
Armors Labs
Fairdesk
2022/5/16

Document information

Project Name:	Fairdesk Exchange
Start and end time:	2022/04/26-2022/05/16

Copyright description

Catalogue

I.Test process

The penetration was carried out in accordance with the authorization of the Fairdesk exchange, and the penetration test was carried out on the related businesses of the Fairdesk exchange from a remote location. How to accurately and effectively discover the real risks and security vulnerabilities existing in the system, and how to avoid and protect them effectively is the main goal of our penetration test.

Since this system is the actual operating system of the Fairdesk exchange, we try to avoid using denial of service attacks during the testing process, so as to avoid affecting the related business operations of the Fairdesk exchange during the testing process.

Information collection and analysis in the early stage is very important for the later penetration testing. Vulnerability discovery is mainly carried out through manual analysis and tool scanning. Security analysis will be evaluated from three aspects: technical dimension, business dimension, and management dimension. The vulnerability disposal process will complete the overall process disposal process from five aspects: vulnerability discovery, vulnerability verification, vulnerability rating, vulnerability retest, and vulnerability closure. The development of remote penetration testing is more conducive to the development of the project, and external network penetration testing can better illustrate the seriousness of security than internal network penetration testing.

a Client entrustment

Client entrustment is a necessary condition for the implementing company to conduct penetration testing. The implementing company will do its best to ensure that Party A is aware of all details and risks of the penetration test, and all processes are carried out under the control of Party A, which is also the implementing company. The professional services of a hacker are different in nature.

The penetration testing power of attorney (authorization letter) should contain the following:

(a). Scope of penetration test (including IP address or domain name);

(b). Penetration test time (including start time and end time);

(c). The power of attorney (authorization) for penetration test shall be submitted in writing and stamped with the official seal of Party A.

b Information collecting

Information collection and analysis is the premise / prelude / basic of almost all intrusion attacks. Through information collection and analysis, attackers (testers) can make corresponding and targeted plans for intrusion attacks, improve the success rate of intrusion and reduce the probability of exposure or discovery.

The methods of information collection include application fingerprint information detection, system service information detection, search engine collection, third-party service information collection, etc. The basis of the analysis after information collection is the key knowledge base of security weaknesses.

c Vulnerability analysis

Vulnerability analysis is a necessary work for vulnerability discovery. Through a comprehensive assessment of the security weaknesses found in the actual business test of the enterprise, and through evaluation and analysis from the aspects of vulnerability utilization conditions, technical implementation difficulty, data sensitivity, business importance, etc., we can ensure the preciseness of the vulnerability discovery process and analysis.

d Risk rating

Risk rating is based on vulnerability analysis. The vulnerability risk rating will be assessed from three dimensions: technical risk, business impact risk, and management risk, which will serve as data support for the implementation of security management work to ensure that relevant risks can be properly handled.

e Report

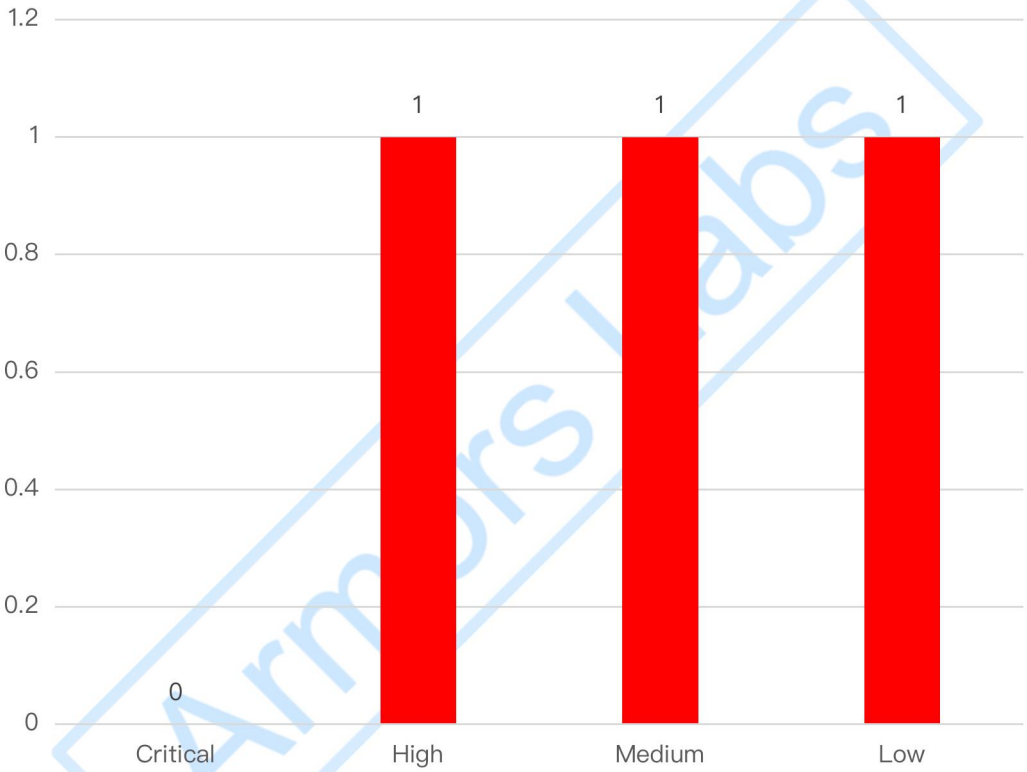
In order to enable customers to better understand and supervise the whole process of penetration test, the penetration test will be recorded in various ways such as written text and key screenshots. The key screenshots can reflect the whole process of penetration test more intuitively and truly than written text.

After the penetration test is completed, the safety experts of Armors (Beijing) Technology Co., Ltd. will write the penetration test report according to the process documents of the penetration test to describe the process and results of the penetration test in detail, and propose solutions to the problems found.

II.Test result

a Vulnerability statistics

Vulnerabilities Number	Critical risk	High risk	Medium risk	Low risk
3	0	1	1	1



b Vulnerability distribution statistics table

ID	Type	Severity	Status
ZA-2022-0426-T9A18p	Parallel ultra vires	Low	Closed
ZA-2022-0501-QihoVv	Remote command / Code execute	High	Closed
ZA-2022-0429-mOq6rH	Remote command / Code execute	Medium	Closed

III.Test summary

This time, we have conducted a comprehensive risk investigation on the Fairdesk exchange and related application businesses, and have further understood the security of Fairdesk exchange. In response to the relevant requirements and confidentiality requirements of the penetration test authorization letter, we tested this time to find problems and verify. The severity of the impact of security risks on the business is mainly, and the security risks are most likely to be discovered within the testing time range. For relevant target platforms we make a table as belows;

Test item	Vulnerability type	Function item detection description
User registration	Any user registration	Attempt to bypass validation for detection
User login	brute force	Brute force cracking by replaying packets
	Error prompt	By constantly trying to input, see the returned error information and judge
	Data not encrypted	Check if important parameter transmission is encrypted transmission by grabbing
	SQL injection	Try universal password login for SQL injection detection
Password Reset	SMS / email verification code	Try brute force cracking of verification code
		The verification code can be obtained by grabbing
		The verification process logic problem is tested by trying to bypass the verification code
		You can try to guess the link parameters, such as simply using the MD5 value of the user name
Verification Code	Timeliness	The validity period of the verification code is too long. You can try brute force cracking
	Randomness	The verification code is not random and inexhaustible. Try to guess the verification code
	Complexity	The verification code is not complex enough, the length is not enough, or the content of the graphic verification code is not mixed, which can be identified by the script program

	Security	Key parameters such as user ID are not bound when sending. Try to send beyond authority or send without restriction
Business operation	Ultra vires	The server failed to restrict or verify the account authority of the requested data, resulting in the attacker being able to manipulate the account of others
	Unauthorized access	The business function permission setting is unreasonable. Try to bypass the login and access directly
Data transmission	Parameter filtering	Through semi manual and semi automated testing of SQL injection, XSS, command execution and other vulnerabilities
	Parameter encryption	Through the form of residential data packet, detect whether plaintext is used to transmit data in HTTP protocol
Session security	Submission method	Detect whether to submit sensitive data in get mode by capturing data packets
	Parameter randomization	The parameters are controllable, can be guessed and replayed
	Disable external entities	By modifying the form of data packet, detect whether there is external entity injection vulnerability
	Authentication information	Try to log in to the system for detection through unexpired session / token and other information
	Random factor	Exploit CSRF in session, random detection without parameters
Program logic	Improper identity authentication	Use SSRF vulnerability to detect Intranet
	Improper function call	By manually trying to access system files and other means, detect whether any file reading, command execution and file inclusion exist
	Data binding	Detect vertical ultra vires and horizontal ultra vires vulnerabilities for verification
API	Customer request interface	Jsonp, request interface, return sensitive information, request controllable VPS to obtain data
Other	Middleware security	Automatic tools are used to detect the parsing vulnerabilities and command execution vulnerabilities of various middleware
	Weak password	Detect by trying common passwords
	Disclosure of sensitive	Try to find git, SVN and other sensitive

	information	information

Through this penetration test of the Fairdesk exchange and related business systems, I have a better understanding of the security status of the related Fairdesk exchange. From the analysis of the vulnerabilities found in the penetration test, the platform mainly has the following problems;

a Disclosure of sensitive information

Sensitive information is leaked in the page or in the returned response package, which provides a lot of useful information for attackers to penetrate.

b Design errors / Logic flaws

Because the code is not fully considered or there are problems in the verification method when the code is written and actual, the attacker can modify the return packet, bypass the front-end verification code, or the back-end verification is incomplete, so that the attacker can call out of sequence by calling The interface bypasses the normal flow.

Since the code layer and database are not locked, an attacker can send a large number of requests in a short time to cause a conditional competition vulnerability, which can break through the maximum limit set by the program.

c SQL injection

In the web program, the parameters submitted by the user are not filtered and directly spliced into the SQL statement for execution. As a result, the special characters in the parameters destroy the original logic of the SQL statement. Attackers can use this vulnerability to execute arbitrary SQL statements, such as querying data and downloading data. , write webshells, execute system commands, bypass login restrictions, etc.

d XSS attack

The parameters submitted by the user are not filtered or are not strictly filtered in the Web program code, resulting in the special characters in the parameters destroying the original logic of the HTML page. Attackers can use this vulnerability to execute malicious HTML/JS code, construct worms, and tamper with pages. Implement phishing attacks, induce users to log in again, and then obtain their login credentials.

Although XSS attack does no direct harm to the web server itself, it spreads through the website, attacks website users, steals website user account identity information, etc., which will also pose a serious threat to the website.

Storage XSS attacks can lead to the following compromises:

(a). Identity theft: a cookie is the user's authentication mark for a specific website. XSS attack can steal the user's cookie, so as to use the cookie to steal the user's operation authority on the website.

(b). Stealing website user information: when the user's cookie is stolen to obtain the user's identity, the attacker can steal the user's operation authority on the website and view the user's privacy information.

(c). Spam sending: in the social networking site community, XSS vulnerabilities are used to borrow the identity of the attacker to send a large amount of spam to a specific target group.

(d). Hijacking users' web behavior: some advanced XSS attacks can even hijack users' web behavior, so as to monitor users' browsing history, sent and received data, etc.

e Over Permission in paralle

Since there is no strict inspection and restriction on the permissions of the user's access role, the current account can perform related operations on other accounts, such as viewing and modifying. The operation of a low-privileged account with a high-privilege account is vertical override, and the operation between accounts with the same privilege becomes horizontal override, also called horizontal override.

f Unauthorized access

Due to the failure to check the login status and access rights of the sensitive pages and interfaces of the website, the attacker can access without authorization, obtain sensitive information and perform unauthorized operations.

IV.Safety advice

Based on the process of penetration testing and the problems found, we propose the following security improvement suggestions for the current security construction of the Fairdesk exchange and related business systems;

a Disclosure of sensitive information

(a). If it is a useless program such as probe or test page, it is recommended to delete it or change it to a name that is difficult to guess.

(b). Delete or prohibit access to pages that disclose sensitive information without affecting business or functions.

(c). Fuzzify the relevant sensitive information on the server side.

(d). Strictly check the data returned by the server to ensure that the query data is consistent with the data displayed on the page.

b Design errors / Logic flaws

Modify the code logic and perform strong verification or multiple verification on the server. For conditional competition vulnerabilities, the best fix is to set locks in the code layer and database.

c SQL injection

The best defense against SQL vulnerabilities in the code layer: query and bind variables with precompiled SQL statements.

(a) When using precompiled statements and PDO, be careful not to splice variables directly into PDO statements. All query statements use the parameterized query interface provided by the database. Parameterized statements use parameters instead of embedding user input variables into SQL statements. At present, almost all database systems provide parameterized SQL statement execution interface, which can effectively prevent SQL injection attacks.

(b) Escape the special characters ('< > & *; etc.) entering the database or code conversion.

(c) Confirm the type of each data. For example, digital data must be digital, and the storage field in the database must correspond to int.

(d) The data length should be strictly regulated to prevent long SQL injection statements from being executed correctly to a certain extent.

(e) The coding of each data layer of the website is unified. It is recommended to use UTF-8 coding. Inconsistent coding between the upper and lower layers may lead to some filtering models being bypassed.

(f) Strictly restrict the operation authority of the website user's database, and provide this user with the authority that can only meet his work, so as to minimize the harm of injection attack to the database.

(g) Prevent the website from displaying SQL error information, such as type error, field mismatch, etc., and prevent attackers from using these error information to make some judgments.

(h) Filter dangerous characters. For example, regular expressions are used to match Union, sleep, and, select, and load_ File and other keywords. If they match, the operation will be terminated.

d XSS attack

XSS vulnerability is essentially a kind of HTML injection, that is, injecting HTML code into web pages. Then the fundamental defense is to do a series of filtering and escape when displaying the code submitted by the user on the page.

(a) Filter the input data and strictly check the dangerous characters such as: """, "", "<", ">", "on *", script, iframe, etc. the input here is not only the input interface that users can interact directly, but also the variables in the cookie in the HTTP request, the variables in the HTTP request header, etc.

(b) Verify not only the type of data, but also its format, length, scope and content.

(c) Not only do data verification and filtering at the client, but the key filtering steps are carried out at the server.

(d) Carry out corresponding coding conversion for the data output to the page, such as HTML entity coding, JS coding, etc. The output data should also be checked. The values in the database may be output in multiple places of a large website. Even if the input is coded, it should also be checked at the output points everywhere.

e Parallel ultra vires

(a). Strictly check and restrict the access rights of users to roles.

(b). In some operations, you can use session to judge and control the user's identity

f Unauthorized access

(a). Strictly control the access rights of the page and check the access rights of the access roles.

(b). You can use session to judge and control the user's identity.

V.Project summary and suggestions

a Project summary

In view of this comprehensive security analysis of the business system provided by Fairdesk exchange, the security of Fairdesk exchange has been basically understood. This project evaluates from the aspects of vulnerability mining, risk analysis, vulnerability verification, etc., and found that the security of

Fairdesk exchange The level of system construction needs to be improved. There are many SQL injection vulnerabilities in the security loopholes found in this analysis, which proves that the development structure is weak. Other problems, such as parallel overreach, design errors/logic flaws, etc., confirm from the side that the website permissions are not perfect and the logic is not rigorous enough. In the future, it is recommended to strengthen the security awareness of developers and improve the overall software development architecture system.

b Suggestions for future safety planning

Based on some of the problems found during our penetration test, we can focus on the following aspects of the Fairdesk exchange;

a) Gradually improve the safety development norms and system construction

Strictly abide by the safety development specifications, and prevent the launch of the disease. For the design of new products and new functions, it is necessary to do a good job in product security design and implementation of security mechanisms, and gradually improve new security risks in combination with existing security development specifications (security coding, permission control, special character filtering, file upload whitelist, etc.) Scenarios, and detailed security inspections are carried out before the products are launched. For different vulnerability types, different function points, and different business scenarios, a multi-dimensional and comprehensive security inspection is carried out to ensure the stable operation of the platform after the product is launched. Timely update and improve the existing security development specifications and system construction for the security risks of new business scenarios.

b) Regularly conduct safety awareness education for all employees and key positions, and strengthen third-party safety responsibility management

Strengthen the network security awareness of all employees, regularly organize network security awareness education and training, and conduct targeted security awareness training for business operation teams, key technical positions, functional personnel, and management teams to improve security awareness of all employees. At the same time, it puts forward security responsibility requirements for third-party partners of the Fairdesk exchange to reduce the risk of third-party outsourcing.

VI.About Me

Armors is the first group of institutions engaged in blockchain security services in the world, mainly engaged in blockchain code audit and smart contracts development. Since its establishment five years ago, it has served more than 1000 customers and the audit partner of OK, Huobi, binance, bitthumb, bitfinex and other exchanges. Protect blockchain assets worth more than $15 billion. In 2021, Armors officially realized full profitability. Compared with competitors, Armors has zero audit accident rate and is better at smart contracts development. At the same time, Armors team has comprehensive ability. In addition to ETH related ecology, it has the ability of safety audit and ecological development for new active public chains such as Solana, ADA, Polygon and Metis, which has been officially recognized.