ALIBABA CLOUD

Alibaba Cloud

游戏盾

Quick Start

Document Version: 20201030

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

Before configuring protection for a game, you must create the game in GameShield. In this step, all of the required settings are configured for the game. When creating a game, you can only configure the name of a game. GameShield automatically configures default settings for the game.

Prerequisites

You have activated GameShield and one or more available app licenses. For more information, see Billing methods.

Procedure

1. Log on to the GameShield console.
2. On the **Homepage**, click **Create Game**.

3. In the **Add Game** dialog box, enter a **Game Name**, and click **OK**.

> **Note** The name can be a maximum of 24 characters in length and contain letters and digits.

GameShield calls automation scripts to complete the following tasks and configure default settings for the game. These tasks include **Check Resource**, **Create Game**, **Creating Business**, and **Configuring Unlimited Protection**. Default GameShield settings include a created game and business. A business includes a node group and anti-DDoS node. The unlimited protection feature is enabled by default.

4. After a game is created and configured, click **OK** in the **Note** dialog box.

Result

A game is created. You can view the new game in the GameShield console.

What's next

Step 2: Add a protection asset

2.Step 2: Add a protection asset

After you create a game in GameShield, you must add a protection asset for the game.

Prerequisites

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

Procedure

1. Log on to the GameShield console.
2. On the **Homepage**, find the required game, and click **Manage**.

3. On the **Protection Target Settings** tab, choose **Protection Target Management > Add Protection Target**.

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

Parameter	Description
Protection Target ID	The ID of a protection asset. Each ID is unique. The ID can be a maximum of 128 characters in length and contain letters, digits, and special characters. **Note** The supported special characters are hyphens (-) and periods (.). If you use SDKs to configure protection, the ID is required to call API operations.
Remark	Optional. The remark about the protection asset. The remark can be a maximum of 128 characters in length and contain letters, digits, and special characters, such as hyphens (-) and periods (.).
Protocol	**TCP** is selected by default and cannot be cleared.
Active Line IP	The IP address of the active line for the game. You can enter a maximum of 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	The IP address of the standby line for the game. You can enter a maximum of 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

A protection asset is added. On the **Protection Target Settings** tab, you can view the new protection asset and click **Edit** or **Delete** to edit or delete it as required.

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 an Android SDK
- Integrate an iOS SDK
- Use a C++ IDE to integrate a Windows SDK
- 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 is created. For more information, see Step 1: Create a game.

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 an Android SDK

This topic describes how to use Android Studio to integrate an Android SDK.

Prerequisites

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

- GroupName: indicates a node group ID. Log on to the GameShield console, open the game management page, and view the target node group ID on the **Basic Settings** tab.

- Protection Target ID. Log on to the GameShield console, open the Games page, and view a protection target ID on the **Protection Target Settings** tab.

Context

Key methods include initEx and getProxyTcpDomain. For more information, see Introduction to core methods.

You can contact GameShield Technical Support to obtain demo applications.

Procedure

1. Open Android Studio.
2. Create a project and use the default settings to complete creation. Name the project yxd_sdk_test.
   The following figure shows the structure of the new project directory.

> **Note** Before proceeding to subsequent operations, you must make sure that the new project is working as expected.

3. Add dependencies
   i. Copy the *yunceng.jar* file from the Android SDK to the *libs* directory. You can drag and drop the file to the libs directory.

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

   iii. Click the plus sign (**+**), select **jardependency**, and specify **yunceng.jar**.

   iv. Click **OK** to complete the configuration.
4. Add the .so file. Under the **src > main** directory, create a subdirectory named *jniLibs*, and copy the *libyunceng.so* file to the *jniLibs* directory.

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

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

6. Add the following code to obtain the IP address of a protection target. The following shows the sample code.

```
// Initialize the SDK. One successful initialization for the SDK is required.
int len = 0;
ret = YunCeng.initEx(getResources().getString(R.string.appkey), "token");if (0 ! = ret) {
    msg_show.setText("sdk init failed " + Integer.toString(ret));
    return;
}msg_show.setText("sdk init success ");
return;

// Use core methods to obtain the IP address of a protection target
ret = YunCeng.getProxyTcpByDomain("Player ID","GroupName", "Protection target ID","Port number o
f the origin server", ip, port);
if (0 == ret) {
    msg_show.setText("get next ip success: " + Integer.toString(ret) + "\nip : " + ip + "port :
" + port);
} else {
    msg_show.setText("get next ip failed. : " + Integer.toString(ret));
}
```

7. (Optional)Configure ProGuard. If you use ProGuard to perform obfuscation, you must add the following statement 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. Integrate an iOS SDK

This topic describes how to use XCode to integrate a GameShield SDK.

Prerequisites

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

- GroupName: indicates a node group ID. Log on to the GameShield console, open the game management page, and view the target node group ID on the **Basic Settings** tab.

  

- Protection Target ID. Log on to the GameShield console, open the Games page, and view a protection target ID on the **Protection Target Settings** tab.

  

Procedure

1. Open XCode.
2. Create a project, select **Single View Application**, and use the default settings to complete the creation of the project. Name the project yxd_sdk_test.

   > **Note** Before proceeding to subsequent operations, you must make sure that the new project is working as expected.

3. Add dependencies. Copy *YunCeng.framework* from the iOS SDK to the directory where the yxd_sdk_test project resides.

   

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

     

5. Modify *ViewController.m*.

   > **Note** For more information about the initEx and getProxyTcpByDomain methods, see Introduction to core methods.

```
// Initialize the SDK. One successful initialization for the SDK is required.
const char appkey[] = "appkeytest"; //Obtain the AccessKey pair from the console.
int ret = [YunCeng initEx: appkey: "token"];
if (0 ! = ret) {
  printf("init failed. \n");
  return;
}
const char groupname[] = "GroupName"; //Obtain the node group ID from the console. Each game has a unique node group ID.
char ip[128] = {0};
char port[32] = {0};

//Call the core methods.
ret = [YunCeng getProxyTcpByDomain: "Player ID": groupname: "Port for the protection target": "Port for the origin server": ip: 128 : port: 32];
if (0 ! = ret) {
  printf("get next ip failed. \n");
  return;
}
printf("get next ip success. %s, port:%s \n", ip, port);
```

Note

- If a message showing **"_OBJC_CLASS_$_CTTelephonyNetworkInfo", referenced from** occurs when you compile the project, we recommend that you add the *CoreTelephony.framework* library by referring to Step 4.
- If a message showing **"_res_9_getservers", referenced from** occurs when you compile the project, we recommend that you add the *libresolv.tbd* library by referring to 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++ IDE to integrate a Windows SDK

This topic describes how to use a C++ IDE to integrate a Windows SDK.

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.
- You can obtain the following information from the GameShield console.
  - GroupName: indicates a node group ID. Log on to the GameShield console, open the game management page, and view the target node group ID on the **Basic Settings** tab.

    
  - Protection Target ID. Log on to the GameShield console, open the Games page, and view a protection target ID on the **Protection Target Settings** tab.

    

Procedure

1. Open a C++ editor in Windows.
2. Create a project of the Console App type. Name the project yxd_windows_sdk_test.

   
3. Add dependencies to the libs directory.
   i. Create a directory named *libs* in the new project directory.
   ii. Copy the *YunCeng-WINDOWS.lib* file that resides in the Windows SDK to the *libs* directory.
   iii. Open the Property page of the project, choose **Linker > General**, and add **./libs** as an **additional library directory**.
   iv. Choose **Linker > Input**, and add **YunCeng.WINDOWS.lib** as an **additional dependency**.
4. Test the init method.

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

5. Copy the .dll file to the directory where the project resides. Move *YunCeng-WINDOWS.dll* to the directory where the project executable file named *yxd_windows_sdk_test.exe* resides.

   
6. Check whether you can obtain the IP address of a protection target.

```
//Call the key methods.
ip_len = 18
ip = create_string_buffer('/0' * ip_len)
port_len = 18
port = create_string_buffer('/0' * port_len)
ret = YunCeng_GetProxyTcpByDomain("Player ID", "GroupName", "Protection target ID", "Port for the
protection target", ip, ip_len, 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.
- You can obtain the following information from the GameShield console.
  - GroupName: indicates a node group ID. Log on to the GameShield console, open the game management page, and view the target node group ID on the **Basic Settings** tab.

    
  - Protection Target ID. Log on to the GameShield console, open the Games page, and view a protection target ID on the **Protection Target Settings** tab.

    

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","Port for t
he 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: Add a protection asset.

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.vivre.cn:8910/login-for-yxd.vivre.cn`. This method helps you fix issues in hostname matching and certificate verification.

Sample code

```
String url = 'https://'+"login-for-yxd.vivre.cn"+':'+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

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.
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.

Parameter	Description
GroupName	The node group name of a game, for example, access.v812vCOE21.xxxxxxxxxxx.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 common error codes of the GameShield SDK.

Descriptions

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 an error of the appkey parameter or initialization occurred.
3000~3999	An error code returned because an error occurred in the GameShield control center.
4000~4999	An error code returned because a data exchange error occurred in the GameShield control center.

Common error codes

Error code	Description	Solution
-1	The error code returned because the group name (groupname) or another parameter was not set.	Set a valid value.
0	No error occurred.	None.

Error code	Description	Solution
2000	The error code returned because the appkey parameter was not set.	Set a valid value.
2001	The error code returned because the format of the appkey value is invalid.	Use a valid format.
2002	The error code returned because the length of the appkey value exceeded the upper limit.	Check whether the appkey parameter is correctly set.
2005	The error code returned because the initialization endpoint was not called.	Call the initialization endpoint first.
3201	The error code returned because the SDK feature is disabled.	Contact GameShield technical support.
3305	The error code returned because the SDK request parameter is invalid.	Check whether the SDK request parameter is valid. If the issue persists, contact GameShield technical support.
3306	The error code returned because the SDK request type is invalid.	Check whether the specified endpoint is correct. An IP address and domain use different endpoints.
3307	The error code returned because the SDK request parameter is invalid.	Check whether the SDK request parameter is valid. If the issue persists, contact GameShield technical support.
3500	The error code returned because no IP address is available in the specified group.	Add IP addresses to the specified group in the GameShield console.
3600	The error code returned because no IP address is 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 group name (groupname) is invalid.	Set a valid value. If the issue persists, contact GameShield technical support.
3702	The error code returned because the protection target was not set.	Set a protection target for unlimited protection against DDoS in the GameShield console.
3703	The error code returned because the forwarding rule was not set.	Set a port for unlimited protection against DDoS in the GameShield console.

Error code	Description	Solution
3800	The error code returned because SDK data was hijacked when it was transmitted over HTTP connections.	If issues such as network hijacking occur, contact GameShield technical support.
3999	The error code returned because the endpoint parameter is invalid.	Check whether the endpoint 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.	If issues such as network hijacking occur, contact GameShield technical support.
9100	The error code returned because the endpoint received simultaneous calls from multiple threads.	Call the endpoint 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.