Alibaba Cloud Crowdsourced Security Testing

User Guide

Issue: 20190516

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. **You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.**
2. **No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.**
3. **The content of this document may be changed due to product version upgrades , adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.**
4. **This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults " and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity , applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use**

or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified , reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates . The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus , page names, and other UI elements.	Click OK.
`Courier font`	It is used for commands.	Run the `cd /d C:/windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *[-all|-t]*

Style	Description	Example
{} or {a|b}	**It indicates that it is a required value, and only one item can be selected.**	`swich {stand | slave}`

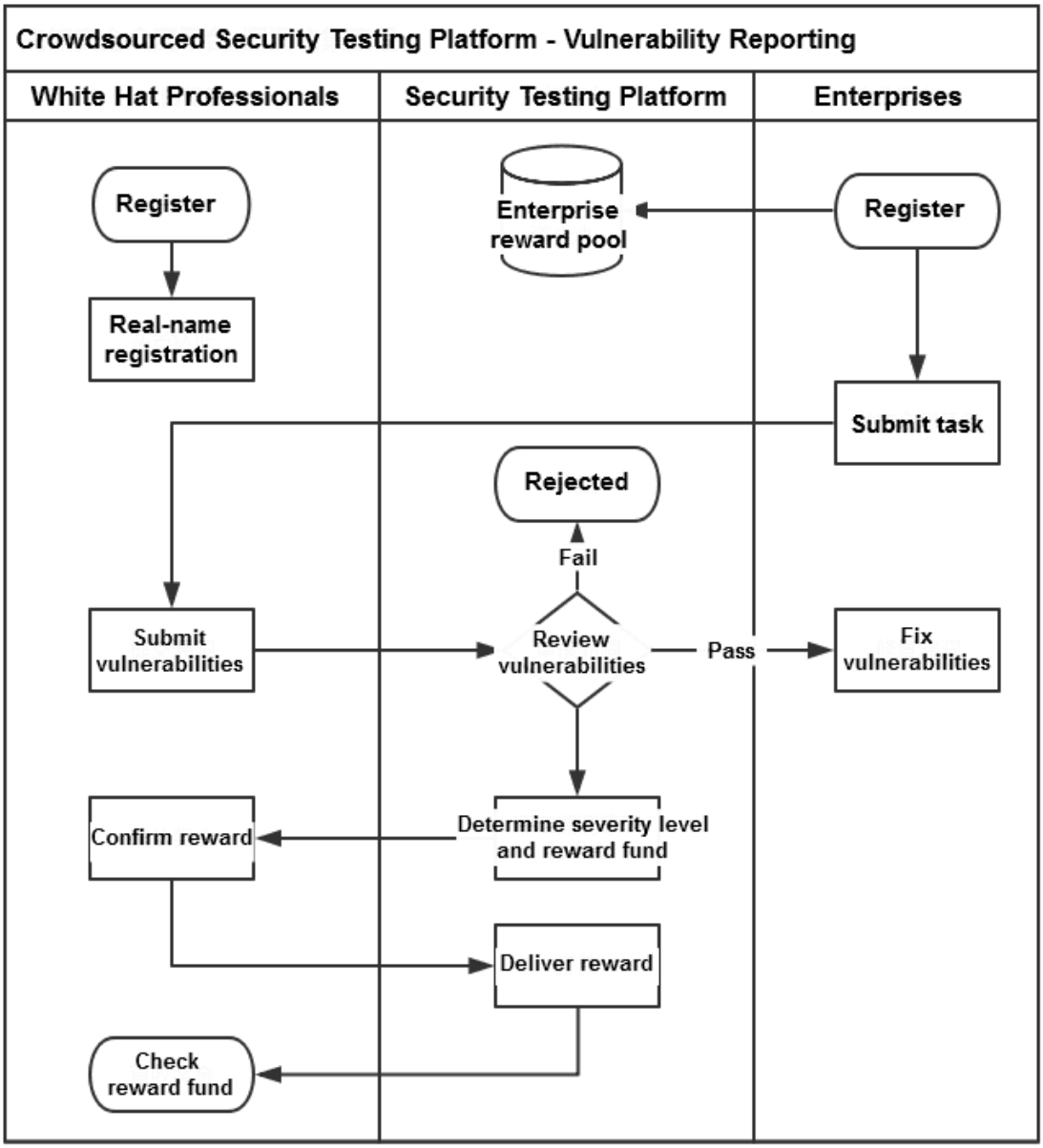
Contents

1 Vulnerability reporting process

This article describes the vulnerability reporting process.

Context

Steps to completing vulnerability reporting



Procedure

1. Log on to your Alibaba Cloud account and complete the personal and enterprise information section.
   - White hat professionals: Use your Taobao account to log on to the Crowdsourced Security Testing platform, complete the information section and real-name registration.

     Note:
     Make sure that all data is up to date.
   - Enterprises: See *Process for Enterprise Registration*.
2. The white hat professionals can follow the instruction on the submission page to submit security vulnerability information.

   Note:
   The vulnerability information should be as specific and detailed as possible. This helps the Crowdsourced Security Testing support team to determine the appropriate rewards amount.
3. Once the vulnerability is submitted, the platform support team conducts an internal evaluation on the vulnerability report within two business days.
   - If the vulnerability does not exist or if it has already been submitted, the support team marks the vulnerability as Rejected and provides a reason for rejection.
   - If the vulnerability description is unclear, the support team marks it as Add More Details and returns the vulnerability. The white hat professionals should add the vulnerability information within 72 hours for re-evaluation. After 72 hours, the platform automatically rejects the vulnerability.
   - If the vulnerability is verified and confirmed, the support team marks it as Verified.
4. Once the vulnerability is confirmed, the support team issues the rewards within three business days, and changes the status to Rewarded.
   - The rewards for general software vulnerabilities is based on the vulnerability rewards standards in Vulnerability Evaluation Standards.
   - The discovery of third-party enterprise vulnerabilities are rewarded based on the custom rewards standards of the company.

5. On the *Crowdsourced Security Testing platform*>My Submit page, the white hat professional can find vulnerability with the Reward Confirmed status, and confirm the vulnerability level and reward fund. Then, the status of the vulnerability is changed to Reward Sending.
   - If the white hat professional admits the reward fund, process the confirmation, and the status of the vulnerability is changed toReward Sending.
   - If the white hat professional has disputes over the reward amount, a ticket can be submitted. The support team will contact the white hat professional, and negotiate for a more appropriate amount.
6. The support team processes the withdrawal request within 24 hours after the white hat has professional confirmed the vulnerability level and reward fund. The status of the vulnerability is changed toRewarded.
7. The reward fund will be issued directly to the specified Alipay account of the white hat professional. The white hat professional can check the reward fund in Alipay.
8. After the enterprise updates the vulnerability status as they fix the vulnerability. The vulnerability is marked as Fixed once it has been fixed..

2 Registration procedure for enterprise

This article describes the registration procedure for enterprise on the Crowdsourced Security Testing platform.

Context

To register your enterprise on the Crowdsourced Security Testing platform, follow these steps:

> Note:
> Your enterprise must have completed the enterprise real-name registration.

Procedure

1. Log on to the *Crowdsourced Security Testing platform application page*.
2. Select the amount of fund to add and complete the prepayment.

   This fund will be directly converted into a reward fund and eventually delivered to the white hat professionals who have contributions.
3. Log on to the *Crowdsourced Security Testing platform console*, complete the enterprise information that includes your enterprise logo and name, and set up a rewards program.

   After completing these steps, your enterprise is successfully registered on the Crowdsourced Security Testing platform. Now, wait for the white hat professionals to submit vulnerabilities for you.

3 Crowdsourced security testing platform procedure for enterprises

This article describes procedure on the Crowdsourced security testing platform for enterprises.

Step 1: Registration

After you have activated the Crowdsourced Security Testing service and added funds to your account, you can log on to the *Crowdsourced Security Testing platform* to complete the enterprise information and set up a rewards program.

Step 2: White hat professionals report vulnerabilities

After the rewards program is set up, the Crowdsourced Security Testing platform updates the status of your company and begins to receive vulnerabilities from the white hat professionals.

Step 3: Review and fix vulnerabilities

After any vulnerabilities are reported, you can log on to the *Crowdsourced Security Testing platform* to view the vulnerabilities and to follow the steps to resolve the vulnerabilities.

4 Regarding vulnerabilities

4.1 Vulnerability types

This article introduces common security vulnerabilities for web servers and clients.

Security vulnerabilities for web servers

SQL injection attacks

An SQL injection attack (also known as injection attack or SQL injection) attempts to obtain control over websites maliciously. It is a security vulnerability that occurs at the database level of applications. The program code does not check for SQL commands contained in users' input strings. The database mistakes the attack commands as regular SQL commands, allowing the database to be attacked. The attack can lead to the theft, tampering, and deletion of data, and it can execute system commands. This further causes the site to be embedded with malicious code, and implanted with backdoor programs.

Common scenarios

- URL parameter submission, mainly GET request parameters.
- Form submission, mainly POST and GET requests.
- Cookie parameter submission.
- Modifiable values in the HTTP request header, such as Referer and User_Agent.
- Peripheral input parameters, such as MP3 files and the file information of image files.

Defense methods

- Use precompiled statements: In general, the best way to defend against SQL injection is to use precompiled statements to bind the variables. This requires changes to the existing code.
- Use stored procedures: The use of secure stored procedures can prevent SQL injection. However, this method does not guarantee full security.
- Check user data: Check the data type and content entered by users. For example, you need to check the data type. When you perform a query by ID, you have to check whether the entered ID is an integer. Check whether the mailbox entered

follows the mailbox format. Time and date values must be entered strictly following the time and date format. For data content, check whether the user submission data contains certain keywords or strings, and check whether it matches certain injection rules. Make sure that you escape all special characters. Even though this method is easy to implement, it is prone to false positives and false negatives, and it is easily bypassed.

- Other methods: Use secure encoding functions, ensure consistency in the data-layer encoding formats (such as always using the UTF-8 format), restrict user access to the database, conduct regular black box and white box scans of code, and avoid displaying backend error messages on the page.

File upload

The program code does not strictly analyze and check the files submitted by the user. As a result, the attacker can upload executable code files to obtain the control (Getshell) of web applications.

Common scenarios

- All scenarios involving the use of the upload function.
- Customizable profile pictures and background images.
- The upload function in rich-text editor.

Defense methods

- Set the upload directory to not be executed.
- Set the upload directory as not executable Check the file type. Use a whitelist instead of a blacklist. Make sure the capitalization is correct. Pay attention to issues related to vulnerabilities found on web servers, such as the file parsing vulnerabilities on Apache, IIS, and Nginx web servers.
- Use random numbers to modify the uploaded file names and file paths.
- Set up a separate file server and domain name.

Access vulnerability

Access control is the granting of access and certain privileges to systems, resources or information for users. This usually includes horizontal access and vertical access . Access control problems are a logical vulnerability that may be generated by all business systems. It is difficult to scan or obtain protection using average security tools. Access control issues often result in the leakage of large amounts of user data.

- Horizontal privilege escalation: Issues related to users having the same access level. For example, user A can gain access to the data of user B without authorization.
- Vertical privilege escalation: Issues related to users having different access levels. For example, users with lower level access can perform management operations without authorization; users who are not logged on can access applications that require authorization.

Common scenarios

- All scenarios involving user-related data, such as user data, addresses, and orders.
- All scenarios involving logon and access control, such as background logon and current user access check.

Defense methods

- For all operations related to user data, check the identity of the current user.
- For all scenarios in which access control is required, check the user access level.

Brute-force

The attacker sends a large number of requests to the target using the traversal or dictionary attack. The attacker identifies the correct authentication information by judging the characteristics of the returned data packet. The verification system is bypassed as a result. In addition, attackers can use data stolen from the databases of many websites to narrow down the samples and have a higher chance of brute-force cracking the passwords.

Common scenarios

- The account and password on the user logon page.
- The CAPTCHA system is easy to bypass, especially when a more easily identifiable verification code is used.
- Text message verification codes such as those for password reset or for two-factor verification purposes.

Defense methods

- Use a strong password, and change it regularly.
- Limit the number of failed logon attempts.
- Use CAPTCHA.
- Limit the access frequency during a certain period of time.

Denial of service attack

A denial-of-service (DoS) attack floods incomming traffic to overload network resources, making them unavailable to their intended users. This can be divided into attacks against web applications and attacks against the clients or mobile applications . The attack.

Common scenarios

- For web applications, DoS attacks are common in scenarios that consume large amounts of resources, such as lookups.
- For clients and mobile applications, DoS attacks are initiated using malicious parameters that result in program crashes.

Defense methods

DoS attacks against web applications

- Limit the request frequency of each client.
- Use CAPTCHA.
- Optimize code performance and network architecture for the application.

DoS attacks against clients and mobile applications

- Delete unnecessary components.
- Filter and check the data entered by users.

Sensitive information leakage

Attackers exploit websites, interfaces, and external storage components to obtain user information, employee information, internal data, and other data. This vulnerability can result in the leakage of massive amounts of user and company information, which is used to attempt fraud and account theft, harming users and businesses. Once the information has been leaked, mitigating the negative impacts is difficult.

Common scenarios

- Websites and client interfaces that obtain user and company information.
- External storage components such as cloud storage and mailbox, which may be accessed by the company.
- All other channels that may leak data.

Defense methods

Check and control the access to data interfaces. Define the security boundary of the company; prohibit the outflow of internal data, for example, by restricting access to external storage. Train employees on data security.

Business logic vulnerability

It is a process or logic vulnerability caused by a flaw in the application. For example, the attacker can reset any user password by exploiting the password reset vulnerability. The attacker can also maliciously increase costs for the company and harass its users by sending unauthorized text messages through the interface. Business logic loopholes are closely related to business operations. Therefore, regular security programs cannot detect these vulnerabilities. Human intervention is required to examine the business scenarios and identify vulnerabilities.

Common scenarios

- All scenarios involving interaction with users.

Defense methods

- Fully examine the business scenarios.

Security configuration flaws

This includes file traversal, source code leakage, and configuration file leakage.

- File traversal: The files under the web directory of the server may be browsed by the attacker, causing important files to be leaked.
- Source code leakage: The source code of the web application may be leaked.
- Configuration file leakage: The configuration files for the web server and for the program code may be leaked.

Defense methods

- Check for all security configuration risks. Optimize the security configuration while satisfying business demands.

Security vulnerabilities for web clients

Cross-site scripting (XSS) attacks

The attacker injects malicious scripts and changes the website using HTML injection. The attacker hijacks the browser when the user browses the webpage affected by cross-site scripting. XSS Attackers exploit this vulnerability to steal identities (

especially targeting administrators), hijack user operations, and initiate Trojan, worm, and phishing attacks. The XSS attack is a key vulnerability for clients.

The XSS attacks include three types based on their effects.

- Reflected XSS: The webpage displays input made by users in the source code of the webpage. To make the attack effective, the attacker has to lure the user into opening on the webpage.
- Stored XSS: The XSS attack code is stored in the server. This type of attack causes more damage because the user may actively browse the compromised webpage.
- DOM-based XSS: The DOM nodes of the webpage are modified.

Common scenarios

- All user-controllable input and output parameters, such as personal information, articles, and comments.

Defense methods

- Use HttpOnly parameters in key cookie fields.
- Check all user-controllable inputs. Check all input parameters. Filter or intercept inputs that do not match the context. Since not all output contexts can be listed, the effect of this method is relatively poor.
- Check all output parameters of input made by users. Because the XSS attack occurs at the output parameters, we need to analyze the environment of all output parameters for input made by users. Check the following parameters: HTML labels , HTML attributes, script labels, events, and CSS. Create different escape or filter rules based on different output parameters.
- Process rich texts. Distinguish rich texts from cross-site scripting in scenarios that use rich texts, such as forum posts. Prohibit all HTML syntax labels and "js events." Use whitelists to filter labels, events, and attributes.

Cross-site request forgery (CSRF)

Cross-site request forgery (CSRF). Forgery). Because all the parameters for key operations can be guessed, the attacker can forge requests with the user's identity. The attacker may publish articles, purchase items, make payments, modify data, and change passwords.

Common locations

- All operations initiated by the users and the administrators.

Defense methods

- Use verification codes. They are the most effective method to combat the CSRF attacks. However, they may also have an impact on user experience. Additionally, not all operations can be protected by verification codes. As a result, verification codes can only be used as a secondary verification method.
- Add random CSRF tokens and update frequently. This can prevent the parameters from being guessed. CSRF tokens are now a common defense method.
- Verify HTTP Referer and refuse insecure sources. However, the server may not always obtain the Referer value.

 Note:
We recommend that you use a combination of the three methods to make your defense more effective.

4.2 Vulnerability rating standards

This article introduces vulnerability rating standards.

General rules for rating standards

1. The rules are only applicable to enterprises that have registered with the Crowdsourced Security Testing platform. They only pertain to products and services for which the company explicitly agree to receive the vulnerabilities . Vulnerabilities not on the acceptance list of the enterprise are rejected. If you make any adjustment to the severity levels, the new levels take effect. For vulnerabilities in your peripheral business, the severity levels are decreased based on the importance of the peripheral business.
2. The contribution of white hat professionals for exploring vulnerabilities is determined by factors such as level of exploitation difficulty and scope of vulnerability impact. If the vulnerability is only triggered in rare cases, including but not limited to cross-site scripting (XSS) vulnerabilities in certain browsers, then the contribution will be lowered accordingly.
3. Multiple vulnerabilities resulting from a single vulnerability source are counted as one vulnerability. Examples include: The same interface can cause multiple security vulnerabilities. The same distribution system can cause vulnerabilities on multiple pages. Inappropriate frameworks can cause site-wide vulnerabilities. Wildcard domain name resolution can cause multiple vulnerabilities.

4. The platform only honors the contribution of the first professional who reports the vulnerabilities in third-party products. The level of vulnerability is no higher than "Medium." Examples include: WordPress and Flash plug-ins, server-side components such as Apache, OpenSSL, and third-party SDKs. Vulnerabilities in different versions of the third-party products are considered one vulnerability.
5. For a single vulnerability, the platform only honors the contribution of the first professional who reports it.
6. Before the vulnerability is resolved, the disclosed vulnerabilities do not count toward contribution.
7. Reporting vulnerabilities that are already disclosed does not count toward contribution.
8. For multiple vulnerabilities in the same report, only the vulnerability with the highest severity level counts toward contribution.
9. The platform does not honor any contribution for the following behaviors: Undermining customer interests by exploiting vulnerabilities, hindering business operations, or stealing user data. Additionally, the Crowdsourced Security Testing platform reserves the right to work with registered enterprises to pursue legal actions against such behaviors.

Resolving disputes

If the white hat professional disputes the vulnerability review process, vulnerability rating or scoring, they can contact the Crowdsourced Security Testing support team . The Crowdsourced Security Testing support team can be contacted through live messaging service, or through commenting on the vulnerability details page. The Crowdsourced Security Testing platform puts the interest of the reporting professional first. It deals with the issue with both the reporting professional and the enterprise . When necessary, external security personnel will be enlisted to help judge the case.

4.3 Vulnerability severity levels

This article introduces vulnerability severity levels.

Rewards program

- We recommend that you set the reward factor to at least 5 to attract more white hats exploring for vulnerabilities.

- We recommend that you set up a tiered reward factor to encourage white hats to focus more on critical vulnerabilities.

For example, you can establish the following reward program:

Security level	Basic score	Reward factor	Reward fund	Security level	Basic score	Reward factor	Reward fund
High level risks	60~100	*5	= 300~500	High level risks	60~100	*20	= 1200~2000
Medium level risks	30~50	*5	= 100~250	Medium level risks	30~50	*10	= 300~500
Low level risks	10~20	*5	= 50~100	Low level risks	10~20	*5	= 50~100

Vulnerability severity levels

The vulnerabilities are categorized into three levels based on severity: high, medium , and low. The Crowdsourced Security Testing platform factors in the severity and difficulty of the vulnerabilities to assign a value and severity level to each vulnerabil ity.

The scoring guide of vulnerability severities and types of vulnerabilities are as follows :

High level risks

Base score: 60-100. High level vulnerabilities include:

- Vulnerabilities that directly obtain system access (server access and PC client access). They include remote command execution, arbitrary code execution, WebShell upload, SQL injection for system access, and buffer overflow (including ActiveX buffer overflow).
- Vulnerabilities that directly result in the denial of service for mobile gateways, APIs, and web applications.
- Vulnerabilities that leak sensitive information, including SQL injection for key business databases and interface vulnerabilities that can obtain a large amount of key business data.
- Severe logical design and process loopholes. They include vulnerabilities that allow mass modification of account password, and logical vulnerabilities that compromise the company's key business.
- Unauthorized access to sensitive information. This includes bypassing authentica tion to access the backend, weak backend password, and server side request forgery (SSRF) that obtains a large amount of intranet sensitive information.

- Unauthorized operations in the company's key business. They include unauthorized modification of key information, and modification of key business configuration.
- Vulnerabilities that affect a large number of users. For example, stored cross-site scripting (XSS) vulnerabilities (and stored DOM-XSS vulnerabilities) on key webpages, which may spread automatically.

Medium level risks

Base score: 30-50. Medium level vulnerabilities include:

- Vulnerabilities that affect users. They include stored cross-site scripting (XSS) vulnerabilities on webpages and cross-site request forgery (CSRF) in key businesses.
- General unauthorized operations, including bypassing restrictions to modify user information and execution of user operations.
- General logical design and process flaws, including unlimited text messages and registration using any phone number and email.

Low level risks

Base score: 10-20. The reward factor can be set to zero. Low level vulnerabilities include:

- Local denial-of-service vulnerabilities, including local denial-of-service vulnerabilities on the client (exceptions that occur during the parsing of file formats and network protocols), and issues related to Android component access exposure and general application access.
- General information leakage, including clear text passwords on the client, web path traversal, and system path traversal.
- Other low-risk vulnerabilities, including reflected cross-site scripting (XSS) vulnerabilities (and reflected DOM-XSS vulnerabilities), general cross-site request forgery (CSRF), and URL redirection vulnerabilities.

5 Using OpenVPN to test vulnerabilities

This article describes how to test vulnerabilities by using the OpenVPN tool.

Context

Certain projects on the Crowdsourced Security Testing platform require that you use a VPN service when you perform testing. On the enterprise page, if certain VPN testing requirement message appears, it means that the company requires testing using a VPN. Without a VPN, your reports cannot pass the vulnerability review.

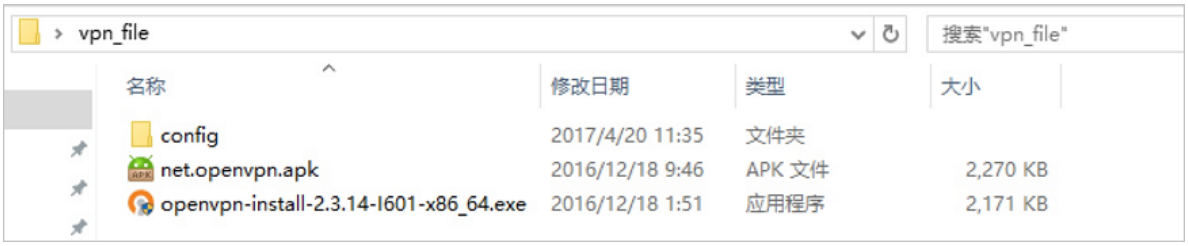
Note:

OpenVPN on the Mac system may not resolve domain names correctly. We recommend that you try the following methods:

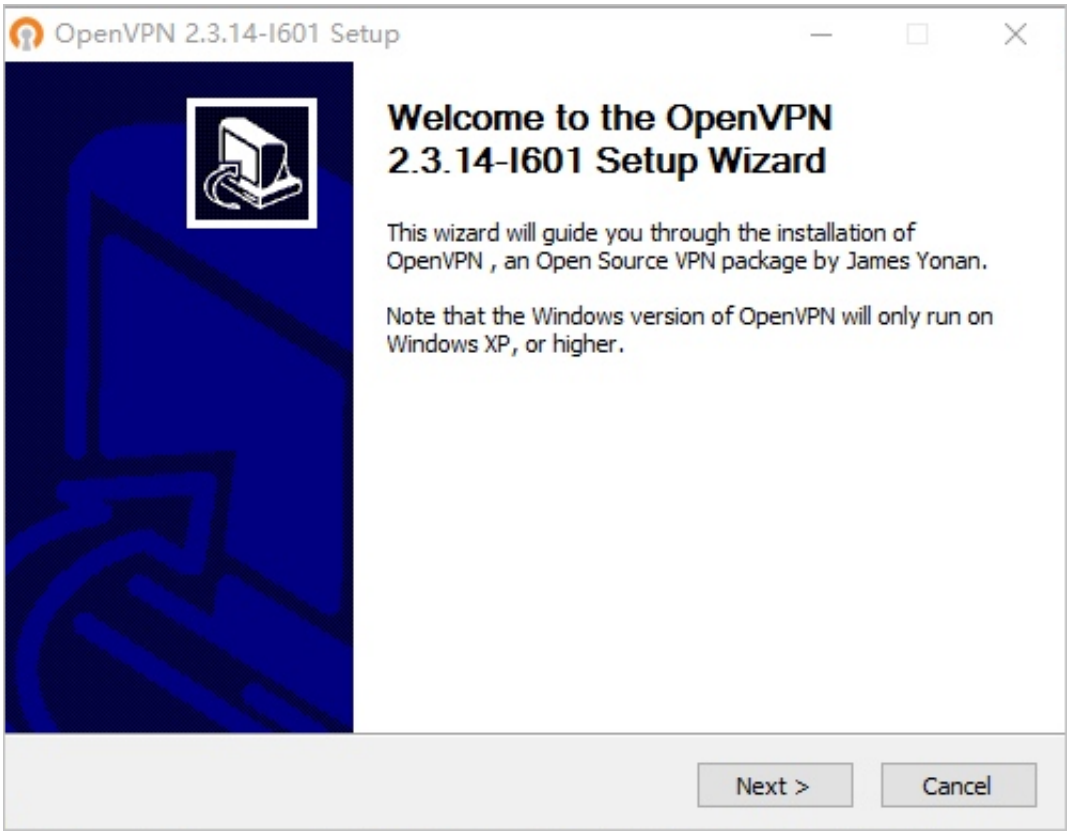
- Directly enter the IP address to access the test target.
- Use a Windows virtual machine to perform testing.

Procedure

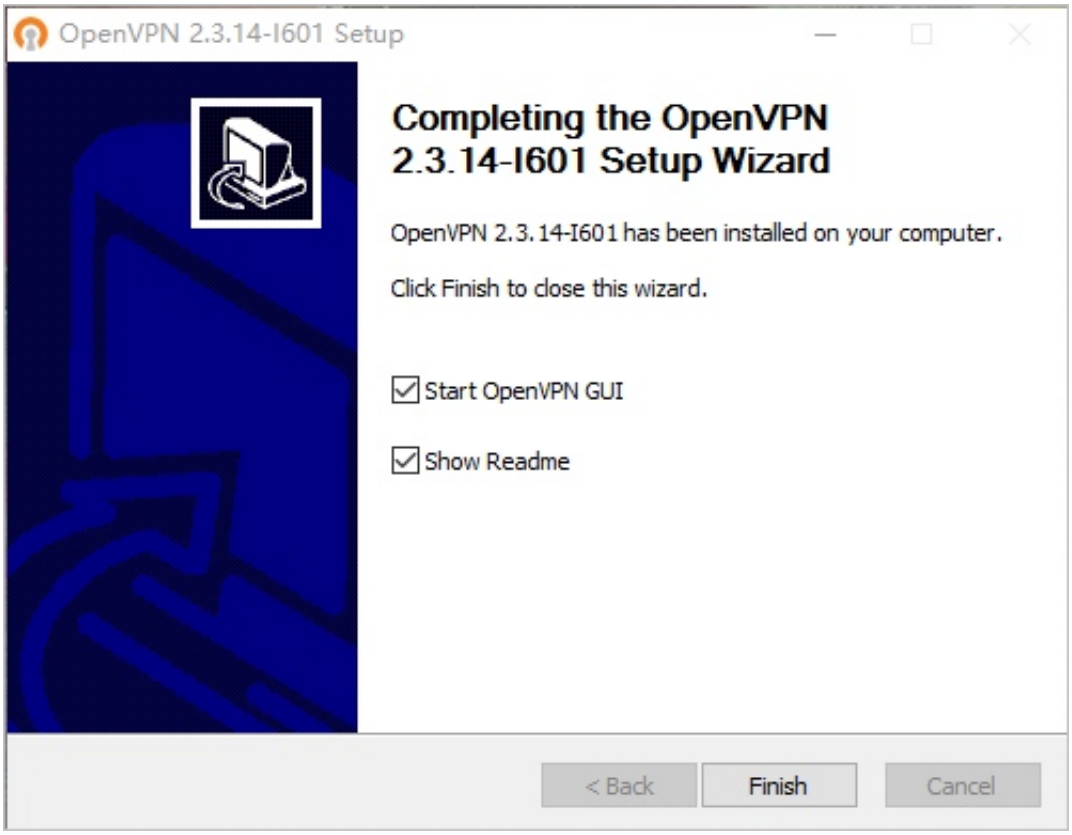
1. Click to download the *OpenVPN compressed file* to a local folder.
2. Extract the compressed file.



3. Run the openvpn-install-2.3.14-I601-x86_64.exe file to install OpenVPN.



4. Run OpenVPN.



5. Enter the user name and password you have received in the password.txt file and save it. The first line is the user name and the second line is the password.
6. Modify the client.ovpn file. Add the IP address and port of the VPN server after `remote`. For example, `remote 1.1.1.1 1194`.

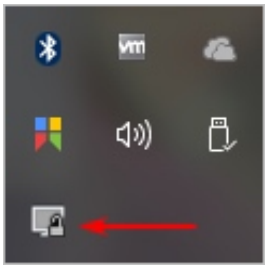
```
client
dev tun
proto udp
##############################################
# The hostname/IP and port of the server.#
# You can have multiple remote entries   #
# to load balance between the servers.   #
##############################################
remote
##############################################
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
ns-cert-type server
comp-lzo
verb 3
auth-user-pass password.txt
```

7. Install the ca.crt certificate file, and add the certificate to the trusted root certificates.

> Note:
> You have to manually import certificates into your mobile device and PC to establish VPN connections.

8. Save the configured ca.crt, client.ovpn, and password.txt files in the config folder of OpenVPN. For example, C:\Program Files\OpenVPN\config.

9. **Double-click the application icon in the lower right-hand corner to run OpenVPN.**



The connection is now established.