Publication	<1%
11	www.professionalqa.com Internet Source	<1%
12	www.cenzic.com Internet Source	<1%
13	Senthamil Preethi K, Murugan A. "Analysis of Vulnerability Detection Tool for Web Services", International Journal of Engineering & Technology, 2018 Publication	<1%
14	www.linuxinsider.com Internet Source	<1%
	Shikha Jain, Rahul Johari, Arvinder Kaur.	

15	"PJCT: Penetration testing based JAVA code testing tool", International Conference on Computing, Communication & Automation, 2015 Publication	<1%
16	readwrite.com Internet Source	<1%
17	Nitin Nagar, Ugrasen Suman. "chapter 11 Prevention, Detection, and Recovery of CSRF Attack in Online Banking System", IGI Global, 2017 Publication	<1%
18	www.cari-loker.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		