Kerberos Server API

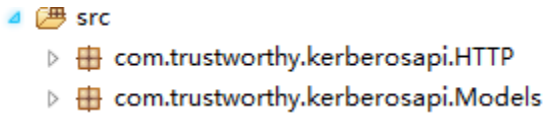
说明文档

1. Project 和 API 简介

本次 Project 的目的是为 Emergency Response System 建立一套安全体系，其中包括用户登录与访问控制、利用 Kerberos 来进行通讯的加密，最终将加密传输过程通过界面明确的显示出来。如果有时间，可以设计另一个黑客模拟程序来试图截取通信，来证明此系统的可靠。

Kerberos Server API 提供了连接本次 Trustworthy Project 中 Server 的所有必要 API。通过这组 API 程序可以直接与 Server 进行数据的传输，如请求 Ticket、发送自己的 Authenticator 等等；API 中还集成了 AES 加密的函数，其加密参数通过调试已与服务器保持一致。

所有与 Server 的通信都可以通过 API 中所提供的函数（将在第 3 节说明）来进行，只是通信内容（如用户名、要传入的 Ticket 或者 Authenticator 等参数）需要由用户自己定义。接收到的回复将保存在各个 Model（将在第 3 节说明）中，方便随时进行读取。API 为一个 Library，先引入到 Workspace 以后再从项目的属性中添加它即可。此 Library 分为两个文件夹：



HTTP 为连接相关的函数，Models 为 Project 中所用到的类。

2. 服务器简介

服务器由 Web API 搭建而成，通过此接口连接到数据库读取数据。所以最简单的访问方法则是通过浏览器直接进行 HTTP 访问。由于服务器位于骏仔的电脑上，所以并不能保证时刻处于运行状态，所以每次需要使用时可以通过以下网址查询：

http://s320880.vicp.net:8080/api/check

若出现“Yes, the server is on!”，如

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Yes, the server is on!</string>
```

则表示服务器正在运行中，否则可以联系我。（我会尽量保持打开电脑的时候就让服务器运行着，不过如果我在外面就木有办法了）

数据库

此 Project 分为三个数据库：AS 数据库、TGS 数据库和 Data 数据库。

AS 数据库是 Authentication Server 所使用的数据库，具体结构如下：

项目名	数据类型	说明
UserId	Int	自动生成，索引
Username	String	用户名
Password	String	密码，密文保存
Authority	Int	权限，0 为最低，数字越大权限越高

C_Time	DateTime	账户建立时间
Detail	String	待定

此数据库在调试阶段可以直接通过以下链接进行访问（下同）：

http://s320880.vicp.net:8080/api/auth *

TGS 数据库是 Ticket Generate Server 所使用的数据库，具体结构如下：

项目名	数据类型	说明
KeyId	Int	自动生成，索引
KeyType	Int	密钥类型，1 为 TGS、2 为 Service
KeyName	String	密钥名称，跟服务名一致
KeyValue	String	密钥值
Details	String	描述

http://s320880.vicp.net:8080/api/TGS

Data 数据库是存储最终数据所用的数据库，也是 Kerberos 系统中的最终 Service 端。在此 Project 中我们假设一共有两种不同的服务：Service_1 记录了每个 POI（如伤员、工作人员）的信息，Service_2 则是发布和查阅公告（如警告、命令等）的服务。

Service_1 数据结构：

项目名	数据类型	说明
Id	Int	自动生成，索引
POIType	Int	1-Victim; 2-Responder; 3-Commander; 4-Hazard
POIName	String	POI 的名字
POILocation	String	位置，若以后使用地图显示则存储其坐标
SecretLevel	Int	最低需要权限，只有权限高于此值的用户才能看到

http://s320880.vicp.net:8080/api/Service1

项目名	数据类型	说明
id	Int	自动生成，索引
MsgTitle	String	消息标题
MsgContent	String	消息内容
SecretLevel	Int	最低需要权限

http://s320880.vicp.net:8080/api/Service2

*以上能直接查询到的内容只是用作写代码和调试，Project 中并不能直接使用这些。

3. Models & Functions

3.1 HTTP.KerberosClient

负责连接主要的类，其中包含了所有与服务器进行通信的函数。注意：由于需要网络连接，所以此类的函数需要调用于线程中。其成员函数为：

Public String **getTimerStamp**()
生成一个本地的 TimeStamp
参数 *无* 返回值 字符串表示的时间戳，格式为 YY/MM/DD HH:MM:SS

Public ASResponse **getTGSTicketFromServer** (String username, String tgsname, String macaddress)	
从服务器获得 TGS Ticket	
参数	
username	用户名
tgsname	TGS 的名称
macaddress	本机的 MAC 地址
返回值 ASResponse，即 Authentication Server 的回应，具体见 ASResponse 类。	

Public boolean **createNewUser** (String username, String password, String detail)	
建立一个新用户	
参数	
username	用户名
password	密码，需要先进行加密再调用此函数
detail	详细信息，随便填什么或不填
返回值 是否建立成功	

Public TGSResponse **getServiceTicket**(String serviceName, TGSTicket ticket, AuthenticatorC authenticator)	
从 TGS 服务器获得 Service Ticket	
参数	
serviceName	所需要连接的 Service 名字
ticket	从 Authentication Server 获得的 TGS Ticket
authenticator	本机的 authenticator，需进行加密
返回值 TGSResponse，即 TGS 的回应，具体见 TGSResponse 类	

Public List<IncidentPOIs> **getAllPOIsFromService_1**(ServiceTicket ticket, AuthenticatorC auth)
从 Service 1 获得所有的 POI 信息
参数 *ticket* 从 TGS 获得的 Service Ticket *auth* 本机的 authenticator，需进行加密 返回值 若验证成功，则获得所有的 POI，否则为 null。具体见 IncidentPOIs 类。

Public List<IncidentMSGs> **getAllMSGsFromService_2**(ServiceTicket ticket, AuthenticatorC auth)
从 Service 2 获得所有的 Message 信息
参数 *ticket* 从 TGS 获得的 Service Ticket *auth* 本机的 authenticator，需进行加密 返回值 若验证成功，则获得所有的消息，否则为 null。具体见 IncidentMSGs 类。

3.2 Models.AESEncryption

负责 AES 加密与解密。使用此类不需要创建其对象，直接调用其成员函数即可：

Public static String **aesDecrypt**(String ciphertext, String key)
AES 解密
参数 *ciphertext* 密文 *key* 密钥 返回值 解密以后的值。如果无法解密将返回 null。

Public static String **aesEncrypt**(String ciphertext, String key)
AES 加密
参数 *ciphertext* 原文 *key* 密钥 返回值 加密以后的值。如果无法解密将返回 null。

3.3 Models.ASResponse

接收从 Authentication Server 返回的数据，默认属于加密状态，需进行解密才能读取其中的内容。其包含的成员变量如下：

变量名	说明	读取方法
key_C_TGS	服务器返回的 Session Key	getKey_C_TGS()
timestamp	此消息产生的时间	getTimestamp()
lifetime	此消息的有效时间长度（分钟）	getLifetime()
ticket	获得的 TGS Ticket	getTicket()
errorCode	返回的错误提示，若无错误则为 0 ***注意：若这个值不为 0，则证明 AS 拒绝发放 ticket，此时其他参数都为 null。在此情况下可以通过查询 errorCode 来确定错误类型。***	getErrorCode()

当获取 AS 的正确回应（errorCode==0）以后，需要进行解密才能读取出其中的数据。解密的过程在此类中进行了封装，只需调用一次即可。

public boolean **unSeal**(String key)
把整个包进行解密，只需调用一次即可解密所有变量
参数 *key* 密钥 返回值 若解密成功则返回 true，否则请检查 key 是否正确。

3.4 Models.AuthenticatorC

产生一个本机的 Authenticator，并进行加密封装。

public **AuthenticatorC**(String username, String macaddr, String timestamp)
构造函数，创建一个 Authenticator
参数 *username* 用户名 *macaddr* 本机的 MAC 地址 *timestamp* 此 authenticator 的创建时间，一般使用 KerberosClient.getTimeStamp()生成 返回值 一个未加密的 Authenticator

public void **seal**(String key)
加密整个 Authenticator，调用一次即可
参数 *key* 密钥 返回值 无

3.5 Models.IncidentMSGs

Incident 中发布的公告消息，读取自 Service 2。比如

Alert
A hurricane attacks this area, please remain indoor for further instruction.

此消息通过 Service 2 发布，也有其不同的权限要求。消息在传输过程中通过 Session Key 进行加密。

成员变量：

变量名	说明	读取方法
id	消息的 id	getId()
title	此消息的标题	getTitle()
message	此消息的内容	getMessage()
secretLevel	此消息的权限要求	getSecretLevel()
str_开头的变量	加密状态时的参数	没必要读取

成员函数：

public boolean **decrypt**(String key)
解密这个消息
参数 *key* 密钥 返回值 解密成功则返回 true，否则请检查 key

3.6 Models.IncidentPOIs

Incident 中 Victim、Responder 等兴趣点的信息，读取自 Service 1，也有权限要求，也进行了加密。

成员变量：

变量名	说明	读取方法
id	此 POI 的 id	getId()
type	此 POI 的类型，可参考此类的常量	getType()
name	此 POI 的名字	getName()
location	此 POI 的位置	getLocation()
secretLevel	查询此 POI 的权限要求	getSecretLevel()
str_开头的变量	加密状态时的参数	没必要读取

成员函数：

public boolean **decrypt**(String key)
解密这个 POI 的内容
参数 *key* 密钥 返回值 解密成功则返回 true，否则请检查 key

3.7 Models.ServiceTicket

从 TGS 获得的连接 Service 的 ticket，需要保存在内存中并在连接 Service 的时候作为参数发送给服务器。

3.8 Models.TGSResponse

同 ASResponse，从 TGS 返回的数据。成员变量：

变量名	说明	读取方法
key_c_v	服务器返回的 Session Key	getKey_c_v()
servicename	所要连接的 Service 的名字	getServicename()
timestamp	此消息产生的时间	getTimestamp()
ticket	获得的 Service Ticket	getTicket()
errorCode	返回的错误提示，若无错误则为 0 ***注意：若这个值不为 0，则证明 TGS 拒绝发放 ticket，此时其他参数都为 null。在此情况下可以通过查询 errorCode 来确定错误类型。***	getErrorCode()

成员函数：

public boolean **unSeal**(String key)
把整个包进行解密，只需调用一次即可解密所有变量
参数 *key* 密钥 返回值 若解密成功则返回 true，否则请检查 key 是否正确。

3.9 Models.TGSTicket

从 AS 获得的连接 TGS 的 ticket，需要保存在内存中并在连接 TGS 的时候作为参数发送给服务器。

3.10 HTTP.KerberosConstants

连接服务器所需要的一些常量。

常量名	值	说明
TGS_NAME	“TGS”	TGS 服务器的名字
SERVICE_NAME_1	“Service_1”	Service 1 服务器(POI)的名字
SERVICE_NAME_2	“Service_2”	Service 2 服务器(公告)的名字

3.11 Example

```
KerberosClient client = new KerberosClient();
ASResponse response_AS = client.getTGSTicketFromServer(username, tgsname, mac);  //Connect AS
response_AS.unSeal(key);          //Unseal
TGSTicket ticket_tgs = response_AS.getTicket();          //Get the TGS Ticket
```