pacman/Package signing (Русский)

< Pacman

Ссылки по теме

- DeveloperWiki:Package signing

Эта страница нуждается в сопроводителе

Статья не гарантирует актуальность информации. Помогите русскоязычному сообществу поддержкой подобных страниц. См. **Команда переводчиков ArchWiki**

Состояние перевода: На этой странице представлен перевод статьи **pacman/Package signing**. Дата последней синхронизации: 21 октября 2015. Вы можете **помочь**синхронизировать перевод, если в английской версии произошли **изменения**.

Основная статья: **pacman (Русский)**.

Чтобы определить, является ли пакет подлинным, *pacman* использует **ключи GnuPG** в модели **сеть доверия** ("web of trust"). В настоящее время существует пять **ключей мастер-подписи**. По меньшей мере три из этих ключей мастер-подписи используются для подписи собственных ключей разработчика и доверенного пользователя, которые потом, в свою очередь, используются для подписи их пакетов. Пользователь также имеет уникальный ключ PGP, который генерируется, когда вы устанавливаете *pacman-key*. Так сеть доверия связывает пользовательский ключ с пятью мастер-ключами.

Примеры сетей доверия:

- **Клиентские пакеты**: вы создали свой пакет сами и подписали его своим собственным ключом
- **Неофициальные пакеты**: разработчик создал пакет и подписал его. Вы используете свой ключ для подписи этого ключа разработчика
- **Официальные пакеты**: разработчик создал пакет и подписал его. Ключ разработчика был подписан мастер-ключами Arch Linux'a. Вы использовали свой ключ для подписи мастер-ключей и доверяете им поручиться за разработчика

Примечание: Протокол HKP использует порт 11371/tcp для связи. Чтобы получить подписанные ключи от серверов (при помощи *pacman-key*), требуется этот порт

Contents

[hide]

Установка

Настройка pacman'a

Опция `SigLevel` в файле `/etc/pacman.conf` определяет, сколько доверия требуется для установки пакета. Чтобы получить подробное объяснение того, что такое `SigLevel`, смотрите **страницу справочного руководства pacman.conf** и комментарии в самом файле. Проверка подписи может быть установлена глобально или для каждого репозитория в отдельности. Если `SigLevel` установлен глобально в разделе `[options]`, требуя подписи всех пакетов, то пакеты, созданные вами, также будет нужно подписывать при помощи *makepkg*.

Примечание: Хотя все официальные пакеты теперь подписаны, на июнь 2012 подписывание баз данных - все еще в процессе. Если установлено значение `Required`, тогда также требуется установить `DatabaseOptional`

Конфигурация по умолчанию может использоваться для того, чтобы устанавливать только те пакеты, которые подписаны доверенными ключами:

```
/etc/pacman.conf

SigLevel = Required DatabaseOptional
```

Это следствие того, что параметр `TrustedOnly` используется в *pacman* по умолчанию, т.е. результат получится таким же, как и в этом примере:

```
SigLevel = Required DatabaseOptional TrustedOnly
```

Вышеупомянутое может быть достигнуто также на уровне репозитария, далее в файле конфигурации:

```
[core]
SigLevel = PackageRequired
Include = /etc/pacman.d/mirrorlist
```

явно добавляет проверку подписи для пакетов репозитария, но не требует подписи от базы данных. Значение `Optional` выключит для данного репозитория используемое глобально значение `Required`.

Важно: Значение SigLevel `TrustAll` существует только для целей отладки и делает слишком легким доверие ключам, которые не были проверены. Вы должны использовать `TrustedOnly` для всех официальных репозитариев

Инициализация связки ключей

Для установки связки ключей *pacman'*а используйте:

```
# pacman-key --init
```

Для этой инициализации требуется **энтропия**. Двигая мышью по кругу, нажимая случайные символы на клавиатуре или производя действия, связанные с диском (например, запустив в другой консоли `ls -R /`, `find / -name foo` или `dd if=/dev/sda8 of=/dev/tty7`), вы генерируете энтропию. Если ваша система не имеет достаточно энтропии, эта ступень может занять часы, а если вы активно генерируете энтропию, это будет выполнено намного быстрее.

Создаваемая случайная последовательность чисел используется для создания связки ключей (`/etc/pacman.d/gnupg`) и подписывающего ключа GPG вашей системы.

Примечание: Если вы запускаете `pacman-key --init` на компьютере, который не генерирует много энтропии (например, удаленный сервер), генерация ключа может занять очень много времени. Для выработки псевдоэнтропии установите на целевую машину **haveged** или **rng-tools**.

Перед запуском `pacman-key --init` от имени суперпользователя **запустите** службу `haveged.service`

Управление связкой ключей

Проверка пяти мастер-ключей

Начальная установка ключей выполняется при помощи команды `pacman-key --populate archlinux` (от имени суперпользователя).

Потратьте время на проверку **подписывающих мастер-ключей** при запросе, поскольку они используются для совместной подписи (и, следовательно, доверия) со всеми остальными ключами разработчика.

Ключи PGP слишком велики (2048 бит или больше) для людей, чтобы работать с ними, поэтому они обычно хешируются, чтобы сделать из них шестнадцатиричный 40-значный отпечаток, который можно использовать для проверки вручную, что два ключа одинаковы. Последние восемь цифр отпечатка служат как имя для ключа, известных как "(краткое) имя или ID ключа" (последние *шестнадцать* цифр отпечатка могут быть "длинное ID имя ключа").

Добавление ключей разработчика

Ключи официальных разработчиков и доверенных пользователей подписываются мастер-ключами, так что вам не нужно использовать *pacman-key*, чтобы подписывать их самостоятельно. Когда *pacman* встречает ключ, который он не может распознать, он предложит скачать его с сервера ключей (`keyserver`), указанного в файле `/etc/pacman.d/gnupg/gpg.conf` или в опции `--keyserver` командной строки. **Список серверов ключей** сопровождается Википедией.

Когда вы скачали ключ разработчика, вам не нужно будет скачивать его снова, и он может быть использован для проверки любого пакета, подписанного этим разработчиком.

Примечание: Пакет **archlinux-keyring**, являющийся зависимостью для **pacman**, содержит последние ключи. Однако, ключи также могут быть обновлены вручную командой `pacman-key --refresh-keys` (от имени суперпользователя). При выполнении этой команды ваш локальный ключ будет также просмотрен удаленным сервером ключей и вы получите сообщение, что он не найден. Об этом вам не стоит переживать

Добавление неофициальных ключей

Вы можете использовать этот способ, например, чтобы добавить свой ключ в связку ключей *pacman*'а или включить подписанный **неофициальный репозиторий**.

Примечание: Возможно, сначала вам потребуется выполнить команду `dirmngr` от имени суперпользователя, смотрите раздел **#gpg: keyserver receive failed: No dirmngr**

Сначала получите ID ключа (`keyid`) от собственника. Потом вам нужно добавить ключ в связку:

- Если ключ найден на сервере ключей, импортируйте его командой:

```
# pacman-key -r keyid
```

- Если у вас есть ссылка на файл ключа, загрузите его и выполните:

```
# pacman-key --add /путь/к/скачанному/файлу/ключа
```

Всегда старайтесь проверить отпечаток, как бы вы делали с мастер-ключом или любым другим ключом, который собираетесь подписать:

```
# pacman-key -f keyid
```

Наконец, вам нужно подписать импортированный ключ на локальном уровне:

```
# pacman-key --lsign-key keyid
```

Теперь вы доверяете этому ключу подписывать пакеты.

Отладка при помощи gpg

Для отладки вы можете получить доступ к связке ключей *pacman*'а напрямую при помощи *gpg*, например, так:

```
# gpg -- homedir /etc/pacman.d/gnupg --list-keys
```

Решение проблем


Эта статья или раздел нуждается в переводе


Примечания: Вступление можно сделать аккуратнее (обсуждение: Talk:Pacman/Package signing (Русский)#)

Важно: Работа *pacman-key* зависит от **времени**. Если системные часы неверны, вы увидите такие ошибки:

```
error: ИмяПакета: подпись от "User <email@archlinux.org>" неверна
error: не удалось совершить транзакцию (неверный или испорченный пакет (подпись PGP))
Случилась ошибка, пакет не обновлен.
```

Не удается импортировать ключи

У этой проблемы могут быть разные причины:

- Устаревший пакет **archlinux-keyring**
- Неправильная дата
- Ваш интернет-провайдер блокировал порт, используемый для импорта ключей PGP
- Кэш *pacman*'а содержит копии неподписанных с предыдущей попытки пакетов

Во время синхронизации обновления вы можете застрять из-за устаревшего пакета **archlinux-keyring**. Сначала попробуйте **обновить систему**, это может помочь.

Если это не помогает, и если ваше системное время в порядке, можете попробовать переключиться на сервер ключей MIT, который предоставляет другой порт. Чтобы это

сделать, отредактируйте файл `/etc/pacman.d/gnupg/gpg.conf`, заменив значение `keyserver` на следующее:

```
keyserver hkp://pgp.mit.edu:11371
```

Если не помогает и это, измените `keyserver` на kjsl, который предоставляет этот сервис через порт 80 (порт HTTP), который всегда должен оставаться разблокированным:

```
keyserver hkp://keyserver.kjsl.com:80
```

Если у вас отключен IPv6, *gpg* не сможет работать, если найдет какой-то адрес IPv6. В этом случае используйте сервер ключей для исключительно IPv4, такой как:

```
keyserver hkp://ipv4.pool.sks-keyservers.net:11371
```

Если вы вдруг забыли выполнить `pacman-key --populate archlinux`, вы можете получить некоторые ошибки при импортировании ключей.

Если ничего из этого не помогает, возможно, ваш кеш *pacman'*а, располагающийся в каталоге `/var/cache/pacman/pkg/`, содержит неподписанные с предыдущих попыток пакеты. Очистите кеш вручную или выполните:

```
# pacman -Sc
```

что удалит все кешированные пакеты, которые не были установлены.

Отключение проверки подписи

Важно: Используйте с осторожностью. Отключение подписи пакета позволит *pacman'*у устанавливать недоверенные пакеты автоматически

Если вас не заботит подпись пакетов, вы можете полностью отключить проверку подписи PGP. Отредактируйте файл `/etc/pacman.conf`, раскомментировав следующую строку в разделе `[options]`:

```
SigLevel = Never
```

Вам нужно также раскомментировать любые установки SigLevel, зависимые от репозитариев, потому что они перевешивают глобальные установки. Это приведет к отсутствию проверки подписи, как было свойственно пакману до четвертой версии. Если вы решите сделать так, вам не нужно устанавливать связку ключей при помощи *pacman-key*. Вы сможете изменить этот выбор позднее, если решите включить проверку пакетов.

Сброс всех ключей

Если вы захотите удалить или сбросить все ключи, установленные в вашей системе, можете удалить каталог `/etc/pacman.d/gnupg` и перезапустить `pacman-key --init` и следуя процедуре, добавляющей ключи предпочтительным образом.

Удаление залежавшихся пакетов

Если те же пакеты продолжают не работать и вы уверены, что сделали правильно все дела *pacman-key*, попробуйте удалить пакеты при помощи команды `rm /var/cache/pacman/pkg/`*`имя_плохого_пакета`*`*`, чтобы они были скачаны заново.

Это может в самом деле оказаться решением, если вы получаете сообщение вида `error: linux: signature from "<Some.Person@example.com>" is invalid` или подобное при обновлении (например, вы можете не быть, наконец, жертвой атаки MITM, ваш скачанный файл просто поврежден).

Обновление ключей через прокси

В *gnupg* есть **ошибка**, не дающая обновлять ключи через http-прокси. Для разрешения этой проблемы вы можете сделать следующее:

1. Измените /etc/hosts:

```
127.0.0.1 pool.sks-keyservers.net
```

2. Установите туннель при помощи **socat**. Запускать его необходимо от имени суперпользователя, потому что вам нужно прослушивание порта 80 TCP:

```
# socat TCP-LISTEN:80,reuseaddr,fork PROXY:localhost:pool.sks-keyservers.net:80,proxyport=3128
```

3. Обновите ключи:

```
# pacman-key --refresh-keys
```

Верните назад прежнюю конфигурацию, когда прокси больше не будет нужно.

gpg: keyserver receive failed: No dirmngr

Для получения дополнительной информации смотрите сообщение об ошибке **FS#42798**.

Запустите:

```
# dirmngr < /dev/null
```

Или по другому: создайте файл `/root/.gnupg/dirmngr_ldapservers.conf` **[1]**.

Смотрите также

- **DeveloperWiki:Package Signing Proposal for Pacman**
- **Pacman Package Signing – 1: Makepkg and Repo-add**
- **Pacman Package Signing – 2: Pacman-key**
- **Pacman Package Signing – 3: Pacman**
- **Pacman Package Signing – 4: Arch Linux**

Categories:

- Package management (Русский)
- Русский