Университет ИТМО, кафедра ВТ

Лабораторная работа №2 по “Системному Программному Обеспечению”

Работу выполнил

студент группы P3200

Рогов Я. С.

Преподаватели:

Дергачёв А.М.

Жмылёв С.А.

Санкт-Петербург, 2016

Задание

1. На техническом русском языке своими словами (не перевод!) написать инструкцию по работе с утилитой, выданной в качестве варианта
2. Подготовить размеченный для форматирования командами troff и nroff текст электронного руководства по данной инструкции
3. Создать в домашнем каталоге необходимую для работы команды man иерархию подкаталогов и разместить в ней подготовленный файл
4. В стартовый командный файл .profile внести необходимые изменения для работы команды man с подготовленным руководством
5. Продемонстрировать работу команды `man вариант'

1. Инструкция по работе с утилитой

jarsigner - создание подписей и их проверка для архивов формата JAR
Создаёт подписи для Java-архивов, JAR, и проверяет правильность подписей подписанных JAR-файлов.
Использование
jarsigner [опции] путь_к_jar_файлу псевдоним_пользователя - подпись архивов
jarsigner -verify [опции] путь_к_jar_файлу - проверка правильности подписи
Описание
Программа jarsigner может использоваться для следующих целей:
1. Подпись Java-архивов, JAR;
2. Проверка подписи у JAR файла, а также его, архива, целостности
JAR файлы предоставляют удобный способ распространения и использования Java-приложений. Для их создания используется программа jar.
Для подписи архивов необходимо иметь пару приватный-публичный ключи.
1. Приватный ключ используется для непосредственной подписи архива. Необходимо хранить в секрете.
2. Публичный ключ используется для верификации архива, т.е. проверки правильности подписи у архива.
Также необходимо иметь не менее одного сертификата, подтверждающего истинность публичного ключа, т.е. подтверждающего, что конкретный публичный ключ принадлежит конкретному пользователю.
Цифровая подпись архива это последовательность бит, которая вычисляется на основе содержимого архива (подобно контрольной сумме) и приватного ключа пользователя.
Помимо подписи, подписанный архив может содержать сертификатq, способный подтвердить правильность публичного ключа.
Подпись выполняет следующие функции:
* Можно проверить её правильность с помощью публичного ключа пользователя, подписавшего архив. Если проверка пройдёт неудачно, это будет означать, что архив - либо подделка, либо повреждён.
* Подпись архива невозможно подделать при условии сохранения приватного ключа в секрете
* Подпись есть результат работы функции, использующей данные архива, а потому она не может быть одинаковой для двух разных архивов.
* Подписанный архив невозможно изменить, т.к. в таком случае архив не будет проходить проверку подписи
jarsigner использует "keystore" - базу данных приватных ключей и сертификатов, подтверждающий соответствующие публичные ключи. Хранилище защищено паролем. Утилита keytool позволяет управлять этими базами данных.
Т.к. JAR - ZIP архив с дополнительным файлом мета-данных (META-INF/MANIFEST.MF), jarsigner может работать как с JAR файлами, так и с ZIP архивами.
Псевдонимы
Данные в keystore доступны по псевдонимам (поле псевдоним_пользователя). Каждой паре ключей соответствует владелец, псевдоним которого и нужно указать при подпси архива. Смотри ключи -keystore и -storepass для работы с хранилищами ключе.
Алгоритмы
По-умолчанию, jarsigner подписывает архив с помощью одной из следующих пар алгоритмов:
* DSA (Digital Signature Algorithm, "Алгоритм Цифровой Подписи") для создания подписи и SHA-1 для вычисления контрольных сумм
* RSA для создания подписи и MD5 для вычисления контрольных сумм
Выбор происходит в зависимости от алгоритма, использовавшегося для генерации ключей пользователя. Если использовался DSA, то используется связка DSA + SHA-1, если RSA - RSA + MD5.
Подписанный JAR файл
Основные отличие подписанного архива от обычного архива:
* файл подписи расширения .SF в директории META-INF

* файл блока подписи расширения .DSA в директории META-INF
Файл подписи .SF
Содержит:
1. Контрольную сумму файла-манифеста
2. Информацию о каждом файле в архиве в следующей форме:
путь алгоритм_хэширования хэш
(Хэширование - вычисление контрольной суммы.) Здесь хэш - контрольная сумма трёх строк из файла-манифеста для данного файла архива.
.SF-файл является подписаным, а реальная подпись хранится в файле .DSA.
Файл блока подписи .DSA
Содержит подпись, а также сертификаты/цепочку сертификатов из хранилища keystore, позволяющих удостовериться в подлинности подписи.

Проверка подписи JAR файла

Подпись JAR файла проходит проверку, если подпись подлинна, а также никакие из файлов архива не были изменены или удалены. Проверка происходит поэтапно:

1. Проверка подписи .SF файла
Происходит проверка целостности и подлинности .SF файла на основе данных, хранящихся в файле .DSA.
2. Проверка подлинности и целостности файла-манифеста
Происходит проверка записей о файлах архива в файле-манифесте на основе сравнения с контрольной суммой каждой такой записи, записанной в .SF файле.
3. Проверка подлинности и целостности файлов архива
Происходит проверка контрольных сумм файлов архива и контрольных сумм тех же файлов, записанных в файле-манифесте.

При возникновении любой серьёзной ошибки при проверке объявляется о неудачной попытке проверки, т.е. то что архив не является подлинным и/или его изменили (или он был повреждён при передаче).

Множественные подписи для JAR файлов

Также доступен механизм множественной подписи архивов. В таком случае, в директории META-INF архива будет несколько пар файлов .SF-.DSA с разными именами. Например:

USER1.SF
USER1.DSA
USER2.SF
USER2.DSA

Ключи:

-keystore путь
Указывает URL к keystore, хранилищу приватных ключей и сертификатов, необходимых для подписи. По-умолчанию используется хранилище, указанное в файле .keystore домашней директории пользователя.
Необходим только при подписи.

-storetype тип_хранилища
Указывает тип хранилища ключей. По-умолчанию используется значение свойства keystore.type из файла свойств безопасности.

-storepass пароль
Указывает пароль для доступа к хранилищу ключей. Если пароль не был передан через эту опцию, пользователю будет предложено ввести его.

-keypass пароль
Указывает пароль, используемый для защиты непосредственно приватного ключа в хранилище.

-sigfile название
Указывает название, используемое для создания файлов .SF и .DSA. Если вызываемый ключ имел вид "-sigfile MyUserName", то файлы будут называться MYUSERNAME.SF и MYUSERNAME.DSA соответственно. Разрешённые символы в поле название - буквы алфавита (любого регистра), цифры, нижнее подчёркивание и тире.
По-умолчанию названием является первые 8 символов псевдонима в верхнем регистре с заменой всех запрещённых символов на нижние подчёркивания.

-sigalg алгоритм
Указывает алгоритм, используемый при генерации подписи архива. По-умолчанию используется одна из пар алгоритмов DSA+SHA-1 или RSA+MD5 на основе типа приватного ключа, используемого для подписи.

-digestalg алгоритм
Указывает алгоритм, используемый при вычислениях контрольной суммы. По-умолчанию используется SHA-1.

-signedjar путь
Указывает путь и название уже подписанного архива. По-умолчанию происходит перезапись неподписанного архива (переданого как аргумент) подписаным.

-verify
Указывает, что архив необходимо проверить на подлинность и целостность, а не подписать. При успешной проверке выведется сообщение "jar verified" (jar подтверждён). При попытке проверить неподписанный архив или архив с подписью, созданной с неподдерживаемым

```
программой алгоритмом - "jar is unsigned. (signatures missing or not parsable)" (jar не
подписан. (подписи отсутствуют или невозможно проверить)).
       -certs
              Указывает о необходимости также выводить информацию о сертификатах для каждого
пользователя, подписавшего архив. Информация включает в себя тип сертификата и, если это
сертификат X.509, имя подписавшего архив пользователя. Используется при указании ключей
-verify и/или -verbose.
       -verbose
              Включает "подробный режим" подписи или проверки, т.е. вывод дополнительной
информации о совершаемом процессе.
Примеры использования
       Подпись архива
              Подпись архива с названием archive.jar, используя ключ пользователя alisa,
защищённого паролем "bob", хранимого в хранилище по пути /working/mystore с паролем от
хранилища "mypass" и сохранением подписанного архива под названием signedarchive.jar
              jarsigner -keystore /working/mystore -storepass mypass -keypass bob -signedjar
signedarchive.jar archive.jar alisa
       Проверка подписанного архива
              Проверка подписанного архива signedarchive.jar с выводом дополнительной
информации о ходе процесса проверки, используя данные хранилища по пути /working/mystore.
              jarsigner -keystore /working/mystore -verify -verbose signedarchive.jar
              Пример вывода при выполнении этой программы:
              198    Fri Sep 26 16:14:06 PDT 1997 META-INF/MANIFEST.MF
                     199    Fri Sep 26 16:22:10 PDT 1997 META-INF/ALISA.SF
                     1013   Fri Sep 26 16:22:10 PDT 1997 META-INF/ALISA.DSA
              smk     2752  Fri Sep 26 16:12:30 PDT 1997 SomeClass.class
              smk     849   Fri Sep 26 16:12:46 PDT 1997 AnotherClass.class
              s - подпись прошла проверку
              m - запись об этом файле содержится в файле-манифесте
              k - найден как минимум один сертификат для проверки подписи данного файла
        jar прошёл проверку.
Смотрите также:
       jar(1), keytool(1)
```

2. Краткое описание назначения команд troff и nroff

Предназначение команд nroff и troff – форматирование текстовых документов для вывода на экран или для печати на принтере. В то время как наиболее распространеная модель для текстовых редакторов с форматированием – WYSIWYG, *roff полагаются на специальные команды и "запросы" , зачастую называемые макросами. Различные макросы позволяют указать, как тот или иной фрагмент текста должен выглядеть. Имеют вид ".AB", где AB – короткая аббревиатура действия, за которое отвечает макрос (обычно длина аббревиатуры – не больше двух символов). Например, .fi включает т.н. "заполнение" (fill) – разбивку текста, следующего за командой на строки, полностью заполняющего отведённую для текста ширину. Возможно использование дополнительных препроцессоров: для создания таблиц (tbl), для отображения математических формул (eqn) и многие другие.
Основное отличие troff от nroff – возможность использования различных шрифтов.

3. Текст руководства с разметкой

```
.TH jarsigner 1

.LP
.SH НАЗВАНИЕ
.LP
jarsigner \- создание подписей и их проверка для архивов формата JAR
.LP
Создаёт подписи для Java\-архивов, JAR, и проверяет правильность подписей подписанных JAR\-
файлов.
.LP
.RS 3
.LP
.SH ИСПОЛЬЗОВАНИЕ
.LP
.nf
jarsigner [ опции ] путь_к_jar псевдоним \- подпись архива
.fi
```

.nf
jarsigner \-verify [опции] путь_к_jar \- проверка подписи
.fi
.SH ОПИСАНИЕ
.sp
.LP
Программа jarsigner может использоваться для следующих целей:
.sp
.RS 3
.TP 3
1.
Подпись Java\-архивов, JAR;
.sp
.TP 3
2.
Проверка подписи у JAR файла, а также его, архива, целостности
.RE
.sp
.LP
JAR файлы предоставляют удобный способ распространения и использования Java\-приложений. Для их создания используется программа jar.
.LP
Для подписи архивов необходимо иметь пару приватный\-публичный ключи.
.RS 3
.TP 3
1.
Приватный ключ используется для непосредственной подписи архива. Необходимо хранить в секрете.
.TP 3
2.
Публичный ключ используется для верификации архива, т.е. проверки правильности подписи у архива.
.RE
.sp
Также необходимо иметь не менее одного сертификата, подтверждающего истинность публичного ключа, т.е. подтверждающего, что конкретный публичный ключ принадлежит конкретному пользователю.
.sp
Цифровая подпись архива это последовательность бит, которая вычисляется на основе содержимого архива (подобно контрольной сумме) и приватного ключа пользователя. Помимо подписи, подписанный архив может содержать сертификата, способный подтвердить правильность публичного ключа.
.sp
Подпись выполняет следующие функции:
.RS 3
.TP 3
*
Можно проверить её правильность с помощью публичного ключа пользователя, подписавшего архив. Если проверка пройдёт неудачно, это будет означать, что архив либо подделан, либо повреждён.
.TP 3
*
Подпись архива невозможно подделать при условии сохранения приватного ключа в секрете
.TP 3
*
Подпись есть результат работы функции, использующей данные архива, а потому она не может быть одинаковой для двух разных архивов.
.TP 3
*
Подписанный архив невозможно изменить, т.к. в таком случае архив не будет проходить проверку подписи
.RE
.LP
jarsigner использует "keystore" \- базу данных приватных ключей и сертификатов, подтверждающий соответствующие публичные ключи. Хранилище защищено паролем. Утилита keytool позволяет управлять этими базами данных.
.LP
Т.к. JAR \- ZIP архив с дополнительным файлом мета\-данных (META\-INF/MANIFEST.MF), jarsigner может работать как с JAR файлами, так и с ZIP архивами.
.SS
Псевдонимы

.LP
.RS 3
Данные в keystore доступны по псевдонимам (поле "псевдоним"). Каждой паре ключей соответствует владелец, псевдоним которого и нужно указать при подпси архива. Смотри ключи \-keystore и \-storepass для работы с хранилищами ключе.
.RE
.SS
Алгоритмы
.LP
.RS 3
По\-умолчанию, jarsigner подписывает архив с помощью одной из следующих пар алгоритмов:
.RS 3
.TP 3
*
DSA (Digital Signature Algorithm, "Алгоритм Цифровой Подписи") для создания подписи и SHA\-1 для вычисления контрольных сумм
.TP 3
*
RSA для создания подписи и MD5 для вычисления контрольных сумм
.RE
.LP
Выбор происходит в зависимости от алгоритма, использовавшегося для генерации ключей пользователя. Если использовался DSA, то используется связка DSA + SHA\-1, если RSA \- RSA + MD5.
.RE
.SS
Подписанный JAR файл
.LP
.RS 3
Основные отличие подписанного архива от обычного архива:
.RS 3
.TP 3
*
файл подписи расширения .SF в директории META\-INF
.TP 3
*
файл блока подписи расширения .DSA в директории META\-INF
.RE
.SS
Файл подписи .SF
.LP
.RS 3
Содержит:
.RS 3
.TP 3
1.
Контрольную сумму файла\-манифеста
.TP 3
2.
Информацию о каждом файле в архиве в следующей форме:
.sp
.nf
путь алгоритм_хэширования хэш
.fi
.sp
(Хэширование \- вычисление контрольной суммы.) Здесь хэш \- контрольная сумма трёх строк из файла\-манифеста для данного файла архива.
.RE
.LP
.SF\-файл является подписаным, а реальная подпись хранится в файле .DSA.
.RE
.SS
Файл блока подписи .DSA
.LP
.RS 3
Содержит подпись, а также сертификаты/цепочку сертификатов из хранилища keystore, позволяющих удостовериться в подлинности подписи.
.RE
.SS
Проверка подписи JAR файла
.LP

.RS 3
Подпись JAR файла проходит проверку, если подпись подлинна, а также никакие из файлов архива не были изменены или удалены. Проверка происходит поэтапно:
.RS 3
.TP 3
1.
Проверка подписи .SF файла
.LP
Происходит проверка целостности и подлинности .SF файла на основе данных, хранящихся в файле .DSA.
.TP 3
2.
Проверка подлинности и целостности файла\-манифеста
.LP
Происходит проверка записей о файлах архива в файле\-манифесте на основе сравнения с контрольной суммой каждой такой записи, записанной в .SF файле.
.TP 3
3.
Проверка подлинности и целостности файлов архива
.LP
Происходит проверка контрольных сумм файлов архива и контрольных сумм тех же файлов, записанных в файле\-манифесте.
.RE
.LP
При возникновении любой серьёзной ошибки при проверке объявляется о неудачной попытке проверки, т.е. то что архив не является подлинным и/или его изменили (или он был повреждён при передаче).
.RE
.SS
Множественные подписи для JAR файлов
.LP
.RS 3
Также доступен механизм множественной подписи архивов. В таком случае, в директории META\-INF архива будет несколько пар файлов .SF\-.DSA с разными именами. Например:
.LP
.nf
.RS 4
USER1.SF
USER1.DSA
USER2.SF
USER2.DSA
.RE
.fi
.SH КЛЮЧИ
.RS 3
.TP 3
\-keystore путь
Указывает URL к keystore, хранилищу приватных ключей и сертификатов, необходимых для подписи. По\-умолчанию используется хранилище, указанное в файле .keystore домашней директории пользователя.
Необходим только при подписи.
.TP 3
\-storetype тип_хранилища
Указывает тип хранилища ключей. По\-умолчанию используется значение свойства keystore.type из файла свойств безопасности.
.TP 3
\-storepass пароль
Указывает пароль для доступа к хранилищу ключей. Если пароль не был передан через эту опцию, пользователю будет предложено ввести его.
.TP 3
\-keypass пароль
Указывает пароль, используемый для защиты непосредственно приватного ключа в хранилище.
.TP 3
\-sigfile название
Указывает название, используемое для создания файлов .SF и .DSA. Если вызываемый ключ имел вид \-sigfile MyUserName", то файлы будут называться MYUSERNAME.SF и MYUSERNAME.DSA соответственно. Разрешённые символы в поле название: буквы алфавита (любого регистра), цифры, нижнее подчёркивание и тире.
.sp
По\-умолчанию названием является первые 8 символов псевдонима в верхнем регистре с заменой всех запрещённых символов на нижние подчёркивания.

.TP 3
\-sigalg алгоритм
Указывает алгоритм, используемый при генерации подписи архива. По\-умолчанию используется одна из пар алгоритмов DSA+SHA\-1 или RSA+MD5 на основе типа приватного ключа, используемого для подписи.
.TP 3
\-digestalg алгоритм
Указывает алгоритм, используемый при вычислениях контрольной суммы. По\-умолчанию используется SHA\-1.
.TP 3
\-signedjar путь
Указывает путь и название уже подписанного архива. По\-умолчанию происходит перезапись неподписанного архива (переданого как аргумент) подписаным.
.TP 3
\-verify
Указывает, что архив необходимо проверить на подлинность и целостность, а не подписать. При успешной проверке выведется сообщение "jar verified" (jar подтверждён). При попытке проверить неподписанный архив или архив с подписью, созданной с неподдерживаемым программой алгоритмом \- "jar is unsigned. (signatures missing or not parsable)" (jar не подписан. (подписи отсутствуют или невозможно проверить)).
.TP 3
\-certs
Указывает о необходимости также выводить информацию о сертификатах для каждого пользователя, подписавшего архив. Информация включает в себя тип сертификата и, если это сертификат X.509, имя подписавшего архив пользователя. Используется при указании ключей \-verify и/или \-verbose.
.TP 3
\-verbose
Включает "подробный режим" подписи или проверки, т.е. вывод дополнительной информации о совершаемом процессе.
.SH ПРИМЕРЫ ИСПОЛЬОВАНИЯ
.SS
Подпись архива
.LP
.RS 3
Подпись архива с названием archv.jar, используя ключ пользователя alisa, защищённого паролем "bob", хранимого в хранилище по пути /working/mystore с паролем от хранилища "mypass" и сохранением подписанного архива под названием sarchv.jar
.nf
.LP
jarsigner \-keystore /working/mystore \-storepass mypass \-keypass bob \-signedjar \
sarchv.jar archv.jar alisa
.fi
.RE
.SS
Проверка подписанного архива
.LP
.RS 3
Проверка подписанного архива signedarchive.jar с выводом дополнительной информации о ходе процесса проверки, используя данные хранилища по пути /working/mystore.
.nf
.LP
jarsigner \-keystore /working/mystore \-verify \-verbose signedarchive.jar
.fi
.LP Пример вывода при выполнении этой программы:
.nf
.fl
          198 Fri Sep 26 16:14:06 PDT 1997 META\-INF/MANIFEST.MF
.fl
          199 Fri Sep 26 16:22:10 PDT 1997 META\-INF/ALISA.SF
.fl
         1013 Fri Sep 26 16:22:10 PDT 1997 META\-INF/ALISA.DSA
.fl
  smk   2752 Fri Sep 26 16:12:30 PDT 1997 SomeClass.class
.fl
  smk   849 Fri Sep 26 16:12:46 PDT 1997 AnotherClass.class
.fl

      s \- подпись прошла проверку
.fl
      m \- запись об этом файле содержится в файле\-манифесте

```
.fl
      k \- найден как минимум один сертификат для проверки подписи данного файла

.fl

             jar прошёл проверку.
.fi
.RE
.SH СМОТРИТЕ ТАКЖЕ
       jar(1), keytool(1)
```

4. Краткое описание каждой использованной в разметке директивы

.TH заголовок [1-9] – название описываемой команды и секция, к которой принадлежит данное описание

.LP – начать новый параграф

.SH строка – заголовок текста

.SS – подзаголовок текста

.nf, .fi – выключить или включить "заполнение текстом" соответственно.

.sp n – пропустить n строк (По-умолчанию - 1)

.RS n, .RE – начать/закончить часть с относительным отступом соответственно. Оступ увеличивается на n

.TP n – начать параграф с отступом длины n. Строка, следующая за этой командой является тэгом этого параграфа. Текст, идущей за ней – текст самого параграфа.

.fl – очистить буфер вывода

5. Внесённые в стартовый командный файл изменения:

В конец файла добавлена строка "MANPATH=~/myman/:$MANPATH"

6. Результат вывода команды man для подготовленного руководства

```
s207220@helios:/home/s207220/myman$ man jarsigner
Reformatting page.  Please Wait... done

User Commands                                          jarsigner(1)

НАЗВАНИЕ
    jarsigner - создание подписей  и  их  проверка  для  архивов
    формата JAR

    Создаёт  подписи  для  Java- архивов,   JAR,   и   проверяет
    правильность подписей подписанных JAR-файлов.

ИСПОЛЬЗОВАНИЕ
    jarsigner [ опции ] путь_к_jar псевдоним - подпись архива
    jarsigner -verify [ опции ] путь_к_jar - проверка подписи

ОПИСАНИЕ
    Программа  jarsigner  может  использоваться  для   следующих
    целей:

       1. Подпись Java-архивов, JAR;

       2. Проверка подписи у JAR файла,  а  также  его,  архива,
          целостности

    JAR файлы предоставляют  удобный  способ  распространения  и
    использования  Java-приложений. Для их создания используется
    программа jar.

    Для  подписи  архивов  необходимо  иметь  пару  приватный  -
    публичный ключи.
```

1. Приватный ключ используется для непосредственной подписи архива. Необходимо хранить в секрете.

2. Публичный ключ используется для верификации архива, т.е. проверки правильности подписи у архива.

Также необходимо иметь не менее одного сертификата, подтверждающего истинность публичного ключа, т.е. подтверждающего, что конкретный публичный ключ принадлежит конкретному пользователю.

Цифровая подпись архива это последовательность бит, которая вычисляется на основе содержимого архива (подобно контрольной сумме) и приватного ключа пользователя. Помимо подписи, подписанный архив может содержать сертификата, способный подтвердить правильность публичного ключа.

Подпись выполняет следующие функции:

*
Можно проверить её правильность с помощью публичного ключа пользователя, подписавшего архив. Если проверка

пройдёт неудачно, это будет означать, что архив либо подделан, либо повреждён.

* Подпись архива невозможно подделать при условии сохранения приватного ключа в секрете

* Подпись есть результат работы функции, использующей данные архива, а потому она не может быть одинаковой для двух разных архивов.

* Подписанный архив невозможно изменить, т.к. в таком случае архив не будет проходить проверку подписи

jarsigner использует "keystore" - базу данных приватных ключей и сертификатов, подтверждающий соответствующие публичные ключи. Хранилище защищено паролем. Утилита keytool позволяет управлять этими базами данных.

Т.к. JAR - ZIP архив с дополнительным файлом мета - данных (META-INF/MANIFEST.MF), jarsigner может работать как с JAR файлами, так и с ZIP архивами.

Псевдонимы

Данные в keystore доступны по псевдонимам (поле "псевдоним"). Каждой паре ключей соответствует владелец, псевдоним которого и нужно указать при подпси архива. Смотри ключи -keystore и -storepass для работы с хранилищами ключе.

Алгоритмы

По-умолчанию, jarsigner подписывает архив с помощью одной из следующих пар алгоритмов:

* DSA (Digital Signature Algorithm, "Алгоритм Цифровой Подписи") для создания подписи и SHA-1 для вычисления контрольных сумм

* RSA для создания подписи и MD5 для вычисления контрольных сумм

Выбор происходит в зависимости от алгоритма,

использовавшегося для генерации ключей пользователя. Если использовался DSA, то используется связка DSA + SHA-1, если RSA - RSA + MD5.

Подписанный JAR файл

Основные отличие подписанного архива от обычного архива:

* файл подписи расширения .SF в директории META-INF
* файл блока подписи расширения .DSA в директории META-INF

Файл подписи .SF

Содержит:

1. Контрольную сумму файла-манифеста
2. Информацию о каждом файле в архиве в следующей форме:

   путь алгоритм_хэширования хэш

   (Хэширование - вычисление контрольной суммы.) Здесь хэш - контрольная сумма трёх строк из файла-манифеста для данного файла архива.

Файл блока подписи .DSA

Содержит подпись, а также сертификаты/цепочку сертификатов из хранилища keystore, позволяющих удостовериться в подлинности подписи.

Проверка подписи JAR файла

Подпись JAR файла проходит проверку, если подпись подлинна, а также никакие из файлов архива не были изменены или удалены. Проверка происходит поэтапно:

1. Проверка подписи .SF файла

   Происходит проверка целостности и подлинности .SF файла на основе данных, хранящихся в файле .DSA.

2. Проверка подлинности и целостности файла-манифеста

   Происходит проверка записей о файлах архива в файле - манифесте на основе сравнения с контрольной суммой каждой такой записи, записанной в .SF файле.

3. Проверка подлинности и целостности файлов архива

   Происходит проверка контрольных сумм файлов архива и контрольных сумм тех же файлов, записанных в файле-

манифесте.

При возникновении любой серьёзной ошибки при проверке объявляется о неудачной попытке проверки, т.е. то что архив не является подлинным и/или его изменили (или он был повреждён при передаче).

Множественные подписи для JAR файлов

Также доступен механизм множественной подписи архивов. В таком случае, в директории META-INF архива будет несколько пар файлов .SF-.DSA с разными именами. Например:

```
USER1.SF
USER1.DSA
USER2.SF
USER2.DSA
```

КЛЮЧИ

-keystore путь
Указывает URL к keystore, хранилищу приватных ключей и сертификатов, необходимых для подписи. По-умолчанию используется хранилище, указанное в файле .keystore домашней директории пользователя. Необходим только при подписи.

-storetype тип_хранилища
Указывает тип хранилища ключей. По - умолчанию используется значение свойства keystore.type из файла свойств безопасности.

-storepass пароль
Указывает пароль для доступа к хранилищу ключей. Если пароль не был передан через эту опцию, пользователю будет предложено ввести его.

-keypass пароль
Указывает пароль, используемый для защиты непосредственно приватного ключа в хранилище.

-sigfile название
Указывает название, используемое для создания файлов .SF и .DSA. Если вызываемый ключ имел вид -sigfile MyUserName", то файлы будут называться MYUSERNAME.SF и MYUSERNAME.DSA соответственно. Разрешённые символы в поле название: буквы алфавита (любого регистра), цифры, нижнее подчёркивание и тире.

По-умолчанию названием является первые 8 символов псевдонима в верхнем регистре с заменой всех

запрещённых символов на нижние подчёркивания.

-sigalg алгоритм
Указывает алгоритм, используемый при генерации подписи архива. По - умолчанию используется одна из пар алгоритмов DSA+SHA-1 или RSA+MD5 на основе типа приватного ключа, используемого для подписи.

-digestalg алгоритм
Указывает алгоритм, используемый при вычислениях контрольной суммы. По-умолчанию используется SHA-1.

-signedjar путь
Указывает путь и название уже подписанного архива. По-умолчанию происходит перезапись неподписанного архива (переданого как аргумент) подписаным.

-verify
Указывает, что архив необходимо проверить на подлинность и целостность, а не подписать. При успешной проверке выведется сообщение "jar verified" (jar подтверждён). При попытке проверить неподписанный

```
          архив или архив с подписью, созданной с
          неподдерживаемым программой алгоритмом - "jar is
          unsigned. (signatures missing or not parsable)" (jar
          не подписан. (подписи отсутствуют или невозможно
          проверить)).

       -certs
          Указывает о необходимости также выводить информацию о
          сертификатах для каждого пользователя, подписавшего
          архив. Информация включает в себя тип сертификата и,
          если это сертификат X.509, имя подписавшего архив
          пользователя. Используется при указании ключей -verify
          и/или -verbose.

       -verbose
          Включает "подробный режим" подписи или проверки, т.е.
          вывод дополнительной информации о совершаемом
          процессе.

ПРИМЕРЫ ИСПОЛЬОВАНИЯ

    Подпись архива

       Подпись архива с названием archv.jar, используя ключ
       пользователя alisa, защищённого паролем "bob", хранимого
       в хранилище по пути /working/mystore с паролем от
       хранилища "mypass" и сохранением подписанного архива под
       названием sarchv.jar

SunOS 5.10              Last change:                        5

User Commands                                       jarsigner(1)

       jarsigner -keystore /working/mystore -storepass mypass -keypass bob -signedjar
sarchv.jar archv.jar alisa

    Проверка подписанного архива

       Проверка подписанного архива signedarchive.jar с выводом
       дополнительной информации о ходе процесса проверки,
       используя данные хранилища по пути /working/mystore.

       jarsigner -keystore /working/mystore -verify -verbose signedarchive.jar

                 198 Fri Sep 26 16:14:06 PDT 1997 META-INF/MANIFEST.MF
                 199 Fri Sep 26 16:22:10 PDT 1997 META-INF/ALISA.SF
                1013 Fri Sep 26 16:22:10 PDT 1997 META-INF/ALISA.DSA
         smk    2752 Fri Sep 26 16:12:30 PDT 1997 SomeClass.class
         smk     849 Fri Sep 26 16:12:46 PDT 1997 AnotherClass.class

            s - подпись прошла проверку
            m - запись об этом файле содержится в файле-манифесте
            k - найден как минимум один сертификат для проверки подписи данного файла

                 jar прошёл проверку.

СМОТРИТЕ ТАКЖЕ
         jar(1), keytool(1)
```

Вывод

В ходе выполнения данной лабораторной работы я ознакомился с основами работы с программами для форматирования текста nroff и troff и научился базовым правилам написания инструкций к программам (страниц man). Помимо этого, была изучена работа утилиты man и предназначения файлов .profile и .kshrc домашнего каталога пользователя.