Kerberos Server API V2.0 资料片

Kerberos 的远征

说明文档

1. 更新介绍

1.1 加入了 Incident 来区分不同的事件

Incident 代表了一个独立的事件，其成员包括 Incident ID，Incident name，Incident detail 和安全级别。Incident ID 代表一个事件的独立 ID，所以即使所有的数据都保存于服务器的同一个表中，当用户查询时也仅会返回当前 Incident ID 的结果。而 Incident 的数据保存于另外一个独立的表中，本次 API 更新中加入了 Incident 的查询和创建功能。

1.2 加入了不同安全等级功能，每一个 Incident 都可设置安全等级

安全等级（Security Level）是本次更新最大的一个改动，它决定了事件的加密机制。当用户创建一个 Incident 时同时必须指定其安全等级，之后将不允许进行更改。安全等级以及其相应的功能如表所示：

Level	访问控制	传输加密	数据库保护
0	用户需要提供用户名便可直接请求数据，服务器仍然会根据用户的权限进行结果的筛选	明文传输	明文保存
1	用户需要提供用户名和密码，并且在密码正确的情况下才能继续进行通信。提供权限管理功能	通过 Kerberos 系统分配一个 Session Key，之后使用它对传输的内容进行加密	明文保存
2	用户需要提供用户名和密码，并且在密码正确的情况下才能继续进行通信。提供权限管理功能	通过 Kerberos 系统分配一个 Session Key，之后使用它对传输的内容进行加密；同时 TGS 会分发数据加密密钥，用于数据库加、解密。*	通过 TGS 分发的密钥进行数据的加密保存，即数据密文保存，同时加密的密钥也不会保存在服务器上。

**Level 1 和 Level 2 客户端代码一样，仅服务器端有所不同*

1.3 加入了创建 POI 和 Message 的函数

加入了函数以实现客户端创建 POI 和 Message 的能力，所有的操作也都将遵循 1.2 所述的机制：Level 0 时需提供用户名，Level 1 和 2 时需要提供 Authenticator 和 Service Ticket。

1.4 Bug Fix 和完善

修复了 *Serializable* 的 Bug，并加入了获取用户权限的函数。

1.5 函数更新

*getAllPOIsFromService_1()*和 *getAllMSGsFromService_2()*将被废除，用于替代的新函数 *getAllPOIsFromService_1V2()*和 *getAllMSGsFromService_2V2()*将包含参数 IncidentId。

2. Models & Functions *Updates*

2.1 HTTP.KerberosClient *Updates*

Public List<Incident> **getAllIncidents**()
得到所有 Incident
参数 *无* 返回值 一个 Incident 的列表，包含了服务器中所有的 Incident

Public boolean **createIncident**(String incidentName, String detail, int securityLevel)
创建一个 Incident，incidentId 将在服务器生成，故创建以后需要重新获取 Incident 列表
参数 *incidentName* 事件名 *detail* 附属说明，可为 NULL *securityLevel* 安全等级，0 到 2 返回值 是否建立成功

Public void **createNewPOI**(int incidentId, ServiceTicket ticket, AuthenticatorC auth, IncidentPOIs poi)
建立一个新的 POI
参数 *incidentId* 当前事件的 ID *ticket* 从 TGS 获得的 Service Ticket，**若当前事件的安全等级为 0 则为 NULL** *auth* Authenticator，在安全等级为 1 和 2 时需要生成并进行加密；安全等级为 0 时**也需生成**，用于将用户名传递到服务器，且只需在 Authenticator 中**指定用户名即可，**且**不需要加密。** *poi* 将要新建的一个 IncidentPOIs 对象 返回值 无

Public void **createNewMSG**(int incidentId, ServiceTicket ticket, AuthenticatorC auth, IncidentMSGs msg)
建立一个新的 MSG
参数 *incidentId* 当前事件的 ID *ticket* 从 TGS 获得的 Service Ticket，**若当前事件的安全等级为 0 则为 NULL** *auth* Authenticator，在安全等级为 1 和 2 时需要生成并进行加密；安全等级为 0 时**也需生成**，用于将用户名传递到服务器，且只需在 Authenticator 中**指定用户名即可，**且**不需要加密。** *msg* 将要新建的一个 IncidentMSGs 对象

返回值
无

Public List<IncidentPOIs> **getAllPOIsFromService_1V2**(int incidentId, ServiceTicket ticket, AuthenticatorC auth)	
从 Service 1 获得所有的 POI 信息（V2.0，用于替代原函数）	
参数	
incidentId	当前事件的 ID
ticket	从 TGS 获得的 Service Ticket，**若当前事件的安全等级为 0 则为 NULL**
auth	Authenticator，在安全等级为 1 和 2 时需要生成并进行加密；安全等级为 0 时**也需生成**，用于将用户名传递到服务器，且只需在 Authenticator 中**指定用户名即可，**且**不需要加密。**
返回值 若验证成功，则获得所有的 POI，否则为 null。	

Public List<IncidentMSGs> **getAllMSGsFromService_2V2**(int incidentId , ServiceTicket ticket, AuthenticatorC auth)	
从 Service 2 获得所有的 Message 信息（V2.0，用于替代原函数）	
参数	
incidentId	当前事件的 ID
ticket	从 TGS 获得的 Service Ticket，**若当前事件的安全等级为 0 则为 NULL**
auth	Authenticator，在安全等级为 1 和 2 时需要生成并进行加密；安全等级为 0 时**也需生成**，用于将用户名传递到服务器，且只需在 Authenticator 中**指定用户名即可，**且**不需要加密。**
返回值 若验证成功，则获得所有的消息，否则为 null。	

废除函数
Public List<IncidentPOIs> ***getAllPOIsFromService_1****(ServiceTicket ticket, AuthenticatorC auth)*
Use **getAllPOIsFromService_1V2()** *instead.*

废除函数
Public List<IncidentMSGs> ***getAllMSGsFromService_2****(ServiceTicket ticket, AuthenticatorC auth)*
Use **getAllMSGsFromService_2V2()** *instead.*

2.2 Models.Incident

代表了一个独立的事件，用于创建新的事件或者得到当前事件。

变量名	类型	说明	支持方法
incidentId	int	事件的 ID，服务器端创建	{ Get; Set; }
incidentName	String	事件的名称	{ Get; Set; }

incidentDetail	String	事件的描述，可为 NULL	{ Get; Set; }
securityLevel	int	当前事件的安全等级，仅在创建时能赋值，范围 0、1、2	{ Get; Set; }

2.3 Models.IncidentMSGs *Updates*

V2.0 支持创建 MSG 和 POI，而在安全等级 1 和 2 的情况下也需要用 Session Key 对将要发送的数据进行加密。API 中提供了加密函数，在调用 **createNewMSG()**之前（Level 1 和 2）需使用它对数据进行加密。

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

2.4 Models.IncidentPOIs *Updates*

由于==V2.0 支持创建 MSG 和 POI，而在安全等级 1 和 2 的情况下也需要用 Session Key 对将要发送的数据进行加密。API 中提供了加密函数，在调用 **createNewPOI()**之前（Level 1 和 2）需使用它对数据进行加密。

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

2.5 Models.ASResponse *Updates*

新增成员变量

int **ASResponse.authority**

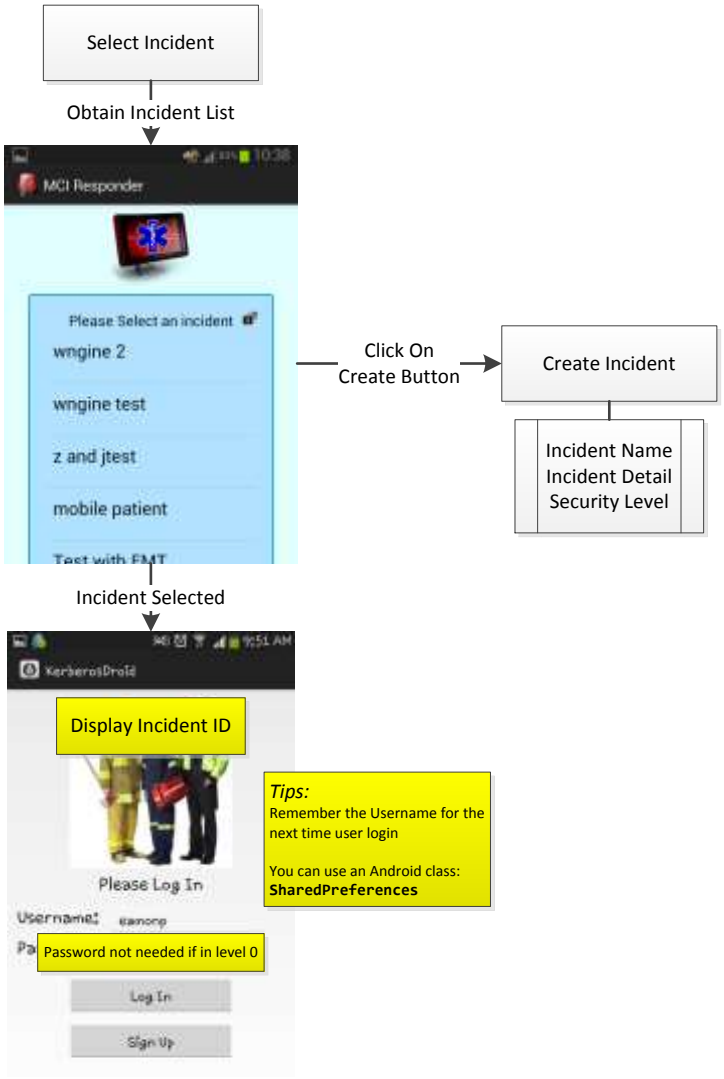
描述：用户的权限值

获取方法：getAuthority()

注意：需要在 ASResponse 解密调用上述函数以后才能得到

3. 2.0 时期的任务

3.1 加入选择、创建 Incident 的界面（主线任务）



3.2 将 API 升级至 V2.0（主线任务）

能够支持不同 Incident、不同 Security Level 的数据读取。

3.3 实现发送 POI 和 MSG 的功能（支线任务）

能够创建新的 POI 和 MSG，同时使它们支持 V2.0 的所有 Security Levels。

3.4 实现 Kerberos 过程的显示

等你完成前面的再说…