


Apt-key is deprecated или управление ключами в современных выпусках Debian и Ubuntu

 interface31.ru/tech_it/2022/09/apt-key-is-deprecated-ili-upravlenie-klyuchami-v-sovremennyh-vypuskah-debian-i-ubuntu.html

6 сентября 2022 г.

```
tc/apt/sources.list.d# apt-key list
Manage keyring files in trusted.gpg.d instead

7CD8 C5E2 2450 0C12 89C0
Viewer GmbH (TeamViewer Linux 2017) <support@

1791 8DA8 4BE5 DEB4 9217
Viewer Germany GmbH (TeamViewer Linux 2020) <

[исключен до: 2024-06-14]
A6AB ABF5 BD82 7BD9 BF62
x signing key <signing-key@nginx.com>

ubuntu-kdiskmark.gpg
-----

0007 C2B0 EBC0 D58C 0CED
chpad PPA for Dmitry Sidorov
```

Многие базовые действия в дистрибутивах Linux не меняются множество лет и для многих стали уже привычкой. А привычки - вещь такая: привыкнуть легко, сложно переучиться. Поэтому изменения базовых вещей многими воспринимается в штыки и вызывает крайне негативные эмоции. Apt-key - утилита командной строки для управления ключами пакетного менеджера APT и когда ее объявили устаревшей, то многим это не понравилось. Однако на то были свои причины, а новая система управления ключами во многом даже проще и удобнее, нужно лишь разобраться и привыкнуть.



Онлайн-курс по устройству компьютерных сетей

На углубленном курсе "[Архитектура современных компьютерных сетей](#)" вы с нуля научитесь работать с Wireshark и «под микроскопом» изучите работу сетевых протоколов. На протяжении курса надо будет выполнить более пятидесяти лабораторных работ в Wireshark.

Коротко о том, что такое ключ репозитория и для чего он нужен. Любой репозиторий содержит пакеты, которые передаются по открытым каналам и перед тем, как устанавливать их в систему нам следует убедиться в их подлинности. Для этого все пакеты в репозитории подписаны закрытым ключом репозитория, а чтобы проверить их подлинность нам потребуется открытый ключ или просто ключ.

Для того, чтобы система могла использовать ключ его нужно установить в системное хранилище, которое располагается в **/etc/apt/trusted.gpg**, для чего использовалась команда:

```
apt-key add repo_signing.key
```

Однако если вы выполните команду в последних версиях Debian, Ubuntu и других основанных на них дистрибутивах, то получите предупреждение:

```
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
```

Ключ при этом добавится и, в принципе, вы можете продолжать пользоваться старым способом не обращая внимание на предупреждения.

Но разработчики не просто так объявили **apt-key** устаревшим, на это есть серьезные причины. Дело в том, что АРТ безоговорочно доверяет **любому** ключу в **trusted.gpg** для **любого** репозитория, что дает возможность загрузить из стороннего репозитория пакеты, подписанные ключом другого репозитория и заменить таким образом любой пакет в системе.

Чтобы устранить потенциальную брешь в безопасности была введена новая система, когда каждый репозиторий доверяет только собственному ключу, а сами ключи помещаются в специальное хранилище, к которому имеет доступ только суперпользователь. В настоящее время это директория **/usr/share/keyrings**, согласно документации там следует размещать ключи, дальнейшее управление которыми предполагается с помощью АРТ или DPKG. Для ключей управляемых локально предназначена директория **/etc/apt/keyrings**. В системах до Debian 12 и Ubuntu 22.04 ее следует создать самостоятельно с разрешением 755.

Ключ должен быть в двоичной форме (без ASCII-Armor) и иметь имя:

```
repo-archive-keyring.gpg
```

Где **repo** - короткая часть имени репозитория.

Допускается также иметь рядом текстовую версию ключа (с ASCII-Armor) с именем:

```
repo-archive-keyring.asc
```

Также в строке подключения репозитория можно явно указать связанный с ним ключ:

```
deb [signed-by=/usr/share/keyrings/repo-archive-keyring.gpg] ...
```

На практике очень часто требование снять ASCII-Armor не соблюдается и в **/usr/share/keyrings** кладется текстовая версия ключа. Однако в документации Debian крайне не советуется так делать.

Далее мы рассмотрим на практике приемы работы с ключами в современных операционных системах.

Определение типа ключа

Выше мы говорили, что ключ может быть двух типов: бинарный и текстовый, т.е. с ASCII-Armor или без. Пусть вас не смущает новый непонятный термин, ASCII-Armor - это привычный всем текстовый формат ключей в кодировке Base64.

```
root@andrey-virtual-machine: /home/andrey# cat nginx_signing.key
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.22 (GNU/Linux)

mQENBES0MnIBCAD+FPYKGrIGGf7NqWKFwCB3cBV01gabgVWQnZbMcFzeW+hMsgxH
W6lImD0RsfZ9oEbfJCPG0CR5Z7ppq5pKamYs2+EJ8Q2ysOFHHwpGrA2C8zyNAS4I
QxnZZiBETgcSwfT0un0XlqPwPZgyuXVn9PAbLZRbfBzm8wR/3SWyqqZBBLdQk5TE
fDR+Eny/M1RVR4xCLECONF9UBB2ejFdI1LD45APbP2hsN/piFByUit7yK2gpFyRt
97WzGHn9MV5/TL7AmRPM4pcr3JacmtCnxXeCZ8nLqedoSuHfuhwyDn1Abu8I1605
XRrfzhrFRJFM1JnIiGmzZi6zBvH0ItfyX6ttABEBAAQ0KW5naW54IHNPZ25pbmcg
a2V5IDxzaWduaW5nLWtleUBuZ2lueC5jb20+iQE+BBMBAgAoAhsDBgsJCAcDAGYV
CAIJCgsEFgIDAQIeAQIXgAUCV2K1+AUJGB4fQQAACRCr9b2Ce9m/YloaB/9XGrol
kocn7L/tsVjaBQCteXKuwsM4XhCuAQ6YAwA1L1UheGOG/aa2xJvrXE8X32tgcTjr
KoYoXWcdxaFjLXGT6jV85qRguUzVM0xxSEM2Dn11setN9p1P0Zz+4rkx8+2vJG
F+eMlrUPXg/zd88NvyLq5gGHEsFRBMVufYmHtNfcp4okC1klwIRISdp4QY1wdrN
10+/oCTL8Bzy6hcHjLIq3aouncLxMjtBoclc/50TioLDw5DfVx7rWYfRhcbZVbwD
oe/PD08A0AA6fxXvwjSxy+dGhEaXoTHjkCbz/l6NxrK3JFyauDgU4K4MytsZ1HDl
MgMW8hZXszoICTTlQEcBBABAgAGBQJOTkelAAoJEKZP1bF62zmo79oH/1XDb29S
YtWp+MTJTPFEwLWRIyRuDXy3wBd/BpwBRIWfWzMs1gnCjNjK0EVBVGa2grvy9Jtx
JKMd6l/PWXVucSt+U/+G08rBkw14SdhqxaS2l14v6gyMeUrSbY3XfToGfWHC4sa/
Thn8X4jFaQ2XN5dAIzJGU1s5JA0tjEzUwCnmrKmyMLXZaoQVrm0RGjCuH0I0aAFk
R50utnB9HppxhCVbs24xXZQnZDNbUQeulFx54uP3OLDBAeCHl+v4t/uotIad8v6J
S093vc1evIje6lguE81HhJn9noxPitvOv5Mb2yPsE8mH4cJHRTFNS5EhPW6ghnlf
Wa9Zw1VX5lgxcvaIRgQEQIABGUCTk5b0gAKCRDs80kLLBcgg1G+AKCnacLb/+W6
cfl1rUjEXgZdUQqoogCeNPVwXlHEIVqlthAM1pdY/gcaQZnIRgQQEQIABGUCTk5f
YQAACRCrN2E5pSTFPnWAZ9gUozyIs+9jf2rJvqnJSEWuCGVRwCcCUFhXRCpQ02Y
Va3l3WuB+rgKjsQ=
=EWMI
-----END PGP PUBLIC KEY BLOCK-----
```

Знакомо, не правда ли? Это и есть ключ с ASCII-Armor. Такие ключи наиболее распространены, так как текстовый формат более удобен при передаче в сетях связи. Убедиться, что перед вами ключ в ASCII формате можно просто прочитав файл, например, командой **cat**:

```
cat repo_signing.key
```

Или при помощи команды:

```
file repo_signing.key
```

В случае текстового формата вы увидите:

```
PGP public key block Public-Key (old)
```

Если это бинарный ключ:

```
OpenPGP Public Key Version 4
```

Почему мы уделяем этому столько внимания? Потому что если делать все по правилам, то ключи должны быть в бинарном формате, а для этого нужно понимать в каком виде мы получили исходный ключ. В большинстве случаев это будет текстовый ключ с ASCII-Armor.

Удаление старых ключей

Перед тем, как устанавливать ключ в новое хранилище нам нужно удалить его из старого хранилища `/etc/apt/trusted.gpg`. Для этого сначала получим список ключей командой:

```
apt-key list
```

В выводе будет полный список установленных в систему ключей.

```
root@andrey-virtual-machine:/etc/apt/sources.list.d# apt-key list
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
/etc/apt/trusted.gpg
-----
pub   rsa4096 2017-03-13 [SC]
      8CAE 012E BFAC 38B1 7A93  7CD8 C5E2 2450 0C12 89C0
uid   [ неизвестно ] TeamViewer GmbH (TeamViewer Linux 2017) <support@teamviewer.com>
sub   rsa4096 2017-03-13 [E]

pub   rsa4096 2020-01-29 [SC]
      D2A5 FEB3 4881 60F0 28CC  1791 8DA8 4BE5 DEB4 9217
uid   [ неизвестно ] TeamViewer Germany GmbH (TeamViewer Linux 2020) <support@teamviewer.com>

pub   rsa2048 2011-08-19 [SC] [годен до: 2024-06-14]
      573B FD6B 3D8F BC64 1079  A6AB ABF5 BD82 7BD9 BF62
uid   [ неизвестно ] nginx signing key <signing-key@nginx.com>

/etc/apt/trusted.gpg.d/jonnagon-ubuntu-kdiskmark.gpg
-----
pub   rsa4096 2020-09-07 [SC]
      3B3C 4A82 F16C E6BC 90F9  0007 C2B0 E8C0 D58C 0CED
uid   [ неизвестно ] Launchpad PPA for Dmitry Sidorov
```

Находим и копируем идентификатор нужного ключа, затем удаляем его командой:

```
apt-key del "KEY-ID"
```

Идентификатор можно взять как есть, с пробелами, только заключить его в двойные кавычки. Либо использовать два последних блока, так для указанного выше ключа можно ввести:

```
apt-key del D58C0CED
```

Кстати, именно эти две команды являются на сегодняшний день допустимыми для использования с `apt-key`.

Получение и установка текстовых ключей OpenPGP (с ASCII-Armor)

Адрес ключа репозитория обычно можно узнать в документации продукта, затем нам потребуется его скачать и установить в новое хранилище. Для скачивания можно использовать **wget**:

```
wget https://example.com/key/repo_signing.key
```

Или **curl**:

```
curl -O https://example.com/key/repo_signing.key
```

Затем его следует преобразовать в бинарный формат и поместить в хранилище, эту команду следует выполнять с правами суперпользователя:

```
gpg --dearmor < repo_signing.key > /usr/share/keyrings/repo-archive-keyring.gpg
```

Можно ли упростить этот процесс, можно, для этого используем конвейер:

```
wget -O- https://example.com/key/repo_signing.key | gpg --dearmor > /usr/share/keyrings/repo-archive-keyring.gpg
```

Или

```
curl https://example.com/key/repo_signing.key | gpg --dearmor > /usr/share/keyrings/repo-archive-keyring.gpg
```

Возможно, многие обратили внимание на чередование ключей в командах. Так **wget** по умолчанию скачивает файл, а **curl** выдает содержимое запроса в стандартный вывод. Поэтому чтобы сохранить файл с исходным именем мы использовали ключ **-O** для curl, и наоборот, чтобы получить содержимое файла в stdout мы запустили wget с ключом **-O-**, который предполагает вывод в файл, но если вместо файла указан дефис, то идет вывод в стандартный поток. Это небольшая мелочь, которую надо всегда помнить при работе с этими утилитами.

Обе команды должны быть запущены под root, если же вы предпочитаете sudo, то следует сделать так:

```
wget -O- https://example.com/key/repo_signing.key | gpg --dearmor | sudo tee /usr/share/keyrings/repo-archive-keyring.gpg >/dev/null
```

Или

```
curl https://example.com/key/repo_signing.key | gpg --dearmor | sudo tee /usr/share/keyrings/repo-archive-keyring.gpg >/dev/null
```

В чем разница? Основная часть работы выполняется с правами обычного пользователя, затем поток вывода передается утилите **tee**, которая записывает его в файл и выводит на экран, чтобы избежать последнего мы перенаправили ее вывод в **/dev/null**. При этом только **tee** запускается с правами суперпользователя.

Получение и установка бинарных ключей OpenPGP (без ASCII-Armor)

Мы бы хотели привести пример, но не можем припомнить чтобы в каком-то репозитории применялись бинарные ключи. Но случаи бывают разные. Для этого вам понадобятся права суперпользователя или sudo:

```
wget -O /usr/share/keyrings/repo-archive-keyring.gpg https://example.com/key/repo_signing.key
```

Или

```
curl -o /usr/share/keyrings/repo-archive-keyring.gpg https://example.com/key/repo_signing.key
```

Здесь все просто, мы сразу помещаем скачиваемый файл в хранилище под нужным именем.

Получение ключей OpenPGP с сервера ключей

Использование серверов ключей достаточно редкий сценарий, но он тоже иногда используется, и вы должны уметь это делать. Для того, чтобы получить ключ надо знать его ID, точнее два его последних блока:

```
gpg --keyserver keyserver.ubuntu.com --recv "KEY-ID"
```

Также вы можете использовать ID ключа целиком, если он содержит пробелы, то оберните его в двойные кавычки.

После выполнения данной операции в домашней директории пользователя будет создано локальное хранилище в скрытой директории **.gnupg/trustdb.gpg**.

Теперь нам нужно экспортировать оттуда ключ в хранилище, команда должна выполняться под тем же самым пользователем, который получил ключ в предыдущей команде, если это root:

```
gpg --export "KEY-ID" > /usr/share/keyrings/repo-archive-keyring.gpg
```

А если нужно получить текстовый ключ:

```
gpg --export --armor "KEY-ID" > /usr/share/keyrings/repo-archive-keyring.asc
```

Если ключ получал непривилегированный пользователь, то работаем через tee и sudo:

```
gpg --export "KEY-ID" | sudo tee /usr/share/keyrings/repo-archive-keyring.gpg  
>/dev/null
```

Можно ли сразу получить и экспортировать ключ? Можно. Если вы внимательно разобрали команды выше, то следующая не покажется вам китайским заклинанием:

```
gpg --no-default-keyring --keyring /usr/share/keyrings/repo-archive-keyring.gpg --  
keyserver keyserver.ubuntu.com --recv "KEY-ID"
```

Данную команду следует выполнять от имени суперпользователя, ключ **--no-default-keyring** предписывает не использовать локальное хранилище в домашней директории пользователя, оттуда не будет ничего считываться и не будет ничего записано.

Удаление ключей из хранилища

Для удаления ключа достаточно удалить соответствующий ему файл, для этого вам потребуются права root:

```
rm -f /usr/share/keyrings/repo-archive-keyring.gpg
```

Как видим, управлять ключами в современных системах несложно. Надеемся, что данная статья будет вам полезна и поможет быстрее перейти к использованию новых и безопасных инструментов.

