HIWA INC. : APPLICATION VULNERABILITIES

Prepared for:

HIWA Inc.

Prepared by:

Angie Diaz

Adelphi University

March 26, 2023

Summary of Findings:

The assessment of the HIWA application is to identify any vulnerabilities that may cause any future cyber attacks. After a thorough investigation, the findings displayed:

- The application is vulnerable to SQL injection attacks, which can allow an attacker to have unauthorized access to sensitive data.
- The application is vulnerable to cross-site scripting(XSS) attacks, which can allow an attacker to inject malicious scripts and hijack a user's account by stealing their credentials and data. Furthermore, this could result in making changes to the way content is presented, or directing the user to a different page or website.
- The application is vulnerable to broken authentication, in which the system allows passwords to be easily guessed. Therefore this allows attackers to spoof authentication.
- The application lacks transport-level security(TLS), which allows for sensitive data exposure.
- The application allows insecure file uploads, which can allow an attacker to execute any code they want, without any restriction or limitation.
- The application allows unprotected configuration files to be accessed by the public.

Thus, these findings display that the HIWA application is at high risk of a cybersecurity attack. The identified vulnerabilities above can allow attackers to have unauthorized access to private data and hijack other users.

Methodology:

To identify the vulnerabilities in the HIWA application, manual testing was conducted to validate these weaknesses in the application and to simulate the type of exploitation a future attacker could do.

The HIWA application is vulnerable to SQL injection attacks. The attacker's method is that an input is put into a browser and then it's sent to the web server. After the web server sees the quantitative data, it's sent to the DBMS due to the SQL associated with it and then the SQL processes this information. In the HIWA inc. login page, a manual test of SQL injection is as follows in the username and password:

- In the username field, the attacker places any random user
- In the password field, the attacker injects ‘ or 1=1 - - , the result of this is that the SQL processes this information as true, therefore giving access to any account. In addition, if the attacker chose to use ‘ or 1 = 1 Limit 1 - - , this would have given access to an admin account.

The manual testing that was conducted that identified the vulnerability of XSS in the product descriptions is as follows:

- In the product page, HTML code was used in the user input. As a result data can be manipulated and changed to give an attacker more leverage, an example would be a trojan login panel.
- Script tags will also be used, an example is <script language="JavaScript">document.location="http://google.com"</script>. This allows the attacker to hijack the user's session.

Broken authentication allows systems to be easily accessed. The manual testing that was conducted is that the HIWA login page is examined and you see that there is no option to use multi-factor authentication, the passwords can be easily guessed, non-existent time-out system for a session, and the poor design of the 'forgot password'. The weak system allows attackers to use methods such as XSS and SQL injection to take over user's accounts.

The manual testing that identified the lack of TLS in the HIWA application is as follows:

- When you go to the HIWA login page, http://192.168.42.99/login.php, the certificate for the page is non-existent. This is found on the address bar on the left side with a lock identified symbol. The information found there states if the connection is safe, private, and secured. Therefore, the connection isn't encrypted. This allows information sent over to be seen/accessed by the public or in this case attackers.

The manual testing that identified the vulnerability of insecure file uploads is as follows:

- The HIWA logo is a png file, therefore since the user holds most of the control in the uploads, the attacker can upload malicious code and change in this case the logo of the web application.
- The system will allow the upload of a completely different file as long as the file type is the same. In the source code, it states that the HIWA logo is a png file, therefore an upload overwrites the HIWA png file, and now all users can see the entire new image that was uploaded.

The manual testing that was executed to identify the vulnerability of insecure files is as follows:

- In the address bar, http://192.168.42.99/login.php, the end of the URL was omitted and then the page was redirected to the URL, http://192.168.42.99/. In this page, we are taken into the directory of the HIWA application and see multiple files that can be accessed. One example is a file named "config.phplib". Although the file is empty, an attacker can access the source code.

Overview of Findings:

1. SQL injection vulnerability:

   If a SQL injection exploit is successful, it can result in unauthorized access to sensitive data stored in the database, modification of database data by inserting, updating or deleting records, execution of administrative tasks on the database such as shutting down the DBMS, retrieval of files stored on the DBMS file system, and in certain cases, issuing commands to the operating system.

2. Cross-Site Scripting(XSS) vulnerability:

   This application is vulnerable as it doesn't verify user input. Therefore, malicious scripts are injected into websites and this could be executed on the client-side and steal sensitive information or perform unauthorized actions.

3. TLS vulnerability:

   This application isn't protected because the connection doesn't have secure encryption. The web application uses http and not http2, therefore you have to manually input security. The application lacks encryption and a certificate.

4. Insecure file upload:

   The file upload functionality of the application is insecure as it allows an attacker to upload malicious files to the server, which could be executed on the server-side and compromise the system.

5. Insecure configuration files:

   The security on the configuration files in the application demonstrates that they were stored in plain text without any permissions required. These vulnerabilities could allow attackers to gain access and perform unauthorized actions, such as modifying critical data or executing malicious code.

Interpretation of Results:

The identified vulnerabilities in the HIWA application are of high risk of being compromised and being the victim of a cyber attack. These vulnerabilities affect the CIA triad or confidentiality, integrity, and availability of the system. These vulnerabilities have many effects which include user session hijacking, stolen sensitive data, and manipulation of data. It is recommended that these risks be confronted immediately to ensure the safety of the system and users.

Recommendations:

1. Creating frameworks that ensure that all variables go through validation and are sanitized is a method to resist XSS and SQL injection attacks. For XSS attacks, developers must remove HTML control characters from the text prior to sending it to the browser.
2. Another method to prevent SQL injection is using parameterized queries with prepared statements so an attacker doesn't change the intent of a query.
3. Implement file type and size validation and sanitization to prevent insecure file uploads.
4. Use http2 because by default it's already encrypted.
5. Use MFA for password security

Conclusion:

After a thorough review of the HIWA application, the application presents with various security breaches that a cyber attacker can manipulate. The security testing demonstrated successful exploitation through XSS and SQL injection attacks, insecure configuration files, insecure uploads, and poor user authentication. Therefore, it is recommended that user input is validated and sanitized, parameterized queries are implemented, and MFA is used to prevent any future cybersecurity attacks. In addition to these new precautions, frequent security tests can guarantee earlier detection of vulnerabilities and mitigation of them.

Overall, it is important to protect the confidentiality, integrity, and availability of the system. Therefore, the HIWA application vulnerabilities are strongly recommended to be mitigated to prevent any present or future attacks to the application.