GLOBAL 2007

Обучение Java

Тема **Безопасность в JAVA**

Часть **Подпись апплетов**

Автор **ASKIL (omendba@gmail.com)**

1.03.2007

Одно из самых важных применений технологии аутентификации состоит в подписании выполняемых файлов. Копируя программу из сети, пользователь вполне обоснованно может потребовать гарантий ее подлинности, поскольку такая программа может нанести вред, если она заражена каким-то вирусом. Поэтому необходимо быть уверенным в том, что программа выслана надежным источником и во время пересылки не была изменена. Если программа написана на языке программирования Java, то, обладая такой информацией, можно принять решение о делегировании ей соответствующих полномочий. Программу можно запустить в "песочнице" как обычный апплет, либо предоставить ей дополнительные права и наложить ограничения. Например, если скопированная из сети программа предназначена для редактирования текста, то ей можно разрешить обращаться к принтеру и к файлам из определенного каталога. Этой программе можно запретить доступ к сетевым соединениям, чтобы она не могла без разрешения рассылать ваши файлы третьим лицам.

Для реализации такой сложной схемы нужно выполнить следующее.

1. С помощью средств аутентификации проверить источник, из которого получен код.

2. Запустить программу с соответствующей политикой защиты, которая определяет права программы.

Апплеты распространяются в зависимости от задач:

1. Доставка в пределах корпоративной сети intranet.

2. Доставка по глобальной сети.

Согласно первому сценарию, системный администратор инсталлирует сертификаты и файлы политики на локальных компьютерах. Когда модуль Java Plug-in загружает подписанный код, он обращается к хранилищу ключей и файлу политики за получением полномочий. Установка сертификатов и политик не составляет труда и проводится однократно на каждом компьютере. После этого любой пользователь сети может запустить подписанный корпоративный апплет вне песочницы. Каждый новый или измененный старый апплет нужно подписать и доставить на Web-сервер, но этот процесс никак не влияет на работу пользовательских систем. Таким образом, данный сценарий очень удобен для доставки компонентов корпоративных приложений на компьютеры внутренней сети.

По второму сценарию поставщики программного обеспечения получают сертификаты, подписанные такими компаниями, как, например, Verisign. Если пользователь посещает Web-страницу, содержащую подписанный апплет, то на экране появляется диалоговое окно, которое идентифицирует поставщика программы и предоставляет пользователю два варианта продолжения работы: загрузить апплет с полным набором привилегий или ограничить его работу песочницей.

Рассмотрим текст апплета, который мы будем исследовать.

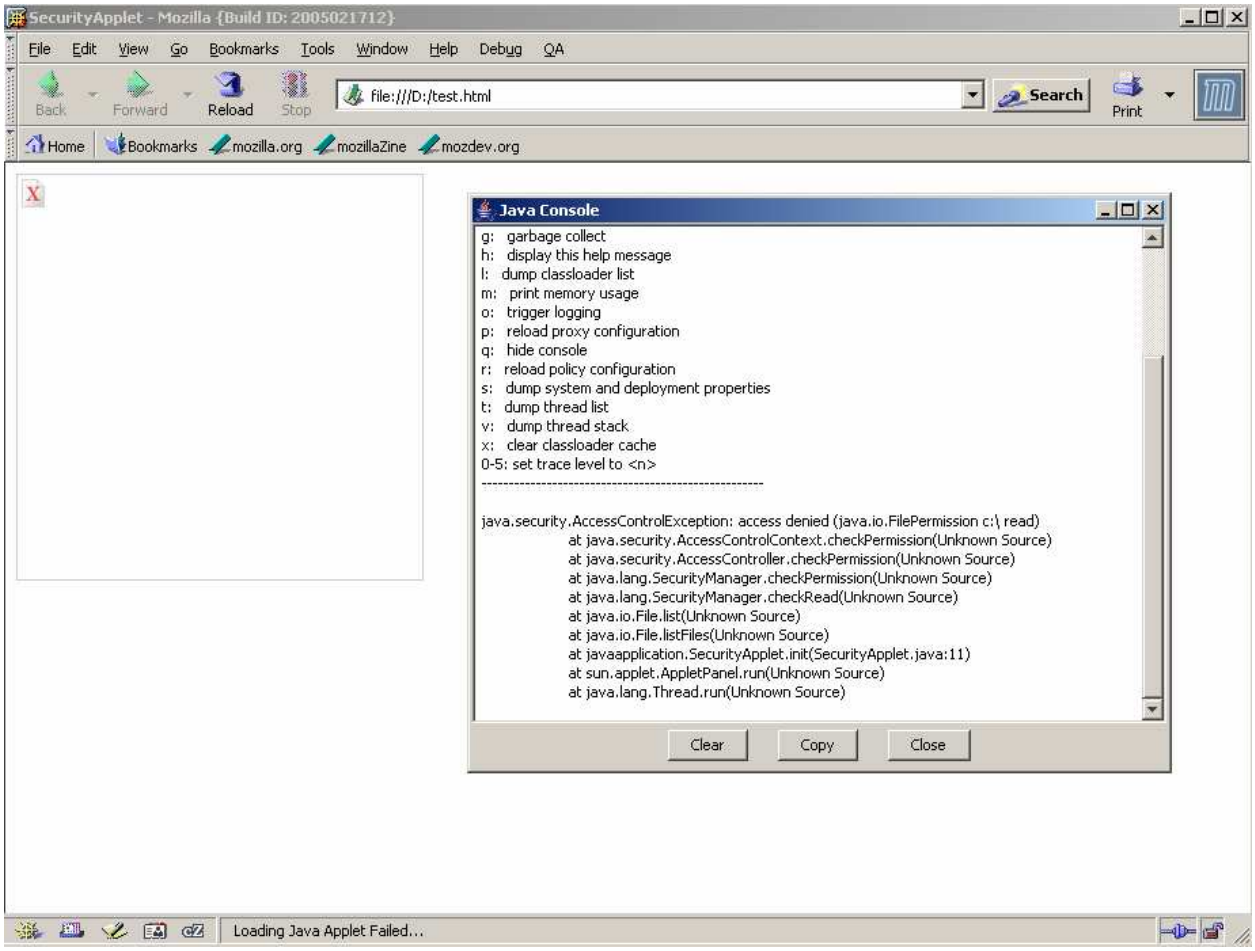
```
package javaapplication;
import java.io.*;
import javax.swing.*;
public class SecurityApplet extends JApplet
{
    public void init()
    {
     JTextArea text = new JTextArea();
     getContentPane().add("Center", new JScrollPane(text));
     File dir = new File("c:/");
     File [] file = dir.listFiles();
     for (int i=0;i<file.length;i++)
     {
         text.append(file[i].getName());
         text.append("\n");
     }
   }
}
```

Данному апплету, по крайней мере, нужны права для чтения локальных файлов. Html-код файла test.html для вызова апплета:

```
<html><head><title>SecurityApplet</title></head>
<body>
<applet code="javaapplication/SecurityApplet" archive="JavaApplication.jar"
        WIDTH="300" HEIGHT="300">
</applet>
</body>
</html>
```

При этом файл JavaApplication.jar должен находиться в той же папке что и test.html. В нашем случае оба файла находятся на диске D.

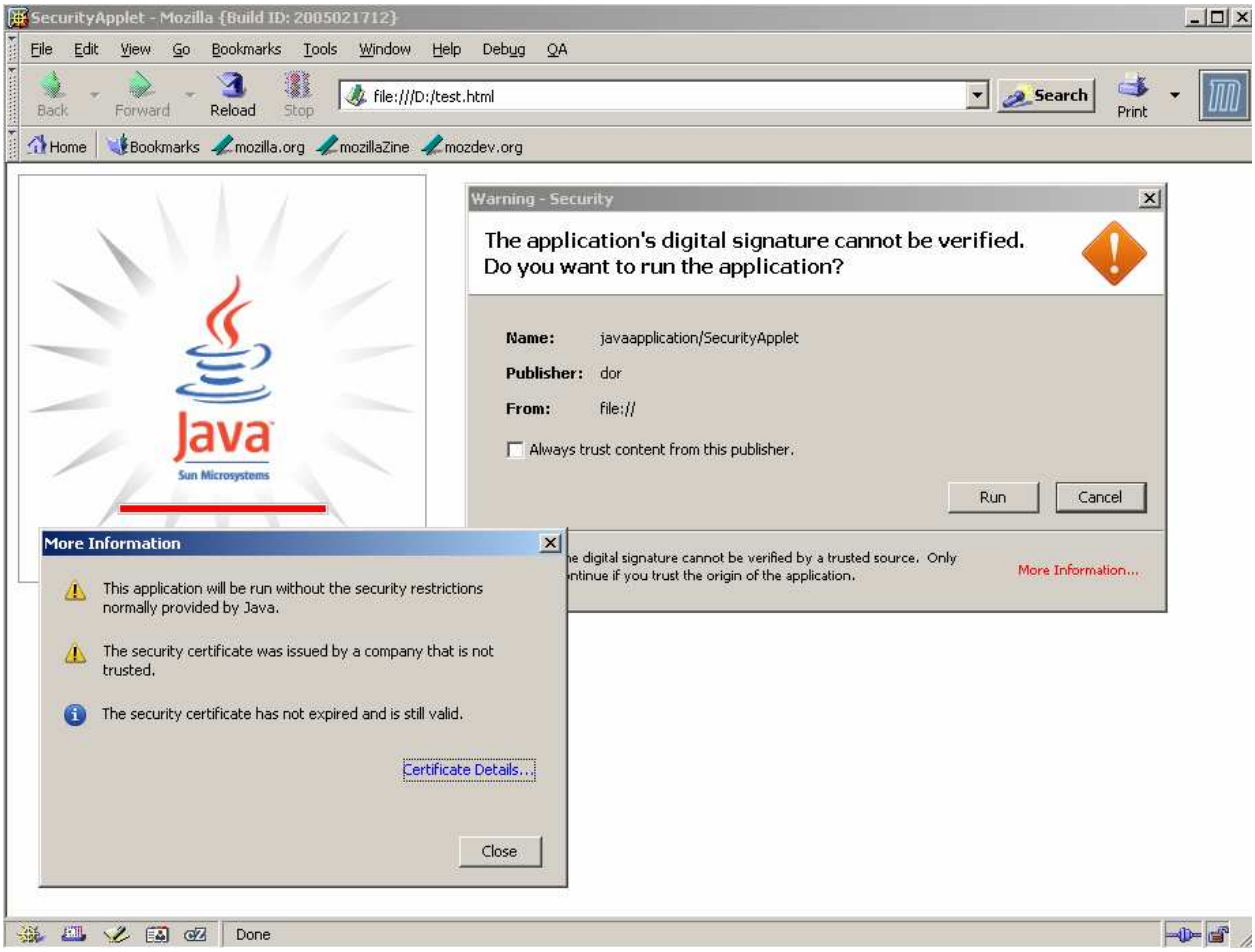
При первом запуске приложения мы увидим следующее окно:

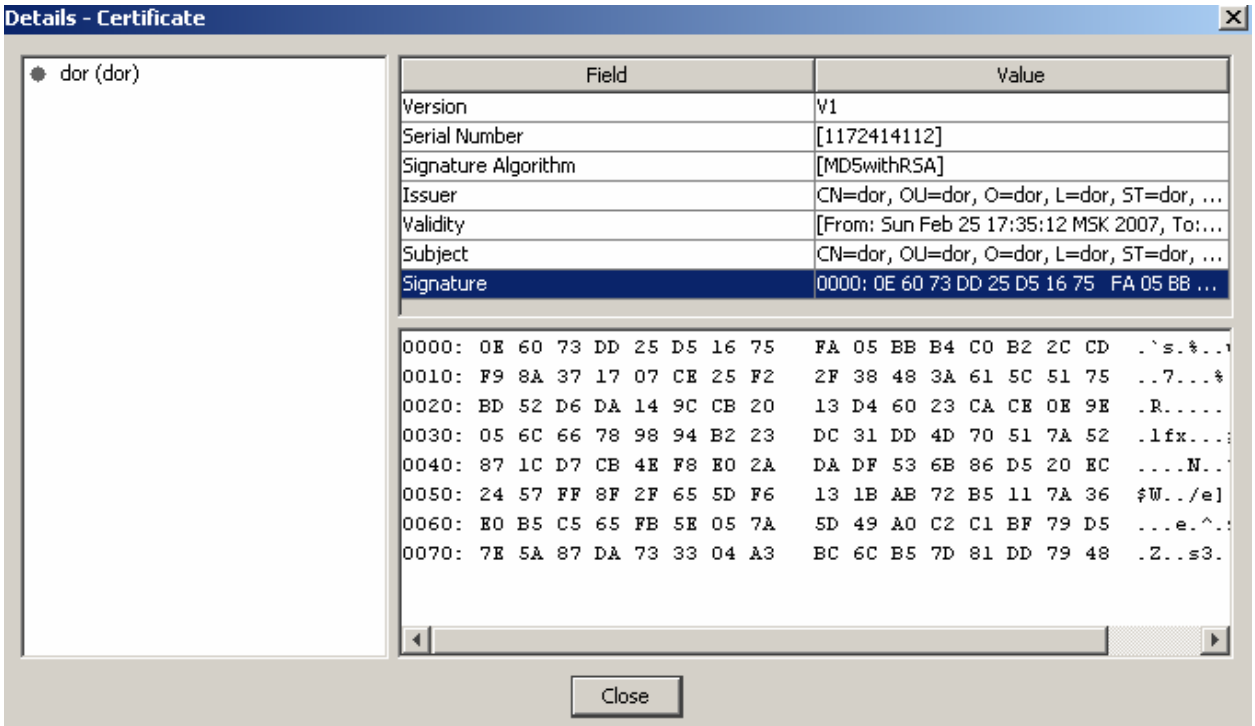


После этого подпишем апплет цифровой подписью:

```
jarsigner –keystore C:\DkeyStore D:\JavaApplication.jar DorAlias
Enter Passphrase for keystore: PASSWORD1
```

Вновь запускаем и видим следующее:





Жмем “Cancel” в окне “Warning - Security”. Апплет не выполнится – прав недостаточно. Если нажать “Run”, то мы соглашаемся с принятием сертификата “dor” и автоматически предоставляем апплету неограниченные возможности. Этого делать нельзя.

Данную проблему можно разрешить, настроив файл политики. Перейдем в каталог “C:\Documents and Settings\EZEN\Application Data\Sun\Java\Deployment”. Здесь находится файл deployment.properties. Добавим к нему строку:

```
#deployment.properties
#Wed Feb 28 17:13:22 MSK 2007
deployment.version=1.5.0
deployment.capture.mime.types=true
deployment.browser.path=C\:\\Program Files\\Internet Explorer\\iexplore.exe
#Java Web Start jre's
#Wed Feb 28 17:13:22 MSK 2007
#Java Plugin jre's
#Wed Feb 28 17:13:22 MSK 2007
deployment.javapi.jre.1.5.0_09.args=
deployment.javapi.jre.1.5.0_09.osname=Windows
deployment.javapi.jre.1.5.0_09.osarch=x86
deployment.javapi.jre.1.5.0_09.path=C\:\\Program Files\\Java\\jre1.5.0_09

deployment.user.security.policy=file:/c:/test.policy
```

Файл политики “C:/test.policy” имеет следующее содержание:

```
keystore "file:/c:/DKeyStore", "JKS";
grant signedBy "DorAlias"
{
  permission java.io.FilePermission "<<ALL FILES>>", "read";
};
```

Теперь при нажатии “Cancel” мы увидим работающий апплет, работающий с правами, определенными в файле политики: