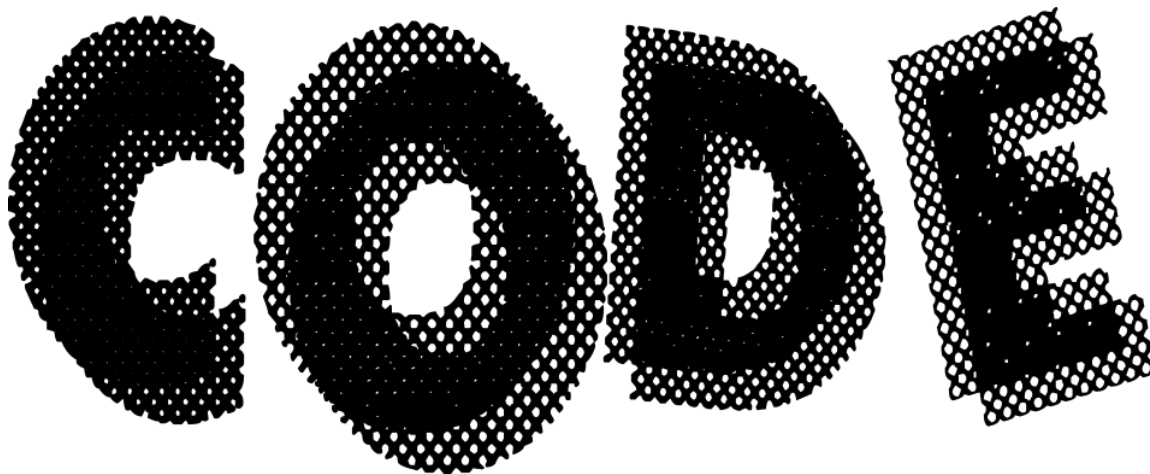


Техно-блог для наших техно-времен



Кертис Колликатт



Информационный бюллетень TIDAL SERIES



LinkedIn



GitHub

gpg-zip

04 мая 2014 г.

У меня есть выделенный сервер, который я использую для размещения нескольких вещей, например, сайтов друзей на WordPress, нескольких личных сайтов и т. д. Одной из услуг, предоставляемых хостинговой компанией, является доступ к резервному серверу по FTP, т. е. они предоставляют мне немного бесплатного места для резервного копирования файлов.


Но я не хочу просто сбрасывать туда файлы резервных копий в открытом виде – я хочу, чтобы они были зашифрованы с помощью gpg. Введите gpg-zip!

gpg-zip – удобная утилита, которая может заархивировать каталог, а затем зашифровать полученный файл. Затем этот файл можно

разместить где угодно, и если gpg работает правильно, то если только у кого-то нет доступа к закрытому ключу открытого ключа, которым он был зашифрован (или они не были указаны в качестве получателя), они не смогут его расшифровать. Важно отметить, что зашифрованный файл безопасен только настолько, насколько безопасен закрытый ключ. Очевидно, что если у кого-то есть доступ к закрытому ключу, то они смогут расшифровать файл.

Во-первых, у меня есть ключ, который я собираюсь использовать для этого примера. Создание ключей и подключей безопасно выходит за рамки этой статьи. Вот хорошая запись в блоге о [создании новых ключей gpg](#) , и еще одна [здесь](#) .

Ниже приведен ключ, который я буду использовать.

A terminal window with a dark background and light-colored text. The command '# gpg --list-keys /root/.gnupg/pubring.gpg' has been executed. The output shows three lines of key information: 'pub 2048R/4FCDA707 2014-05-04', 'uid curtis <curtis-backups@server', and 'sub 2048R/25AEB942 2014-05-04'. The terminal has a scrollbar at the bottom.

```
# gpg --list-keys
/root/.gnupg/pubring.gpg
-----
pub  2048R/4FCDA707 2014-05-04
uid  curtis <curtis-backups@server
sub  2048R/25AEB942 2014-05-04
```

Важно отметить, что этот ключ сам по себе не зашифрован – у него нет установленной парольной фразы, поэтому его может использовать любой, у кого есть доступ к копии ключа. Это нехорошо делать с важным ключом, но в этом случае я собираюсь автоматизировать этот процесс, и нет хорошего способа, насколько мне известно, чтобы автоматический процесс расшифровывал ключ без сохранения парольной фразы в открытом виде. При этом шифрование файла не требует пароля для разблокировки закрытого ключа, но расшифровка в автоматическом режиме потребует.

Давайте создадим тестовый каталог с несколькими файлами в gpg-zip.

```
# cd /tmp; mkdir test
# for i in $(seq 1 100); do echo "hi $i" > test/$i
# ls test | wc -l
100
```

Теперь, когда у нас есть каталог, заполненный файлами для резервного копирования, его можно зашифровать с помощью gpg. По сути, я шифрую файл, указав себя в качестве получателя, т. е. сообщение самому себе.

```
# gpg-zip --encrypt --recipient curtis-backup@serverascode.com test.tar.gz
# file test.tar.gz.gpg
test.tar.gz.gpg: data
```

Я также могу использовать gpg-zip для получения списка файлов (но, очевидно, я смогу сделать это только в том случае, если смогу их расшифровать):

```
# gpg-zip --list-archive test.tar.gz.gpg | tail
gpg: encrypted with 2048-bit RSA key, ID 25AEB942,
      "curtis <curtis-backup@serverascode.com>"
test/57.txt
test/26.txt
test/84.txt
test/6.txt
test/16.txt
test/4.txt
test/18.txt
test/20.txt
test/45.txt
test/76.txt
```

И я тоже могу восстановить файлы. Я сделаю это в каталоге восстановления.

```
# mkdir restore; cd restore
# gpg-zip --decrypt /tmp/test.tar.gz.gpg
gpg: encrypted with 2048-bit RSA key, ID 25AEB942,
      "curtis <curtis-backup@serverascode.com>"
test/
test/22.txt
test/74.txt
test/47.txt
test/5.txt
SNIP!
```

Выглядит хорошо. Теперь, когда я понял, как работает gpg-zip, я могу начать автоматизировать резервное копирование, шифровать их и отправлять на удаленный ftp-сервер.

Заключение

gpg-zip – это простой способ шифрования каталогов. Полученный файл затем может быть отправлен по (незашифрованному) проводу и сохранен на удаленной системе, и я должен быть достаточно уверен, что если кто-то тайно не получит доступ к моему серверу и не украдет мои ключи gpg, файл останется в безопасности. Конечно, есть несколько оговорок, но в целом я считаю этот рабочий процесс разумным.

Если вы заметили какие-либо ошибки или другие проблемы в этом посте, пожалуйста, дайте мне знать в комментариях. :)

253