Java Keytool

Выполнили:

Липинский Леонид, Мурыгин Евгений.

Утилита Keytool

- Java Keytool — это инструмент командной строки, который может генерировать пары открытый ключ / закрытый ключ и сохранять их в хранилище ключей.
- Команды для работы с ключами, сертификатами и хранилищами

```
PS C:\Users\leoni> keytool
Key and Certificate Management Tool

Commands:

 -certreq            Generates a certificate request
 -changealias        Changes an entry's alias
 -delete             Deletes an entry
 -exportcert         Exports certificate
 -genkeypair         Generates a key pair
 -genseckey          Generates a secret key
 -gencert            Generates certificate from a certificate request
 -importcert         Imports a certificate or a certificate chain
 -importpass         Imports a password
 -importkeystore     Imports one or all entries from another keystore
 -keypasswd          Changes the key password of an entry
 -list               Lists entries in a keystore
 -printcert          Prints the content of a certificate
 -printcertreq       Prints the content of a certificate request
 -printcrl           Prints the content of a CRL file
 -storepasswd        Changes the store password of a keystore
 -showinfo           Displays security-related information

Use "keytool -?, -h, or --help" for this help message
Use "keytool -command_name --help" for usage of command_name.
Use the -conf <url> option to specify a pre-configured options file.
```

Генерация пары ключей (genkeypair)

- Сгенерированная пара ключей вставляется в файл KeyStore как пара ключей с собственной подписью.
- Общий формат команды для генерации пары ключей, подробнее об аргументах в конце:

```
-genkeypair
    -alias alias
    -keyalg keyalg
    -keysize keysize
    -sigalg sigalg
    -dname dname
    -keypass keypass
    -validity valDays
    -storetype storetype
    -keystore keystore
    -storepass storepass
    -providerClass provider_class_name
    -providerArg provider_arg
    -v
    -protected
    -Jjavaoption
```

Импорт и экспорт сертификата

Пример команды для импорта сертификата в хранилище :

```
C:\Program Files\Java\jdk1.8.0_121\bin> \
  keytool -importcert -keystore keystore.jks \
  -file veriSignclass1g3ca.cer

Enter keystore password: \
```

Пример команды для экспорта сертификата из хранилища:

```
C:\Program Files\Java\jdk1.8.0_121\bin>
  keytool -exportcert -alias veriSignclass1g3ca -keystore \
  "C:\Program Files\Java\jdk1.7.0_67\jre\lib\security\cacerts" \
  -file veriSignclass1g3ca.cer
```

List

- Чтобы вывести список записей в хранилище ключей, вы можете использовать команду list.

```
-list
    -alias alias
    -storetype storetype
    -keystore keystore
    -storepass storepass
    -providerName provider_name
    -providerClass provider_class_name
    -providerArg provider_arg
    -v
    -rfc
    -protected
    -Jjavaoption
```

List

```
"C:\\Program Files\Java\jdk1.8.0_111\bin\keytool"
    -list
    -storetype JKS
    -keystore keystore.jks
    -storepass abcdef
```

```
Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

testkey, 19-Dec-2017, PrivateKeyEntry,
Certificate fingerprint (SHA1): 4F:4C:E2:C5:DA:36:E6:A9:93:6F:10:36:9E:E5:E8:5A:6E:F2:11:16
```

List + alias

```
"C:\\Program Files\Java\jdk1.8.0_111\bin\keytool"
    -list
    -alias testkey
    -storetype JKS
    -keystore keystore.jks
    -storepass abcdef
```

```
testkey, 15-Dec-2017, PrivateKeyEntry,
Certificate fingerprint (SHA1): 71:B0:6E:F1:E9:5A:E7:F5:5E:78:71:DC:08:80:47:E9:5F:F8:6D:25
```

Delete

- Так же в утилите `keytool` имеется команда, которая может удалить запись из хранилища ключей: `delete` Вот формат этой команды:

```
-delete
    -alias alias
    -storetype storetype
    -keystore keystore
    -storepass storepass
    -providerName provider_name
    -providerClass provider_class_name
    -providerArg provider_arg
    -v
    -protected
    -Jjavaoption
```

- Пример вызова команды `delete.` Эта команда удаляет запись хранилища с псевдонимом `testkey` храня щегося в файле `keystore.jks:`

```
"C:\\Program Files\Java\jdk1.8.0_111\bin\keytool"
    -certreq
    -alias testkey
    -keypass 123456
    -storetype JKS
    -keystore keystore.jks
    -storepass abcdef
    -file certreq.certreq
```

Генерация запроса на сертификат

- Утилита `keytool` может генерировать запрос сертификата с помощью команды `certreq`. Запрос сертификата — это запрос к центру сертификации (ЦС) на создание публичного сертификата для вашей организации. После создания запроса на сертификат он должен быть отправлен в центр сертификации, в котором вы хотите создать сертификат (например, Verisign, Thawte или какой-либо другой центр сертификации).
- Пример запроса:

```
-certreq
    -alias alias
    -sigalg sigalg
    -file certreq_file
    -keypass keypass
    -storetype storetype
    -keystore keystore
    -storepass storepass
    -providerName provider_name
    -providerClass provider_class_name
      -providerArg provider_arg
    -v
    -protected
    -Jjavaoption
```

```
"C:\\Program Files\Java\jdk1.8.0_111\bin\keytool"
    -certreq
    -alias testkey
    -keypass 123456
    -storetype JKS
    -keystore keystore.jks
    -storepass abcdef
    -file certreq.certreq
```

Аргументы (часть 1)

- `-alias` Псевдоним записи в хранилище ключей. Помните, псевдоним может указывать только на один ключ.
- `-keyalg` Название алгоритма, используемого для генерации ключа. Обычно используется RSA.
- `-keysize` Размер ключа в битах. Обычно размеры ключа кратны. Кроме того, различные алгоритмы могут поддерживать только определенные предварительно заданные размеры ключей.
- `-sigalg` Алгоритм подписи, используемый для подписи пары ключей.
- `-dname` Уникальное имя из стандарта X.500. Это имя будет связано с псевдонимом для этой пары ключей в хранилище ключей, также используется в качестве полей «эмитент» и «субъект» в самозаверяющем сертификате.
- `-keypass` Пароль ключевой пары, необходимый для доступа к этой конкретной паре ключей в хранилище ключей.
- `-validity` Количество дней, в течение которых сертификат, приложенный к паре ключей, должен быть действительным.
- `-storetype` Формат файла, в котором должно быть сохранено хранилище ключей. По умолчанию используется JKS. Другим вариантом является формат PKCS11.
- `-keystore` Имя файла хранилища для хранения сгенерированной пары ключей. Если файл не существует, он будет создан

Аргументы (часть 2)

- `-file` Имя файла для чтения или записи сертификата или запроса сертификата.
- `-storepass` Пароль от хранилища ключей, всем, кто захочет работать с ним, понадобится этот пароль.
- `-rfc` Если включить этот флаг, то утилита будет использовать текстовый формат, а не двоичный формат, например для экспорта или импорта сертификатов. Значение -rfc относится к стандарту RFC 1421.
- `-providerName` Имя провайдера криптографического API, который вы хотите использовать при создании пары ключей. Имя провайдера должно быть указано в файлах свойств безопасности Java.
- `-providerClass` Имя корневого класса провайдера криптографического API, который вы хотите использовать. Используется когда имя провайдера не указано в файлах свойств безопасности Java.
- `-providerArg` Аргументы, передаваемые собственному криптографическому провайдеру при инициализации (если это необходимо провайдеру).
- `-v` Сокращенное от verbose, утилита Keytool будет выводить много дополнительной информации в командную строку в удобочитаемом формате.
- `-protected` Определяет, должен ли пароль хранилища ключей предоставляться каким-либо внешним механизмом, например, аппаратный токен. Допустимые значения: true и false.
- `-Jjavaoption` Строка опций для Java VM которая генерирует пару ключей и создает хранилище.