ISSN (online): 2349-784X

Study on Web Penetration Testing, Vulnerability Assessment and Preventive Measures

Jamyang Tashi
Associate Lecturer
Department of Information Technology
Jigme Namgyel Engineering College, Royal University of Bhutan, Bhutan

Abstract

The Study on web penetration testing and vulnerability assessment focus on the evaluation of the various vulnerabilities, and tools required to penetrate these vulnerabilities. It focuses on the development of making web applications secure before the intruder tries to attack the web application. It also provides the idea to assess the vulnerabilities and introduce different preventive measures that will help in preventing intruders from accessing sensitive information. The experiments are done using open-source software which is freely available on the internet. OWASP WAP (Damn Vulnerable Web Application) and RIPS (Buggy Web Application) already have the vulnerabilities and are mainly used for the study purpose and analyses of the result. With this study, one can understand how ethical hacking activities are performed and also place necessary security measures in protecting the organization. A similar study practice can be performed over real-life websites and networks for testing the vulnerability and carry out the assessments.

Keywords: Web Penetration, Testing, Vulnerability, Assessments, Preventive Measures

I. INTRODUCTION

Web penetration testing and vulnerability assessment are to find certain flaws and problems(vulnerabilities) in web applications. Vulnerability assessment and testing identify vulnerabilities present in the websites and how much they affect the system. Since the evolution of the internet has led to many cyber-crimes without the authorization of the organization or the owner, unauthorized access has been practiced to fulfill their goals. The carelessness to look for the vulnerabilities when setting up the system has provided the hackers to find ways to attack the technical assets. According to ARN [1], the creator of the first computer worm which was transmitted through the internet was introduced by Morris, who was a student at Cornell University, USA, and his intention was not to harm. The worm replicated rapidly and infected various other computers which resulted in Denial of Service (DoS). It resulted in an estimated $10-$100 million dollars in repair pair. Another case was done by 15 years old, Michael Calce (MafiaBoy) in 2000, he performed a DDoS attack on commercial websites. Later, it resulted in a $US1.2-billion-dollar damage bill. The biggest fraud case in US history was done by T Gonzales from Miami in 2009, he sealed the millions of credit and debit card numbers from over 250 financial institutions and hacked the payment card network from companies. Due to these cases, security should be imposed on the network of the organization. Network security mainly focuses on the protection of valuable information from unauthorized users, so it usually starts with the authorization of the users to access the data in the network. It consists of the rules and regulations or policies implemented by the network administrator to prevent the attack in the network.

Ethical hacking means penetrating the network so that the authorized person can find the vulnerabilities before the hackers and implement strong security on the network. Penetration testing is a process where the authorized tester with a renowned license and who has signed a contract with the organization or company can simulate the attack of the network or computers and provide the output as a report to provide security [2].

Therefore, there are ways an individual or an organization can prevent unauthorized access to the system before a hacker attempts to launch the attack on the particular system. Therefore, the study was aimed to study

A. Aim

- To study various common vulnerabilities, the tools required to exploit those vulnerabilities and also to provide preventive measures to secure the web application.

B. Objectives:

- To study and identify the most common web application vulnerabilities.
- To study the vulnerability assessment of the application.
- Perform penetration testing on the application.
- Provide preventative measures to secure the application.

C. *Problem Statement.*

Network security has become a major need in every organization due to the evolution of various hackings and exploitation of the valuable information of the organization. In today's technology-centric society, threats continue to plague businesses and governments. Threats have gotten more complicated with a change in the type of criminal. Threats come when there are vulnerabilities in the system, and most vulnerabilities are unpatched and misconfigured systems. As better ways are found to defend against attacks, attackers develop new and better ways to bypass this protective technology. Most of the attackers are motivated by financial gain and a desire to control through the use of criminal activities.

Since Bhutan is developing country, most of the Network Administrators doesn't know much on the different kind of threats thus making it vulnerable to cyber-attacks. The knowledge of one's adversaries has always been a key aspect of protecting and defending against attackers.

Therefore, the penetration testing is a process where the authorized tester with a renowned license and who has signed a contract with the organization or company can simulate the attack of the network or computers and provide the output as a report to provide security [2]

II. LITERATURE REVIEW

Any piece of data or information has become an important asset in every organization. Since almost all the information is maintained in the digital form, there is a high possibility of those sensitive data being violated by the intruders which may lead to cybercrimes and loss to the organization or an individual. According to the case study carried out by Reddy1 and Reddy2, the survey of U.S. technology and healthcare executives nationwide, Silicon Valley found that cybercrimes have been a serious threat to the company's data, 98% of companies are focusing and maintaining on increasing their cybersecurity resources, majority of the companies are preparing for when the cyber-attack occurs, and only one-third of the companies are confident of their company's security measures [3]. According to CVE details, the total number of known vulnerabilities is 122,774 till 2019 with the highest number of vulnerabilities of 16,556 in 2018. Out of all vulnerabilities, the code execution has the highest number of known vulnerabilities of 32,718 with a percentage of 26.6 among the other vulnerabilities till 2019 [4].

According to Bhutan Computer Incident Response Team, in 2018 the incident was increased by 8% as compared to 2017 taking the total of resolved incidents 275. While doing the awareness in all the dzongkhags about cybersecurity 50 incidents were reported by constituents. March month is the busiest month for a team with 76 incidents as the highest record in the year 2018. SQL injection, Broken Authentication, Sensitive Data Exposure, Broken Access Control, Security Misconfiguration, Cross-Site Scripting, and Insufficient Logging Monitoring are categorized as the top 10 Web application vulnerabilities in 2020 [5].

The open-source, static web application vulnerability analysis tools, OWASP WAP (DVWA), and RIPS (bWAPP) are used in this project. Damn Vulnerable Web App (DVWA) which is programmed in PHP/MYSQL is too vulnerable and it is mainly used for ethical hackers to test their skills and run this application in a legal environment. It also helps them to understand the vulnerabilities and the processes of securing web applications in a safe environment. The main purpose behind this is to practice various common web vulnerabilities with different difficulty levels [5]. DVWA embeds various vulnerabilities such as SQL injection and Blind SQL injection, and Reflected and Stored XSS which are commonly used to attack current web applications. It also embeds different types of security levels: low (no protection at all), medium (less secured than the high level), and high (properly secured version) and it is also possible to view and compare the source code of each security level [6].

Buggy Web application is an open-source vulnerable web application that is also deliberately kept vulnerable to help developers, security enthusiasts, and students to make them understand and discover the web vulnerabilities. It contains 140 different web vulnerabilities in it including all the major known web vulnerabilities from the OWASP Top 10 project. It is developed in PHP and uses MySQL database [7]. Thus, with the knowledge of network security, every organization can effectively protect its assets that contain sensitive data with an ultimate level of security. The network administrator can study and identify the vulnerabilities before the intruders attack their system and violate their data providing loss to the company or organization. They can perform penetration testing on their system ethically with the various open-source applications/software available or using Kali Linux as the attacking tool since Kali Linux contains tools that are used for penetration testing. Thus, they can identify the vulnerabilities and implement the necessary security to the required fields.

III. METHODOLOGY

Due to the evolution of various cyber threats that have been taking an active role and creating various threats to the sensitive information of the users, this project will mainly focus on the study of the most common vulnerabilities and evaluating those vulnerabilities with various tools available. This particular study validates our findings of what is being said by the other studies. Since it involves the research of other works and verification, penetration testing activities were performed. Therefore, it is to exploit the vulnerabilities and determine whether unauthorized access or other malicious activity is possible. Penetration testing usually includes network penetration testing and application security testing as well as controls and processes around the network and applications.

The study was carried out by setting up a virtual lab - the Oracle Virtual Machine and GNS3 in the Windows 10 (host) with the minimum specification of 8GB RAM and a 64-bit operating system. All the experiments of the common vulnerabilities will be

done in the Virtual Network Lab. Identifying the web vulnerabilities, selection of open-source tools for penetration, testing, vulnerability assessments, and documenting the findings are the process adopted in the successful completion of the study.

IV. DEMOSTRATION AND TESTING

A. SQL Injection.

SQL Injection. Structured Query Language (SQL) is a database language that is designed mainly for managing data in relational database management systems (RDBMS). The commands written in the SQL will add, remove, edit or retrieve information from a database. SQL Injection is a security threat to web applications that will exploit the database of the applications which contain sensitive information. The attacker gains access to the database and can manipulate the information, thus corrupting the system with false information.

In this vulnerability, attackers target the database servers which contain confidential data. The primary objective of this attack is to obtain the information from the database while accessing the table which contains the sensitive information. The cause of SQL injection vulnerabilities is mainly due to the insufficient validation of user input

1) Exploiting the SQL Injection using DVWA.

When we typed number 1 and clicked on the submit button noticed that it returned the first name and the surname of the user with ID=1. To check whether the particular website is vulnerable to SQL Injection, we must add' after the id=1 as shown in the figure below:

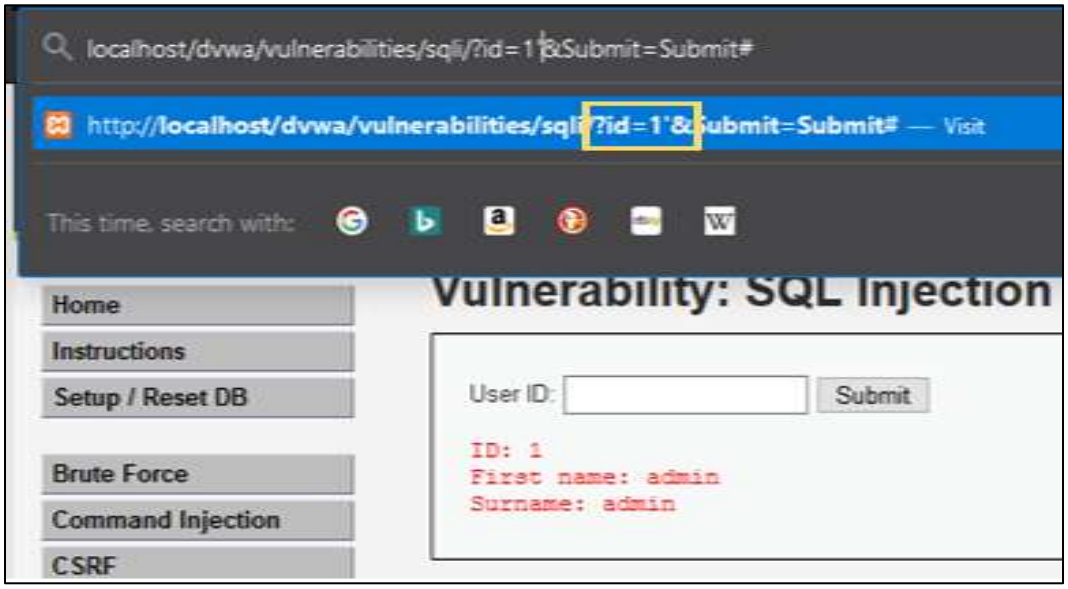

Fig. 4.1 1 : SQL Injection Interface

The following error syntax indicates that it is vulnerable to SQL Injection.

```
You have an error in your SQL syntax; check the manual that corresponds to your

MariaDB server version for the right syntax to use near ''1''' at line 1
```

Figure 4.1 2 : Syntax Error

Now, the attacker can log in using another username and password other than the admin.

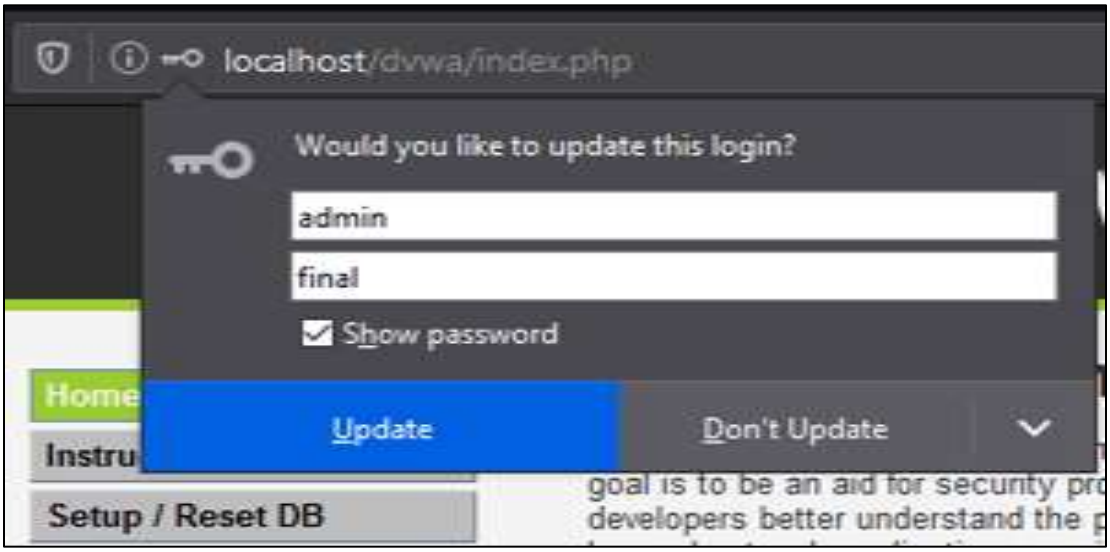

Figure 4.1 3: Another user login

B. Broken Authentication.

Broken Authentication is a web vulnerability in the web application which occurs when there is misconfiguration of session management. Whenever a user login into its account, after the authentication process is completed, a session will be created and activated for the data communication between the server and the client. So, when the attacker can get access into the active session of a user bypassing the authentication process, it is specified as a broken Exploiting Authentication problem of the particular application.

When the user login into the system, the user credentials will be provided. When the request is sent from the client-side to the server, the server will initiate a query to the database for verifying the credentials provided by the user. After the validation is successful, the user will be provided with a specific session ID for communication. And the user can access the provided services. The user credentials of the authenticated users will be stored in browsers. With the help of applications such as cookie manager, eat my cookie and advanced cookie manager+, the intruder can intrude into the other's active session [8]

2) Exploiting the broken authentication vulnerability using the BWAPP.

BWAPP (buggy web application) is a free and open-source insecure web application which are mainly used for practicing penetration for amateur ethical hackers. In the broken authentication, if the session is active, the attacker will capture the cookie from the browsers using applications and then accessing the source code of the particular web page which will show the username and password in clear text, get the information of the login username and passwords.

Figure 4.2 1: Broken Authentication Result

C. Broken Access Control.

Access control means limiting users on certain sections or pages to view or access, depending on their needs. When the authenticated users are not properly restricted on certain privileges or to access certain documents or when the access control is broken then attackers can exploit these flaws to access unauthorized functionality and/or data.

1) Exploiting Broken Access Control using the BWAPP.

Bypassing the access control to gain access to the restricted content is known as broken access control. On this website, only authorized users can access the listed documents. The attacker will copy the URL (http://localhost/bwapp/documents/) and then paste it into the new tab.

Fig. 4.3 1: Folder containing restricted documents

Then it will redirect to this content, where the user can access the restricted content.

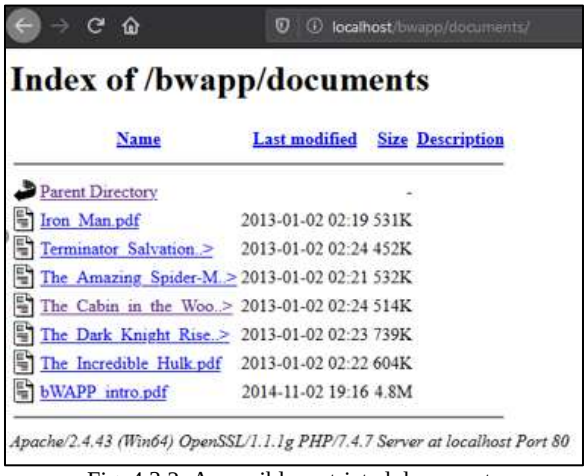

Fig. 4.3 2: Accessible restricted documents

D. Security Misconfiguration.

Security misconfiguration is the most commonly seen issue which occurs mainly due to a default configuration, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. All operating systems, frameworks, libraries, and applications should be securely configured, and also, they must be patched/upgraded now and then. When security is misconfigured then attackers can easily exploit these flaws and access the system.

1) Exploiting vulnerability using DVWA.

In this vulnerability, the source code has been written in such a way that it can accept any file when we upload rather than selecting only the image which it has mentioned. In fig. when the user tried to upload a file with extension .docx, it reads that it has been successfully uploaded and in fig when the user again tried to upload an image file, it has successfully uploaded. The vulnerability here is that the source code hasn't mentioned or restricted a file other than the file with the image extension.

Fig. 4.4 1: Upload of a file with any extension

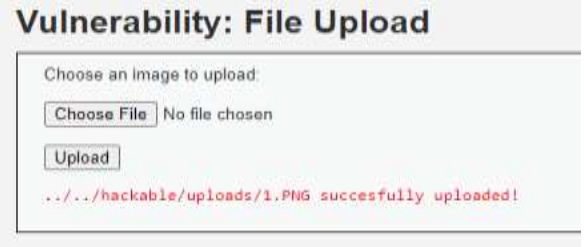

Figure 4.4 2: Upload of a file with image extension

E. Cross-Site Scripting (XSS).

Cross-site scripting XSS is untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface websites, or redirect the user to malicious sites.

1) Exploiting the XSS using XAMPP.

This vulnerability will allow the attackers to add the JavaScript alert which will change the response of the particular page injecting the webpage to respond abnormally.

In the figure below, the attacker can write the JavaScript into the message field and this will be saved into the database, and when you submit the 'Sign Guestbook' button.

Fig. 4.5 1: Exploiting the website

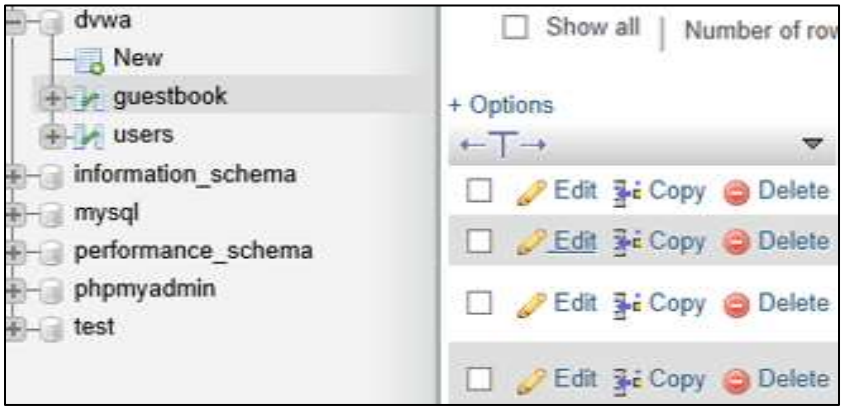

Fig. 4.5 2: Accessing the database

This message popped up indicating it has violated the website.

Figure 4.5 3: Successful XSS Attack

F. Sensitive Data Exposure.

Sensitive Data Exposure is a type of security vulnerability where a web application fails to protect the confidential data of an organization and hence exposes flaws to attackers for an attack. the study found that many web applications and APIs do not protect sensitive data that are sensitive such as financial, healthcare etc. An attacker may steal information or modify information that is weakly protected. Information such as credit card details and their fraud, identity theft, or other cybercrimes has been raised over the period. Therefore, sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.
Exploiting vulnerability using BWAPP.

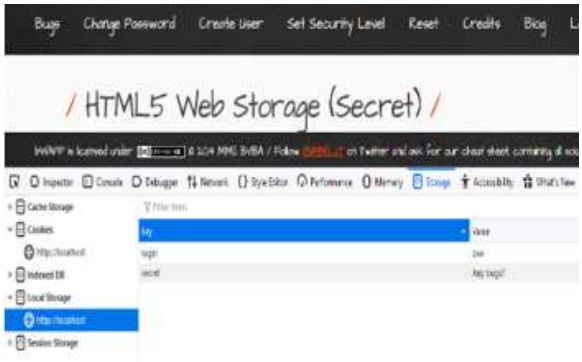

Figure 4.6 1: Exploiting Sensitive data exposure

G. Denial of Service (DoS)

A DoS attack is an attempt performed by a hacker to flood a user's or an organization's system that makes the system busy and denied to the services. Service on the system is unavailable to a user because it's kept busy trying to respond to an exorbitant number of requests made against the back end of the system. This kind of attack doesn't gain the hacker access to any information but rather annoys the target and interrupts their service. DoS attacks can be devastating and have a substantial impact when sent from multiple systems at the same time (DDoS attacks).
2) Exploiting vulnerability using LOIC (Low Orbit Ion Cannon) and hosted website.
This vulnerability is tested using the LOIC application which is mainly designed for the DoS attack. With the help of the URL of a particular website, it can impose an attack on that website and make it busy and delay the intended users from receiving their request.

After entering the URL and then pressing 'Lock On', and then press the button on the right corner, it will generate the IP address of the particular URL. Then select the Method under the Attack options.

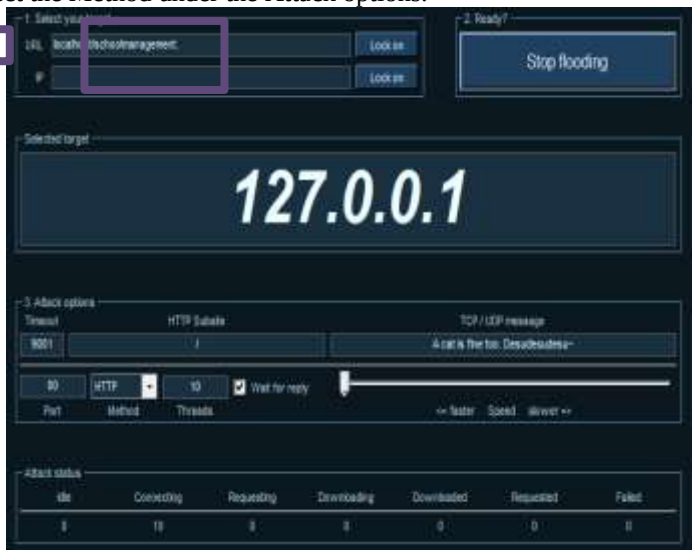

Fig. 4.7 1: Result of DoS attack

Note: we cannot successfully attack on our hosted website since the website has not been publicized. If you used it to attack on the real website, then it will send lots of request to the server, keeping it busy and making it unavailable to the valid users.

H. Cross-Site Request Forgery (CSRF).

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated.

3) Exploiting CSRF vulnerability using XAMPP.

This vulnerability will provide the end-user to execute the unwanted actions on a web application in which they are currently active which will result in modifying the sensitive information such as e-mail address, username or password of the user. The below fig. is an extracted portion of the HTML code of the login page and pasted and saved in .html extension in notepad, which will create a form to reset the password in the form of some sort of advertisement or anything the attacker wants to fool the valid users. In this figure, the new password is given 'final' and when the user clicks the button, it will successfully change the password.

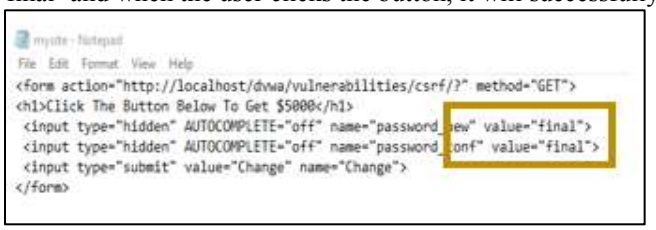

Fig. 4.8 1: Changing the password

In this figure, it has been seen that without the admin's permission, the password has been successfully changed.

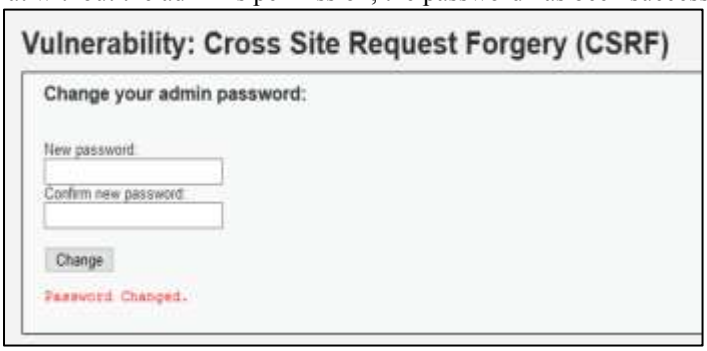

Fig. 4.8 2: Successfully password changed interface

V. ASSESSMENT & PREVENTIVE MEASURES

A. SQL Injection.

SQL injection is vulnerable because SQL Injection is very common with PHP and ASP applications due to the prevalence of older functional interfaces. It normally happens when poor SQL commands are used to check user names and passwords, which may be possible to connect to a system as another user with no previous knowledge of the password. It is also vulnerable when user-supplied data is not validated, filtered, or sanitized by the application. Malicious data is used or concatenated, such that the SQL or command contains both structure and hostile data in dynamic queries, commands, or stored procedures. To protect this vulnerability, use a safe API that avoids the use of the interpreter entirely or use positive or "whitelist" server-side input validation and always escape special characters. SQL structures such as table names, column names, and so on cannot be escaped, and thus user-supplied structure names are dangerous. We can use the LIMIT and other SQL controls within queries to prevent the mass disclosure of records in case of SQL injection.

B. Broken Authentication.

Broken Authentication is vulnerable when leaving the login page for admins publicly accessible to all visitors of the website. When the application allows automated attacks such as credential stuffing, where the attacker will come with a list of valid usernames and passwords. Permits brute force or other automated attacks or default, weak, or well-known passwords, such as" Password1″ or "admin/admin.″ It is also vulnerable when the user uses weak or ineffective credential recovery and forgot-password processes, such as "knowledge-based answers," which cannot be made safe and using plain text, encrypted, or weakly hashed passwords and also exposes session IDs in the URL (e.g., URL rewriting). To prevent these vulnerabilities do not deploy systems with default credentials and always check for a list of weak passwords. Harden all authentication-related processes like registration and credential recovery and limit or delay failed login attempts.

C. Broken Access Control.

Broken Access Control is vulnerable if user Bypass access control checks by modifying the URL, internal application state, or the HTML page, or simply using a custom API attack tool. Allowing the primary key to be changed to another's user's record, permitting viewing or editing someone else's account. To prevent this vulnerability, deny access by default, except for public resources, and employ lease privileged, don't allow users to create, read or delete any record. Remove unnecessary services off the server and log failed access attempts should give an alert to the admins. similarly, the access control weaknesses also happen due to the lack of automated detection, and lack of effective functional testing by application developers.

D. Security Misconfiguration.

Application is vulnerable if the application has unnecessary features are enabled or installed (e.g. unnecessary ports, services, pages, accounts, or privileges) and default accounts and their passwords still enabled and unchanged. For upgraded systems, the latest security features are disabled or not configured securely, and when the software is out of date or vulnerable. To prevent security misconfiguration vulnerabilities automated scanners are useful for detecting misconfigurations, use of default accounts or configurations, unnecessary services, legacy options, and all, and always remove or do not install unused features and frameworks.

E. Cross-Scripting Scripting (XSS).

The application was vulnerable to attack when the application or API includes invalidated user input as part of HTML output and when the application or API stores unsensitized user's input that is viewed at a later stage by another user or an administrator. To prevent XSS vulnerabilities used frameworks that automatically escape XSS. One must study the limitations of each framework's XSS protection and appropriately handle the use cases which are not covered and escape untrusted HTTP request data based on the context in the HTML output (body, attribute, JavaScript, CSS, or URL). Enable a Content Security Policy (CSP)as a defense-in-depth mitigating control against XSS. It is effective if no other vulnerabilities exist that would allow placing malicious code via a local file that includes the path traversal overwrites or vulnerable libraries from permitted content delivery networks.

F. Sensitive Data Exposure.

Web Storage is one of HTML's new local storage solutions, but that is not a standard for replacing cookies. Cookies are absolutely necessary for handling client and server communication as part of the HTTP protocol. Sessions are dependent on implementation. Web Storage intends to solve the local storage that should not be done with cookies but has to use cookies. However, in this case, important information such as secrets should not be stored locally in Web Storage. The administrator or an individual should classify the data processed, stored or transmitted by an application and identify which data is sensitive according to privacy laws, regulatory requirements, or business needs. It is recommended to apply controls as per the classification. All sensitive data should be encrypted and should ensure up-to-date and strong standard algorithms, protocols, and keys are in place. The data should be encrypted with security protocols such as TLS with perfect forward secrecy (PFS) ciphers, cipher prioritization by the server, and security parameters. Therefore, it is recommended to enforce encryption using directives like HTTP Strict Transport Security and disable caching for the response that contains sensitive data

G. Denial of Service (DoS).

A Denial-of-Service Attack (DoS attack) is a cyber-attack that perpetrator seeks to make a machine or network resource unavailable to its intended use by temporarily or indefinitely disrupting the service of a client connected to the internet. Denial of service is typically accomplished by flooding the targeted machine or resource with many requests in an attempt to overload systems. LOIC does not rely on any vulnerabilities. Therefore, vulnerability scanners and network scanners cannot be used to protect against it. Web application firewalls (WAF) work well for most DoS/DDoS attacks but intrusion detection/prevention systems (IDS/IPS) are the best tool to use to protect against such attacks in general.

H. Cross-Site Request Forgery.

To prevent this, the use of an Anti-CSRF token (also known as to request verification tokens) must be utilized. These tokens are simply randomly generated values included in any form/request that warrants protection. We should take a note that this value should be unique for every individual session which guarantees that every form/request is tied to the authenticated user and, therefore, protected from CSRF.

VI. CONCLUSION

To conclude the findings, it is important to know that there are many vulnerabilities and among many, few of them were tested as covered above. Therefore, it is important for Network Administrators and Network Engineers to know the most common vulnerabilities and their preventive measures. The internet was introduced in Bhutan in 1999, it has developed so fast that everything thing in the government and private business depends heavily on internet resources. With this, there are cybersecurity issues that have become a major threat to the Information Technology world. In today's technology-centric society, threats continue to plague businesses and governments. This study was carried out on the OWASP WAP (DVWA) and RIPS (bWAPP) since it was developed as the platform for learners to perform those penetration testing. This platform has provided knowledge to the people on the web penetration that can be done on the website that has been hosted (live website) so that the administrator would know how secure their website is. Although exploiting vulnerabilities and vulnerability assessments required a lot of knowledge of web security, the project has been successful and satisfying and experiences based on the web penetration testing were gained.

REFERENCES

[1] "ARN," [Online]. Available: https://www.arnnet.com.au/slideshow/341113/top-10-most-notorious-cyber-attacks-history/. [Accessed March 2020].
[2] M. N. Mirjalili, "A Survey on Web Penetration Test," ACSIJ Advances in Computer Science: an International Journal. [Online]. [Accessed July 2020].
[3] G. N. &. R. G. J. Reddy, "A Study of Cyber Security Challenge and its Emerging Trends on Latest Technologies," ResearchGate, [Online]. Available: https://www.researchgate.bet/publication/260126665_A_Study_Of_Cyber_Security-Challenges_And_Its_Emerging_Trends_On_Latest_Technologies. [Accessed July 2020].
[4] "The Ultimate security vulnerability datasource," CVE details, [Online]. Available: https://www.cvedetails.com/vulnerabilities-by-types.php. [Accessed July 2020].
[5] Avishek, "What is DVWA and why ethical hacker love this!," Khanna Security Blog, 27 June 2018. [Online]. Available: https://khannasecurityblog.com/blog/what-is-dvwa-and-why-ethical-hacker-love-this/. [Accessed June 2020].
[6] F. L. Lebeau, "Model-Based Vulnerabiltiy Testing for Web Applications," HAL archives-ouvertes, January 2013. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00935070/document. [Accessed June 2020].
[7] S. &. K. Tyagi, "Evaluation of Static Web Vulnerabilty Analysis Tools," 2018. [Online]. Available: 1-6.10.1109/PDGC.2018.8745996. [Accessed June 2020].
[8] M. V. &. J. A. P, "Network Security and Types of Network Attacks in Network," ResearchGate, 2015. [Online]. Available: https://www.researchgate.net/publication/277723629_Network_Security_and_Types_of_Attacks_in_Network. [Accessed June 2020].
[9] "OWASP Top Ten," OWASP, [Online]. Available: https://owasp.org/www-project-top-ten/. [Accessed March 2020].