ALIBABA CLOUD

Alibaba Cloud

游戏盾

Quick Start

Document Version: 20220601

Alibaba Cloud

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 because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should 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 this document based on the "status quo", "being defective", and "existing functions" of its products and services. 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 take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, 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 contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, 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 this document 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 directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice 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.
Warning	A warning notice 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 restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	**Notice:** If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	**Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Step 1: Create a game instance

Before you add a game to GameShield, you must create a game instance in the GameShield console. In this step, the required settings are configured for the game.

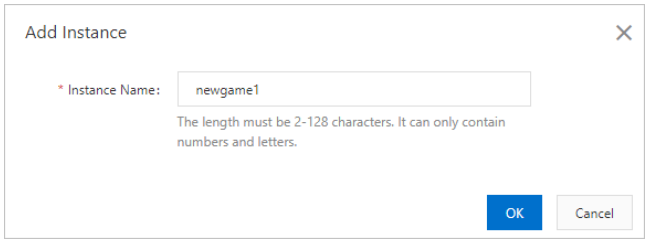
Prerequisites

GameShield is activated and has sufficient instance quota. For more information, see Billing methods.

Procedure

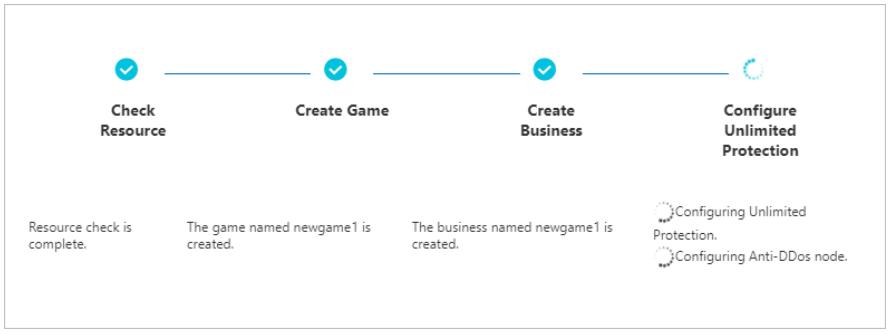
1. Log on to the GameShield console.
2. On the **Homepage**, click **Create Instance**.
3. In the **Add Instance** dialog box, configure **Instance Name** and click **OK**.

   > **Note** The name can be up to 24 characters in length and can contain letters and digits.

   GameShield runs an automation script to complete the following tasks and configure default GameShield settings for the game: **Check Resource**, **Create Game**, **Create Business**, and **Configure Unlimited Protection**.

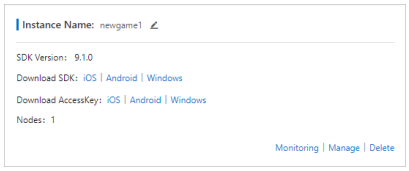
   The default settings include the game and business that are created. In addition, a node group and an anti-DDoS node are created for the business. By default, the unlimited protection feature is enabled.

4. After the game instance is automatically created and configured, click **OK** in the **Tips** message.

Result

After a game instance is created, you can view the instance on the **Homepage**.



What's next

Step 2: Configure a protection object

2.Step 2: Configure a protection object

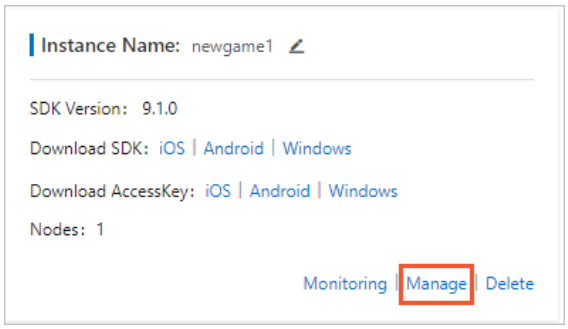
After you create a game instance, you must configure a protection object for the instance.

Prerequisites

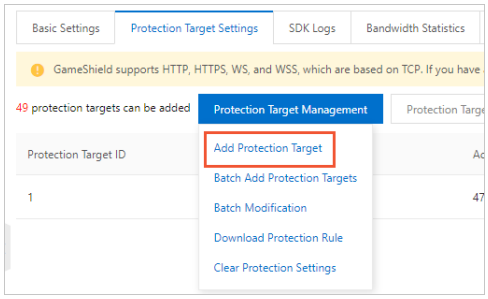
A game instance is created. For more information, see Step 1: Create a game instance.

Procedure

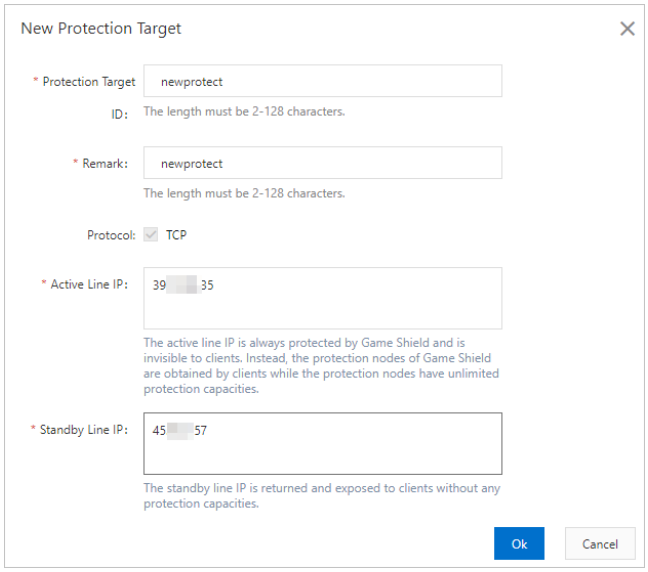
1. Log on to the GameShield console.
2. On the **Homepage**, find the instance that you want to manage and click **Manage**.



3. On the **Protection Target Settings** tab, click **Protection Target Management** and select **Add Protection Target**.



4. In the **New Protection Target** dialog box, configure the parameters and click **OK**. The following table describes the parameters.



Parameter	Description
Protection Target ID	Enter an ID for the protection object. The ID can be up to 128 characters in length, and can contain letters, digits, periods (.), underscores (_), and hyphens (-). If you use SDKs to configure protection, the ID is required to call API operations.
Remark	Enter remarks. The remarks can be up to 128 characters in length, and can contain letters, digits, periods (.), underscores (_), and hyphens (-).
Protocol	Use the default setting for this parameter. Only **TCP** is supported.
Active Line IP	Enter the IP address of the active line for the game. You can enter up to 20 IP addresses. Separate multiple IP addresses with commas (,). The IP addresses of the active line are always protected by GameShield and are not exposed to clients. Clients can access only the protection nodes that are provided by GameShield. This design provides unlimited protection against attacks for a game.
Standby Line IP	Enter the IP address of the standby line for the game. You can enter up to 20 IP addresses. Separate multiple IP addresses with commas (,). The IP addresses of the standby line are returned and exposed to clients. These IP addresses are not protected.

Result

After the protection object is added, you can view the object on the **Protection Target Settings** tab. You can click **Edit** or **Delete** to edit or delete the protection object.

What's next

Step 3: Add your application to GameShield

3.Step 3: Add your application to GameShield

GameShield allows you to add your application by using an SDK. This way, your application is switched to the protection mode within seconds after your application experiences a DDoS attack. GameShield also protects your application against HTTP flood attacks. This topic describes how to add your application to GameShield by using an SDK.

Procedure

After you add your application by using an SDK, GameShield provides several benefits: scheduling within seconds, link detection, and intelligent scheduling. When your application experiences DDoS attacks, GameShield proactively isolates malicious clients and performs scheduling in seconds. Protocol-level data encryption avoids false positives and false negatives, and protects against HTTP flood attacks.

Before you add your application by using an SDK, you must obtain an SDK and AccessKey pair. For more information, see Obtain an SDK package and AccessKey pair. Then, you can add your application by using the SDK. For more information, see the following topics:

- Use Android Studio to integrate GameShield SDK for Android
- Use Xcode to integrate GameShield SDK for iOS
- Use a C++ compiler to integrate GameShield SDK for Windows
- Use a Python IDE to integrate a Windows SDK
- Use Unity to integrate an SDK into Ubuntu

After you add your application, you can use the SDK to retrieve the IP address and the port that are mapped by GameShield for your application. The method that GameShield uses to map an IP address and a port for your application varies based on your service type. For details about the methods, see the following topics:

- TCP applications
- HTTP and HTTPS applications
- HTTP and HTTPS applications with the Browser/Server (B/S) architecture

4.SDK integration

4.1. Obtain an SDK package and AccessKey pair

After you create a game in GameShield, you can obtain an official SDK package and AccessKey pair from the GameShield console. GameShield provides you with different SDK packages and AccessKey pairs that are specific to your operating system, such as Android, iOS, and Windows.

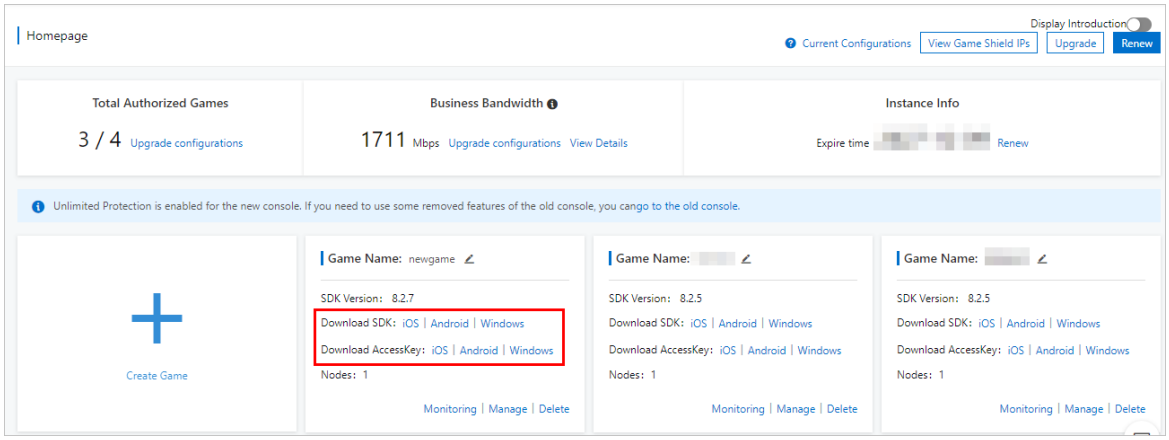
Prerequisites

A game instance is created. For more information, see Step 1: Create a game instance.

Procedure

1. Log on to the GameShield console.
2. In the **GameShield console**, find the target game, and click **iOS**, **Android**, or **Windows** in the **Download SDK** field to download an SDK package that is specific to your operating system.

   For example, you can click **iOS** in the **Download SDK** field to download an SDK package for iOS.



> **Note**
>
> - You can download official SDK packages from the GameShield console.
> - SDKs of GameShield 5.2.5 and later versions support SDK encryption tunnels to defend against Challenge Collapsar (CC) attacks of the protocol simulation type.

3. Find the target game, and click **iOS**, **Android**, or **Windows** in the **Download AccessKey** field to download an AccessKey pair that is specific to your operating system.

   For example, you can click **iOS** in the **Download AccessKey** field to download an AccessKey pair for iOS.

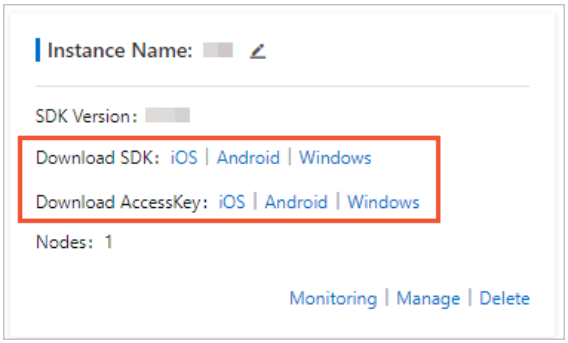
4.2. Integrate an SDK that is specific to your operating system

4.2.1. Use Android Studio to integrate GameShield SDK for Android

This topic describes how to use Android Studio to integrate GameShield SDK for Android.

Prerequisites

The SDK package and AccessKey for Android is obtained from the GameShield console. The AccessKey is required when you set the appkey parameter. For more information, see Obtain an SDK package and AccessKey pair.



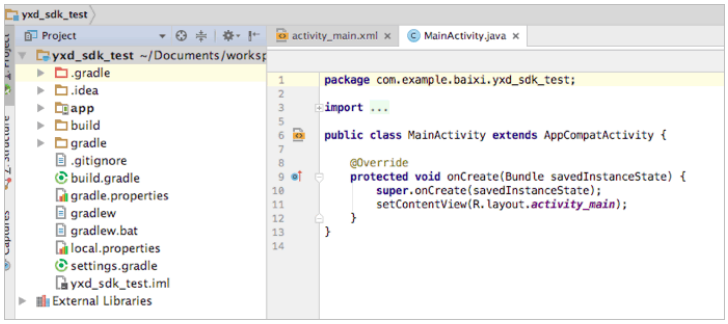
Context

A GameShield SDK includes two import methods: initEx and getProxyTcpDomain. For more information, see Introduction to core methods.

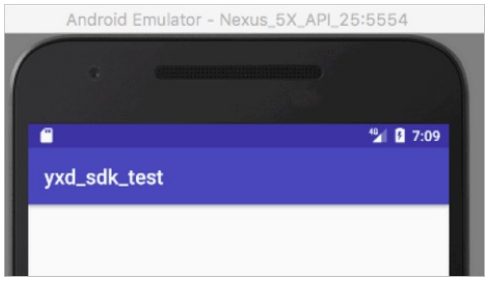
If you want to obtain a demo, contact after-sales technical support of GameShield.

Procedure

1. Open Android Studio.
2. Create a project and use the default settings to complete the creation of the project. In this example, the project is named *yxd_sdk_test*.
   The following figure shows the structure of the project directory.

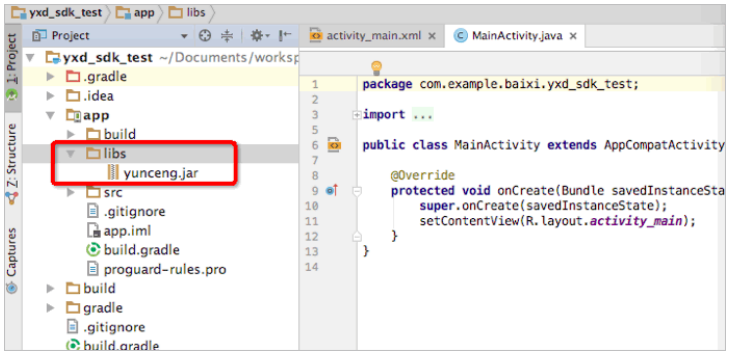


> **Note** Before you perform the next steps, you must ensure that the new project is running normally.

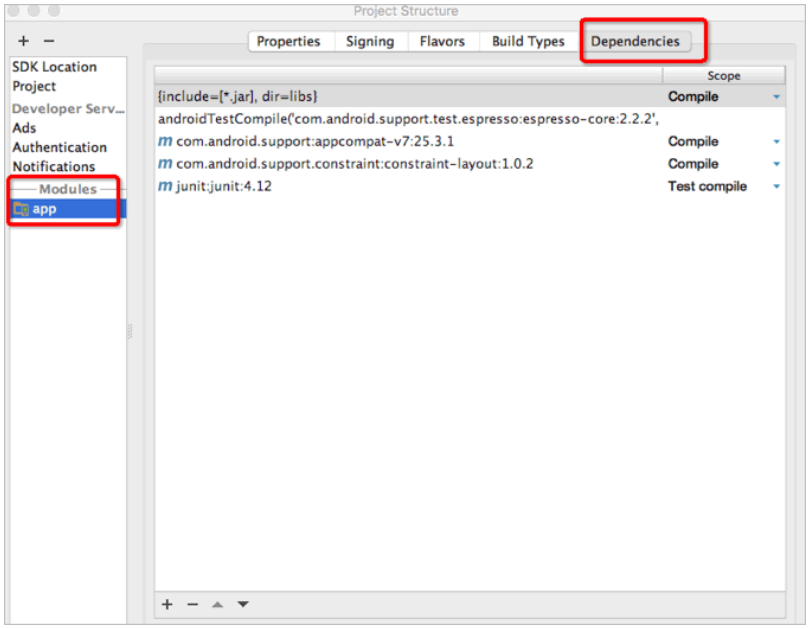


3. Add a JAR package.

   i. Copy the *yunceng.jar* file from the Android SDK package to the **libs** directory of the project.

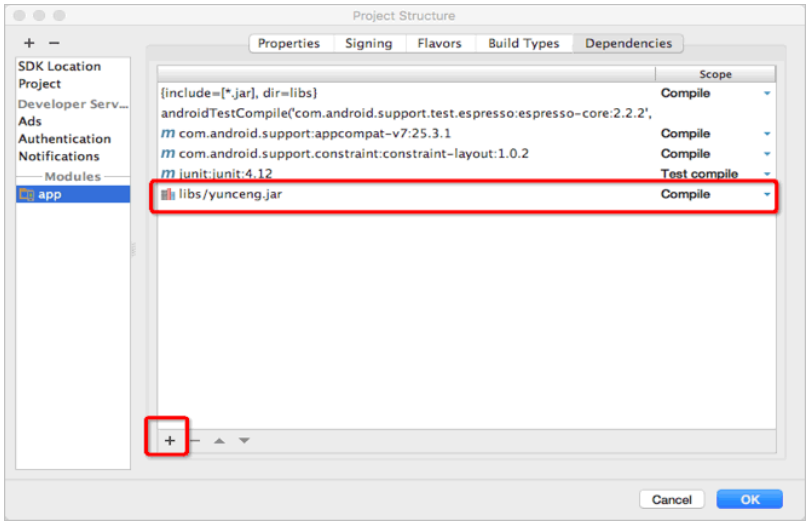


   ii. Open Android Studio, choose **File > Project Structure**, click **app**, and then click the **Dependencies** tab.



iii. Click the

icon, select **jar dependency**, and then add the **yunceng.jar** file.



iv. Click **OK**.

4. Add an SO file. Go to the **src > main** directory and create a directory named **jniLibs** in the main directory. Then, copy the SO file from the Android SDK package to the **jniLibs** directory.

5. Configure access permissions. Open the *AndroidManifest.xml* file and add the following code to the file, as shown in the following figure.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
```

6. Initialize the SDK.

> **Note** You need to complete the initialization of the SDK only once.

Before you run the following code, you must replace the following variables:

- Replace `<yourAppKey>` with the AccessKey that you obtained in the GameShield console.
- Replace `<token>` with the unique ID of a player in your game. If the ID is unavailable, you can use the default value. You cannot leave <token> unspecified.

```
public int sdk_init() {
    String appkey = "<yourAppKey>";
    String token = "<token>";
    int ret = YunCeng.initEx(appkey, game);
    if (ret == 0) {
        Log.d("yuncengsdk", "sdk init success");
    } else {
        Log.e("yuncengsdk", "sdk init failed, ret " + ret);
    }
    return ret;
}
```

7. (Optional)Configure ProGuard.

   If you use ProGuard to implement obfuscation, you must add the following code to the ProGuard configuration file.

```
-keep class com.aliyun.security.yunceng.** {*;}
```

What's next

After you add your application, you can use the SDK to retrieve the IP address and the port that are mapped by GameShield for your application. The method that GameShield uses to map an IP address and a port for your application varies based on your service type. For details about the methods, see the following topics:

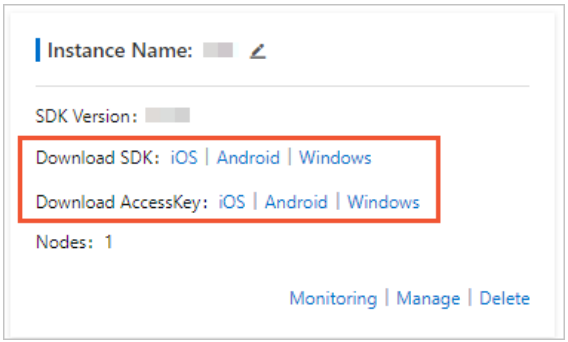
- TCP applications
- HTTP and HTTPS applications
- HTTP and HTTPS applications with the Browser/Server (B/S) architecture

4.2.2. Use Xcode to integrate GameShield SDK for iOS

This topic describes how to use Xcode to integrate GameShield SDK for iOS.

Prerequisites

The SDK package and AccessKey for iOS is obtained from the GameShield console. The AccessKey is required when you set the appkey parameter. For more information, see Obtain an SDK package and AccessKey pair.

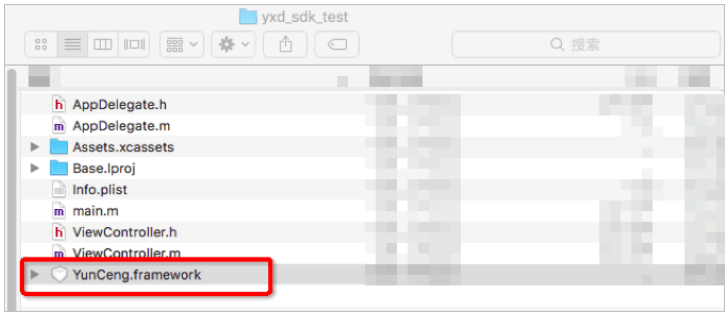


Procedure

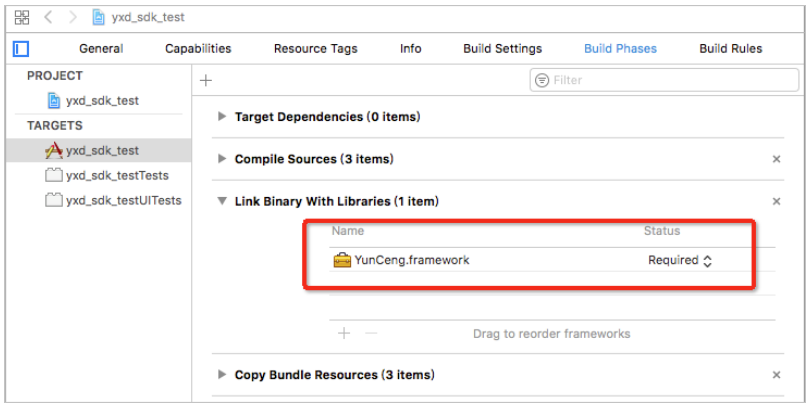
1. Open Xcode.
2. Create a project, select **Single View Application**, and then use the default settings to complete the creation of the project. In this example, the project is named *yxd_sdk_test*.

   > **Note** Before you perform the next steps, you must ensure that the new project is running normally.

3. Add dependencies. Copy the *YunCeng.framework* file from the iOS SDK package to the directory where the new project resides. In this example, the project is *yxd_sdk_test*.



4. Modify the project settings.
   - Change **Build Phases** to **Link Binary With Libraries**.
   - Add the *YunCeng.framework* file and its associated frameworks.



5. Initialize the SDK.

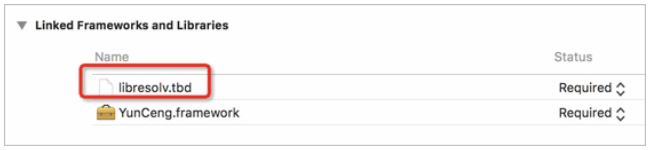
Before you run the following code, you must replace the following variables:

- Replace `<yourAppKey>` with the AccessKey that you obtained in the GameShield console.
- Replace `<token>` with the unique ID of a player in your game. If the ID is unavailable, you can use the default value. You cannot leave <token> unspecified.

```
- (int) SDKInit {
    const char * appkey = "<yourAppKey>";
    const char * token = "<token>";
    int ret = YunCeng_InitEx(appkey, token);
    if (ret == 0) {
        NSLog(@"sdk init success");
    } else {
        NSLog(@"sdk init failed, ret %d", ret);
    }
    return ret;
}
```

Troubleshooting:

- If the **"_OBJC_CLASS_$_CTTelephonyNetworkInfo", referenced from** error message appears when you compile the code, we recommend that you add the *CoreTelephony.framework* library. For more information, see Step 4.
- If the **"_res_9_getservers", referenced from** error message appears when you compile the code, we recommend that you add the *libresolv.tbd* library. For more information, see Step 4.



What's next

After you add your application, you can use the SDK to retrieve the IP address and the port that are mapped by GameShield for your application. The method that GameShield uses to map an IP address and a port for your application varies based on your service type. For details about the methods, see the following topics:

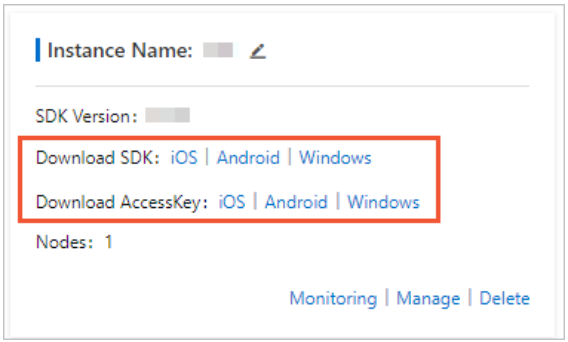
- TCP applications
- HTTP and HTTPS applications
- HTTP and HTTPS applications with the Browser/Server (B/S) architecture

4.2.3. Use a C++ compiler to integrate GameShield SDK for Windows

This topic describes how to use a C++ compiler in Windows to integrate GameShield SDK for Windows.

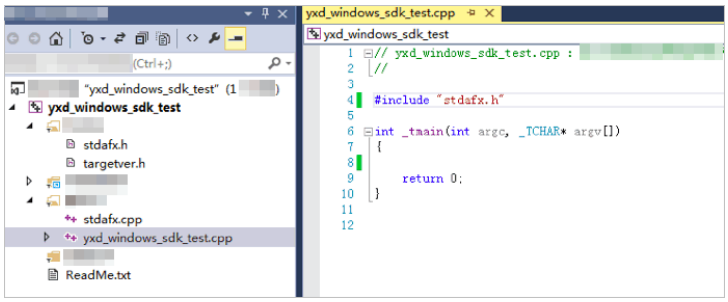
Prerequisites

The SDK package and AccessKey for Windows is obtained from the GameShield console. The AccessKey is required when you set the appkey parameter. For more information, see Obtain an SDK package and AccessKey pair.



Procedure

1. Open a C++ compiler in Windows.
2. Create a project of the Console App type. In this example, the project is named *yxd_windows_sdk_test*.



3. Add dependencies to the libs directory.
   i. Create a directory named *libs* in the project directory.
   ii. Copy the *YunCeng-WINDOWS.lib* file from the Windows SDK package to the *libs* directory.
   iii. Open the Property page of the project, choose **Linker > General**, and then add **./libs** as an **additional library directory**.
   iv. Choose **Linker > Input** and add **YunCeng.WINDOWS.lib** as an **additional dependency**.
4. Add header files and write test code.

```
char appkey[] = "testgroupid";
eAlSdkRet ret = YunCeng_InitEx(appkey, "token");
if (ret != cAlSdkOK) {
        printf("init sdk failed.\n");
        return -1;
   }
```

5. Move the *YunCeng-WINDOWS.dll* file to the directory where the project executable file resides. In this example, the executable file is *yxd_windows_sdk_test.exe*.

6. Check whether you can obtain the IP address of a game client.

```
/*
 ret = YunCeng_GetProxyTcpByIp("token","groupId", "192.168.0.1","80", ip,ip_len,  port,
port_len);  */
ret= YunCeng_GetProxyTcpByDomain("token","groupId", "www.aliyundoc.com","80", ip, ip_le
n, port,port_len);
if (ret != cAlSdkOK) {
    printf("get next ip failed. \n");
   } else {
    printf("get next ip success. %s %s\n", ip, port);
   }
```

4.2.4. Use a Python IDE to integrate a Windows SDK

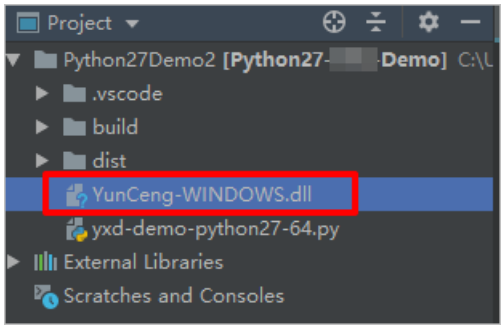
This topic describes how to use a Python IDE to integrate a GameShield SDK into Windows. It uses PyCharm as an example.

Prerequisites

- You can obtain an SDK and AccessKey pair for Windows from the GameShield console. For more information, see Obtain an SDK package and AccessKey pair.
-

Procedure

1. Open a Python editor. The following uses PyCharm as an example.
2. Create a 64-bit project.
3. Copy the *YunCeng-WINDOWS.dll* file from the Windows SDK to the directory where the project resides.



4. Add the following statement to the code to load the *YunCeng-WINDOWS.dll* file.

```
g_Dll = cdll.LoadLibrary("YunCeng-WINDOWS.dll")
```

5. Use the following code to check whether you can retrieve the IP address of a protection target.

```
Init_result = g_Dll.YunCeng_InitEx(access_key,token)//Initialize the Windows SDK.
#Return value
ip_len=18
ip = create_string_buffer('/0'*ip_len)
port_len=18
port = create_string_buffer('/0'*port_len)
ret=g_Dll.YunCeng_GetProxyTcpByDomain("Player ID","GroupName","Protection target ID","P
ort for the protection target", ip, ip_len, port,port_len);
if  ret == 0 ://A return value of 0 for the ret parameter indicates a success retrieval
.
```

4.2.5. Use Unity to integrate an SDK into Ubuntu

GameShield does not provide an SDK for Ubuntu. If you want to integrate a GameShield SDK into Ubuntu, you can use Unity to work with Android Studio or XCode to enable the SDK on Ubuntu.

We recommend that you determine the optimal access method to a game based on your business requirements.

4.3. Use SDKs to integrate applications into GameShield

4.3.1. TCP applications

This topic describes how to integrate TCP applications into GameShield. These TCP applications include game logon services and game servers. The access method over TCP is the simplest among all access protocols. To access a TCP application, you only need the IP address and port that is generated by GameShield for the application. No extra step that is required to access a TCP application.

The following code shows how to establish a persistent connection.

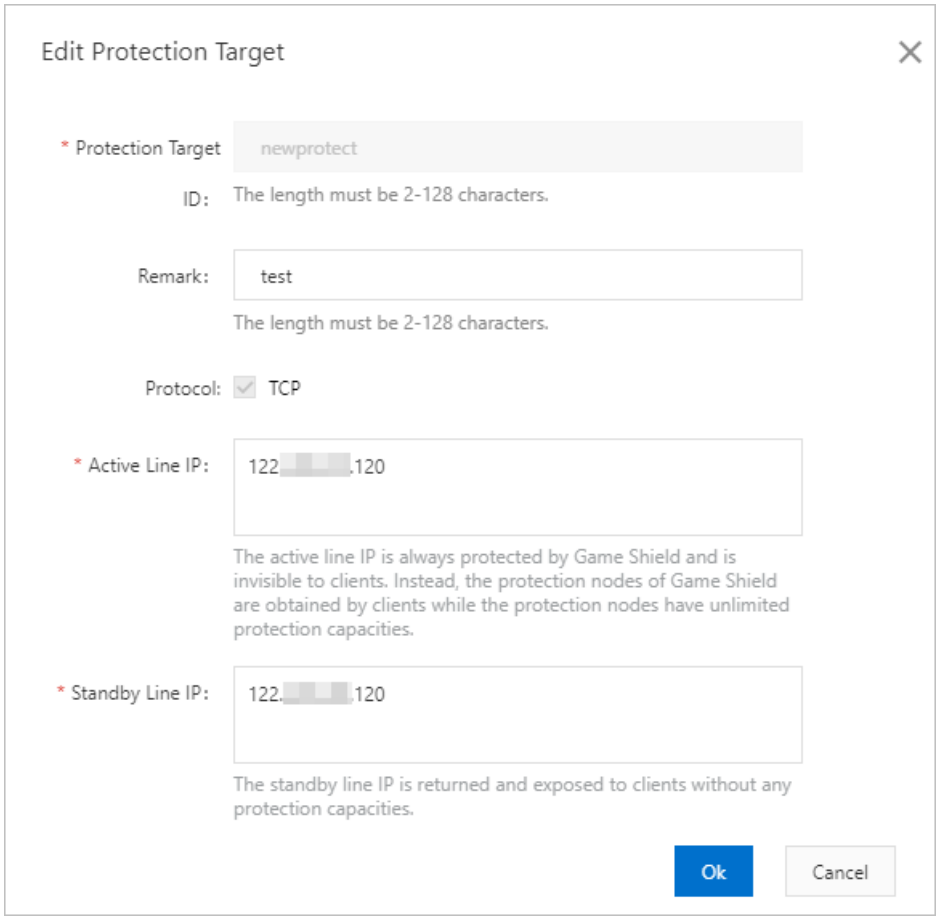
```
Socket socket = new Socket("127.0.0.1", target_port.toString());
//Establish a TCP persistent connection by using a socket.
//The target_port.toString() method returns a local random port number from GameShield.
```

4.3.2. HTTP and HTTPS applications

This topic describes how to use SDKs to integrate HTTP and HTTPS applications into GameShield. These HTTP and HTTPS applications include APIs, and websites for user logon and data retrieval.

Configure a protection target in the GameShield console

When you add a protection target to GameShield by using the console, you must specify a standard domain name as the **protection target ID**. The new domain name must point to the IP address 127.0.0.1. The following figure shows how to configure a protection target.



For more information about how to add a protection target, see Step 2: Configure a protection object.

Call an SDK by using a game client

GameShield concatenates and converts the IP address 127.0.0.1 and port 8901 (a random port number) from a protection target ID to an HTTP address. For example, `http://127.0.0.1:8910`.

For HTTPS applications, you must replace 127.0.0.1 with a standard domain name that has an SSL certificate configured. The domain name must point to 127.0.0.1. An example of the domain name is `http://login-for-yxd.aliyundoc.com:8910/login-for-yxd.aliyundoc.com`. This method helps you fix issues in hostname matching and certificate verification.

Sample code

```
String url = 'https://'+"login-for-yxd.aliyundoc.com"+':'+target_port.toString();//The URL of an HTTPS short-lived connection request.
```

The `target_port.toString()` method returns a local random port number.

References

For more information, see Best practice for dealing with HTTPS business. This topic provides instructions about how to integrate an HTTPS application into GameShield.

4.3.3. HTTP and HTTPS applications with the Browser/Server (B/S) architecture

You can use SDKs to integrate HTTP and HTTPS applications with the Browser/Server (B/S) architecture into GameShield. These applications provide services, such as administration console, customer services, and website services for adding funds. For more information, see the following topic: HTTP and HTTPS applications. This topic describes how to access HTTP and HTTPS applications after integrating these applications into GameShield.

Method	Feature	Description
Use a browser to access an endpoint that is generated by GameShield for a game.	Cost-effective and low compatibility	Potential issues: For iOS systems, a game client is switched to the background after calling a browser to access a game. Then, the IP address and port that are generated by GameShield immediately become unavailable. With a browser, you may experience compromised performance when accessing HTTP and HTTPS applications that have infrequent user interactions, such as websites for adding funds. However, you may have difficulty in accessing HTTP and HTTPS applications that have frequent user interactions, such as websites that provide administration consoles. You can resolve this issue by using WebView to replace the browser. For more information, see the next method.
A game client calls the WebView framework that is provided by Tencent inside the client rather than opening a browser.	High costs and better compatibility	We recommend that you determine the optimal access method to a game based on your business requirements.
Recommended. Configure a local proxy for WebView. You can map a remote proxy to the localhost by using GameShield. This method transfers all traffic that is initiated from WebView to GameShield.	High costs and optimal compatibility	This method exposes an origin server to the Internet. The origin server must reside in the China (Hangzhou) region. If an origin server that resides in the region is exposed, GameShield continues to forward data without unexpected interruptions. We recommend that you determine the optimal access method to a game based on your business requirements.

4.4. Obtain the real IP address of a game client

4.4.1. Overview

This topic describes how to obtain the real IP addresses of clients that attempt to access an application after it is integrated into GameShield.

Background information

GameShield adopts the FullNat proxy mode. After receiving a request from a client, GameShield replaces the IP address of the client with the IP address of GameShield. This topic provides a solution for obtaining the real IP address of a client.

Implementation

GameShield uses the options field of a Transmission Control Protocol (TCP) packet to store and transfer the IP address of a client. In most cases, this method is called TCP Options Address (TOA). The TOA method is provided by GameShield. You can only obtain the IP address of a client after integrating a TOA module to an origin server. You can integrate a TOA module by using application hooks. No code change is required.

- Linux

  Use application hooks to integrate a TOA module. For more information, see Linux.

- Windows

  Windows provides application hooks for some applications to integrate a TOA module. For more information, see Windows.

Deployment of origin servers

Scenario	Supported architecture	Unsupported architecture
Obtain the real IP address of a client when the client transfers data over TCP	• Data flows from GameShield to Alibaba Cloud Elastic Compute Service (ECS) instances that host origin servers or to third-party origin servers. • Data flows from GameShield and distributed at Layer 4 by using Alibaba Cloud Server Load Balancer (SLB). Data is then forwarded to Alibaba Cloud ECS instances that host origin servers.	Data flows from GameShield and distributed at Layer 4 by using third-party load balancing services. Data is then forwarded to third-party origin servers.

Scenario	Supported architecture	Unsupported architecture
Obtain the real IP address of a client when the client transfers data over HTTP or HTTPs	• Data flows from GameShield to Alibaba Cloud ECS instances that host origin servers or to third-party origin servers. • Data flows from GameShield and distributed at Layer 4 by using Alibaba Cloud SLB to Alibaba Cloud ECS instances that host origin servers.	• Data flows from GameShield to Web Application Firewall (WAF) or Anti-DDoS Pro and distributed at Layer 7 by using Alibaba Cloud SLB. Data is then forwarded to Alibaba Cloud ECS instances that host origin servers. • Data flows from GameShield and distributed at Layer 4 or Layer 7 by using third-party load balancing services. Data is then forwarded to third-party origin servers.

Note Based on Layer 4 data forwarding, GameShield does not manage HTTPS certificates. GameShield cannot retrieve data details that are contained in a HTTPS data stream. When a client accesses GameShield over HTTP or HTTPS, GameShield retrieves the real IP address of the client by using a TOA module that is installed on an origin server. You cannot obtain the real IP address of a client from the X-Forwarded-For (XFF) header field of an HTTP or HTTPS request.

4.4.2. Linux

This topic describes how to obtain real IP addresses of clients that access a game running on a Linux server.

Integrate a TOA module by using application hooks

1. Run the *install.sh* script to install services that relate to the toa_server.
2. Specify *preload.so* when starting an application service. If the name of an application service is nginx, you can use the following command to start the application server.

   ```
   LD_PRELOAD=./preload.so ./nginx
   ```

   Note You must find the entry point of your program and include the parameter in the preceding command to start the service.

   nginx service

   i. Run the *install.sh* script.

   ii. Check whether the */usr/lib/systemd/system/nginx.service* file exists.

   iii. Use the following command to update the *mynginx.sh* script.

   ```
   cat > /root/mynginx.sh
   ```

iv. Add the following statements to the *mynginx.sh* file and replace `path-to-preload.so` with the full path of the *preload.so* file.

```
#! /bin/bash
LD_PRELOAD=path-to-preload.so /usr/sbin/nginx
```

v. Use the following command to modify the permissions for the mynginx.sh file.

```
chmod +x /root/mynginx.sh
```

vi. Use the following command to edit the *nginx.service* file.

```
vi /usr/lib/systemd/system/nginx.service
```

vii. Replace `ExecStart=/usr/sbin/nginx` with `ExecStart=/root/mynginx.sh`.

viii. Use the following command to restart the nginx service.

```
service nginx restart
```

ix. Use the following command to enable automatic start for the nginx service at startup.

```
systemctl enable nginx.service
```

> **Note** You can also start the nginx service by adding the following statements to the *nginx_reload.sh* script.
>
> ```
> killall nginx
> LD_PRELOAD=path-to-preload.so /usr/local/nginx/sbin/nginx
> ```
>
> Then, you must add the directory where the *nginx_reload.sh* file resides to the *rc.local* startup configuration file.

3. Check whether *preload.so* is loaded by the nginx service.

   The followin example command are provided for your reference.

   - To check whether a port is enabled: `netstat -ntulp |grep 48888`
   - To check whether the nginx service is running and view its PID: `ps -ef | grep nginx`
   - To check whether preload.so is loaded: `cat /proc/PID/maps | grep preload.so`

Integrate a TOA module by modifying application code

The TOA module of GameShield installs a process on an origin server. The process listens on UDP port 48888. A game process passes non-real port numbers and IP addresses to port 48888 by using a specific format and retrieves port numbers and real IP addresses from the process.

Precautions

- You must make sure that UDP port 48888 is not disabled by the firewall on the localhost of 127.0.0.1.
- When you configure settings to allow access to the endpoint of 127.0.0.1:48888 through UDP, you must specify a timeout period. This setting helps avoid denial of services due to unexpected issues.
- You can retrieve real IP addresses by using the bypass method. In theory, retrieving real IP addresses seldom fails. However, you still need to prepare related solutions when an application process fails to retrieve real IP addresses.

- In a server-client connection, data that relates to IP addresses and ports is deleted when one of the parties closes the connection socket. You must re-establish a connection between the server and the client to create data that relates to IP addresses and ports.

For more information, see the instructions that are provided in the TOA archive. You can also contact GameShield Technical Support.

4.4.3. Windows

This topic describes how to obtain real IP addresses of clients that access a game running on a Windows server.

Procedure

1. You can use Visual Studio 2013 or a later version to open and compile the *toaservice.sln* file.
2. Run the *toaservice.exe* file that is compiled from toaservice.sln. This application runs as a backend process that listens on a network interface card (NIC) and allows access to API operations from UDP port 48888.

Integrate a TCP Options as Address (TOA) module by using application hooks

> **Note** This method is only applicable to C++ and not applicable to other languages, such as C#.

You can follow these steps to integrate the module.

1. Start the target application. Then, use the `LoadLibraryA("GetSourceName.dll")` command to load the *GetSourceName.DLL* file as soon as possbile.

   > **Note** For more information, see the TestGetSourceName method in the *toaservice.sln* file.

2. After the *GetSourceName.DLL* file is loaded, a backend process that is linked to the .dll file intercepts messages from the getpeername and accepts methods to retrieve the real IP addresses of clients.

Integrate a TOA module into application code

Similar to the method that is applied in Linux, you can integrate the TOA module by using code. The TOA module of GameShield installs a process on an origin server. The process listens on UDP port 48888. A game process passes non-real port numbers and IP addresses to port 48888 by using a specific format and retrieves port numbers and real IP addresses from the process.

Precautions

- Microsoft no longer offers support for Windows Server 2008.
- During tests, you must make sure that the firewall on an origin server is disabled. Otherwise, you fail to obtain real IP addresses because the WSASocket SOCK_RAW socket cannot sniff incoming data packets.
- You can retrieve real IP addresses by using the bypass method. In theory, retrieving real IP addresses seldom fails. However, you still need to prepare related solutions when an application process fails to

retrieve real IP addresses.

- You cannot use FreeLibrary to release the *GetSourceName.dll* file after you use the LoadLibrary method to load the *GetSourceName.dll* file. Otherwise, a Windows crash issue occurs.

  > **Note** To resolve this issue, you can purchase the Microsoft Detours package and replace the mhook library.

- The *GetSourceName.dll* file, the *toaservice.exe* file, and the executable file that loads *GetSourceName.dll* must reside in the same directory.
- You must have the administrator permissions to run the executable file that is used to load the *GetSourceName.dll* file.
- In terms of performance optimization, you must bind an NIC when a TOA module need to sniff data packets. If several NICs exists on an application server, you can modify the parameters in lines 131 to 139 of the *toa_service/win/toaservice/toaservice/Sniffer.cpp* file to bind the target NIC.

For more information, see the instructions that are provided in the TOA archive. You can also contact GameShield Technical Support.

4.5. Introduction to core methods

A GameShield SDK includes two core methods: initEx and GetProxyTcpByDomain. This topic describes how to use core methods. It also includes the details of each method.

initEx

You can call the initEx method to initialize an SDK when using the SDK for the first time. We recommend that you repeat the call for the method until a value of 0 is returned.

The following table illustrates the parameters of the initEx method. It also includes the description of each parameter.

Parameter	Description
access_key	The AccessKey pair that is used to access GameShield. You can download the AccessKey pair from the GameShield console. For more information, see Obtain an SDK package and AccessKey pair.
token	The player ID in the game. You can use the parameter to identify a malicious player or attacker when a game application is experiencing DDoS attacks. Default value: *Default*.

GetProxyTcpByDomain

You can call the GetProxyTcpByDomain method to retrieve callback IP addresses. It can also be used to retrieve random port numbers by using synchronized blocks.

The following describes the feature of the GetProxyTcpByDomain method.

- The IP address and port number for each return value are the same when the specified request parameters for each call are the same. This occurs within the lifecycle of an application after you start the application.
- The converted IP address and port number that are returned from a call remain unchanged. This occurs within the lifecycle of an application after you start the application. GameShield automatically

checks the availability of returned IP addresses and port numbers. Based on the results, GameShield automatically switches between nodes.

- The returned IP address and port number changes after you start the application again.
- The return value of the Target_ip parameter is set to 127.0.0.1. The IP address that is returned from the GetProxyTcpByDomain method remains unchanged. However, the port number that is returned is randomly generated.

The following table describes the parameters of the GetProxyTcpByDomain method.

Parameter	Description
Token	The player ID in the game. It is used to identify the malicious gamers or hackers who conduct DDoS attacks. Default value: Default.
GroupName	The node group name of a game, for example, access.v812vCOE21.aliyundoc.com. In the GameShield console, you must configure node groups after you add a game and an application. For each node group, you determine the number of nodes based on the number of simultaneous gamers. You can specify multiple node groups for each game.
Dip	The protection target ID, which is the IP address of the origin server. GameShield converts the IP address to a fixed IP address. You can obtain a protection target ID on the homepage of the GameShield console.
Dport	The port number of the origin server. The port is randomly generated. You do not need to configure the port number in the GameShield console.
target_ip	The returned IP address. The IP address is set to 127.0.0.1.
target_port	The randomly generated port number that is returned.

4.6. SDK error codes

This topic describes the common error codes defined by the GameShield SDK.

Overview

Error code	Description
0	No error occurred.
1000-1999	An error code returned because a network communication failure occurred.
2000-2999	An error code returned because the appkey parameter fails the verification or the initialization of the GameShield SDK fails when you attempt to integrate the GameShield SDK.
3000-3999	An error code returned because an error occurred in the GameShield server when you integrate the GameShield SDK.

Error code	Description
4000-4999	An error code returned because an error occurred during the data exchange between the GameShield SDK and the GameShield server.

Common error codes

Error code	Description	Solution
-1	The error code returned because the group name (groupname) or another parameter was not set.	Enter a valid value.
0	No error occurred.	N/A.
2000	The error code returned because the appkey parameter was not set.	Enter a valid value.
2001	The error code returned because the value of the appkey parameter is in an invalid format.	Use a valid format.
2002	The error code returned because the value of the appkey parameter exceeds the maximum length.	Check that the value of the appkey parameter does not exceed the maximum length.
2005	The error code returned because the API operation that is used to initialize the GameShield SDK was not called.	Call the initialization API operation first.
3201	The error code returned because the Gameshield SDK is not enabled.	Enable the Gameshield SDK.
3305	The error code returned because the SDK request parameters are invalid.	Check that the SDK request parameters are valid. If the issue persists, contact GameShield technical support.
3306	The error code returned because the SDK request type is invalid.	Check that the specified API operation is correct. If the issue persists, contact GameShield technical support.
3307	The error code returned because the SDK request parameters are invalid.	Check that the SDK request parameters are valid. If the issue persists, contact GameShield technical support.

Error code	Description	Solution
3500	The error code returned because no IP addresses are configured in the specified group.	Add IP addresses to the specified group in the GameShield console.
3600	The error code returned because no IP addresses are available in the specified group.	Add IP addresses to the specified group or enable unlimited protection against DDoS.
3700	The error code returned because the value of the groupname parameter is invalid.	Enter a valid value. If the issue persists, contact GameShield technical support.
3702	The error code returned because the protection target was not specified.	Set a protection target for unlimited protection against DDoS attacks in the GameShield console.
3703	The error code returned because the forwarding rule was not specified.	Set a port for unlimited protection against DDoS attacks in the GameShield console.
3800	The error code returned because SDK data was hijacked when it was transmitted over HTTP connections.	Contact GameShield technical support.
3999	The error code returned because the API parameter of the GameShield SDK is invalid.	Check that the API operation parameter is valid. If the issue persists, contact GameShield technical support.
4000	The error code returned because SDK data was hijacked when it was transmitted over HTTP connections.	Contact GameShield technical support.
9100	The error code returned because the API of the Gameshield SDK received simultaneous calls from multiple threads.	Call the API from one thread at a time.

Note If the issue persists, contact GameShield technical support.

4.7. Troubleshoot SDK issues

This topic describes how to troubleshoot SDK issues. Issues may occur when you use an SDK to access GameShield.

Context

A game calls the GetProxyTcpByDomain method to schedule available IP addresses through network security services and return these IP addresses to clients. These clients access IP address pools for different security networks based on different requested routes. These requests are then sent to origin servers.

Recommendations for debugging

You must list all request and response parameters for debugging. This facilitates troubleshooting. If some issues cannot be resolved, we recommend that you use the Wireshark tool to capture packets. You can submit abnormal packets that are detected by the tool to the GameShield team for further analysis of communication issues.

Procedure

1. Troubleshoot version-related issues.

   You can troubleshoot version-related issues based on SDK error codes. Based on the instructions that are provided by error messages, you can change settings in the GameShield console or change parameters when calling methods.

2. Confirm the status of origin servers.

   You can confirm the status of origin servers by checking whether the IP address and port of an application server are accessible.

   You can contact GameShield engineers for assistance to confirm the status of communication between the origin server and GameShield. The status indicates whether firewall policies on the origin server reject requests from back-to-origin IP addresses of GameShield. This method helps obtain a more accurate result than if you troubleshoot the issue by exploring online solutions.

3. Check request parameters again.

   You must check the IP address and port number by using an SDK. You must also check the target URL that is to be accessed. Check request parameters that are passed from GameShield. Use caution: Requests parameters are case-sensitive.

4. Confirm the protocol.
   - TCP: No specific action is required. In most cases, the returned IP address and port are accessible.
   - HTTP or HTTPS: You may need to handle host match issues. We recommend that you seek further assistance by contacting the game administrator.
   - WS or WSS: You may need to handle host match issues. We recommend that you seek further assistance by contacting the game administrator.