Penetration Testing Report

Client Information

Target: HackThisSite.org
Challenges: Basic Levels 1-11
Penetration Report Date: October 2, 2024
Penetration Test Date: October 1, 2024

Executive Summary

This report details the results of penetration testing undergone to identify and exploit vulnerabilities found in the "Basic" web challenge from HackThisSite.org - https://www.hackthissite.org/missions/basic/. These challenges illustrate how web application vulnerabilities can be exploited. They provide examples of how an attacker might compromise a web application, potentially leading to a breach of confidentiality, integrity, and availability that could result in business, financial or reputational loss for an organization. The objective of the penetration testing report is to identify vulnerabilities associated with this web application, attempt to exploit these vulnerabilities, and provide recommendations for better securing the web application.

Key Findings

Basic Level 1

Vulnerability Description and Findings:
The login form contains hardcoded credentials in the HTML source code. An attacker can view the source and retrieve the credentials.

Remediation:
Avoid storing passwords in plaintext in the source code, they should be stored as a hashed value in a separate file. If the password is not hashed, it should be kept in a properly encrypted file.

Basic Level 2

Vulnerability Description and Findings:
The absence of a password file prevents password verification. That results in a submission of an empty password field, allowing for successful authentication.

Remediation:
Always inspect for vulnerabilities by testing the application with empty fields, commonly used credentials and default login credentials submissions.

Basic Level 3

Vulnerability Description and Findings:
The location of the password is found in the source code. Hackers exploring the directory or file path structure allows an attacker to potentially retrieve the password within the source code.

Remediation:
Directory structure of a web application should be mapped out when testing it. In order to get a better understanding of how the application works. And for discovery of interesting or overlooked elements such as login pages.

Basic Level 4

Vulnerability Description and Findings:
The credentials are hardcoded into the application, allowing the script of a web page to be manipulated to redirect a password to the desired recipient.

Remediation:
Sensitive information should not be found on client-side code in order to protect it from being accessed by unauthorized parties.

Basic Level 5

Vulnerability Description and Findings:
The credentials are hardcoded into the application, allowing the script of a web page to be manipulated to redirect a password to the desired recipient.

Remediation:
Sensitive information should not be found on client-side code in order to protect it from being accessed by unauthorized parties.

Basic Level 6

Vulnerability Description and Findings:
Weak encryption is used, allowing an attacker to determine the type of encryption used. The predictable encryption method is easily recognized and exploitable to attackers.

Remediation:
Use stronger encryption, apply hashing, add salt, and constantly change the encryption key.

Basic Level 7

Vulnerability Description and Findings:
The web application is using a Perl script that allows for command injection. The script can be used for exploitation, to list the current directory and execute arbitrary commands on the host system.

Remediation:
Web applications should check and sanitize user input before executing it.

Basic Level 8

Vulnerability Description and Findings:
Server-side injection is present when testing the input value result. While files are being stored in the tmp directory when they should be not. Directories are viewable, which allows for the file containing the password to be located.

Remediation:
User input should be validated and sanitized. A web application firewall (WAF) can be used while web servers and web applications should be kept up-to-date. Web servers should be configured to not allow server-side injection execution within user-controlled directories.

Basic Level 9

Vulnerability Description and Findings:
Server-side injection is present when testing the input value result. While files are being stored in the tmp directory when they should be not. Directories are viewable, which allows for the file containing the password to be located.

Remediation:
User input should be validated and sanitized. A web application firewall (WAF) can be used while web servers and web applications should be kept up-to-date. Web servers should be configured to not allow server-side injection execution within user-controlled directories.

Basic Level 10

Vulnerability Description and Findings:
Cookie information is easily modifiable. Incorrect implementation of cookies leads to a variety of security issues such as session hijacking, cross-site scripting, insecure storage of sensitive information, and misconfiguration.

Remediation:
Yes/no cookies should not be used for authenticating users. Cookies should be used to maintain a user's authenticated state after the initial authentication process has been completed.

Basic Level 11

Vulnerability Description and Findings:
The .htaccess file that configures the Apache web server, has incorrect file permissions. Which allows the files to be read by anyone. This poses a security vulnerability as it may contain sensitive information.

Remediation:
Correct the file permissions on the .htaccess file to prevent unauthorized access. Also implement both authentication and access controls to ensure proper protection for the directories.

Conclusion

In conclusion, the penetration testing report identified several vulnerabilities in HackThisSite.org's "Basic" web challenges, including hardcoded passwords, hidden credentials in the source code, weak encryption, and command injection. These issues pose risks to confidentiality, integrity, and availability, potentially leading to business, financial, or reputational damage. The report provided recommendations for improving web application security and emphasized the importance of regular penetration testing and vulnerability assessments to address security flaws before attackers can exploit them.