From eloy@uco.es Mon Jan 28 17:27:44 2002
Date: Wed, 13 Sep 2000 10:43:05 +0200 (CEST)
From: eloy <eloy@uco.es>
Reply-To: ersanz@uco.es
To: licor@licor.ods.org
Subject: [Licor] Curso GPG -- 1

Saludos, Pingüínos.

Como dijimos ayer en la reunión, vamos a hacer unas cuantas prácticas con GPG, el sustituto libre de PGP.

En primer lugar, aseguráos de que tenéis instalada la versión 1.0.1 al menos (parece que la 1.0.2 ya está publicada). En Debian viene seguro. En las demás creo que también. De todas formas, tenéis el código fuente en http://www.gnupg.org .

Cuando lo tengáis instalado, lo primero es crear un par de claves propio. No vamos a entrar en el funcionamiento de los sistemas de clave pública, pero baste decir que para usarlos necesitamos un par de claves: la privada y la pública. La pública hay que distribuirla lo más posible, ya que si quieren enviarnos un mensaje deben usarla. La privada, por supuesto, hay que guardarla bien.

Para crear la pareja de claves, ejecutad

```
gpg --gen-key
```

Posiblemente os diga que lo ejecutéis de nuevo para releer el fichero de opciones que acaba de crear. En fin... Os pedirá a continuación el tipo de clave que queréis. Seleccionad claves DSA y ElGamal (opción 1, por defecto). Lo siguiente es el tamaño de clave. Con 1024 bits es suficiente. Decidle que la clave es permanente (no expira).

Luego os va a pedir el identificador de usuario en tres partes: vuestro nombre, vuestra dirección de correo y un comentario.

A continuación os pedirá la frase de paso para proteger vuestra clave privada. Elegid una buena, tipo 'sesamo'... es broma ;-).

Entonces comienza la generación de la clave. Saldrán un montón de puntos y '+' y otras cosas mientras se obtienen datos aleatorios para la generación. Cuando esté terminada, ya tenéis vuestro par de claves. Bienvenidos a la seguridad.

Vamos a confirmar que estan ahí:

```
gpg --list-keys
```

debe mostraros la clave pública con su subclave (pub y sub) y

```
gpg --list-secret-keys
```

os mostrará la clave privada con su subclave (sec y ssb).

Realmente, estas dos órdenes lo que muestran son los anillos (almacenes) público y privado. Pronto tendréis más claves en vuestro anillo público.

No sirve de nada tener una clave pública si no se publica. Vamos a ello:

```
gpg --keyserver www.rediris.es --send-keys tu_identif
```

envía tu clave pública al servidor de claves que hay en rediris. Debes (claro) estar conectado. Puedes referirte a tu clave (tu_identif) de varias formas distintas: con tu dirección de correo electrónico, con tu nombre, con el identificador de clave (míralo en gpg --list-keys, después del '1024D/')...

Para confirmar que está enviada, conecta con http://www.rediris.es/cert/keyserver/ y consúltalo.

Otra forma de exportar tu clave pública, esta vez a un fichero, para ponerla (por ejemplo) en tu página:

```
gpg --armor --export tu_identif  >  clave-gpg.txt
```

Creo que con esto tenemos para empezar. Os propongo lo siguiente: cread vuestras claves, exportadlas a rediris, ponedlas en vuestra página y enviadlas a la lista (adjuntanto el fichero clave-pgp.txt). Cuando tenga unas cuantas, continuaremos con el 'cursillo'. ¿OK?.

Animáos.

-- Eloy

---- Eloy Rafael Sanz Tapia -- ersanz@uco.es -- ma1satae@uco.es -----
------------------ http://www.uco.es/~ma1satae ----------------------
----------- GPG ID: 190169A0 / finger eloy@rabinf50.uco.es ----------
Córdoba _ España _____ Debian 2.2 GNU/Linux 2.2.16 rabinf50

From eloy@uco.es Mon Jan 28 17:27:52 2002
Date: Mon, 18 Sep 2000 09:37:46 +0200 (CEST)
From: eloy <eloy@uco.es>
Reply-To: ersanz@uco.es
To: licor@licor.ods.org
Subject: [Licor] Curso GPG -- 2

Volvamos a la carga.

En la entrega pasada del curso explicamos cómo crear una pareja de claves (pública y privada), cómo ver el contenido de nuestros anillos (público y privado) y cómo exportar claves públicas a un fichero o a un servidor de claves en Internet. En este segundo 'capítulo' veremos cómo importar nuevas claves para introducirlas en nuestro anillo público y cómo encriptar ficheros.

Tenemos como mínimo dos formas de obtener la clave pública de alguien a quien queramos enviar un mensaje: podemos obtenerla de un servidor de claves (como www.rediris.es) o podemos tener un fichero texto con la clave pública. Este fichero texto puede que nos lo hayan enviado por correo, como habéis hecho ya varios, o puede que lo hayamos bajado de la página web del propietario de la clave.

En el primer caso, para tomar la clave pública de un servidor e insertarla en nuestro anillo público, necesitamos el identificador de la clave. El identificador es el que aparece justo después del '1024D/' en el listado de 'gpg --list-keys'. ¿Cómo obtener el identificador de clave de alguien cuya clave pública no tenemos?. Pues probablemente porque lo tenga puesto en su firma de correo (mirad en la mía ahí abajo). Una vez que tenemos el identificador de la persona cuya clave queremos, cogemos la clave:

```
gpg --keyserver www.rediris.es --recv-keys ID
```

donde ID es el identificador de clave que acabamos de obtener.

¿Y si tenemos la clave en un fichero?. Fácil. Supongamos que el fichero se llama 'clave.gpg'. Para importarlo en nuestro anillo público sólo tenemos que escribir:

```
gpg --import clave.gpg
```

Y ya está.

Y ahora vamos a lo divertido. Vamos a encriptar un mensaje. De momento, os voy a pedir que encriptéis uno con un único destinatario: el que subscribe.

Así que aseguráos de que tenéis mi clave pública, coged cualquier fichero que tengáis a mano (¡que no pese cinco megas!) y encriptadlo. Para encriptar un mensaje en un sistema de clave pública, como los usados por gpg, hay que especificar el destinatario. Y el destinatario es la única persona que, usando su clave privada, puede abrir el mensaje.

Vamos, que lo que tendríais que escribrir es algo como:

```
gpg --recipient eloy --encrypt fichero
```

Y esto os generaría un fichero con nombre 'fichero.gpg' encriptado con la clave pública de Eloy. Donde pone 'eloy' podéis poner cualquier cosa que identifique al usuario: su identificador de clave, su dirección de correo electrónico, su nombre...

Puede que gpg se queje de que no es seguro que la clave pública de Eloy que tenéis sea realmente la suya. ¿Quién os dice que no os han engañado y os han dado una clave pública de alguien que está interceptando el correo de Eloy y que, por tanto, ahora ya sería capaz de desencriptar los mensajes que enviéis a Eloy?. Más adelante veremos algo sobre las relaciones de confianza y la firma de claves.

Con esto terminamos por hoy. Encriptad esos ficheros y enviádmelos.

Saludos.

-- Eloy

---- Eloy Rafael Sanz Tapia -- ersanz@uco.es -- ma1satae@uco.es -----
---------- http://www.uco.es/~ma1satae ----------
---------- GPG ID: 190169A0 / finger eloy@rabinf50.uco.es ----------
Córdoba _ España _____ Debian 2.2 GNU/Linux 2.2.16 rabinf50

From eloy@eloy.ayrna.org Mon Jan 28 17:28:06 2002
Date: Mon, 16 Oct 2000 20:22:11 +0200 (CEST)
From: eloy <eloy@eloy.ayrna.org>
Reply-To: ersanz@uco.es
To: licor@licor.ods.org
Subject: [Licor] Curso GPG -- 3

Tras un paréntesis, vamos a continuar con el curso de GnuPG.

He recibido algunos ficheros encriptados y los he podido desencriptar sin problemas. En el capítulo anterior vimos cómo se podían obtener claves públicas para incorporarlas a nuestro anillo. Podíamos recogerlas de un servidor de claves o tomarlas de un fichero que nos hubieran enviado.

¿Cómo hacemos para desencriptar un mensaje que hayamos recibido?. La forma más directa consiste en guardar el mensaje en un fichero (llamémoslo f.txt) y ejecutar el siguiente comando:

```
gpg -o f.txt.decodif f.txt
```

La opción -o y su argumento 'f.txt.decodif' no son necesarios, pero si el fichero de entrada no tiene la extensión '.gpg' son aconsejables.

Hecho esto, tendremos un fichero llamado 'f.txt.decodif' con el mensaje decodificado. Así de fácil.

Hay que decir que gpg tiene un comportamiento especial cuando le pedimos que decodifique un fichero y encuentra que no tiene la clave pública de la persona que nos lo envió. En ese caso, trata de conectarse a un servidor de claves para obtenerla. Si no estamos conectados a Internet, la decodificación fallará. Hay que tener en cuenta ese comportamiento.

Vamos a pasar a los dos últimos puntos del minicursillo. Primero trataremos brevemente el asunto de la confianza en las claves y por último explicaremos cómo hacer para revocar una clave pública, en caso de que nunca la vayamos a volver a usar.

¿Cómo estamos seguros de que una clave que hemos obtenido a través de un servidor de claves corresponde realmente a su propietario (llamémoslo A)?. Podría haber sido puesta en el servidor por alguien (llamémoslo B) que trata de hacerse pasar por A, y si cogemos la clave sin más y enviamos mensajes encriptados con ella, B podría interceptarlos y decodificarlos. Claro está que nos daríamos cuenta del engaño en cuanto A nos escribiera diciéndonos que

a) no recibe nuestro mensaje, o
b) lo recibe pero no puede decodificarlo (claro: lo intenta decodificar con una clave que no es la correcta).

Pero para entonces el daño ya estaría hecho: B tendría nuestro mensaje.

Lo que necesitamos es una forma de asegurarnos de que la clave de A que tenemos en nuestro anillo público es realmente la suya. La mejor manera de hacer esto es obtener la huella de la clave usando el comando

```
gpg --fingerprint <id-usuario-A>
```

Esto genera algo como:

```
pub  1024D/190169A0 1999-09-10 A. Nónimo <anon@yoquese.es>
       Key fingerprint = A707 4004 B956 37BD B3C1  1F47 26C8 851C 1901 69A0
sub  1024g/CFD3A2E8 1999-09-10
```

Pues el número hexadecimal de 20 bytes (=40 dígitos hexadecimales) que hay en la segunda línea es la huella de la clave pública de ese usuario. Teniendo eso, podemos llamar a A por teléfono y que nos diga los dígitos que componen su huella, para comprobar que coinciden con los que tenemos en nuestro anillo. Una vez que estemos seguros de que tenemos la clave correcta, podemos firmarla. Firmar una clave es avalar su autenticidad. Con la firma estamos diciendo 'yo avalo que ésta es la clave pública de A'. Para firmar una clave, usamos

```
gpg --lsign-key <id-usuario-A>
```

Existen diferencias entre las opciones '--lsign-key' y '--sign-key'. La primera firma de manera local, para uso privado, digamos. La segunda permite que se exporte a Internet la clave de A con nuestro aval. Si alguien confía bastante en nosotros y ve que hemos firmado la clave de A, confiará también en que esa es la clave real de A. Como veis, esto se puede complicar de forma enorme. Vamos a dejarlo aquí, mencionando sólo que a través de estas relaciones de confianza se puede tejer una red que permita asegurar de forma rápida si una clave nueva que nos llegue pertenece o no a la persona que esperamos.

Y ahora, tras un buen tiempo usando nuestra querida clave, un buen día pasa algo y decidimos empezar a usar una nueva. Puede que hayamos cambiado de dirección de correo (pero la verdad es que entonces podríamos añadir una nueva identidad a la clave existente, algo que no vamos a ver). O puede que alguien nos haya entrado en el ordenador y nos haya robado la clave privada. En fin... Que creamos una nueva clave y ya está. ¡No!. Nuestra anterior clave sigue circulando por los servidores de Internet y si alguien nos escribe usándola, el mensaje puede caer en manos del tipo que nos robó la clave secreta, y éste podría desencriptarlo.

Lo que tenemos que hacer es revocar nuestra clave anterior. El proceso de revocación implica varios pasos. Como llevo un buen rato pegándome con el manual, prefiero dejarlo para...

¡LA ULTIMA PARTE DEL MINICURSO!
Próximamente en sus pantallas

Saludos.

-- Eloy

---- Eloy Rafael Sanz Tapia -- ersanz@uco.es -- ma1satae@uco.es -----
------------------ http://www.uco.es/~ma1satae ----------------------
---------- GPG ID: 190169A0 / finger eloy@rabinf50.uco.es ----------
Córdoba _ España _____ Debian 2.2 GNU/Linux 2.2.16 rabinf50