#bash #GnuPG #GPG #encrypt #decrypt #gpa #kgpg #kleopatra #GPG4Win

Понадобилось мне ассиметричное шифрование / дешифрование, и GPG. как всегда выручает.

Для всех кому лень пользоваться терминалом существуют GUI утилиты для взаимодействия с GnuPG.

Для Linux: gpa, kgpg, kleopatra.

Для Windows: GPG4Win.

Для всех тех, кому необходимо немного больше пространства для манёвров, т.е. больше управления будем использовать терминал.

Подписывайтесь на канал, ставьте лайки, комментируйте.
Ну а с вами как всегда был Shadow.
Всем Добра и Удачи!

Немного GnuPG.

Продолжаем изучать GPG.

Сегодня затронем тему шифрования файлов и директорий, а также безопасный способ передачи данных от вас, например, на работу или заказчику.

Возьмем в пример мою ситуацию. На работе согласно датам происходит плановая замена ключей доступа**.** И так вышло, что у меня и учебный отпуск и ежегодный оплачиваемый идут подряд друг за другом.

Вся проблема заключается в том, что приватные ключи доступа никоим образом нельзя перекачивать на заданные USB носители через сеть интернет. При малейшем признаке компрометирования данных и на организацию могут наложить немаленький штраф.

На помощь как всегда приходит **GPG**.

GPG — **это** инструмент шифрования и электронного подписывания. В его работе используется ассиметричное шифрование, основанное на двух ключах: приватный и публичный.

Для всех кому лень пользоваться терминалом существуют **GUI** утилиты для взаимодействия с **GnuPG**.

Для **Linux**: gpa, kgpg, kleopatra.

Для **Windows**: GPG4Win.

Для всех тех, кому необходимо немного больше пространства для манёвров, т.е. больше управления читайте статью до конца. Для последнего будем использовать **терминал**.

В **Linux**-е gpg и gpg2 по умолчанию уже должны быть установлены. Если нет:

$ sudo apt install gnupg

sudo apt install gnupg2

А вот **gnupg2** не во всех системах можно вот так поставить. Но если сразу заработает командой «**gpg2»** — отлично. Если нет, то это не смертельно.

$ sudo pacman -S gnupg

sudo pacman -S gnupg2

В **Windows**-е можете воспользоваться, например git-bash-ем или cygwin-ом.

Для всех систем команды будут одинаковые. Вместо shred **воспользуйтесь обычной командой удаления файла(ов)** (rm -rf, rmdir or rd)**. Вместо** tar **просто используйте имеющийся у вас архиватор.**

Если что-то не работает, замените короткий идентификатор на адрес введённой в ключ электронной почты.

Часть №1. Подготовка почвы.

Для начала примера создадим 2 ключа в 2 разных директориях. Т.е. база данных **GnuPG** ключей будет находится в разных папках. Так у нас основные ключи системы будут отдельно от данных тестовых. Но вы вполне можете сделать тоже самое прямо сразу на вашу флеш-карту. И в последствии можете сразу использовать её на другом ПК в другой консоли или командной строке.

В качестве примера 2 директории как раз и будут характеризовать **домашний ПК** и **рабочий ПК**. Но вы вполне можете **не использовать** изменение домашней директории хранения базы данных **GnuPG**-ключей:

--homedir ./homelab

--homedir ./worklab

Тогда ваши **GPG**-ключи будут храниться в системе по пути по-умолчанию.

В **Linux** это «~/.gnupg/». В **Windows** это «C:\\Users\Пользователь\.gnupg\». В **Windows**-е бывают и другие директории.

Создаём каталоги **homelab** и **worklab** и не забываем про правильные права доступа к этим каталогам.

$ mkdir -p ./homelab ./worklab

$ chown -R mikl:users ./homelab

$ chown -R mikl:users ./worklab

$ chmod 0700 ./homelab

$ chmod 0700 ./worklab

Переходим к GPG.

$ gpg2 --homedir ./homelab --full-generate-key

$ gpg2 --homedir ./homelab -k --keyid-format 0xshort

--keyid-format 0xshort — отображает ключ в коротком формате **0x**.

Сразу экспортируем публичный и приватные ключи. Разумеется короткие идентификаторы у вас будут свои.

$ gpg2 --homedir ./homelab --output ./home-public.gpg --armor --export 0xB8F76730

$ gpg2 --homedir ./homelab --output ./home-private.gpg --armor --export-secret-keys

Не забываем сразу поменять уровень доверия к ключу и также экспортировать его. Учтите, что если вы планируете загрузить ключ на публичный сервер ключей, то рекомендуется при генерации ключа добавить короткий комментарий, а уровень доверия выставить не выше 4-его.

$ gpg2 --homedir ./homelab --edit-key 0xB8F76730

trust

4

y

save

$ gpg2 --homedir ./homelab --export-ownertrust > home-trust.txt

Также обязательно генерируем сертификат отзыва ключа. Данный сертификат следует импортировать только в случае если ваш приватный ключ был скомпрометирован, и сразу после этого отправить изменения на публичный сервер ключей. Или если вам больше не нужен данный ключ, или при сбое системы был утерян приватный ключ, например. Ситуаций бывает много. Во всех остальных случаях он скорее всего просто не понадобится.

$ gpg2 --homedir ./homelab -a --gen-revoke 0xB8F76730 > home-revoke.gpg

Делаем тоже самое для второго ключа.

$ gpg2 --homedir ./worklab --full-generate-key

$ gpg2 --homedir ./worklab -k --keyid-format 0xshort

$ gpg2 --homedir ./worklab --output ./work-public.gpg --armor --export 0xDF679B3B

$ gpg2 --homedir ./worklab --output ./work-private.gpg --armor --export-secret-keys

$ gpg2 --homedir ./worklab --edit-key 0xDF679B3B

trust

4

y

save

$ gpg2 --homedir ./homelab --export-ownertrust > work-trust.txt

$ gpg2 --homedir ./worklab -a --gen-revoke 0xDF679B3B > work-revoke.gpg

Всё. Публичными ключами и файлами уровня доверия можно обмениваться между пользователями.

Импортируем публичные ключи для разных директорий (пользователей, ПК) и подписываем их. Про подпись читайте ниже. Вы вполне можете обойтись без подписывания ключа, даже если будете загружать его на публичный сервер ключей. Но с подписью у вас появляются некоторые преимущества.

homelab:

$ gpg2 --homedir ./homelab --import work-public.gpg

$ gpg2 --homedir ./homelab --import-ownertrust work-trust.txt

$ gpg2 --homedir ./homelab -k --keyid-format 0xshort

$ gpg2 --homedir ./homelab --sign-key 0xDF679B3B

worklab:

$ gpg2 --homedir ./worklab --import home-public.gpg

$ gpg2 --homedir ./worklab --import-ownertrust home-trust.txt

$ gpg2 --homedir ./worklab -k --keyid-format 0xshort

$ gpg2 --homedir ./worklab --sign-key 0xB8F76730

Экспортируем подписанные ключи.

$ gpg2 --homedir ./worklab --output ./home-sign-public.gpg --armor --export 0xB8F76730

$ gpg2 --homedir ./homelab --output ./work-sign-public.gpg --armor --export 0xDF679B3B

Обмениваемся между пользователями и заново импортируем.

$ gpg2 --homedir ./worklab --import work-sign-public.gpg

$ gpg2 --homedir ./homelab --import home-sign-public.gpg

По умолчанию любой **PGP-ключ** на начальном этапе, который должен быть загружен на публичный сервер ключей имеет уровень доверия 3— ограниченно доверен.

Между двумя ключами **PGP** существует путь доверия. Путь доверия — это цепочка подписей, где **A** подтверждает, что личность **B** является доверенной, **B** подтверждает, что **C** является доверенной, **C** подтверждает **D** и так далее. По этому пути мы можем сделать вывод ключ **PGP** контролируется настоящим человеком.

Поэтому если вы всё-таки планируете загрузить свои публичные ключи на публичный сервер, то уровень доверия

необходимо всегда ставить **не выше 4-го**. О публичных серверах читайте в части №3.

Таким образом ваш уровень доверия рано или поздно автоматически возрастёт к 5-му, абсолютному. Но такие ключи достаточно редкие. Чаще всего такие встречаются только у пользователей с личным использованием, не публичным.

Часть №2. Практика.

Подпишем и зашифруем архив в консоли для одной домашней директории ключей и расшифруем в другой.

$ gpg2 --homedir ./homelab -se -r 0xDF679B3B archive.tar.gz

-s — подпись,

-e — шифрование,

-r — указать User-ID.

Я же использовал короткую версию идентификатора. Полученный gpg файл уже свободно можно отправлять второму человеку, чей публичный ключ мы использовали для шифрования.

Для теста удаляю исходный архив.

$ shred -u archive.tar.gz

Человек получил файл и расшифровывает его своим приватным ключом. Одновременно происходит проверка подписи.

$ gpg2 --homedir ./worklab -d archive.tar.gz.gpg > archive.tar.gz

Удаляю не нужные шифрованные данные.

$ shred -u archive.tar.gz.gpg

Если вам нужно только зашифровать файл, то воспользуйтесь следующей командой. А далее можете поступить более хитро, и создать отделённую подпись на сам архив, а не на зашифрованный файл или сразу шифровать и подписывать. Вот пример.

$ gpg2 --homedir ./homelab -e -r 0xDF679B3B archive.tar.gz

Обратите внимание, что в любом варианте с шифрованием или без подпись происходит вашим ключом, а шифрование чужим публичным!

$ gpg2 --homedir ./homelab -b -r 0xB8F76730 archive.tar.gz

Для теста удаляю исходный архив.

$ shred -u archive.tar.gz

И теперь также **gpg** и **sig** файлы можно отправлять конечному пользователю, чей публичный ключ, мы использовали.

Теперь о преимуществах. Сначала расшифровываем файл, затем проверяем расшифрованный файл на соответствие, что файл в процессе передачи к конечному пользователю никоим образом не был изменён.

Человек получил файл, расшифровывает и проверяет его.

$ gpg2 --homedir ./worklab -d archive.tar.gz.gpg > archive.tar.gz

$ gpg2 --homedir ./worklab --verify archive.tar.gz.sig archive.tar.gz

Если файл в процессе передачи был как-то изменён или вы пересоздали архив а подпись пересоздать забыли, то подпись и файл не будут соответствовать друг другу и подпись к файлу будет **ПЛОХАЯ**. Это хороший способ отслеживания изменений, что вы ничего не забыли сделать.

Если вы хотите зашифровать директорию, сначала создайте **tar** архив, т.е. вы просто объединяете ваши директории в один большой файл. И затем уже можете шифровать и расшифровывать.

$ tar -cf test.tar test/

$ tar -xvf test.tar

Часть №3. Публичные сервера ключей.

Не обязательно обмениваться публичными ключами только через электронную почту или какое-нибудь облачное хранилище. Ими можно обмениваться через публичный сервер ключей. Сервера легко можно на**Googl**-ить. По имеющимся у вас коротким идентификаторам **0xShort**, **fingerprint** отпечаткам или **адресу электронной почты** вы легко можете как выгрузить любой ваш публичный **GPG-**ключ на публичный сервер или наоборот скачать.

Для выгрузки на сервер, например директория **homelab**:

$ gpg2 --homedir ./homelab --keyserver pgp.mit.edu --send-keys 0xB8F76730

Для загрузки с сервера, например директория **homelab**:

$ gpg2 --homedir ./homelab --keyserver pgp.mit.edu --receive-keys 0xB8F76730

Можете также загрузить себе мой публичный **GPG**-ключ:

$ gpg2 --keyserver pgp.mit.edu --receive-keys 0x96E3C043

gpg2 --keyserver pgp.mit.edu --receive-keys maximalis171091@yandex.ru

gpg2 --keyserver pgp.mit.edu --receive-keys 1DEC8548C828F98EBDE8D46ABE2BA3C896E3C043

$ gpg2 -k --keyid-format 0xshort

man gpg

Ну а сегодня на этом всё. Надеюсь я вас заинтересовал. Всем Удачи! Пока-пока!