Manual de GPG: cifra, firma y envía datos de forma segura

NOTA: Está práctica es una adaptación del manual de GENBETA (https://www.genbeta.com/desarrollo/manual-de-gpg-cifra-y-envia-datos-de-forma-segura) a Ubuntu Server 20.04.

¿Qué es GnuPG?

Antes de empezar con lo interesante tenemos que saber que es GPG (GNU Privacy Guard), que es un derivado libre de PGP y su utilidad es la de cifrar y firmar digitalmente, siendo además multiplataforma (podéis descargarlo desde la página oficial) aunque viene incorporado en algunos sistemas Linux, como en Ubuntu (será con el sistema que haré todos los ejemplos, en Windows se encuentra solo con gestor gráfico).

Anillo de claves

GPG tiene un repositorio de claves (anillo de claves) donde guarda todas las que tenemos almacenadas en nuestro sistema, ya sean privadas o públicas.

Más adelante cuando veamos un anillo de claves debemos de recordar que pub hace referencia a la clave pública y sub hace referencia a la privada (y que tenemos que tener a buen recaudo).

Servidores de claves

Para que nos cifren un mensaje tenemos que compartir la clave pública de nuestro par de claves para cifrar, y como es un poco engorroso difundir una clave a muchas personas existen los servidores de claves PGP (compatibles con GPG), donde subiré una clave pública para el que quiera probar los ejemplos.
Unos ejemplos de servidores son estos: pgp.rediris.es (español, aunque falla algunas veces) o pgp.mit.edu (americano, del MIT y a mi no me ha dado problemas).

Cifrado simétrico

Como ya sabéis el cifrado simétrico es el tipo de cifrado más sencillo que hay, es más rápido de procesar y por desgracia menos seguro que el cifrado asimétrico.

Para empezar la prueba tenemos que tener un archivo de cualquier tipo e introducir en la terminal de Linux el comando gpg con el parámetro -c para cifrar y -d para descifrar.

```
pedro@ubuntu:~/gpg$ echo "Genbeta Dev" > texto.txt`
pedro@ubuntu:~/gpg$ gpg -c texto.txt
```

Tras crear un archivo de texto usamos el comando gpg -c [archivo], nos aparecerá un cuadro que nos pide la contraseña y se generará un archivo .gpg. Y después lo descifraremos con el comando gpg -d [archivo] (e introduciendo la clave de alta seguridad, en este caso qwerty).

```
pedro@ubuntu:~/gpg$ gpg -d texto.txt.gpg
gpg: AES256 encrypted data
gpg: encrypted with 1 passphrase
Genbeta Dev
```

Podéis probar a descifrar este archivo usando la clave qwerty.

Cifrado asimétrico

Generar las claves

Para poder cifrar asimétricamente primero tenemos que crear la pareja de claves (pública y privada) con el comando gpg —full-generate-key.

```
pedro@ubuntu:~/gpg$ gpg --full-generate-key
gpg (GnuPG) 2.2.12; Copyright (C) 2018 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
Your selection?
```

GPG nos permite elegir el tipo de clave que queremos usar, hay opciones que solo permiten firmar y otras que permiten firmar y cifrar, en este caso usaremos DSA y Elgamal (2).

```
DSA keys may be between 1024 and 3072 bits long.
What keysize do you want? (2048)
```

Nos piden el tamaño de la clave que puede variar entre 1024 bits y 3072, esto es de libre elección, yo tomaré el término medio que es el que propone por defecto (2048).

A partir de aquí todo es más trivial, nos pide la fecha en la que expirará la clave (si ponemos 0 no caducará), el nombre del emisor de la clave, su correo, un comentario si queremos poner información extra y por último nos pedirá la contraseña que salvaguarda la clave privada.

Tras generar las claves podemos verlas con el comando gpg -k que nos muestra nuestro anillo de claves, lo importante de este paso es que veremos la identificación de cada una (uid), que es necesaria para poderlas exportar y enviar.

```
pedro@ubuntu:~/gpg$ gpg -k
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
/home/david/.gnupg/pubring.kbx
------------------------------
pub   dsa2048 2019-12-13 [SC]
      E18907E201530CD65CF293F7AACB4DEBB8B30D17
uid           [ultimate]  Pedro Gutiérrez <info@xitrus.es>
sub   elg2048 2019-12-13 [E]
```

Exportar y enviar la clave privada

El objetivo de esta pareja de claves es que cualquiera nos pueda mandar un archivo cifrado que solo veremos nosotros y esto se hace difundiendo la clave pública que acabamos de crear (la pública, nunca la privada), para exportarla en un archivo usaremos el comando gpg -output [archivo destino] —export [Nombre del emisor] (la clave pública generada antes tiene el Nombre Pedro Gutiérrez).

```
pedro@ubuntu:~/gpg$ gpg --output CPub.gpg --export "Pedro Gutiérrez"
pedro@ubuntu:~/gpg$ ls
CPub.gpg
```

Este archivo ahora se puede difundir por el medio que queramos, tenemos que tener en cuenta que el único problema de seguridad que habría en difundir la clave es que alguien se hiciese pasar por otro al mandarnos un mensaje, algo que pasaría igual si no estuviese cifrado, por eso el que nos envíe algo lo debería de firmar (si fuese pertinente).

Subir una clave pública a un servidor de claves

Los servidores de claves suelen ser de acceso público (al no haber mucho problema por difundir una clave pública) y en este caso subiremos una clave a los servidores del MIT (pgp.mit.edu) usando el comando gpg —send-keys —keyserver [Dirección del servidor] [ID de la clave pública] (en este caso el ID de antes es E18907E201530CD65CF293F7AACB4DEBB8B30D17).

```
pedro@ubuntu:~/gpg$ gpg --send-keys --keyserver pgp.rediris.es E18907E201530CD65CF293F7AACB4DEBB8B30D17
gpg: sending key AACB4DEBB8B30D17 to hkp://pgp.rediris.es
```

A partir de este momento la clave estará accesible desde este servidor específico.

Importar la clave desde el archivo o servidor de claves

Para poder usar la clave pública para cifrar o comprobar la identidad del remitente tenemos que importar previamente la clave, desde un archivo debemos de usar el comando gpg —import [Archivo de la clave pública] (el que hemos descargado anteriormente).

```
pedro@ubuntu:~/gpg$ gpg --import CPub.gpg
gpg: key AACB4DEBB8B30D17: "Pedro Gutiérrez <info@xitrus.es>" not changed
gpg: Total number processed: 1
gpg:              unchanged: 1
```

Al tener la clave ya en mi anillo de claves me contesta que no hay cambios.
Para realizar la importación desde el servidor tenemos que usar el comando gpg —keyserver [Dirección del servidor] —recv-keys [ID de la clave].

```
pedro@ubuntu:~/gpg$ gpg --keyserver pgp.rediris.es --recv-keys AACB4DEBB8B30D17
gpg: key AACB4DEBB8B30D17: "Pedro Gutiérrez <info@xitrus.es>" not changed
gpg: Total number processed: 1
gpg:              unchanged: 1
```

Como podemos ver al tener ya la clave nos devuelve el mismo mensaje.

Cifrar con la clave pública

Ahora tenemos que pensar que hemos importado una clave pública, por ejemplo de nuestro jefe y tenemos que mandarle un documento, para cifrar el documento usaremos el comando gpg —encrypt —recipient [ID de la clave] [Archivo]

```
pedro@ubuntu:~/gpg$ echo "Genbeta Dev" > documento.txt
pedro@ubuntu:~/gpg$ gpg --encrypt --recipient AACB4DEBB8B30D17 documento.txt
pedro@ubuntu:~/gpg$ ls
documento.txt documento.txt.gpg
```

Y ya tenemos el archivo listo para mandarlo de forma segura.

Descifrar un archivo con la clave privada

Y ahora es el momento de descifrar con nuestra clave privada el documento tras recibirlo, con el comando gpg -d [Archivo] e introduciendo la contraseña que creamos para salvaguardar la clave privada.

```
pedro@ubuntu:~/gpg$ gpg -d documento.txt.gpg
gpg: encrypted with 2048-bit ELG key, ID C7614DD38546B3DB, created 2019-12-13
      "Pedro Gutiérrez <info@xitrus.es>"
Genbeta Dev
```

Y el resultado nos lo muestra a continuación (Genbeta Dev), aunque si queremos especificar la salida debemos de usar el parámetro -o [Archivo de salida].

Firmar archivos

Una de las medidas de seguridad básicas al pasar un mensaje es asegurarnos que el emisor es quien dice ser, para asegurarnos de esto digitalmente existe la firma digital, podemos cifrarlo y a su vez firmarlo, que es lo que haremos con el comando gpg -u [ID de la clave] —output [Archivo resultante] —sign [Archivo para firmar] e introduciendo la contraseña de la clave privada.

```
pedro@ubuntu:~/gpg$ echo "Genbeta Dev" > firmar.txt
pedro@ubuntu:~/gpg$ gpg -u AACB4DEBB8B30D17 --output firmar.txt.gpg --sign firmar.txt
```

Y ahora para asegurarse la confidencialidad del documento (ahora que esta firmado por nosotros) deberíamos de cifrarlo con la clave pública del destinatario.

Verificar y descifrar un archivo firmado

Cualquiera con la clave pública asociada a la que ha firmado el documento puede leerlo, de la misma forma que desciframos un archivo (gpg -d [Archivo]) o verificándolo únicamente con el comando gpg —verify [Archivo].

```
pedro@ubuntu:~/gpg$ gpg --verify firmar.txt.gpg
gpg: Signature made vie 13 dic 2019 10:52:06 UTC
gpg:                using DSA key E18907E201530CD65CF293F7AACB4DEBB8B30D17
gpg: Good signature from "Pedro Gutiérrez <info@xitrus.es>" [ultimate]
```

Y el resultado es la información del remitente.

Resumen

GPG es una herramienta de cifrado muy potente y fácil de usar, que en principio, a la mayoría no nos hace falta, pero puede que se nos presente la necesidad de enviar algo por medio inseguros (porque no haya más remedio), haciéndolo de esta forma podremos hacerlo sin miedo ha que lean el contenido del archivo o nos den el cambiazo.
Este es un tema bastante extenso y si no lo acabas de entender puede ser un lío, pero para preguntar están los comentarios.