Anton Baskov <*ab@architecturebureau.org*>

Рекомендации по работе с PGP

Церемония подписания ключей PGP

ENOG 13, Санкт-Петербург

23 – 24 Мая 2017 года

Церемония подписания ключей на первой встрече ENOG в июне 2001 года

Церемония подписания ключей на тринадцатой встрече ENOG в мае 2017 года

СОЗДАНИЕ КЛЮЧЕЙ

Простая инструкция

- Создаём новый ключ
- Устанавливаем срок действия
- Пользуемся
- Периодически продлеваем срок действия ключа

- Что произойдет при возможной компрометации ключа? Например, при утере ноутбука?
  - Вы отзываете ключ и теряете всю цепочку доверия
  - Или пользуетесь ключом и рискуете

Сложная инструкция

- Создаём основной ключ и храним его в сейфе
- Создаём один или несколько дополнительных ключей с ограниченным сроком действия и распределяем их по устройствам
- В случае компрометации отзываем дополнительные ключи
- Основной ключ используется
  - для создания, продления срока действия и отзыва дополнительных ключей
  - для подписания других ключей

Создание основного ключа

- gpg --homedir /the-secret-place --gen-key
  - Type: RSA & DSA (as ECC key support is not widely distributed)
  - Size: 4096
  - Valid: 0 (does not expire)

- N.B. Основной ключ удобнее сразу создать на внешнем носителе и положить его в безопасное место

- gpg --gen-revoke EB0069AC
  - Обязательно создайте сертификат отзыва для основного ключа и сохраните его в надёжном месте

```
pub   4096R/EB0069AC 2017-05-17
uid                  Example <example@example.com>
sub   4096R/2C7BD7E1 2017-05-17
```

Добавление идентификаторов

- К одному ключу можно добавить несколько идентификаторов (UID), например, личный и рабочий адреса электронной почты

- gpg --edit-key EB0069AC
  > adduid
  > uid <number>
  > trust
    (5 = ultimate trust)
  > uid <number>
  > primary
    (для выбора основного идентификатора)
  > save

```
pub   4096R/EB0069AC 2017-05-17
uid                  Example <example@example.net>
uid                  Example <example@example.com>
sub   4096R/2C7BD7E1 2017-05-17
```

Удаление идентификаторов

- Удалить идентификатор
  - Используйте, когда нужно исправить ошибку при том, что открытый ключ не отправлен
  - Если вы отправили ключ в реестр, то его можно только отозвать

  > *deluid*

- Отозвать идентификатор
  - Идентификатор более не действителен, например, нужно убрать старый рабочий адрес
  - Информация о старых идентификаторах может сохраниться в ключе и скорее всего сохраниться в реестрах

  > *revuid*

Дополнительные ключи

- Существует несколько видов ключей
  - для сертификации (тип C) (обычно только основной ключ)
  - для подписи (тип S)
  - для шифрования (тип E)
  - для аутентификации (тип A)

Дополнительный ключ подписи

- Можно сделать сколько угодно дополнительных ключей подписи
  - например, по ключу подписи на каждое устройство
- Рекомендуется установить разумный срок действия для каждого дополнительного ключа подписи и периодически их менять
- gpg --edit-key EB0069AC
  > addkey
  > save

```
pub  4096R/EB0069AC  created: 2017-05-17  expires: never       usage: SC
                     trust: ultimate      validity: ultimate
sub  4096R/2C7BD7E1  created: 2017-05-17  expires: never       usage: E
sub  4096R/D63BAAC4  created: 2017-05-17  expires: 2018-05-17  usage: S
```

Дополнительный ключ шифрования

- Рекомендуется иметь лишь один действительный ключ шифрования
  - В ином случае сообщение вам могут зашифровать любым из действительных ключей

- По умолчанию вам уже создан ключ шифрования, нужно лишь ограничить срок его действия

- gpg --edit-key EB0069AC
  > key <key id>
  > expire
  > save

```
pub  4096R/EB0069AC  created: 2017-05-17  expires: never       usage: SC
                     trust: ultimate      validity: ultimate
sub  4096R/2C7BD7E1  created: 2017-05-17  expires: 2018-05-17  usage: E
sub  4096R/D63BAAC4  created: 2017-05-17  expires: 2018-05-17  usage: S
```

Ключ для аутентификации

- В основном используется для SSH
  - Обратите внимание, что срок действия ключа, будучи единожы записанным в authorised_keys, обычно сам не обновляется

- gpg --expert --edit-key EB0069AC
  > addkey

  выбрать пункт 8: RSA, own capabilities

  выбрать Authentication, снять Sign и Encryption

  > save

```
pub  4096R/EB0069AC  created: 2017-05-17  expires: never       usage: SC
                     trust: ultimate      validity: ultimate
sub  4096R/2C7BD7E1  created: 2017-05-17  expires: 2018-05-17  usage: E
sub  4096R/D63BAAC4  created: 2017-05-17  expires: 2018-05-17  usage: S
sub  4096R/1F7BDBC7  created: 2017-05-17  expires: 2020-05-17  usage: A
```

Резервные копии

- Экспорт открытой части основного ключа и дополнительных ключей
  - `gpg --armor --export EB0069AC`
- Экспорт закрытой части ключей
  - `gpg --armor --export-secret-keys EB0069AC`
- Импорт ключей
  - `gpg --import`

Список ключей

- gpg --list-keys example@example.com
- gpg --fingerprint --list-key EB0069AC
  - Указывайте полный отпечаток ключа в подписи и на визитке

```
pub   4096R/EB0069AC 2017-05-17
      Key fingerprint = 10B2 1DB6 7D3A A10D 6064  76E4 C5F0 1377 EB00 69AC
uid                  Example <example@example.com>
uid                  Example <example@example.net>
sub   4096R/2C7BD7E1 2017-05-17 [expires: 2018-05-17]
sub   4096R/D63BAAC4 2017-05-17 [expires: 2018-05-17]
sub   4096R/1F7BDBC7 2017-05-17 [expires: 2018-05-17]
```

Отпечатки ключей

- Полный отпечаток (2^{160}):
  - 10B2 1DB6 7D3A A10D 6064 76E4 C5F0 1377 EB00 69AC
- Длинный (2^{64}):
  - C5F0 1377 EB00 69AC
- ~~Короткий (2^{32})~~:
  - EB00 69AC
  - Меньше минуты на генерацию ключа с аналогичным отпечатком!
- Решение
  - gpg --keyid-format long --list-keys ...
  - echo keyid-format long >> ~/.gnupg/gpg.conf

Список закрытых ключей

- `gpg --list-secret-keys`

```
sec   4096R/EB0069AC 2017-05-17
uid                  Example <example@example.com>
uid                  Example <example@example.net>
ssb   4096R/2C7BD7E1 2017-05-17
ssb   4096R/D63BAAC4 2017-05-17
ssb   4096R/1F7BDBC7 2017-05-17
```

Экспорт в рабочее окружение

- Нам необходимо экспортировать все ключи, за исключением закрытой части основного ключа
  - `gpg --homedir /Volumes/Key --armor --export-secret-subkeys | gpg --import`

- После импорта закрытой части дополнительных ключей и открытой части ключей должно получится следующее (обратите внимание на выделенную часть):
  - `gpg --list-secret-keys`

```
sec#  4096R/EB0069AC 2017-05-17
uid                  Example <example@example.com>
uid                  Example <example@example.net>
ssb   4096R/2C7BD7E1 2017-05-17
ssb   4096R/D63BAAC4 2017-05-17
ssb   4096R/1F7BDBC7 2017-05-17
```

Ошибка при экспорте?

- Если вы ошиблись при экспорте закрытой части ключей, например, экспортировали лишний дополнительный ключ, то удалите все закрытые ключи и импортируйте их заново по одному
  - `gpg --delete-secret-keys EB0069AC`

ПУБЛИКАЦИЯ КЛЮЧЕЙ

Публикация ключей

- В реестры
  - gpg --keyserver <keyserveraddress> --send-key EB0069AC
- На визитку и в подпись
  - Указывайте полный отпечаток ключа
  - gpg --fingerprint --list-key EB0069AC

Sincerely yours,

Dr. Example <example@example.org>

10B2 1DB6 7D3A A10D 6064 76E4 C5F0 1377 EB00 69AC

Публикация в DNS

- Подготовка ключа
  - gpg --armor --export-options export-minimal --export A71049CA
- CERT RR
  - RFC 2538, RFC 4398
- OPENPGPKEY RR
  - RFC 7929

ПОДПИСЬ КЛЮЧЕЙ

Подготовка к церемонии

- Загрузите ваш ключ в keyring по указанному на последующих слайдах адресу
  - gpg --armor --export EB0069AC
- Возьмите с собой распечатку с полным отпечатком основного ключа
  - Удобно, если отпечаток ключа уже есть у вас на визитке
- Возьмите документ удостоверяющий личность
  - В случае если написание имени в идентификаторе отличается от указанного в документе, приложите любое иное подтверждение правильности транслитерации (например, банковскую карточку)
  - Но проще взять с собой загранпаспорт

Церемония подписания ключа

- Получите распечатку
  - Проверьте отпечаток своего ключа
- Зачитайте отпечаток своего ключа со своей визитки
  - Если вы хотите, чтобы вам подтвердили иной идентификатор, отличный от основного, скажите об этом
- Отметьте проверенные ключи и документы
  - Если необходимо, укажите уровень доверия

Уровни доверия

- 0x10 (sig): Уровень доверия не определён
- 0x11 (sig1): Проверки не производилось, обычное доверие
  - Так можно подтвердить псевдоним
- 0x12 (sig2): Обычная проверка
  - Вы проверили имя по официальным документам
  - Вы проверили указанный адрес электронной почты
- 0x13 (sig3): Тщательная проверка
  - Вы уверены в предъявленных документах: вы знаете средства защиты указанных документов и проверили их
  - Имя указано так же как и в предъявленном официальном документе (например, в соответствии с загранпаспортом)
  - Вы давно знаете того, чей ключ вы подписываете

После церемонии

- Загрузите список ключей (keyring) и импортируйте их
  - из файла: gpg --import from_keyring.txt
- Проверьте отпечаток ключа
  - gpg --fingerprint --list-keys A71049CA
- Определите идентификатор, который вы хотите подписать
  - Если не оговорено, выберите основной идентификатор

Подпись ключей

- Подпишите ключ
  - gpg --ask-cert-level --sign-key A71049CA
    - Really sign all user IDs? No
    - Выберите идентификатор для подписания
    - Введите sign
    - Выберите уровень доверия
    - Save
- Отправьте подписанный ключ зашифрованным по электронной почте
  - Таким образом вы можете подтвердить адрес электронной почты, указанный в идентификаторе
  - gpg --armor --export A71049CA | gpg --encrypt -r A71049CA --armor

Подпись ключей

- В случае нескольких идентификаторов для проверки принадлежности адреса электронной почты вам нужно будет подписать каждый идентификатор по отдельности, т.е.
  - экспортировать ключ в отдельный файл,
  - подписать один из идентификаторов,
  - экспортировать ключ,
  - отправить его в зашифрованном виде на указанный адрес электронной почты,
  - удалить ключ,
  - импортировать ключ из файла (в нем не будет предыдущей подписи),
  - продолжить подписание следующего идентификатора,
  - или **использовать caff** и иные подобные инструменты.

Замечания

- Удостоверяйте только те идентификаторы, в которых вы уверены
  - Не уверены в фотографии?
  - Не уверены в адресе электронной почты?
  - Не подписывайте!
- Вы можете описать свои правила подписи
  - --set-policy-url
- Для указания уровней доверия в с использованием сторонних инструментов, таких как caff, добавьте *ask-cert-level* в конфигурационный файл *.gnupg/gpg.conf*
- Используйте trust/sign с осторожностью!

Полезные инструменты

- GNU Privacy Guard (gpg), gnupg.org
- GPG Suite, GPGTools.org
- CAFF
  - signing-party package from debian
  - https://pgp-tools.alioth.debian.org/

Anton Baskov <*ab@architecturebureau.org*>

Вопросы?