

互娱通用签名二次验证接口



腾讯科技（深圳）有限公司

版权所有 侵权必究

修订记录

日期	修订版本	描述	作者
2009-12-03	1.0	创建	hardway
2009-12-03	1.1	修改 对,配置文件和接口 作了重新定义	hardway
2009-12-04	1.2	修改 对配置字段,和数据 结构做了细化说明	hardway
2009-12-10	1.3	增加返回值定义,细 化了日志说明,增加 了FAQ说明	hardway
2009-12-17	1.4	发布tsigapi版本1.2, 增加IP验证不一致, 时戳不过期的配置。	hardway
2009-12-30	1.5	支持 windows(vc6,vc7,vc8) 版本	hardway

目录

1. 概述.....	3
1.1 概述.....	3
1.2 功能.....	3
1.3 词汇表.....	4
2. 配置.....	5
3. 接口.....	5
3.1 初始化句柄及读取配置文件.....	5
3.2 刷新配置文件.....	6
3.3 对签名中信息和当前连接信息作二次验证.....	7
3.4 释放句柄.....	8
4. FAQ.....	9
4.1 原来解签名逻辑需要改变么?	9
4.2 业务 ID 和客户端版本信息这些字段在 0x82 协议中没有定义怎么办?	9
4.3 日志目录的结构是怎么样的?	9

1. 概述

1.1 概述

提供一个通用的接口,根据游戏签名的输出结果和用户连接信息作二次对比验证。

1.2 功能

- 支持对 0X82/0XDDE 协议签名结果和当前连接信息作二次验证

- 支持对比验证条件可配置,可以动态 reload
- 支持日志功能(日志目录可以配置,按日期建目录,每天写一个文件,对于验证未通过,记录未通过类别,或者特殊情况下通过也记录日志)

日志格式:

时间 QQ 号 类别 连接中信息 签名中信息 原因

- 时间:当前时间
- QQ 号:签名中的 QQ 号
- 类别:未验证通过字段类别,如签名时戳过时,或 UIN 不一致或者 IP 不一致等。
 - 1:UIN
 - 2:时戳
 - 3:IP
 - 4:业务 ID
 - 5:客户端版本号
- 连接中信息:连接中该字段信息
- 签名中信息:签名中该字段信息
- 原因:表示验证未通过,但是满足一些特殊原因也认为验证合法,目前只对 IP 类别有效不一致

◆ IP 类别:

原因 1: 时戳还未过期,日志将记录原因为 1

原因 2: IP 属于同一个网段

1.3 词汇表

词汇	解释
0X82	0X82 QQ 签名协议
0XDE	0XDEQQ 签名协议

2. 配置

配置文件使用文本文件描述,,配置文件结构包括以下项,对应开关配置 1 为打开, 0 为关闭。
后续不再描述。

```
[General]      #签名结果验证字段

CheckUin=1      #必选项,检查签名 Uin 是否跟当前一致,如果解密签名中未作检查,必
                #须打开开关

CheckSigTime=1  #必选项,检查签名有效期,如果解密签名中未作检查,必须打开开关

TimeOutValue=300 #签名过期时间,打开检查签名有效期检查开关时有效

CheckCltIP=1    #必选项,检查签名中 IP 是否跟当前一致, 必须打开.

AllowedTimeGap=180 #当签名 IP 不一致时,签名有效期不超过指定时间,也可以认为
                  #IP 合法,打开检查 IP 开关时有效,如果 AllowedTimeGap 配为
                  #零,则签名不过期,可以先收集 IP 不一致的数据。

NetMaskBitCount=24 #子网掩码位数当签名 IP 不一致时,两者在同一个网段,也可以认
                  #为 IP 合法, 打开检查 IP 开关时有效

CheckServiceID=0 #可选项,检查签名业务 ID 是否跟当前一致,对于 0x82 不支持该开关
                 #对于 0XDE 建议打开业务 ID 检查

CheckCltver=0    #可选项,检查客户端版本是否跟签名中一致,对于 0X82 不支持该开
                 #关,对于 0XDE 可以根据业务是否对客户端版本做验证具体情况配
                 #置

[log]

logdir=./siglog  #统计日志目录,按日期记流水日志。
```

3. 接口

3.1 初始化句柄及读取配置文件

/**

*@brief:初始化句柄,读取配置文件

```

*@param ppSigCtx[IN]:                结构句柄指针地址
*@param szConfilePath[IN]:           配置文件
*@retval
*
*      = 0 for success
*
*      <0 failed,返回错误码
*
*      -1: 输入参数错误
*
*      -2: 读取 CheckSigTime 失败
*
*      -3: 读取 TimeOutValue 失败
*
*      -4: 读取 CheckUin 失败
*
*      -5: 读取 CheckCltIP 失败
*
*      -6: 读取 AllowedTimeGap 失败
*
*      -7: 读取 NetMaskBitCount 失败
*
*      -8: 读取 CheckServiceID 失败
*
*      -9: 读取 CheckCltver 失败
*
*      -10:读取 logdir 失败
*
*      -11:创建 logdir 失败
*
*      -12:创建句柄失败
*/

int tsigapi_init(IN LPTQQSIGHANDLE *ppSigCtx,
                 IN const char *szConfilePath);

```

3.2 刷新配置文件

```

/**
*@brief:reload,刷新配置文件
*@param pSigCtx[IN]:                结构句柄指针
*@param szConfilePath[IN]:           配置文件
*@retval
*
*      = 0 for success
*
*      <0 failed,返回错误码

```

```

*          错误码同 tsigapi_init
*/

int tsigapi_reload(IN TQQSIGHANDLE *pSigCtx,IN const char *szConfilePath);

```

3.3 对签名中信息和当前连接信息作二次验证

```

/**
*@brief:对签名中的信息和当前信息进行二次验证
*@param pSigCtx[IN]:          结构句柄指针
*@param  pstCltInfo[IN]      当前连接的信息
*@param  pstSigInfo[IN]      签名中的信息
*@retval
*          = 0 for success
*          <0 failed,返回错误码
*          -1:   输入参数验证不合法
*          -2:   uin 验证不一致
*          -3:   时戳超时
*          -4:   IP 验证不合法
*          -5:   serviceid 不一致
*          -6:   版本不一致
*/

/*结构定义*/
/*
struct tagTQQConnInfo
{
    unsigned int uin;          //必填项,用户 QQ
    unsigned int uCltIP;       //必填项,客户端 IP
    unsigned int uAppID;       //推荐项,最好填上,业务 ID
    unsigned int uCltVer;      //可选项,用户客户端版本,根据业务是否需要对版本验

```

证决定是否填写此项和打开验证开关

```
};
```

```
struct tagTQQSigCheckList
```

```
{
```

```
    unsigned int utimestamp;        //必填项,签名中时戳信息
```

```
    unsigned int uin;               //必填项,签名中用户 QQ
```

```
    unsigned int uClIP;             //必填项,签名中客户端 IP
```

```
    unsigned int uAppID;            //可选项,签名中业务类型,对于 0x82 签名结构该字段可以不填,对于 0xDE 需要填写
```

```
    unsigned int uClVer;            //可选项,签名中客户端版本,对于 0x82 签名结构可以不填,对于 0xDE 可以填写,是否验证根据业务具体情况配置
```

```
};
```

```
*/
```

```
int tsigapi_check(IN TQQSIGHANDLE *pSigCtx,
```

```
                  IN TQQConnInfo *pstConnInfo,
```

```
                  IN TQQSIGCHECKLIST *pstSigInfo);
```

3.4 释放句柄

```
/**
```

```
*@brief:释放句柄
```

```
*@param pSigCtx[IN]:                结构句柄指针
```

```
*@retval
```

```
*          = 0 for success
```

```
*          <0 failed,返回错误码,待定
```

```
*/
```

```
int tsigapi_fini(IN LPTQQSIGHANDLE *ppSigCtx);
```


4. FAQ

4.1 原来解签名逻辑需要改变么？

A:二次验证接口不关心解密签名的逻辑,只需要把解开后签名中信息传给接口作二次验证。

4.2 业务 ID 和客户端版本信息这些字段在 0x82 协议中没有定义怎么办？

A:对应 0x82 协议二次验证,在配置文档中不要验证业务 ID 和客户端版本信息.同时在传递给接口的结构这些字段可以不填。

4.3 日志目录的结构是怎么样的？

A:默认日志目录的结构如下,每天建一个目录,每一个小时建一个文件,其中根目录可以在配置文件中修改。

`./siglog`

`./siglog/20091214`

`./siglog/20091214/2009121410`

`./siglog/20091214/2009121411`

`/siglog/20091215`

`./siglog/20091215/2009121410`