PMS 中间件接口程序员手册

目录

1 说明

权限管理系统（PMS）可以有效的控制和管理各种应用的权限，但是针对每一个不同的应用都需要做一定的客户化开发，才能与 PMS 无逢的结合。中间件就是各种不同的应用和 PMS 进行衔接的桥梁。通过中间件，应用系统可以得到 PMS 对不同系统用户所签发的权限列表，从而为系统的访问控制提供了判断的依据。

2 JitAcComp 类

2.1 setPKICertificate

定义：

public void setPKICertificate(X509Certificate gaX509Cert)

功能：

设置登录人身份证书

参数：

X509Certificate gaX509Cert 参数为 java.security.cert.X509Certificate 格式

X509Certificate 类是 JDK 对 X509 身份证书的一种实现方式，具体如何创建及使用此对象类请参见 JDK 文档。

注释：

身份证书的格式必须正确。

2.2 setPrivilegeSetType

定义：

public void setPrivilegeSetType(int privilegetype)

功能：

设置获得权限列表的类型

参数：

int privilegetype 获得权限列表类型

权限列表分三种类型：

第一种：自主权限列表。

第二种：公共角色权限列表；

第三种：自定义角色权限列表。

参数设置为－1　　返回第一种权限列表。

参数设置为 1　　判断第二种权限列表是否存在，存在则返回，否则返回第三种权限列表。

参数设置为 0　　判断第二种权限列表是否存在，如果存在返回第一、第二种权限列表的组合，否则返回第一、第三种权限的组合。

参数设置为 2　　返回所有的权限

注释：

如果设置－1、0、1、2 以外的值将不会获得权限列表。

2.3 setBaseDN

定义：

public void setBaseDN(String baseDN)

功能：

设置所查询 LDAP 的基 DN 值。

参数：

String baseDN LDAP 的基 DN 值。

基 DN 形式大致如下：某应用的应用码为 123456 则此处为 cn=123456,c=cn

注释：

错误的基 DN 会导致无法返回正确的权限列表。

2.4 setAuditParameter

定义：

public void setAuditParameter(String auditIP,String auditPort) throws ParameterException

功能：

设置行为审计服务器的 IP 地址和端口号

参数：

String auditIP 行为审计服务器的 IP 地址。

String auditPort 行为审计服务器的端口号。

注释：

如果参数错误将会导致无法进行正确的行为审计

2.5 setAuditMethod

定义：

public void setAuditMethod(String methodName)

功能：

设置行为审计的审计方法

参数：

String methodName 行为审计方法名，支持以下两种方法名，udp,http

注释：

如果输入的方法名不是 udp 或 http 而是其它非法的字符串，则系统默认采用 udp 方法进行审计。

2.6 setDownLoadInterval

定义：public void setDownLoadInterval(long interval)

功能：设定从 ldap 上下载 crl 的时间间隔

参数：long interval 下载的时间间隔，以毫秒为单位

注释：如果不调用此方法来设置时间间隔，默认的下载周期为 CA 发布 CRL 的周期。

2.7 setdefualtTime

定义：public void setdefualtTime(long userTime)

功能：设定从 ldap 上下载 xml 匹配规则的时间间隔

参数：long userTime 下载的时间间隔，以毫秒为单位

注释：如果不调用此方法来设置时间间隔，默认为 1 小时下载一次 xml 匹配规则

2.8 setParameter

定义：public void setParameter(String ip, String port) throws ParameterException, PKILDAPException

功能：设定 PKI 目录服务器的参数

参数：String ip　　目录服务器 IP

String port　　目录服务器 port

2.9 setClientIP

定义：public void setClientIP(String clientIp)

功能：设定用户的 IP 地址

参数：String clientIp　用户的 IP 地址

2.10 isCheckCertPath

定义：public void isCheckCertPath(boolean status)

功能：此为开关方法，确定是否进行证书链的验证，默认情况下不验证。

参数：boolean status　　true 为验证，false 为不验证

注释：如果此方法设为 true, 则要进行证书链的验证，在进行证书链的验证时，有一个前提条件，请参阅<<中间件测试程序使用说明.doc>>文档的前提条件.

2.11 isCheckCRL

定义：public void isCheckCRL (boolean status)

功能：此为开关方法，确定是否进行 CRL 的验证，默认情况下不验证

参数：boolean status　　true 为验证，false 为不验证

注释：如果此方法设为 true, 则要进行证书 CRL 的验证，在进行证书 CRL 的验证时，有一个前提条件，请参阅<<中间件测试程序使用说明.doc>>文档的前提条件.

2.12 getPrivilegeList

定义：

public String getPrivilegeList(String ip,String port,String appCode,String localCode)throws LDAPException,GACertParseException,GACertCRLException,GACertSignException,GACertTimeException,Exception

功能：

获取用户的权限列表

参数：

String ip　　属性证书所在 LDAP 的 IP 地址或地址列表。
例如：
171.16.1.215（只有一个 LDAP 服务器的情况）
172.16.1.215,172.16.1.216（同时有多个 LDAP 的情况）

String port　　访问的端口。
例如 ：
389（只有一个 LDAP 的情况）
389,390（同时有多个 LDAP 的情况，但必须与 IP 设定顺序符合）

String appCode　　应用码

String localCode　本地码

注释：

调用此方法可以获取到相应的权限列表，但在调用之前必须先设置以上各项值，设置的值对于本方法返回的结果会有很大影响。返回的权限列表是 String 字符串的形式。各个权限之间以“:”分割，例如：“权限一:权限二:权限三”。

3 JitCertVerify 类

3.1 SetParameter

定义： public void setParameter(String ip, String port) throws ParameterException,PKILDAPException

功能： 设定 PKI 目录服务器的参数，如果设置多个 ip 和 port，则多个 ip 地址间要以”,” 分隔，多个 port 间也要以”,” 分隔。

参数： String ip　　目录服务器 IP
String port　　目录服务器 port

3.2 setBaseDN

定义：

public void setBaseDN(String baseDN)

功能：

设置所查询 LDAP 的基 DN 值。

参数：

String　baseDN　LDAP 的基 DN 值。

3.3 Verify

定义： public void verify(X509Certificate pkiCert, boolean isCheckCrl, boolean isCheckChain) throws LDAPException, GACertTimeException, GACertSignException, GACertCRLException, Exception

功能： 验证证书的有效性，如果证书验证有效，则不返回任何值，如果证书验证无效，则弹出相关的异常。

参数： X509Certificate pkiCert　　待验证的证书
boolean isCheckCrl　　是否做 CRL 验证，true 为验证，false 为不验证
boolean isCheckChain　　是否做证书链验证，true 为验证，false 为不验证

4 异常定义

4.1 PMILDAPException

说明：PMI 目录服务器操作异常
详细描述：

- 无法连接 PKI 目录服务器
- 下载 XML 匹配规则出错
- 下载属性证书出错

4.2 PKILDAPException

说明：PKI 目录服务器操作异常

详细描述：

- 无法连接 PKI 的目录服务器
- 下载 CRL 列表出错
- 下载二级根证出错

4.3 ParameterException

说明：方法参数异常

详细描述：

方法参数错误，参数为""或 null

4.4 GACertParseException

说明：公安证书解析异常

详细描述：

- 无法正确解析证书
- 证书不符合公安规范
- 没有符合证书条件的匹配结果
- 生成公安角色匹配实例出错
- ACParser 解析属性证书出错

4.5 GAIOException

说明：读取本地文件异常

详细描述：

- 无法正确读取本地缓存文件
- 根证书解析出错
- 无法读取到根证书

4.6 GAAuditException

说明：发送审计信息异常

4.7 GAPrivilegeExcepion

说明：获取该用户类型的权限出错

4.8 GACertTimeException

说明：证书过期

4.9 GACertSignException

说明：证书签名无效

4.10 GACertCRLException

说明：证书被注销

5 JAVA 版例子程序

5.1 JitAcComp 类调用示例

```
import java.security.cert.*;
import java.io.*;
import java.util.*;
```

```
import java.security.KeyStore;

public class TestCrlCheck{
   X509Certificate cert = null;
   boolean status = false;
   private TestCrlCheck() {
      try {
         InputStream is = new FileInputStream("PA 审核.cer");
         CertificateFactory cf = CertificateFactory.getInstance("X.509");
         cert = (X509Certificate) cf.generateCertificate(is);
      }
      catch (CertificateException ex) {
      }
      catch (FileNotFoundException ex) {
      }
   }

   public void verify() throws Exception {
      JitAcComp jit = new JitAcComp();
      jit.setPKICertificate(cert);
      jit.setParameter("172.16.8.147", "389");
      jit.isCheckCRL(true); //不设定则不做 Crl 的验证
      jit.isCheckCertPath(true);//不设定则不做证书链的验证
      jit.setBaseDN("c=cn");
      jit.setAuditParameter("172.16.1.12", "3000");不设定则不执行审计操作
      jit.setAuditMethod("udp");可以设定 udp 和 http 两种字符串
      jit.setClientIP("127.0.0.1"); 设定用户的 IP 地址
      String code = jit.getPrivilegeList("172.16.8.63", "389", "23000", "111111");
      System.out.println(code);
   }

   public static void main(String[] args) throws Exception {
      TestCrlCheck test = new TestCrlCheck();
      test.verify();
   }
      }
```

5.2 JitCertVerify 类调用示例

```
   import java.security.cert.*;
   import com.jit.attr.*;
   import java.security.cert.*;
   import java.io.*;

   public class TestCertVerify {
```

```
    public static void main(String[] args){
        X509Certificate cert = null;
        try {
            InputStream is = new FileInputStream("ZhuXiaoZqTest.cer");
            CertificateFactory cf = CertificateFactory.getInstance("X.509");
            cert = (X509Certificate) cf.generateCertificate(is);

            jitCertVerify ver = new jitCertVerify();
            ver.setParameter("172.16.8.147","389");//单 ip、port 情况
            //ver.setParameter("172.16.8.147,127.0.0.1","389,389");多 ip、port 情况
            ver.verify(cert,false,true);
        }
        catch (Exception ex) {
            System.out.println(ex.getMessage());
        }
    }
}
```