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 is exceedingly dependent on the Internet and Web applications are using very extensively by every organization and human for information sharing, business purposes like online sales, money transfer etc and services Exchange. Nowadays, web security 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 effected, then daily business and reputation will be loosed. As almost all organizations have using the web application service to share or store sensitive information about their clients. So Web applications are inclined to security attacks and new security vulnerabilities have grown in the last two decades in a web application. So web applications become a well known and important target for security attacks by attackers. So it is very vital to secure a web application from attacks. Major security issues in the web application will occur due to improper input validation at the client side which will take advantage of an attacker to inject payloads. This paper reviews the vulnerability assessment and pretesting steps and types, an area of web application security vulnerabilities like SQL injection, Cross-site scripting, file inclusion, and broken authentication

Index Terms- Cross-site scripting (XSS), 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 most important source for delivering information and service to the world. Web application advancements give a promising system of coordinating numerous useful segments over the web and therefore 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, perform money related exchanges, and have fun and communicate and to socialize themselves

Web application grew tremendously in the last few decades and it has brought great benefits to the people, however, these benefits are associated with some challenges like security which will be very important. Security in a web application refers to the threat which occurs due to flaws in software design, coding, testing, and implementation. Web application services are more prone to cyber-attacks due to their public access. And web applications are increasingly used to deliver security-critical services so they become a valuable target for security attacks. Most web applications will use a database that helps to manage data easily and this data will also contain sensitive information, if the web application is hacked, a large amount of information 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 on the server-side, web browser and client-side technologies. For developers with insufficient security vulnerabilities, knowledge or awareness results in a high rate of web applications sent on the Web are uncovered to security vulnerabilities. According to a report by the Internet Application Security Consortium, around 49% of the internet applications being looked into contain vulnerabilities of a tall hazard level and more than 13% of the websites can be compromised naturally. 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 requirement, design, coding or operation that could accidentally occur or intentionally violated and result in a security failure. In the last few years, the number of reported web application security vulnerabilities has increased. Some commonly found web application vulnerabilities include SQL injection, cross-site scripting, command-line injection, cross-site request forgery, and malicious file upload and execution.

this paper is presented as follows.

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] A web application consists of a series of scripts, style sheets, HTML pages, images, etc. that reside on a web server and communicate with dynamic content repositories or other sources. Using Internet infrastructure, web applications enable service providers and consumers to exchange and manipulate information. For a good

introduction to the web application from the penetration tester's perspective, the information can be gathered from [3]. Web application development technologies include PHP, Active Server Pages (ASP), Perl, Common Gateway Interface (CGI), Java Server Pages (JSP), JavaScript, VBScript, HyperText Markup Language (HTML), and with large categories of web application technologies include different communication protocols, formats, server-side and client-side scripting languages, browser plug-ins, and web server API.
A web application has a distributed n-tiered architecture. Typically, there is a client (web browser), a web server, an application server (or several application servers), and a back-end (database). 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 client and web server for additional security.

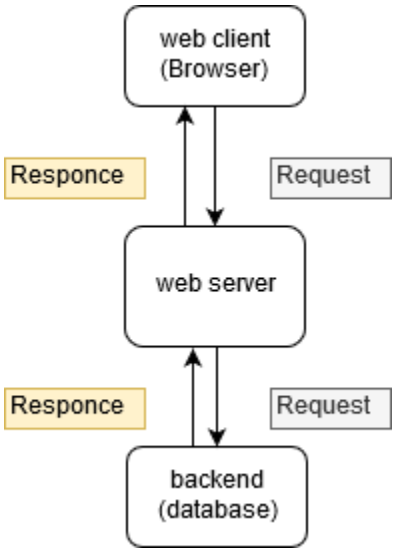


Figure 1. Web Application Environment.

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 for the organization, as well as all legal issues. In this process, organizational assets will have to categorize based on assets in scope (assets on which penetration testing has to perform) and assets out of scope (not included in penetration testing) based on gathered information on all organizational assets

2) Reconnaissance:

Reconnaissance or Open Source Intelligence gathering is one of the important steps in penetration testing. A pen tester aims to gather as much as information and the potential targets for exploit required about client organization. Depending on the type of penetration test chosen, the penetration 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 address, 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 pen testers 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 somewhat in the control of organization and 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 completion of this phase, the information gained is 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 into your 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 gain access to system and sometimes the system can have the secure network which contains 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 gain 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 ensure that all installed backdoors or rootkits should be removed, and it should ensure 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 available in the system and their Business impact on the system. In the technical team, they will have information like different vulnerabilities exist in the system with its remediation.

The pen test report will begin with an executive summary outlining organizational business-related penetration test plan, 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 determine what to correct and what problems pose an appropriate level of risk. The remaining part is a technical detail, which will be descriptive, specific and generic or ambiguous statements will help the technical team to resolve security issues.

7) Resolution & Re-Testing:

In this phase, the technical team tries to resolve the issues and technical team will get some assistant from team to solve the issues and Once vulnerabilities have been remediated, the client has to retest their systems to ensure that fixes were successful and has to test whether new vulnerability was created as a result of remediation or not. And the pen test 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 prior knowledge of evaluating infrastructure so pen tester will act as a real-world hacker in this case. 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 currently available resources in the systems. A black-box penetration tester should know both automated and manual penetration testing methodologies. Generally, this is the best approach because it helps pen testing team to think out of the box and carry out testing in all levels according to their realistic knowledge and expertise and they will also use all the techniques and methodologies available for them by simulating the level of persistence, knowledge, and expertise that a real-world hacker can perform.

2) White box pen testing:

In white-box, pen-testing Pen testers will have full knowledge of the infrastructure and its internal design and working, it is also known as clear box testing. The penetration tester seeks to obtain as much input as they can to gain 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 the vast amount of data and extract data to identify possible vulnerability points, making it the most time-consuming method of penetration testing. 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 will have full access to the system for performing audits on the high-risk area.

3) Grey box pen testing:

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

---

IV. WEB APPLICATION VULNERABILITIES

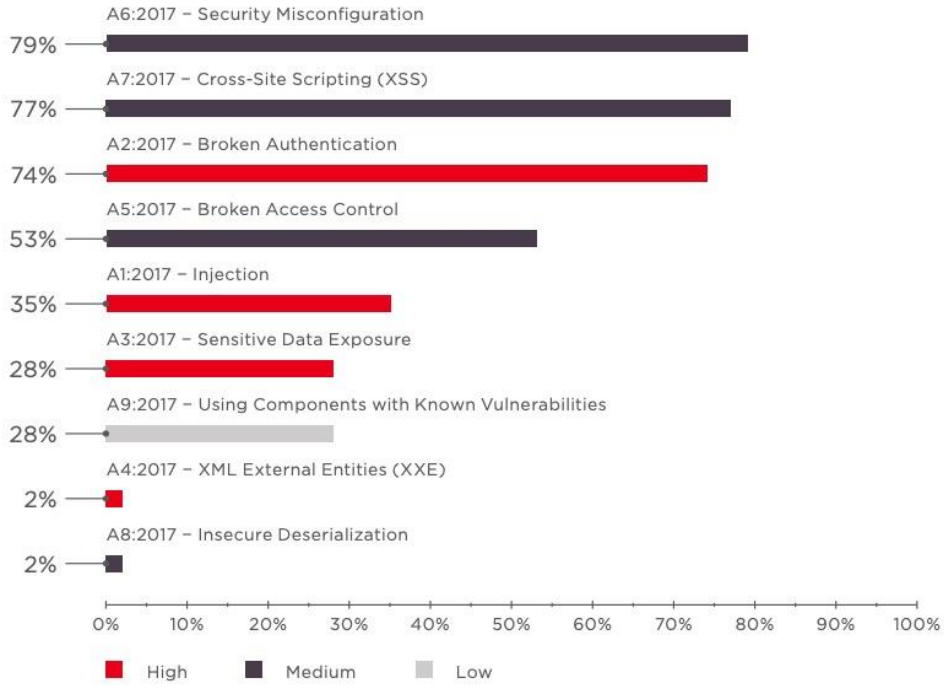
Vulnerability is a weakness in the application which can be a design flaw or implementation bug that allows an attacker to cause harm to stakeholders of an application. Formally, vulnerability is defined as “The existence of a weakness, design, or implementation error that can lead to an unexpected, 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 padigarimage, configuration mismanagement, the complexity of software, accepting unsanitized input from the user, weak password management and features misuse. The impact of vulnerabilities depends on the impact of the vulnerability for example if an attacker obtains the confidential details of a user, he can misuse this information (like account number, account balance, loan amount, etc.) and can also alter the data to cause harm to the concerned user then the impact will be very high. Vulnerability assessment for web applications is done periodically for identifying, classifying, remediating and mitigating vulnerabilities. the testing guidelines for web applications will be given by the organization like OWASP, OSSTMM, ISSAF, Microsoft, etc and will classify the vulnerability based on their risk rating, exploitability, detectability, prevalence, and impact. The OWASP standard is the commonly used standard classification of a web application vulnerability based on OWASP standards is given in below table and OWASP will also release the testing guidelines with a checklist

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

Table1. OWASP (2017) Top Ten 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

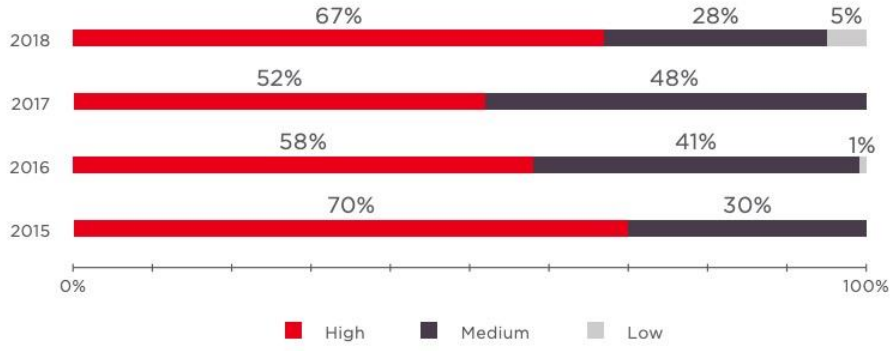


Figure 2. Percentage of Web Application Affected yearly

The different high-level vulnerabilities are disused below:

A. SQL Injection:

The attacker poisons dynamic SQL statements in the SQL Injection Attack to comment on some components of the declaration or to add a condition that will always be valid. The attacker uses the design faults to exploit SQL statements by implementing malicious SQL code is poorly designed web applications and SQL Injection is also one of a type of injection attack [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 tries to control a database server to retrieve data from a database and also be used to bypass authentication.

SQL injection vulnerability can found in web applications that are using the SQL databases like MySQL, Oracle, SQL Server, or others. SQLI is a common attack vector by this means the attacker can pass the malicious payloads to SQL parsers to manipulate or access confidential, sensitive information like customer information, personal data etc that are stored in a database. SQL Injection attack is one of the most common and dangerous Vulnerability.

An attacker must first discover vulnerable user inputs fields and parameters in the web application to perform an SQL Injection attack and such user input fields will be used to pass SQL Injection 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 is the main component of the attack. After the attacker modifies the request parameters with malicious payloads than database executes malicious SQL commands and gives an output with database content relevant to the payload. The malicious queries can be inserted by the attacker via a web form or by attaching them directly to the end of the URL or HTTP headers.

SQL is a query language for managing data stored in relational databases. And it can be used to access, edit, and delete data. Many websites and web applications manage all the data in SQL databases. You can also use SQL commands to execute an operating system 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. These credentials can then be used for impersonated the other users. The impersonated user can be an administrator with all the privileges of the database as well.
- SQL allows you to select and display information in the database. An SQL Injection vulnerability could give the attacker full access to all information on a database server.
- SQL also allows you to change information and add new information to a database. For instance, an attacker could 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 the database is restored. 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 such a case, an attacker might use an SQL Injection to attack the internal network.

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 attacker executes activities that cause error messages to be generated by the database. The attacker may use the data supplied by the error messages to collect information about the database structure.
- **Union Based SQL Injection:** This method uses the UNION SQL operator to fuse various select statements generated by the database to obtain a single HTTP response. This result may include information that the attacker can leverage.

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]. Blind SQL injections depend on the

server's response and behavior patterns, to perform these types of attacks typically consume 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 weather condition is true or false. Based on the result, the HTTP response data will change or remain unchanged. by using this change in response based on the condition the attacker can retrieve the database content
- **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 time delay in the response, the attacker can guess some kind of 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 too slow or unstable to perform such activities. in this method, the attacker will relay 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 to end-user on the web application database server.

The best Preventive measures for SQL Injection Attacks are:

1. Instead of placing user-supplied 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 to prevent SQL injection.
2. Sanitize user-provided inputs to properly escape the special characters and should verify that the input data is in a pre-defined format or not.
3. Don't leave sensitive data in plaintext while storing in the database instead of Encrypt or hash the confidential data which will provide 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. Displaying the common database error messages directly to end-user should be avoided instead the custom error messages should be displayed for end-user
6. By using the Web Application Firewall (WAF) 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 the attacker tries to execute the malicious scripts in the victim's browser by injecting malicious payload in the legitimate web application [4]. Every time the users accessing the web pages that are injected with the malicious script then the real attack will happen.

The web application becomes a means for delivering the 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 that are vulnerable to cross-site scripting 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 in 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

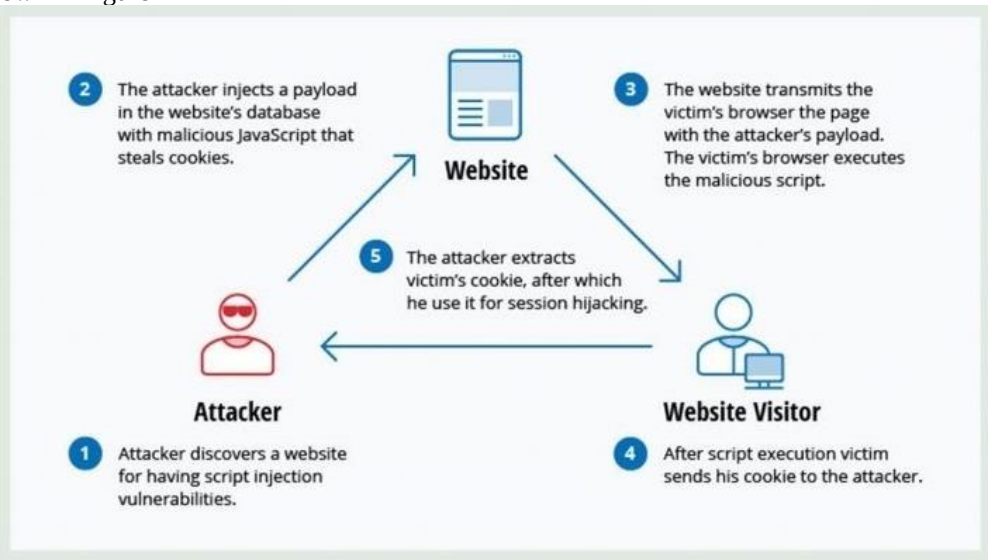


Figure 1. Demonstration of an XSS attack.

By using the Cross-site Scripting 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 website or website with malicious contents Vulnerability is regarded to have less impact than SQL injection vulnerability. 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 user's OS and files.

But if JavaScript is used as part of malicious content, it can still be dangerous as Malicious JavaScript can also access to all objects that remaining web pages can access. Which includes access to the User cookies often used to store data related to the session? If an intruder succeeds in obtaining the session cookie of a user, the attacker can get access the user account and perform 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 which can also use HTML5 APIs in modern browsers. For instance, gaining access to specific files from the user's file system to the geolocation, webcam, and microphone. Most of these APIs require granting permission from the user to execute in the browser, but the attacker may use social engineering to address that restriction.

1) Reflected Cross-Site Scripting:

In Reflected cross-site scripting, the attacker will inject the malicious script in the request and pass that request to the server then the server will process that request and sent back the response with injected payload in an unsafe way which will lead to the execution of script code in the browser. Reflected XSS is also known as non-persistent or TYPE II XSS [8]. In reflected XSS the malicious script will be executed on the same place or page where it is injected so in generally the 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:

In stored XSS 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 XSS or type I XSS [8]. In stored XSS all the users whoever visits the infected page will be effected sometimes admin can also be infected. the payload injected on infected pages will be executed continuously until it has been deleted from its source.

3) DOM Based Cross-Site Scripting:

DOM-based XSS is an advanced XSS attack. It is possible if the web application's client-side scripts write data provided by the user to the Document Object Model (DOM). The data is subsequently read from the DOM by the web application and outputted to the browser. 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].

Dom based XSS 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 most popular sources and sinks are given in the image below.

table

A DOM-based XSS attack is a client-side attack because the malicious payload injected will be injected in client-side DOM and shows response immediately so payload will not be sent to the server. which makes DOM-based XSS more difficult to detect by the Web Application Firewalls (WAFs) or at the server-side.

The Impact of Cross-Site Scripting (XSS) is

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

The impact will be critical on the web application that containing sensitive data, such as banking transactions, emails and health records, the impact will typically be significant and based on rights the compromised has for example If the compromised user has admin privileges then an attacker can take full control over application and can get any kind of data

- Session Hijacking:

The most popular XSS attack vectors are stealing the victim's session cookies to hijack the victim's accounts. This enables attackers 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 customer credentials, instead of obtaining their cookies cloning the login page of the web application and then using XSS attacker steal credentials from the user. This situation is even more useful from an attacker's view, as they 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 (e.g. private identifiable data or cardholder data) or to conduct unauthorized activities.

  - **Key logger:** 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 input forms that can 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.

- Web site defacement:

Changing the visual 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 cross-site scripting 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 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 js frameworks can also be to prevent XSS

C. File inclusion:

File inclusion vulnerability is a web application vulnerability that will arise due to dynamic linking or execute the files or code from the web server and it is a high-risk vulnerability. On the file inclusion vulnerability, 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 gives input or inject payloads that will lead to remote code execution, loading confidential files from the server, deface the website, etc. There are two types of file inclusion vulnerability.

1) Local file inclusion (LFI):

Local file Inclusion is an attack targeting in a web application that exist in the input fields (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 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 arise in PHP web applications but can in all kinds of web applications. Based on the functionality of the application the LFI will lead to executing the input (file or command) 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 web 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 web application from specific directories only.

2) Remote file inclusion (RFI):

Remote file inclusion is an attack targeting in the web application that exists in the input fields (id field, text boxes, text fields, URL parameters, etc) which dynamically reference external script and does not sanitize input fields properly. Using the remote file inclusion the attacker can include or load the file stored in a remote location. Almost all types 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 malicious code stored in a remote location the malicious code will include backdoors, web shells, code execution at OS level, etc. By successfully exploiting the RFI the attacker can gain sensitive information from the server, take over the application or server, server hijacking, etc. The main causes for remote file inclusion are programming mistakes, misconfiguration of the respective programming language functionality. The best way to eliminate the RFI is to completely avoid the dynamically including the files based on user inputs or maintain the white list of filenames that can be included in user inputs.

D. Cross-site request forgery:

Cross-site request forgery will also be as csrf, **one-click attack** or **session riding**, here the attacker will try to send the forged request and makes the victim submit the request on behalf of him the csrf will be possible if the user is logged in the same browser where he is submitting the request. the impact of csrf will depend on the access rights that victim has which includes change password of victims, creating new user accounts, the attacker can also take full control of application data and functionality if the victim is a privileged user

The cross-site request forgery will be possible in the following cases if the application is using cookie-based session handling, unpredictable request parameters should not be generated 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 related to the particular website will be included in the request of browser from other domain if a user is authenticated because the browser cannot distinguish between the forged and legitimate request so csrf is possible and it is a serious vulnerability even though it is not included in OWASP top 10. The best mitigation for csrf is to implement the csrf token which has to be validated in every request and that token should be unpredictable, random and also implementing the same site cookie.

E. Broken authentication

Broken authentication and improper session management is a web application and stands in the second position of OWASP top 10 vulnerability that allows an attacker to gain unauthorized access to the website by stealing the user credentials or by forging the session data. commonly the application will implement the login functionality to provide functionalities based on the user's role and will track the users based on session id. if this functionality is not implemented correctly then an attacker will try to gain unauthorized access.

If the authentication mechanism is not implemented, then the attacker can brute force with a list of usernames and passwords and can get the credentials or if the application is using the default usernames and password, the attacker can easily guess, the attacker can also bypass using Boolean based using injection. If the session is not managed correctly, then the attacker can guess the session id or session

hijack or session reuse. By successfully attack using broken authentication and session management the attacker can gain access to victim user accounts sometimes attackers can also get access to the admin account.
The best methods for avoiding broken authentication-related issues is to implement 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 if possible implementing the two-factor authentication[13]. for avoiding the session management related issues is to not expose the session id in URLs, the session-id generating should be unpredictable, properly validating the sessions id with every request, a session should be terminated after user logged out, the new session should be generated with every new login and user session should be terminated after an inactive time (maximum 5 to 10 minutes).

V. CONCLUSION

This paper is disused about the 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 very important and we conclude that the VAPT is the important way for identifying and eradication the security-related vulnerabilities. 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.