A Mitigation Plan to Encounter Discovered Vulnerabilities in Educational Web Application

[1]Shabana, [2]Dr.Umesh Kumar Singh, [3]Abhishek Raghuvanshi, [4]Vandana Rathore

[1]Institute of Computer Sciences, Vikram University, Ujjain, India
[2]Director, Institute of Computer Sciences, Vikram University, Ujjain, India
[3]Department of Computer Science & Engineering, Mahakal Institute of Technology, Ujjain, India
[4]Government College, Jhiran, India

Abstract: University systems may become exposed to direct attacks on networking hardware, advanced persistent threats, and cyberattacks where hostile attackers use network connections to gain access to restricted resources. One of these tasks examines the susceptibility of a university network to network-based attacks. Network and system administrators are responsible for keeping networks and systems safe from threats coming from both inside and outside the company. There is always a chance that someone from outside the academic community may hack into a university's network infrastructure and take advantage of network weaknesses. Network and system administrators, however, frequently have trouble identifying security holes, solutions, or security patches that can be applied to alleviate the vulnerabilities. An essential component of security assessment is determining the level of compromise that vital servers in a network can withstand. Two host networks were screened using Acunetix, Zaproxy, and OpenVAS scanners. These host networks belongs to educational institutions of North India. This scanning helped to discover vulnerabilities present in the network with their severity level. This manuscript presents a mitigation plan for cross site scripting and SQL injection vulnerabilities in educational institutions network.

Keywords: Vulnerability, Attack, Threats, Cross Site Scripting, SQL Injection, Mitigation Plan,

1. Introduction

Direct attacks on networking devices, advanced persistent threats, and cyber-attacks during which malicious attackers obtain access to restricted resources via network connections may end up making university systems vulnerable. One such task focuses on evaluating a university network's vulnerability to network-based attacks. Malicious outsiders on the World Wide Web and malicious actors directly connected to private systems are examples of these. Both types of attackers can exploit flaws in network systems and infrastructure such as servers (e.g., data centre, web applications, mail servers, and so on), routers, access points, and firewalls. Protection against network-based attacks is difficult since compromising one network provides a surface for attackers to launch additional attacks [1].

Figure 1: Relationship between Vulnerability, threat and risk

It is the role of network and system administrators to maintain both systems and networks protected from both threats from within and outside the organization. There is always a probability of someone outside the university world breaking into a university's network infrastructure and exploiting network vulnerabilities. Internal to the organization, an institution has a broad segment of users, like students, university employees, administrators, contract employees, visitors, or guests, who, despite having authorized access to certain system resources or performing certain duties throughout the systems, can induce all sorts of issues for the system administration [2].

In practice, however, network and system administrators commonly struggle to spot security flaws, fixes, or security patches that can be used to mitigate the vulnerabilities. Determining the degree of compromise that critical servers in a network can tolerate is an important part of security assessment. Moreover, this is a difficult task. The answer is determined by the network topology, the network's security policy, the layout or guideline sets used in equipment such as firewalls, routers, and switches, as well as vulnerabilities in protocols and systems [3] .

This type of analysis is generally carried out using penetration testing, which includes actively investigating a network and running tests exploits that compromise systems. Penetration testing can be highly efficient in detecting network vulnerabilities, but it is labour intensive, time taking, and can interrupt system operations. This paper describes a method for evaluating the overall security of a campus network using penetration testing that interacts with network operations minimally and requires no additional network traffic beyond that required for normal host security testing [4].

This manuscript presents a mitigation plan for cross site scripting and SQL injection vulnerabilities in educational institutions network.

2. Web Security

Security It is a degree of resistance to, or protection from any variety of harm. It means safety or measures taken to be safe or protected. It should be of person, social groups, objects, any educational institutions, ecosystems or the other entity susceptible to unexpected change. In terms of Information Technology it is protection of digital data and IT things against external, internal, accidental and hostile threats. This defense includes detection, prevention and response to threats through the employment of security policies, software tools and IT services [5].

Nowadays almost bussiness, corporate totally depend upon the web appliances to communicate between consumer and sell product. Different task of web site can be handled by the technologies. Websites are used to provide different types of services also they are directly connected with databases so at that time valuable asset targeted by the attacker. Which directly affects on the bussiness strategy like loss, crash the system. Finding software related bugs within web appliances are happens to the system with bad things. Normally, many web appliances have been developed in the mind only for competative or functionality basis. With the help of such an application gives better future [6] [7].

2.1 Types of Attacks

- Passive Attack: In this attack attacker observe the contents of message and also observe pattern of message. It doesn‟t harm the system. Passive attacks never change the resources of system. But in this attack victim does not get informed about the attack. It is danger for confidentiality.
- Active Attack: In this attack attacker change system resources and also effects their operations. It involves modifications of messages, repudiation, replay attack and denial of service attack.
- Distributed Attack: It is also called as cyber attack in which network resources are unavailable to the intended users by temporarily.
- Phishing Attack: This type of attack is often used to steal sensitive data such as user‟s names their password, card information such as credit card details.
- Spyware Attack: It is malicious software which is installed on device without end user‟s knowledge. It steals sensitive data also internet usage data. It specially designed to obtain sensitive data and also damage the system without any knowledge.
- Insider Attack: Insider attack is an accidental attack. This type of attack is launched by internal users who are authorized of the system. This attack mainly target within organisation.
- Wireless Attack: this type of attack means malicious action against any wireless technology, network. Denial of service, penetration and sabotage are the example of this attack.
- Email Attack: This attack includes phishing, identity theft, virus and spam.
- Password Attack: In this type of attack, attacker trying to guess password locally or remotely. It is the process of recovering passwords from data this process is known as cryptanalysis.
- Buffer Attack: when hackers intentionally hold extra data with specific instruction for actions is called as Buffer –overflow attack.
- Exploit Attack: When any attack takes some benefits of vulnerabilities in the application or network or hardware this type of attack is called exploit attack.
- Spoof Attack: In this attack a person trying to gain access in the system by illegitimate.

3. Discovered Vulnerabilities

A vulnerability assessment strategy was applied to the hosts detected throughout the data collection phase, which facilitated the scanning procedure. Even while manual methods still have their place alongside automatic scanners, the former require much longer perfecting the scan and pinpointing security holes. However, in order to fully comprehend the vulnerabilities that might have compromised the system or network, it is recommended to employ both human and automated scanning approaches. Assume that the networks or systems to be tested have a large network with hundreds of systems. A manual approach would be ineffective and inefficient. Automated scanners were utilized instead of conventional scanners during this phase because manual techniques take longer to perfect the scan [8].

Acunetix [9], Zaproxy [10] and OpenVAS [11] were chosen as algorithmic scanning and security vulnerabilities scanners. The above scanners were deployed to determine what operating systems and services were running on the target hosts, as well as which hosts and utilities were vulnerable

In experimental work, two host networks were screened using Acunetix, Zaproxy, and OpenVAS scanners. These host networks belongs to educational institutions of North India. For security point of view, names of all hosts are not disclosed here. Discovered vulnerabilities with their severity level are shown in table 1.

Table 1: Vulnerabilities discovered in Host XYZ-Educational Web Application using Acunetix Scanner

Vulnerability Discovered	Severity Level
Blind SQL Injection	High
Cross site scripting	High
Development configuration file	Medium
Error message on page	Medium
HTML form without CSRF protection	Medium
Slow HTTP Denial of Service Attack	Medium
Source code disclosure	Medium
Clickjacking: X-Frame-Options header missing	Low

Documentation file	Low
Login page password-guessing attack	Low
Possible sensitive directories	Low
Session token in URL	Low

4. Confirmed Vulnerabilities

Penetration testing of educational institute's network using SQL map exploitation tool has confirmed existence of SQL injection, cross site scripting, Slow HTTP Denial of Service Attack, Documentation file and HTML form without CSRF protection. It is shown in figure below in figure 2.

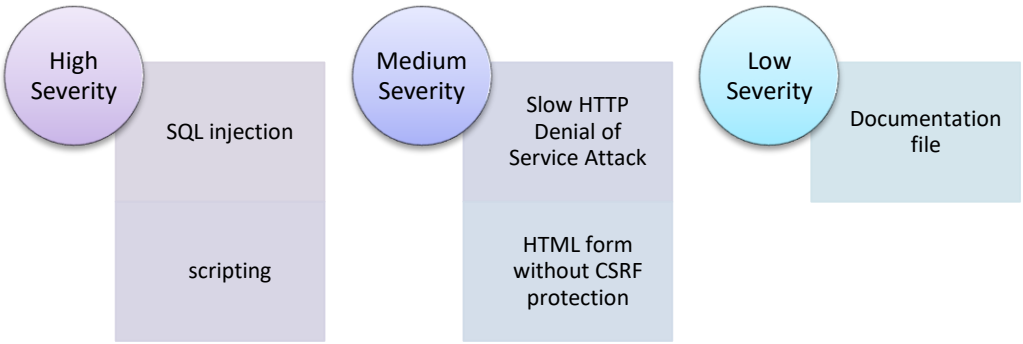

Figure 2: Vulnerabilities Confirmed by SQL map in Host XYZ-Educational Web Application

5. Mitigation Strategies against Cross Site Scripting and SQL attacks

This section presents mitigation strategies for Cross Site Scripting and SQL attacks.

5.1 Cross Site Scripting Mitigation Strategies

Because cross-site scripting [12] is a form of exploitation rather than just a vulnerability that may be exploited, phishing attacks, which are false attempts to obtain information, are very effective. During an attack of this kind, information is often gathered through the use of a hyperlink that contains malicious content. The web application, once it has received the data, will then construct an output page for the user, which will contain the malicious data that was given to it. If information that is stored in user cookies is revealed through an attack known as cross-site scripting, then there is a potential for a serious breach of security and confidentiality. An adversary now has the capability of stealing sensitive session data such as user IDs, passwords, and credit card numbers. By changing the user's settings in this manner, attackers have the ability to take control of the user's account, steal cookies, or instal harmful advertisements. When other vulnerabilities are combined with cross-site scripting, it is possible for arbitrary code to be executed on the machine of the victim. A cross-site scripting attack can be triggered in two different ways: by malicious pages or by parameter values. Cross-site scripting can be identified with a high degree of accuracy if the source of the attack, when it occurred, and the different kinds of signatures used are known. Cross-site scripting is a type of attack in which the attacker gets around security measures by exploiting a vulnerability in the scripting language that enables remote code execution. This type of attack is also known as XSS. A significant number of intrusion protection systems are incapable of detecting even the most fundamental types of assault, such as cross-site scripting. Many people who work in the security industry consider it to be only a moderate risk.

In order to execute a cross-site scripting attack, the server must display the user's input as HTML. Methods to prevent cross-site scripting are addressed below:

- Input Validation: One frequent countermeasure is to verify the legitimacy of the input data before using it.
- HTTP Cookies: The possibility of a cross-site scripting attack is mitigated by including a cookie in the HTTP response header that is delivered back to the client. These characteristics indicate that a cookie cannot be accessed by script.
- HTML Encoding: All users get HTML non-alphanumeric characters through encoding (HTML quotation), which prevents them from being read as HTML.
- Intrusion detection system monitoring Intrusion detection Technique is a type of monitoring or managing the security aspects of the system. An Intrusion Detection System (IDS) collects and examines the traffic entering and leaving a Web application. Identified security breaches are either Intrusions of Misuse type or Anomaly type. Intrusions are in the form of entry of malicious script into the database of the Web System by the attackers who are referred as masqueraders in trying to tamper the sensitive data.
- URL analysis- The URL request Ui and the Request status Si are the inputs and the results, respectively, of this method. It starts by accepting the ith user's URL and parsing it to see whether it matches the expected session ID. The anticipated session ID is then used to conduct a verification check. If the request is approved, it is sent to the server and the user is credited with making the request. Otherwise, the URL request is deemed invalid, and the request status is recorded as "Rejected.".
- Hash verification- The browser extension must be installed in the web browser of each user of the web application. It validates the apps' signatures before launching them. Web content verification is described using pseudo code. The verification certificate for the web server contains the site public key..

5.2 SQL Injection Attack Mitigation Strategies

In order for SQL injection [13] to work, software flaws must exist. If a hacker knows a SQL server is running a susceptible database application, they may exploit the server's flaws and inject malicious code to bypass the login process and obtain access to the database without the owner's knowledge. Botnets, or private networks of infected computers, sometimes referred to as a "Zombie army," are often responsible for launching such attacks. If the attacker is successful, he or she will be able to take control of the

web server, execute instructions on the system, access sensitive information, change data in databases, and more. SQL injection attacks are launched through botnets consisting of thousands of infected computers. Millions of URLs across a wide variety of websites have been compromised due to SQL injection by botnets.

SQL injection attack countermeasures techniques are as follows:

- Firewall must be used: Using a web application firewall, either one built into the system or one that is installed on a separate server, may assist to prevent harmful material from being.
- Dynamic SQL should be avoided: Take use of prepared statements, stored procedures, and parameterized queries.
- Decrease the size of attack surface: Don't leave any hacking openings by disabling unnecessary database features.
- Regularly Update Applications: Applications and databases should have security updates and fixes implemented as quickly as feasible if they have any known vulnerabilities.
- Validate Data: Validate and sanitise every user-submitted data on the assumption that it is harmful.
- Use Appropriate Privileges to users: Instead of using an account with full administrative rights, use a limited one. It reduces vulnerabilities and might restrict the damage that could be done by a hacker.
- Implement dotDefender Firewall: To prevent SQL injection attacks, this technology may be included into a website's application.

6. Conclusion

Administrators of networks and systems are accountable for ensuring the integrity of these resources in the face of dangers that can originate from both within and outside of the organisation. There is always the possibility that someone who is not part of the academic community will attempt to hack into the network infrastructure of a university and take advantage of any holes in the network. However, network and system administrators frequently have problems locating security holes, remedies, or security patches that may be implemented to alleviate the vulnerabilities. This is because of the complexity of modern computer systems. Determining the extent of the amount of penetration that crucial servers in a network are able to sustain is an important part of any security evaluation. Scanners from Acunetix, Zaproxy, and OpenVAS were utilised in order to examine the two host networks. The educational institutions located in Northern India own and operate these host networks. This scanning assisted in locating vulnerabilities that were present in the network along with the severity degree of each vulnerability. A mitigation plan for cross-site scripting and SQL injection vulnerabilities in educational institutions' networks is presented in this publication. This will help network administrators to encounter security attacks.

References

1. U. K. Singh, and C. Joshi, “Network Security Risk Level Estimation Tool for Information Security Measure”, IEEE PIICON 2016, 7th Power India International Conference, IEEE Computer Society, Bikaner Rajasthan, India, November 25-27, 2016.
2. Prabhakar S. Network Security in Digitalization: Attacks and Defence. Int. J. Res. Comput. Appl. Robot. 2017;5:46–52.
3. Conti M., Dragoni N., Lesyk V. A survey of man in the middle attacks. IEEE Commun. Surv. Tutor. 2016;18:2027–2051. doi: 10.1109/COMST.2016.2548426.
4. U. K. Singh, and C. Joshi, “Quantifying Security Risk by Critical Network Vulnerabilities Assessment”, International Journal of Computer Application (IJCA 0975 – 8887), Volume 156(13), December 2016, ISBN 973-93-80883-35-9, pp 26-33, impact factor: 0.782.
5. Z. Li, et Al., “VulPecker: an automated vulnerability detection system based on code similarity analysis”, ACM, Proc. of the 32 Annual Conference on Computer Security Applications, pp. 201213, 2016.
6. S. O. Uwagbole, W. J. Buchanan, & L. Fan, “Applied machine learning predictive analytics to SQL injection attack detection and prevention”, IEEE, Symposium on Integrated Network and Service Management (IM), 2017 IFIP/IEEE, pp. 1087-1090, 2017.
7. Z. Guojun, et. Al., “Design and application of intelligent dynamic crawler for web data mining” IEEE, In Automation (YAC), 2017 32nd Youth Academic Annual Conference of Chinese Association, pp. 1098-1105, 2017.
8. I. Medeiros, N. Neves, & M. Correia, “Detecting and removing web application vulnerabilities with static analysis and data mining”, IEEE, IEEE Transactions on Reliability, Vol 65, Issue 1, pp. 54-69, 2016.
9. https://www.openvas.org/
10. https://www.acunetix.com/vulnerability-scanner/
11. https://www.zaproxy.org/
12. M. Singh, P. Singh and P. Kumar, "An Analytical Study on Cross-Site Scripting," 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, 2020, pp. 1-6, doi: 10.1109/ICCSEA49143.2020.9132894.
13. Z. C. S. S. Hlaing and M. Khaing, "A Detection and Prevention Technique on SQL Injection Attacks," 2020 IEEE Conference on Computer Applications(ICCA), Yangon, Myanmar, 2020, pp. 1-6, doi: 10.1109/ICCA49400.2020.9022833.