Aibox

APPLICATION PENETRATION TEST REPORT

March 2025

ExVul

www.exvul.com

Table of Contents

1. EXECUTIVE SUMMARY

Exvul Web3 Security was engaged by **Aibox** to conduct a penetration test on the application. The assessment was conducted in accordance with our systematic approach to evaluate potential security issues based upon customer requirement. The report provides detailed recommendations to resolve the issue and provide additional suggestions or recommendations for improvement.

The outcome of the assessment outlined in chapter 3 provides the system's owners a full description of the vulnerabilities identified, the associated risk rating for each vulnerability, and detailed recommendations that will resolve the underlying technical issue.

1.1 Rating Methodology

To standardize the evaluation, we define the following terminology based on OWASP Risk Rating Methodology [10] which is the gold standard in risk assessment using the following risk models:

- Likelihood: represents how likely a particular vulnerability is to be uncovered and exploited in the wild.
- Impact: measures the technical loss and business damage of a successful attack.
- Severity: determine the overall criticality of the risk.

Likelihood can be: High, Medium and Low and impact are categorized into for: High, Medium, Low, Informational. Severity is determined by likelihood and impact and can be classified into five categories accordingly, Critical, High, Medium, Low, Informational shown in table 1.1.

Likelihood				
High	*Informational*	Medium	High	Critical
Medium	*Informational*	Low	Medium	High
Low	*Informational*	Low	Low	Medium
	Informational	*Low*	*Medium*	*High*

IMPACT

Table 1.1 Overall Risk Severity

2. FINDINGS OVERVIEW

2.1 Project Info And Contract Address

Project Name: Aibox

Penetration Test Time: March 21, 2025 – March 28, 2025

Target	Link
Aibox	https://apifox.com/apidoc/shared-de73e4e3-8c7e-4f84-af10-ade3f72ae71b
Aibox	iOS/Android APP

2.2 Summary

Severity	Found	
Critical	0	
High	1	🟥
Medium	1	🟨
Low	1	🟦
Informational	0	

2.3 Key Findings

ID	Severity	Findings Title	Status	Confirm
NVE- 001	High	allowBackup Attribute Enabled (Insecure Backup Exposure)	Fixed	Confirmed
NVE- 002	Medium	APK expo Certificate Weak Password	Fixed	Confirmed
NVE- 003	Low	Unsafe HTTP Methods Enabled on Server	Fixed	Confirmed

Table 2.3: Key Test Findings

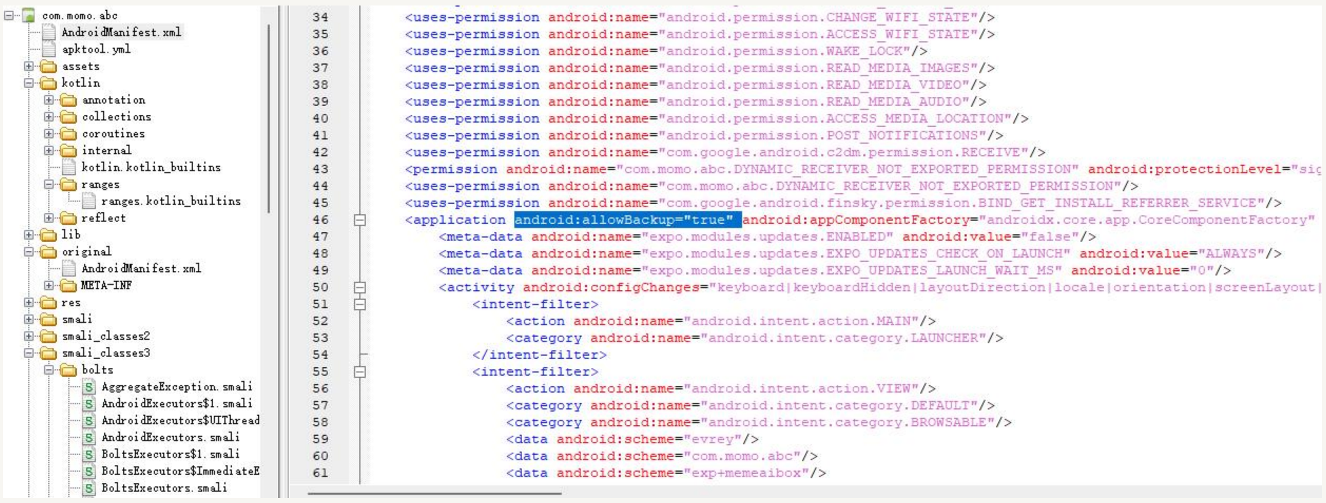
3. DETAILED DESCRIPTION OF FINDINGS

3.1 allowBackup Attribute Enabled (Insecure Backup Exposure)

ID:	NVE-001	**Location:**	AndroidManifest.xml
Severity:	High	**Category:**	Business Issues
Likelihood:	Medium	**Impact:**	High

Description:

In the AndroidManifest.xml file of the Android app, if android:allowBackup="true" (the default setting), the app's data (including private data, SharedPreferences, databases, etc.) can be exported via Android's backup functions (such as ADB backup or cloud backup). For Web3 transaction apps, this data may contain: sensitive information like mnemonics, private keys, Keystore files, and transaction records; user credentials such as login tokens, API keys, and wallet passwords (if not encrypted); and configuration data like node RPC URLs and chain IDs.



```
com.momo.abc
    AndroidManifest.xml
    apktool.yml
    assets
    kotlin
        annotation
        collections
        coroutines
        internal
        kotlin.kotlin_builtins
        ranges
            ranges.kotlin_builtins
        reflect
    lib
    original
        AndroidManifest.xml
        META-INF
    res
    smali
    smali_classes2
    smali_classes3
        bolts
            AggregateException.smali
            AndroidExecutors$1.smali
            AndroidExecutors$UIThread
            AndroidExecutors.smali
            BoltsExecutors$1.smali
            BoltsExecutors$ImmediateE
            BoltsExecutors.smali
```

```xml
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.READ_MEDIA_IMAGES"/>
<uses-permission android:name="android.permission.READ_MEDIA_VIDEO"/>
<uses-permission android:name="android.permission.READ_MEDIA_AUDIO"/>
<uses-permission android:name="android.permission.ACCESS_MEDIA_LOCATION"/>
<uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
<permission android:name="com.momo.abc.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION" android:protectionLevel="sig
<uses-permission android:name="com.momo.abc.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION"/>
<uses-permission android:name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
<application android:allowBackup="true" android:appComponentFactory="androidx.core.app.CoreComponentFactory"
    <meta-data android:name="expo.modules.updates.ENABLED" android:value="false"/>
    <meta-data android:name="expo.modules.updates.EXPO_UPDATES_CHECK_ON_LAUNCH" android:value="ALWAYS"/>
    <meta-data android:name="expo.modules.updates.EXPO_UPDATES_LAUNCH_WAIT_MS" android:value="0"/>
    <activity android:configChanges="keyboard|keyboardHidden|layoutDirection|locale|orientation|screenLayout|
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
        <intent-filter>
            <action android:name="android.intent.action.VIEW"/>
            <category android:name="android.intent.category.DEFAULT"/>
            <category android:name="android.intent.category.BROWSABLE"/>
            <data android:scheme="evrey"/>
            <data android:scheme="com.momo.abc"/>
            <data android:scheme="exp+memeaibox"/>
```

Recommendations:

Add amount validation to ensure the amount parameter matches withdraws[withdrawId].amount for accurate withdrawals.

1. Disable backup functionality

- Explicitly disable backup in the <application> tag of AndroidManifest.xml:
    - android:allowBackup="false"
    - android:fullBackupOnly="false" (necessary for Android 12+ to prevent automatic cloud backup).

2. Extra protection for sensitive data

- Encrypt private keys/mnemonics using AndroidKeyStore
- Avoid storing highly sensitive information in SharedPreferences or databases in plaintext

3. Backup content whitelist (if partial backup is needed)

- Create backup_rules.xml under res/xml to allow only non-sensitive data:
- <full-backup-content> - <exclude domain="sharedpref" path="sensitive_prefs.xml"/> - <exclude domain="database" path="encrypted_wallet.db"/></full-backup-content>

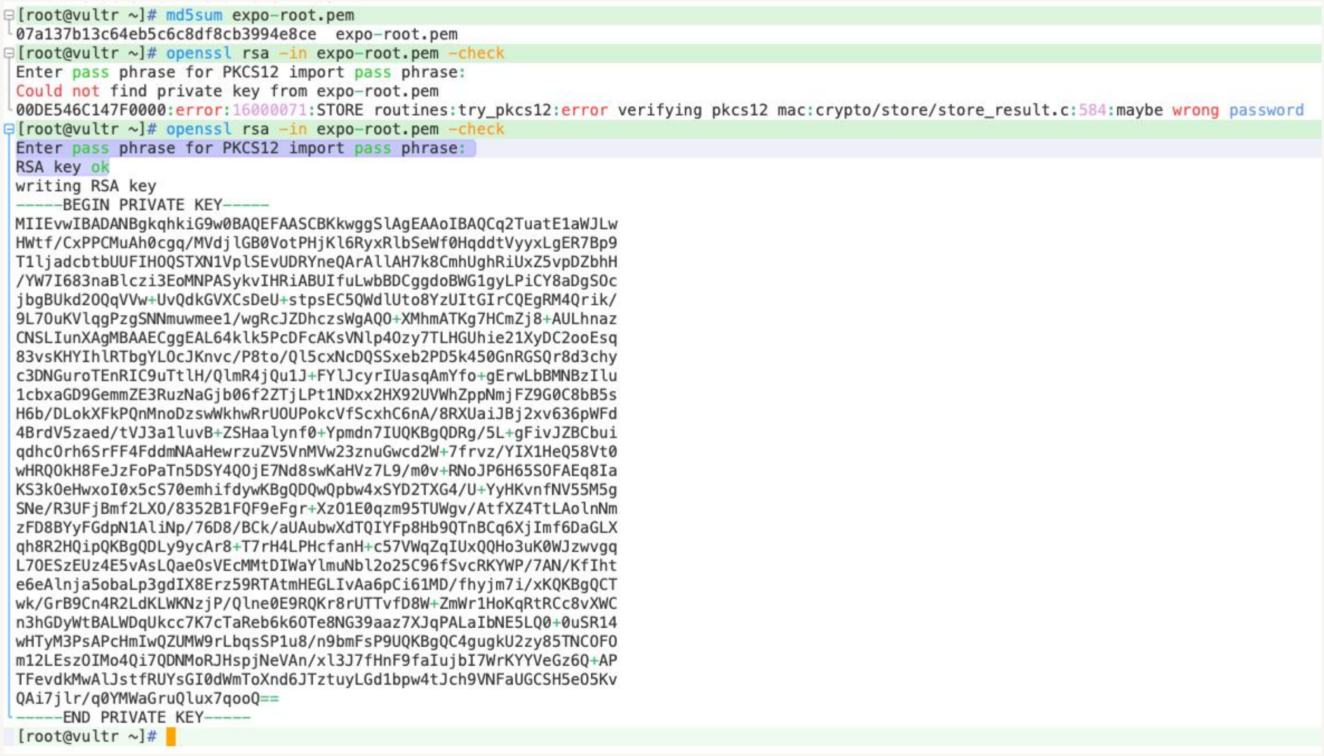
Result: Confirmed

3.2 APK expo Certificate Weak Password

ID:	NVE-002	**Location:**	APK
Severity:	Medium	**Category:**	Business Issues
Likelihood:	High	**Impact:**	Medium

Description:

A weak password (123456) was found for the expo-root.pem certificate in the APK. Attackers can extract and decrypt the certificate using the known password, enabling code signing bypass. If this certificate is used to verify app updates or dynamically loaded code, attackers could forge valid code packages and bypass security checks.



```
[root@vultr ~]# md5sum expo-root.pem
07a137b13c64eb5c6c8df8cb3994e8ce  expo-root.pem
[root@vultr ~]# openssl rsa -in expo-root.pem -check
Enter pass phrase for PKCS12 import pass phrase:
Could not find private key from expo-root.pem
00DE546C147F0000:error:16000071:STORE routines:try_pkcs12:error verifying pkcs12 mac:crypto/store/store_result.c:584:maybe wrong password
[root@vultr ~]# openssl rsa -in expo-root.pem -check
Enter pass phrase for PKCS12 import pass phrase:
RSA key ok
writing RSA key
-----BEGIN PRIVATE KEY-----
MIIEvwIBADANBgkqhkiG9w0BAQEFAASCBKkwggSlAgEAAoIBAQCq2TuatE1aWJLw
HWtf/CxPPCMuAh0cgq/MVdjlGB0VotPHjKl6RyxRlbSeWf0HqddtVyyxLgER7Bp9
T1ljadcbtbUUFIHOQSTXN1VplSEvUDRYneQArAllAH7k8CmhUghRiUxZ5vpDZbhH
/YW7I683naBlczi3EoMNPASykvIHRiABUIfuLwbBDCggdoBWG1gyLPiCY8aDgSOc
jbgBUkd2OQqVVw+UvQdkGVXCsDeU+stpsEC5QWdlUto8YzUItGIrCQEgRM4Qrik/
9L7OuKVlqgPzgSNNmuwmee1/wgRcJZDhczsWgAQO+XMhmATKg7HCmZj8+AULhnaz
CNSLIunXAgMBAAECggEAL64klk5PcDFcAKsVNlp4Ozy7TLHGUhie21XyDC2ooEsq
83vsKHYIhlRTbgYLOcJKnvc/P8to/Ql5cxNcDQSSxeb2PD5k450GnRGSQr8d3chy
c3DNGuroTEnRIC9uTtlH/QlmR4jQu1J+FYlJcyrIUasqAmYfo+gErwLbBMNBzIlu
1cbxaGD9GemmZE3RuzNaGjb06f2ZTjLPt1NDxx2HX92UVWhZppNmjFZ9G0C8bB5s
H6b/DLokXFkPQnMnoDzswWkhwRrUOUPokcVfScxhC6nA/8RXUaiJBj2xv636pWFd
4BrdV5zaed/tVJ3a1luvB+ZSHaalynf0+Ypmdn7IUQKBgQDRg/5L+gFivJZBCbui
qdhcOrh6SrFF4FddmNAaHewrzuZV5VnMVw23znuGwcd2W+7frvz/YIX1HeQ58Vt0
wHRQOkH8FeJzFoPaTn5DSY4QOjE7Nd8swKaHVz7L9/m0v+RNoJP6H65SOFAEq8Ia
KS3kOeHwxoI0x5cS70emhifdywKBgQDQwQpbw4xSYD2TXG4/U+YyHKvnfNV55M5g
SNe/R3UFjBmf2LXO/8352B1FQF9eFgr+XzO1E0qzm95TUWgv/AtfXZ4TtLAolnNm
zFD8BYyFGdpN1AliNp/76D8/BCk/aUAubwXdTQIYFp8Hb9QTnBCq6XjImf6DaGLX
qh8R2HQipQKBgQDLy9ycAr8+T7rH4LPHcfanH+c57VWqZqIUxQQHo3uK0WJzwvgq
L7OESzEUz4E5vAsLQaeOsVEcMMtDIWaYlmuNbl2o25C96fSvcRKYWP/7AN/KfIht
e6eAlnja5obaLp3gdIX8Erz59RTAtmHEGLIvAa6pCi61MD/fhyjm7i/xKQKBgQCT
wk/GrB9Cn4R2LdKLWKNzjP/Qlne0E9RQKr8rUTTvfD8W+ZmWr1HoKqRtRCc8vXWC
n3hGDyWtBALWDqUkcc7K7cTaReb6k6OTe8NG39aaz7XJqPALaIbNE5LQ0+0uSR14
wHTyM3PsAPcHmIwQZUMW9rLbqsSP1u8/n9bmFsP9UQKBgQC4gugkU2zy85TNCOFO
m12LEszOIMo4Qi7QDNMoRJHspjNeVAn/xl3J7fHnF9faIujbI7WrKYYVeGz6Q+AP
TFevdkMwAlJstfRUYsGI0dWmToXnd6JTztuyLGd1bpw4tJch9VNFaUGCSH5eO5Kv
QAi7jlr/q0YMWaGruQlux7qooQ==
-----END PRIVATE KEY-----
[root@vultr ~]#
```

Recommendations:

Replace the weak password with a strong, complex one.

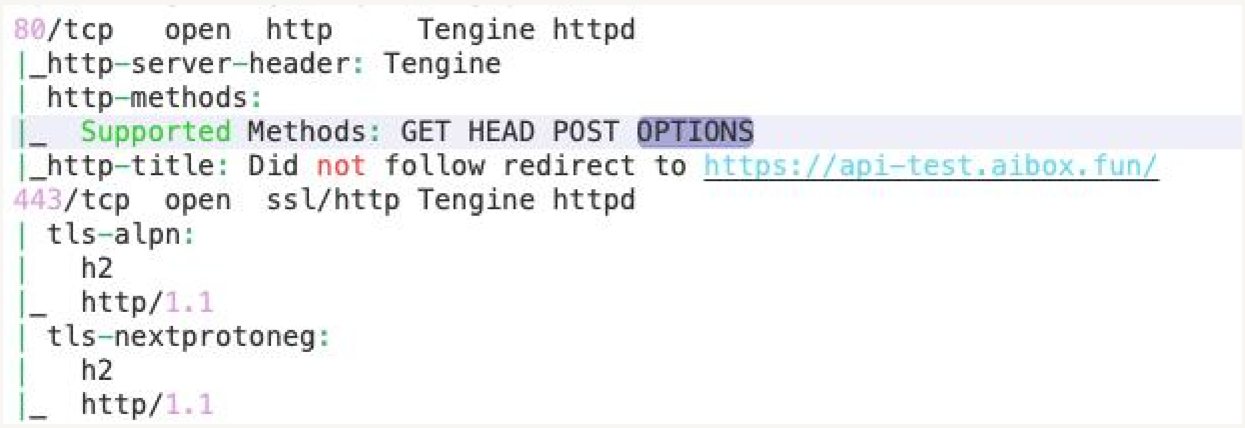
Result: Confirmed

3.3 Unsafe HTTP Methods Enabled on Server

ID:	NVE-003	**Location:**	Web servers
Severity:	Low	**Category:**	Business Issues
Likelihood:	Medium	**Impact:**	Low

Description:

Web servers or API interfaces of Web3 transaction applications improperly restrict OPTIONS requests, allowing attackers to enumerate supported HTTP methods or exploit permissive CORS configurations. This could expose sensitive endpoints or enable follow-up attacks like CSRF or unauthorized access, potentially leaking blockchain interaction endpoints in Web3 environments.



```
80/tcp   open  http     Tengine httpd
|_http-server-header: Tengine
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Did not follow redirect to https://api-test.aibox.fun/
443/tcp  open  ssl/http Tengine httpd
| tls-alpn:
|   h2
|_  http/1.1
| tls-nextprotoneg:
|   h2
|_  http/1.1
```

Recommendations:

Restrict allowed HTTP methods to essentials.

Result: Confirmed

4. CONCLUSION

In this penetration, we thoroughly analyzed the implementation of **Aibox** application. The problems found are described and explained in detail in Section 3. The problems found in the penetration have been communicated to the project leader. We therefore consider the penetration result to be **PASSED**. To improve this report, we greatly appreciate any constructive feedbacks or suggestions, on our methodology, penetration test findings, or potential gaps in scope/coverage.

5. APPENDIX

5.1 Vulnerability explanation and repair plan

5.1.1 SQL injection vulnerability

Vulnerability descripFon:

SQL Injection (SQL Injection) is a common Websecure vulnerability that occurs between

applications and databases. Due to lack of checks in improperly designed programs, the user's input is not effectively filtered. As a result, the attacker can successfully submits malicious SQL query code to the server, which the program incorrectly receives as part of the query statement and executes, causing the original query logic to be changed, additionally executing the malicious code carefully constructed by the attacker, thus bypassing the authentication and permission check to achieve the purpose of obtaining hidden data, or overriding key values, or even performing database host operating system commands.

Repair plan:

1. Strictly limit the operation permission of the database of the web application, and provide this user with only the minimum permission to satisfy its work, thereby minimizing the damage of the injection attack on the database.

2. Check if the input data has the required data format and strictly limit the type of the variable.

3. Escape the special characters ('"\<>&*; etc.) entering the database, or transcode them.

4. Do not directly assemble SQL statements. For all query statements, it’s recommended to use the parameterized query interface provided by the database. Use parameterized statements instead of embedding user input variable into SQL statements.

5. Before the application is released, it is recommended to use professional SQL injection detection tools to detect and timely repair the SQL injection vulnerability.

6. Avoid websites print out SQL error information, such as type error, field mismatch, etc.

5.1.2 XSS Cross-Site Scripting Vulnerabilities

Vulnerability descripFon:

Cross-site scripting (XSS), a common Web secure vulnerability. Because the Web application does not adequately filter user submitted data, which allows users to incorporate HTML code in submitted data (mainly ">" and "<"), and output the un-escaped malicious code to the browser of the third-party user to interpret and execute, resulting in

XSS vulnerability. After the attack is completed, the attacker may be given higher rights (such as performing some operations), private web content, sessions, and cookies.

Repair plan:

1. Defense against request parameters and user-controllable data and encode the output.

2. Validate each input on the server, allowing only common symbols, letters, and numbers to be entered.

3. Add the HttpOnly flag to the cookie.

5.1.3 CSRF cross-site request forgery

Vulnerability descripFon:

Cross-site request forgery (CSRF) is a common Web secure vulnerability. Because all parameters of important operations in a Web request can be guessed, through some technical means, an attacker can deceive the user's browser to visit a website that the user once authenticated, so that an attacker can use the identity of the user to perform malicious operations.

Repair proposal:

1. Add a random number or letter verification code to the form to effectively suppress CSRF attacks by forcing users to interact with the application.

2. If the inspection is found to be a non-normal page submission request (as judged by Referer), it is most likely a CSRF attack.

3. Add a random token parameter to the requested parameter, and it cannot be guessed.

4. Sensitive operations should use POST, rather than GET, and submit in the form of form to avoid token leaks.

5.1.4 Broken Access Control Vulnerabilities

Vulnerability descripFon:

Breach Access (Broken Access Control, BAC for short) is a common Web secure vulnerability. This kind of vulnerability refers to flaws in the application when checking the authorization, which allows an attacker to use some methods to bypass the permission check; therefore he can access or operate the interface that originally he was not authorized to access. In the actual code secure review, such vulnerabilities are often difficult to automate detection through tools, so it is quite harmful in actual application.

Repair plan:1. The user identity is stored in the session and authenticated. It cannot be placed in the form of parameters in the HTTP request, it should be placed in the session, and only through the session can the user's identity be verified.

2. It is forbidden to determine the user group to which the user belongs from the Cookie parameter. It is necessary to determine the user group to which the user belongs by reading the sessions.

3. When downloading a file, it is forbidden to use a continuous ID that can be guessed as a parameter to download the file. When downloading the file, whether the current user has the authority to download the target file should also be determined.

4. Strictly manage the authority for the non-ordinary user operation page. Verify the current user authority for the addition, deletion, modification and search operations.

5.1.5 Unrestricted File Upload Vulnerability

Vulnerability descripFon:

Unrestricted File Upload Vulnerability is a common Web secure vulnerability. Due to the lack of necessary checks on uploaded files when the Web application executes the file upload function, the attacker can upload unrestricted files. Using this vulnerability, an attacker can directly upload various dangerous files such as Webshell (WebShell is a command execution environment in the form of web files such as asp, php, jsp, or cgi, and can also be referred to as a web page backdoor), viruses, and malicious scripts, which may lead to server permissions being directly obtained, thereby jeopardizing the secure operation of the entire system.

Repair plan:

1. Use the white list to limit the file suffixes uploaded by the user and rename the file.

2. Limit the file upload directory, and the directory does not allow parsing of dynamic script files.

3. Update the web server version and configure it correctly to prevent the parsing of vulnerabilities.

5.1.6 Unrestricted File Download Vulnerability

Vulnerability descripFon:

Unrestricted File Upload is a common web security vulnerability. Due to security defects in the functions of viewing downloaded files, downloading attachments, as well as other functions provided by Web applications, it is possible to view and download unrestricted files by modifying the file path. Files include: source code files, system files (/etc/passwd, C:/boot.ini, etc.), configuration files (config.php, /WEB-INF/web.xml, web.config, etc.), causing site-sensitive data exposure, which poses a serious threat to website security.

Repair plan:

1. The server filters special characters such as: ....// ....\/ ..../\ ....\\.

2. Determine if the format of the parameters entered by the user is legal.

3. Specify a white list of file types (eg, jpg, gif, png, rar, zip, pdf, doc, xls, ppt, etc.) to prevent users from reading or downloading files other than the white list.

4. Specify the download path to prevent users from reading or downloading files other than those listed in the specified directory

5.1.7 Unrestricted Code Execution Vulnerability

Vulnerability descripFon:

Unrestricted Code Execution vulnerability (Unrestricted Code Execution) is a common Websecure vulnerability. Because web applications fail to filter execution functions, thus in case a web application is used to call some functions that can convert strings into commands (such as eval (), system (), exec () in php.), some security restrictions are not taken into consideration, which results in the construction of special code; when executes operating system commands, resulting in an attacker to obtain the website server permissions.

Repair plan:

1. If remote code execution vulnerabilities (such as remote code execution of struts2, deserialization code execution of jboss, and weblogic) are caused by the use of frameworks or middleware, upgrade the framework and middleware.

2. Filter the special function entry executable in the code, try to strictly check all submitted statements that may execute the command or control the external input. The system command executes the function and external parameters aren't allowed to pass.

3. All the filtering steps are performed on the server side, and verify not only the type of data, but also the format, length, range, and content.

5.1.8 Registration module design flaws

Vulnerability descripFon:

Register module design flaws (Registration module design flaws) is a common Web secure vulnerability. The design flaws of the website registration module will result in the following secure loopholes:1. Unrestricted user password recovery.

2. Registered users in brutal enumerate websites.

3. Crack user password with brute force.

4. Universal password login.

5. SQL injection.

The above security issues will bring risks such as theft of user passwords, disclosure of personal information, leakage of website databases, and intrusion of websites.

Repair plan:

1. If you retrieve the password using email authentication, reset the password token to be unpredictable, and use the encrypted encryption token instead of constructing it yourself; set the reset password session expiration time. Do not obtain the user name that needs to be reset the password from the request during the process.

2. If you use the SMS verification method to retrieve the password, the verification message should be at least 6 digits, and the expiration time of the message cannot exceed 10 minutes. Add the confused Figure-shaped verification code to the send SMS page, and set the frequency of sending text messages within the unit time in the backend.

3. Limit the number of authentication errors per unit of time.

4. In the user registration page and login interface, add a reliable robot identification function, such as a Figure-type verification code or a short message verification code, and the next operation can be performed only after verification.

5.1.9 SMS interface design flaws

Vulnerability descripFon:

SMS interface design flaws (SMS interface design flaws) are common Web secure vulnerabilities. SMS interfaces are generally used for registration verification, login verification, and verification of other sensitive operations. However, due to improper design, there are usually the following security issues:

1. Send a large number of SMSs within a short time.

2, SMS verification code is too short and can be guessed.

3. When the SMS verification code is sent multiple times, multiple verification codes are valid at the same time.

4. The SMS verification code is returned to the client in the corresponding HTTP package.

Repair plan:

1. In the Send SMS interface, set up a robot recognition mechanism, such as an obfuscated Figure-type verification code, and send a short message after verification.

2. The verification code message used for verification should be at least 6 digits. Only one verification code can be valid within the expiration time, and the valid time should not exceed 10 minutes.

3. Do not return the SMS verification code to the client.

5.1.10 URL redirection vulnerability

Vulnerability descripFon:

URL redirection vulnerability (URL redirection vulnerability) is a common Websecure vulnerability. Because the URL redirection function of the website is not designed properly and there is no verification of whether the target URL of the jump is valid, users can jump to any website through this vulnerability. This can lead to the jump to the site where Trojans and viruses exist or the phishing site, which can harm the rights of users and the reputation of the site.

Repair plan:

1. The jump target URL should not be obtained from the content requested or filled by the user. The jump URL should be set at the back end.

2. Verify the target URL that needs to be redirected. If the redirected URL is not allowed, jumping is prohibited.

3. Prompt user and display jump target URL address when making URL and ask if jump.

5.1.11 Denial-of-service attack Vulnerability

Vulnerability descripFon:

Denial-of-service attack (abbreviation: DoS attack, DoS), also known as flood attack, is a kind of network attack technique. Its purpose is to exhaust the network or system resources of the target computer and temporarily interrupt or stop the service. , causing its normal users to be inaccessible. Denial-of-service attacks may also result in attacks on other computers on the same network as the target computer. The bandwidth between the Internet and LAN will be attacked and lead to a lot of consumption, not only affect the target computer, but also affect other computers in the LAN. If the scale of the attack is large, the entire region's network connection may be affected.

Repair plan: The defense methods of denial-of-service attacks are usually intrusion detection, flow filtering and multi-authentication, which is designed to block the network bandwidth will be filtered and normal flow can pass.

1. Network equipment. Firewalls can set rules, such as allowing or denying specific communication protocols, ports, or IP addresses. When attacks originate from a few unusual IP addresses, you can simply use a deny rule to block all traffic originating from the attacking source IP.

2. Black hole guidance/flow cleaning. Black hole guidance refers to sending all the communications of all attacked computers to a "black hole" (empty interface or non-existent computer address) or a network operator with sufficient ability to handle the flood, so as to avoid the network from being greatly affected. When traffic is sent to the

DDoS protection cleaning center, normal traffic and malicious traffic are separated by using anti-DDoS software processing. In this way, the normal operation of the site can be ensured, and the legitimate flow brought by the real user visiting the website can be handled.

3. Web server. Upgrade the Web server to avoid denial of service vulnerabilities, such as the HTTP.sys (MS15-034) denial of service vulnerability.

5.1.12 Session fixation attack

Vulnerability descripFon:

Session fixation attack is a common Websecure vulnerability that uses the server's session invariant mechanism to gain authentication from other people and impersonates others for malicious operations.

Repair plan:

To guard against such attacks, first of all, it should be considered that session hijacking only has practical use when the user logs in or has higher authority. Therefore, when the authority changes, the session identifier is regenerated (for example, the user name or password is modified), which effectively removes the risk of a fixed session attack.

5.1.13 Dangerous HTTP request method

Vulnerability descripFon:

The dangerous HTTP method (Dangerous HTTP request method) is a common security vulnerability caused by configuration errors. In addition to the standard GET and POST methods, HTTP requests use various other methods. Many of these methods are mainly used to accomplish uncommon and special tasks. If low-privileged users have access to these methods, they can use it to execute effective attacks on the application. Here are some noteworthy ways:

1. PUT, put additional files to the specified directory.

2. DELETE, delete the specified resource.

3. COPY, copy the specified resource to the location specified by the Destination header.

4. MOVE, move the specified resource to the location specified by the Destination header.

5. SEARCH, search for resources in a directory path.

6. PROPFIND, get information about the specified resource, such as author, size, and content type.

7. TRACE, trace the original request received by the server in the response.extensions of the HTTP protocol, through which Web server content can be centrally edited and managed.

Repair plan:

In addition to the GET and POST methods, the dangerous HTTP methods listed above should be disabled.

5.1.14 Sensitive Data Exposure

Vulnerability descripFon:

Sensitive Data Exposure (Sensitive Data Exposure) is a common Web secure vulnerability. Due to the negligence of the site's operation and maintenance personnel, sensitive documents are leaked or the sensitive information is leaked due to errors in website operations. The vulnerability of sensitive data exposure can bring about the following hazards: The attacker can directly download the relevant information of the user, including the absolute path of the website, the user's login name, password, real name, ID number, phone number, email, QQ number, and the like. The attacker triggers the system web application to report an error by constructing a special URL address, and obtains the website sensitive information in the echo content. The attacker uses leaked sensitive information to obtain sensitive information such as Web server path of the website to facilitate further attack.

Repair plan:

1. Make a unified return to the website error message and obfuscate it.

2. Encrypt sensitive files and store them properly to avoid leakage of sensitive information.

5.1.15 HTTP transmission without encryption

Vulnerability descripFon:

HTTP transmission without encryption is a common Web secure problem. As the HTTP protocol passes data in plain text, the data is directly transmitted without any processing. Therefore, when the “Man-in-the-middle” performs a sniffing attack, all the transmitted content was completely acquired by the attacker, resulting in problems such as the theft of the user's password as well as sensitive information, and other security issues.

Repair plan:

Use secure HTTPS protocol and encrypt sensitive (e.g. password, personal information) data for transmission. Use POST to transfer encrypted and sensitive data.

5.1.16 Business logic vulnerabilities

Vulnerability descripFon:

Business logic vulnerabili'es are common Websecure vulnerabili'es. Due to the fact that at the beginning of the system design, the security problem was not fully considered. During the normal opera'on of the system, aLackers use the service features to perform illegal opera'ons, resul'ng in users' personal informa'on leakage, property damage and other serious security issues. Savvy aLackers pay special aLen'on to the logical approach used by the target applica'on and try to understand the assump'ons made by designers and developers to consider how to break through these assump'ons.

Repair plan:

1. Perform signature verifica'on on the data packet to prevent the data packet from being tampered with during transmission.

2. The user's opera'ons are all verified on the server, instead of simply checking through js.

3. Important func'ons to increase confirma'on opera'ons or re-authen'ca'on, such as payment transac'ons, modify mobile phone numbers, etc.

4. Use strong random tokens for protection in each session.5. Use a strong algorithm to generate the session ID, which must be random and unpredictable and must be at least 128 bits long. Set session expiration time. Set two properties of Cookie: secure and HttpOnly to prevent sniffing and blocking JS operations.

6. Verify all the parameters from the client, the emphasis of which is on permissions related parameters, such as user ID or role permission ID.

7. The session ID is bound to the authen'ca'on token and placed in the server's session. It is not sent to the client.

8. For requests involving user unique informa'on aher login, check the ownership every me, and add a random number parameter to the sensi've informa'on page to prevent the browser from caching the content.

9. The user should correctly follow the business process to complete the detec'on mechanism of each step, so as to prevent hackers from skipping, bypassing, and repea'ng process checks in any business process. When developing this part of the business logic, some useless or misuse test cases should be tested. When fails to complete the correct steps in the correct order, the business process cannot be successfully completed.

5.2 Introduction to the vulnerability Test Tool

5.2.1 BurpSuite Agent Tool

Burp Suite is an integrated plaiorm for aLacking web applica'ons. It contains many tools and many interfaces are designed for these tools to facilitate the process of aLacking the applica'on. All tools share a powerful extensible framework that can handle and display HTTP messages, persistence, authen'ca'on, proxies, logs, and alerts.

5.2.2 Sqlmap Injection Tool

Sqlmap is a free injec'on tool abroad, based on Python development, supports almost all databases now, supports get, post, cookie injec'on, can add cookies and user-agent, supports blind, error injec'on, DNS injec'on and so on as well as many other injec'on methods; support agents, op'mize algorithms, and be more efficient; judge the database by fingerprint recogni'on technology.

5.2.3 BeEf cross-site vulnerability utilization tool

BeEf is currently the most popular Web framework aLack plaiorm in Europe and America. Its full name is the Browser exploita'on framework project. In the past two years, various foreign hackers' conferences have introduced it. Many pentesters speak highly of this tool. Through the simple Vulnerability of XSS, BeEF can control the target host's browser through a piece of compiled java script, get all kinds of informa'on through the browser and scan the internal network informa'on, and can further penetrate the host with metasploit.

5.2.4 Hydra password cracking tool

An open source brute cracking tool of Hydra's famous hacker group thc, which can be used to crack mul'ple passwords. Its official website hLp://www.thc.org, the test version of this ar'cle is the latest version 5.8, which can support: TELNET, FTP, HTTP, HTTPS, HTTP PROXY, SMB, SMBNT, MS-SQL, MYSQL, REXEC, RSH, LOGIN , CVS, SNMP, SMTP-AUTH, SOCKS5, VNC, POP3, IMAP, NNTP, PCNFS, ICQ, SAP/R3, LDAP2, LDAP3, Postgres, Teamspeak, Cisco Auth, Cisco enable, AFP, LDAP2, Cisco AAA and other password crackers .

5.2.5 Firefox hackbar plugin

Simple security audi'ng and penetra'on tes'ng widgets, including some commonly used tools (SQL injec'on, XSS, encoding, encryp'on, decryp'on, etc.).

6. DISCLAIMER

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without ExVul's prior written consent.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts ExVul to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. ExVul's position is that each company and individual are responsible for their own due diligence and continuous security. ExVul's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

7. REFERENCES

[1] MITRE. CWE- 191: Integer Underflow (Wrap or Wraparound).

https://cwe.mitre.org/data/ definitions/191.html.

[2] MITRE. CWE- 197: Numeric Truncation Error.

https://cwe.mitre.org/data/definitions/197. html.

[3] MITRE. CWE-400: Uncontrolled Resource Consumption.

https://cwe.mitre.org/data/ definitions/400.html.

[4] MITRE. CWE-440: Expected Behavior Violation.

https://cwe.mitre.org/data/definitions/440. html.

[5] MITRE. CWE-684: Protection Mechanism Failure.

https://cwe.mitre.org/data/definitions/ 693.html.

[6] MITRE. CWE CATEGORY: 7PK - Security Features.

https://cwe.mitre.org/data/definitions/ 254.html.

[7] MITRE. CWE CATEGORY: Behavioral Problems.

https://cwe.mitre.org/data/definitions/438. html.

[8] MITRE. CWE CATEGORY: Numeric Errors.

https://cwe.mitre.org/data/definitions/189.html.

[9] MITRE. CWE CATEGORY: Resource Management Errors.

https://cwe.mitre.org/data/ definitions/399.html.

[10] OWASP. Risk Rating Methodology.

https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology