ASEGURANDO DATOS MEDIANTE CIFRADO

Peso	3
Topico Cubierto	110.3 Asegurando datos mediante cifrado
Descripcion	Los candidatos deberán ser capaces de usar técnicas de llave publica para asegurar datos y comunicaciones.
Temas	* Realizar configuraciones y usos básicos de GnuPG
Ejemplos	* gpg
	~/.gnupg/*

*

Peso: Indica el valor de importancia que tiene este tópico en la certificacion.

Tópico Cubierto: Indica según el programa de certificacion LPI que topico le corresponde a este tema.

Descripción: Un resumen de lo que se verá.

Temas: Un resumen de los conceptos primordiales que están cubiertos.

Ejemplos: Palabras claves que se tienen que tener en cuenta.

INTRODUCCIÓN

En este topico vamos a explicar sobre como usar la herramienta GnuPG y su funcionamiento.

GnuPG

GnuPG es una aplicación que sirve para encriptar mensajes, documentos … GnuPG usa un sistema de claves públicas lo que quiere decir que cada usuario tiene una clave privada y una clave pública.
La clave privada es la que se usa para desencriptar aquello que nos envían encriptado con nuestra clave pública, La clave privada es una clave que solo ha de conocer el propietario ya que si alguien más la conociese podría desencriptar lo que nos mandan encriptado.
La clave pública es la que se da a la gente para que nos manden cosas encriptadas y usarán para encriptar aquello que nos quieran pasar.

FUNCIONAMIENTO

GNU Privacy Guard (GPG) es una implementación de código abierto de la norma OpenPGP (RFC 4880) ,que le permite cifrar y firmar digitalmente los datos y las comunicaciones.

Por ejemplo, usted puede cifrar los archivos en el sistema de archivos Linux. También puede cifrar y firmar digitalmente mensajes de correo electrónico.

GPG cifra los mensajes usando pares de claves individuales asimétricas generadas por los usuarios. Las claves públicas pueden ser compartidas con otros usuarios de muchas maneras, un ejemplo de ello es depositándolas en los servidores de claves. Siempre deben ser compartidas cuidadosamente para prevenir falsas identidades por la corrupción de las claves públicas. También es posible añadir una firma digital

criptográfica a un mensaje, de esta manera la totalidad del mensaje y el remitente pueden ser verificados en caso de que se desconfíe de una correspondencia en particular.

GnuPG también soporta algoritmos de cifrado simétricos, por ejemplo CASTS.

GPG no usa algoritmos de software que están restringidos por patentes, entre estos se encuentra el algoritmo de cifrado IDEA que está presente en PGP casi desde sus inicios. En su lugar usa una serie de algoritmos no patentados como ElGamal, CAST5, Triple DES (3DES), AES y Blowfish. También es posible usar IDEA en GPG descargando un plugin extra, sin embargo este puede requerir una licencia para usuarios de algunos países en donde esté patentada IDEA.

GPG es un software de cifrado híbrido que usa una combinación convencional de criptografía de claves simétricas para la rapidez y criptografía de claves públicas para el fácil compartimiento de claves seguras, típicamente usando recipientes de claves públicas para cifrar una clave de sesión que es usada una vez. Este modo de operación es parte del estándar OpenPGP y ha sido parte del PGP desde su primera versión.[2]

GPG utiliza muchos algoritmos de encriptación, incluyendo los siguientes:

Cifrado simétrico

§ AES

§ 3DES

§ Blowfish

Cifrado asimétrico

§ Elgamal

§ RSA

§ Hashes:

§ MD5

§ SHA-1 and -2

§ RIPEMD-160

Las firmas digitales

§ DSA

§ RSA

CONCEPTOS BÁSICOS

Sistemas de Claves Públicas

Para poder entender mejor el sistema de codificación usado por los sistemas de claves asimétricas (ie. claves públicas y privadas), es necesario entender las diferencias con los sistemas de claves simétricas (ie. claves secretas).

Los sistemas de cifrado con clave simétrica son aquéllos en los que la clave que se usa para cifrar una serie de datos, es la misma que la que se usará para descifrar estos datos. En el caso del correo electrónico, el remitente cifraría el mensaje con una clave secreta, y para que el destinatario pueda descifrarlo, necesitaría haber obtenido previamente esta misma clave de un modo «seguro», o sea de modo que la clave no haya

podido ser interceptada durante la entrega. Si no tenemos la completa seguridad de que el intercambio de la clave ha sido seguro, la validez de este sistema es nula.

Por el contrario, los sistemas de cifrado con claves asimétricas usan claves distintas para el cifrado y posterior descifrado de los datos. En un caso como el anterior, el remitente usaría la clave pública del destinatario para cifrar el mensaje, y el destinatario descifraría el mensaje con su propia clave privada. Así pues, la clave privada no debe ser accesible para nadie que no sea el propio dueño de la misma, mientras que la clave pública, puede ser entregada a cualquier persona. En un sistema de cifrado bien implementado, la clave privada no debe derivar nunca de la clave pública.

Firmas Digitales

El concepto de la firma digital se basa en la verificación de la autoría de un mensaje. Esto quiere decir que se puede comprobar que el destinatario del mensaje puede comprobar que el «supuesto» remitente es quien afirma ser. Para ello, el remitente, una vez compuesto el mensaje, lo firma usando su propia clave privada. El destinatario, una vez ha recibido el mensaje, comprobará la veracidad de éste, esto es, lo verificará usando la clave pública del remitente.

Este método es de especial utilidad para reducir riesgos de seguridad en nuestros sistemas (nos podrían enviar un supuesto parche para un programa, y éste en realidad ser un virus o un troyano); también podrían enviarnos información o datos, como provenientes de una fuente lícita o fiable. En ambos casos, no sería muy difícil falsificar la dirección y nombre del remitente, pero sí imposible falsificar la firma digital de éste.

Como ya hemos dicho, la verificación de un mensaje firmado digitalmente se lleva a cabo mediante el uso de la clave pública del remitente sobre el texto del propio mensaje. De este modo no sólo podemos verificar la identidad del autor, sino que también podemos comprobar la integridad del mensaje, ya que la firma digital ha sido generada con el texto y la clave privada. Así pues, una alteración o modificación del texto «a posteriori», o cualquier manipulación del mensaje (especialmente si hacemos uso de las especificaciones MIME/PGP), daría como resultado un error en la verificación.

Anillos de Confianza

Un punto flaco en los algoritmos de clave asimétrica es la transmisión del código público. Es posible que una persona ponga en circulación código con un identificador de usuario falso. Si se codifican mensajes con este pseudo código, el intruso los puede decodificar y leerlos.

La solución PGP (y por consiguiente la solución GnuPG) está en firmar los códigos. La clave pública de un usuario puede estar firmada con las claves de otros usuarios. El objetivo de estas firmas es el de reconocer que el UID (identificador de usuario) de la clave pertenece al usuario a quien dice pertenecer. A partir de ahí es un problema de cada usuario de GnuPG el decidir hasta qué punto se puede fiar de la firma. Una clave se puede considerar fiable cuando se confía en el remitente y cuando se sabe con seguridad que dicha clave pertenece a éste. Sólo cuando se puede confiar plenamente en la clave del firmante, se puede confiar en la firma que acompaña a la clave de un tercero. Para tener la certeza de que la clave es correcta hay que compararla con la huella digital por medio de canales fiables (por ejemplo, podríamos buscar el teléfono en la guía y llamarle, y que nos la dijera de palabra para poder compararla), antes de darle una confianza absoluta.

Límites de Seguridad

Si lo que se desea es mantener la confidencialidad de los datos que se poseen, no basta con determinar qué algoritmo de cifrado se va a usar; también es necesario pensar en la seguridad general del sistema. En principio, PGP está considerado como suficientemente seguro, y hasta el momento no se sabe que haya habido ningún incidente en el que una clave PGP haya sido descodificada. Pero eso no significa que todo lo cifrado sea seguro; si la NSA (Agencia de Seguridad Nacional de los EE.UU.) hubiera conseguido descodificar una clave PGP mediante criptoanálisis, análisis del código, o cualquier otro modo, no es probable que lo hicieran público. Pero aún en el caso de que las claves PGP fueran a todas luces imposibles

de decodificar, otros tipos de ataques a la seguridad pueden ser utilizados. A principios de Febrero fue detectado un troyano que buscaba las claves PGP en el disco duro, y las transfería al atacante mediante FTP. Si en este caso hubiéramos escogido una contraseña débil o fácil, un simple análisis que consistiera en un «ataque de diccionario» la descubriría en poco tiempo.

Otra posibilidad técnica, aunque más difícil, es la de los troyanos que recogen entradas de teclado y las transmiten al asaltante. También es posible, aunque muy difícil, pasar el contenido de una pantalla a otra. En este último caso no sería necesario ningún análisis sobre datos cifrados, ya que se obtendrían «pre-cifrados».

Por todo esto es necesaria una planificación de la seguridad que esté bien prevista y que minimice los riesgos.

La idea no es la de recrear una atmósfera de paranoia entre la gente, sino dejar claro que para implementar un sistema seguro no basta con la instalación de un programa criptográfico, que si bien es un paso hacia un sistema más seguro, no es una solución completa. Troyanos como el aparecido en Marzo de 1999 (Melissa) probaron que muchas compañías no se encuentran preparadas en temas de seguridad.

USO

Creación de claves

Lo primero que hay que hacer una vez que se tiene GnuPG instalado es crear nuestra clave pública y privada. Para hacerlo hay usar el comando gpg --gen-key.

gpg --gen-key

gpg (GnuPG) 1.4.12; Copyright (C) 2012 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?

Al ser la primera vez que se ejecuta nos crea un directorio en el que guardará el fichero de configuración así como los archivos secring.gpg y pubring.gpg. En el primero se almacenarán las claves privadas y en el segundo las claves públicas.

La primera pregunta que hace es qué tipo de clave queremos. Lo normal suele ser seleccionar la primera opción (DSA and ElGamal) que nos permite encriptar y firmar.

```
las claves RSA pueden tener entre 1024 y 4096 bits de longitud.
¿De qué tamaño quiere la clave? (2048) 2048
El tamaño requerido es de 2048 bits
```

La siguiente pregunta es el tamaño de las claves que se puede elegir entre 1024 y 4096 bits. Por defecto se recomienda 2048, a mayor tamaño más segura es la clave. También a mayor tamaño más tiempo lleva encriptar y desencriptar.

```
Por favor, especifique el período de validez de la clave.
         0 = la clave nunca caduca
      <n>  = la clave caduca en n días
      <n>w = la clave caduca en n semanas
      <n>m = la clave caduca en n meses
      <n>y = la clave caduca en n años
¿Validez de la clave (0)? 0
La clave nunca caduca
¿Es correcto? (s/n) s
```

La siguiente pregunta es cuanto tiempo de validez queremos que tenga la clave. La periodicidad se puede poner que no caduque nunca, que dure ciertos semanas, meses o años. En el caso de poner que caduque al cabo de cierto tiempo habrá que volver a generar las claves y volver a mandar la nueva clave pública a aquellos que usaban la que ha caducado. Por defecto viene la opción 0 que es que no caduque nunca.

```
Necesita un identificador de usuario para identificar su clave. El
programa
construye el identificador a partir del Nombre Real, Comentario y
Dirección
de Correo Electrónico de esta forma:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Nombre y apellidos: Juan Educacionit
Dirección de correo electrónico: juan@educacionit.com.ar
Comentario: EducacionIt
Ha seleccionado este ID de usuario:
    "Juan Educacionit (EducacionIt) <juan@educacionit.com.ar>"

¿Cambia (N)ombre, (C)omentario, (D)irección o (V)ale/(S)alir? V
```

Ahora pregunta nuestro nombre y apellidos, dirección de correo y un comentario para la llave. Una vez introducidos todos los datos nos muestra cual es nuestro ID de usuario que lo crea a partir de los datos que le hemos introducido antes. Luego pregunta si queremos cambiar algún dato o si están bien los datos. Si están correctos respondemos “V” y sigue adelante el proceso.

```
Necesita una frase contraseña para proteger su clave secreta.

Es necesario generar muchos bytes aleatorios. Es una buena idea
realizar
alguna otra tarea (trabajar en otra ventana/consola, mover el ratón,
usar
```

```
la red y los discos) durante la generación de números primos. Esto da
al
generador de números aleatorios mayor oportunidad de recoger
suficiente
entropía.
.............+++++
.+++++
Es necesario generar muchos bytes aleatorios. Es una buena idea
realizar
alguna otra tarea (trabajar en otra ventana/consola, mover el ratón,
usar
la red y los discos) durante la generación de números primos. Esto da
al
generador de números aleatorios mayor oportunidad de recoger
suficiente
entropía.
...+++++
.................+++++
gpg: /home/mockbuild/.gnupg/trustdb.gpg: se ha creado base de datos de
confianza
gpg: clave 2320CE4F marcada como de confianza absoluta
claves pública y secreta creadas y firmadas.

gpg: comprobando base de datos de confianza
gpg: 3 dudosa(s) necesarias, 1 completa(s) necesarias,
modelo de confianza PGP
gpg: nivel: 0  validez:   1  firmada:   0  confianza: 0-, 0q, 0n, 0m,
0f, 1u
pub   2048R/2320CE4F 2012-01-29
      Huella de clave = 8391 070D B54A 2D40 8D80  7325 10D9 0352 2320
CE4F
uid                  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>
sub   2048R/211B0879 2012-01-29

[mockbuild@oc6127656113 ~]$
```

Por último se pregunta cuál va a ser el password para nuestra clave privada. Al introducir el password no se ve nada de lo que se escribe ni se ve avanzar el cursor. Después de introducirla nos vuelve a preguntar el password y si coincide con el primer password comienza la generación de las claves. Cuando se produce el proceso de generación de las claves es buena idea reproducir mp3, mover el ratón … para que se generen números aleatorios y se creen antes las claves.

Ver claves públicas disponibles

Para ver las claves públicas que tenemos disponibles hay que hacerlo con el comando gpg –list-keys. Esto lo que haces listar las claves que hay disponibles dentro del fichero pubring.gpg.

```
[mockbuild@oc6127656113 ~]$ gpg --list-keys
/home/mockbuild/.gnupg/pubring.gpg
---------------------------------
pub   2048R/2320CE4F 2012-01-29
uid                  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>
sub   2048R/211B0879 2012-01-29

pub   2048R/C26C23A2 2012-01-29
uid                  Linux EducacionIT (}) <linux@educacionit.com.ar>
sub   2048R/0821FEAA 2012-01-29

[mockbuild@oc6127656113 ~]$
```

El identificador de las claves es lo que hayamos metido en el nombre, en el apellido, en la dirección de correo o el número que aparece después del 1024D al hacer gpg –list-keys. Si en algún caso coincide el ID se mostrarán los que coinciden.

```
[mockbuild@oc6127656113 ~]$ gpg --list-keys Linux
pub   2048R/C26C23A2 2012-01-29
uid                  Linux EducacionIT (}) <linux@educacionit.com.ar>
sub   2048R/0821FEAA 2012-01-29

[mockbuild@oc6127656113 ~]$ gpg --list-keys EducacionIT
pub   2048R/2320CE4F 2012-01-29
uid                  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>
sub   2048R/211B0879 2012-01-29

pub   2048R/C26C23A2 2012-01-29
uid                  Linux EducacionIT (}) <linux@educacionit.com.ar>
sub   2048R/0821FEAA 2012-01-29

[mockbuild@oc6127656113 ~]$ gpg --list-keys linux@educacionit.com.ar
pub   2048R/C26C23A2 2012-01-29
uid                  Linux EducacionIT (}) <linux@educacionit.com.ar>
sub   2048R/0821FEAA 2012-01-29

[mockbuild@oc6127656113 ~]$ gpg --list-keys 0x211B0879
pub   2048R/2320CE4F 2012-01-29
uid                  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>
sub   2048R/211B0879 2012-01-29

[mockbuild@oc6127656113 ~]$ gpg --list-keys "EducacionIt"
pub   2048R/2320CE4F 2012-01-29
uid                  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>
sub   2048R/211B0879 2012-01-29

pub   2048R/C26C23A2 2012-01-29
uid                  Linux EducacionIT (}) <linux@educacionit.com.ar>
sub   2048R/0821FEAA 2012-01-29

[mockbuild@oc6127656113 ~]$
```

Ver claves privadas disponibles

Para ver las claves privadas que tenemos disponibles hay que hacerlo con el comando gpg –list-secret-keys. Esto lo que haces listar las claves que hay disponibles dentro del fichero secring.gpg.

```
[mockbuild@oc6127656113 ~]$ gpg --list-secret-keys
/home/mockbuild/.gnupg/secring.gpg
----------------------------------
sec   2048R/2320CE4F 2012-01-29
uid                  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>
ssb   2048R/211B0879 2012-01-29

sec   2048R/C26C23A2 2012-01-29
uid                  Linux EducacionIT (}) <linux@educacionit.com.ar>
ssb   2048R/0821FEAA 2012-01-29

[mockbuild@oc6127656113 ~]$
```

Borrar claves de los anillos

Se llama anillos a los archivos en los que se guardan las claves públicas y las privadas. Generalmente donde se guardan las claves públicas es el archivo pubring.gpg y en el que se guardan las claves secretas **secring.gpg**. Si se quiere borrar alguna clave primero hay que borrar la clave privada y después la

pública. Si se intenta borrar primero la clave pública y esta tiene asociada una clave privada da un mensaje de error.

```
[mockbuild@oc6127656113 ~]$ gpg --list-keys
/home/mockbuild/.gnupg/pubring.gpg
---------------------------------
pub   2048R/2320CE4F 2012-01-29
uid                  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>
sub   2048R/211B0879 2012-01-29

pub   2048R/C26C23A2 2012-01-29
uid                  Linux EducacionIT (}) <linux@educacionit.com.ar>
sub   2048R/0821FEAA 2012-01-29

[mockbuild@oc6127656113 ~]$ gpg --list-secret-keys
/home/mockbuild/.gnupg/secring.gpg
---------------------------------
sec   2048R/2320CE4F 2012-01-29
uid                  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>
ssb   2048R/211B0879 2012-01-29

sec   2048R/C26C23A2 2012-01-29
uid                  Linux EducacionIT (}) <linux@educacionit.com.ar>
ssb   2048R/0821FEAA 2012-01-29

[mockbuild@oc6127656113 ~]$ gpg --delete Linux
gpg: Option "--delete" is ambiguous
[mockbuild@oc6127656113 ~]$ gpg --delete-keys Linux
gpg (GnuPG) 1.4.11; Copyright (C) 2010 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.

gpg: ¡hay una clave secreta para esta clave pública! "Linux"!
gpg: use antes la opción "--delete-secret-key" para borrarla.
[mockbuild@oc6127656113 ~]$
```

Para borrar claves privadas se hace con el comando gpg –delete-secret-key ClaveID

Para borrar claves públicas se hace con el comando gpg –delete-key ClaveID

```
[mockbuild@oc6127656113 ~]$ gpg --list-secret-keys
/home/mockbuild/.gnupg/secring.gpg
--------------------------------
sec   2048R/2320CE4F 2012-01-29
uid                  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>
ssb   2048R/211B0879 2012-01-29

sec   2048R/C26C23A2 2012-01-29
uid                  Linux EducacionIT (}) <linux@educacionit.com.ar>
ssb   2048R/0821FEAA 2012-01-29

[mockbuild@oc6127656113 ~]$ gpg --delete Linux
gpg: Option "--delete" is ambiguous
[mockbuild@oc6127656113 ~]$ gpg --delete-keys Linux
gpg (GnuPG) 1.4.11; Copyright (C) 2010 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.

gpg: ¡hay una clave secreta para esta clave pública! "Linux"!
gpg: use antes la opción "--delete-secret-key" para borrarla.
[mockbuild@oc6127656113 ~]$ gpg --delete-secret-keys Linux
gpg (GnuPG) 1.4.11; Copyright (C) 2010 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.


sec  2048R/C26C23A2 2012-01-29 Linux EducacionIT (})
<linux@educacionit.com.ar>

¿Eliminar esta clave del anillo? (s/N) s
¡Es una clave secreta! ¿Eliminar realmente? (s/N) s
[mockbuild@oc6127656113 ~]$ gpg --delete-keys Linux
gpg (GnuPG) 1.4.11; Copyright (C) 2010 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.


pub  2048R/C26C23A2 2012-01-29 Linux EducacionIT (})
<linux@educacionit.com.ar>
```

```
¿Eliminar esta clave del anillo? (s/N) s
[mockbuild@oc6127656113 ~]$ gpg --list-keys
gpg: comprobando base de datos de confianza
gpg: 3 dudosa(s) necesarias, 1 completa(s) necesarias,
modelo de confianza PGP
gpg: nivel: 0  validez:   1  firmada:   0  confianza: 0-, 0q, 0n, 0m,
0f, 1u
/home/mockbuild/.gnupg/pubring.gpg
---------------------------------
pub   2048R/2320CE4F 2012-01-29
uid                  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>
sub   2048R/211B0879 2012-01-29

[mockbuild@oc6127656113 ~]$ gpg --list-secret-keys
/home/mockbuild/.gnupg/secring.gpg
---------------------------------
sec   2048R/2320CE4F 2012-01-29
uid                  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>
ssb   2048R/211B0879 2012-01-29

[mockbuild@oc6127656113 ~]$
```

Ver huella de la clave

Las claves están identificadas por lo que se llama huella. La huella es una serie de números que se usa para verificar si una clave pertenece realmente al propietario. Si se recibe una clave podemos ver cuál es su huella y luego pedirle a su propietario que nos diga su huella. Si ambas coinciden la clave es correcta y no ha sido manipulada. Si no fuera igual es que ha sido modificada. La huella es como el md5 que verifica que un archivo no ha sido manipulado.

```
[mockbuild@oc6127656113 ~]$ gpg --fingerprint juan@educacionit.com.ar
pub   2048R/2320CE4F 2012-01-29
      Huella de clave = 8391 070D B54A 2D40 8D80  7325 10D9 0352 2320
CE4F
uid                  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>
sub   2048R/211B0879 2012-01-29

[mockbuild@oc6127656113 ~]$
```

Exportar claves

Las claves se pueden exportar a ficheros para que las podamos distribuir entre la gente que queremos que nos encripte o firme cosas o bien porque vamos a formatear el equipo y necesitamos salvarlas.

Para exportar la clave pública se hace poniendo gpg –armor –output fichoeDeSalida –export ClaveID

```
[mockbuild@oc6127656113 ~]$ cat educacionit-public-key.asc
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.11 (GNU/Linux)

mQENBE8k1gkBCADao0UF4bRSWqUotYNHcvHxiDJBieiJZ6ArgQxtchdhEsxmhZ48
vFmxeLkqlx9ARW11iTs1wB//7DF2IuEokTZslHEIDilE1EjgXVJA2eHCT5YwgMvA
X+dHPH4tCZ0wmJyY2p4fW/mtcZ0KnehUy0gZXyOfrHIYM80GDKpYtkC2+U/UwZ3Y
katRXkkh/PDELnfaQ+bsZgY9h04cmBFF4XVuQhjF4ALo+g73C8CE9S0S4K9yXrwe
lpF+naeMQpLXq82OILXZvQj6CI+03UWY6QaxikLG6nfZbF/UXnorw1uFkoumAZDE
ID4l5GcjOW0Wnm2AB7WsPpZ/WEk5VFwRH1OjABEBAAG0OEp1YW4gRWR1Y2FjaW9u
aXQgKEVkdWNhY2lvbkl0KSA8anVhbkBlZHVjYWNpb25pdC5jb20uYXI+iQE4BBMB
AgAiBQJPJNYJAhsDBgsJCAcDAgYVCAIJCgsEFgIDAQIeAQIXgAAKCRAQ2QNSIyDO
T+frB/wNYZgPjv26H/r8a5RD+SOaQ88z3RJDBG8zjqM5/yDqmOck8fHLdacET3my
M2TFoN0RaYO9iT4sE+FVi4U7q4F29NzXCSzJVJW1THSLSZorcL+ypjF+cGvF1+ag
D08cvY/Jb+jTMn+TzDcbsW8GmvtUD7uqg1xjqhasfSRICsGgTg9C0YJ8wG6gTjOU
OfPYajVcjM/S/iHNy9P04U0SUrtlI+QHFGWjJwxcMvWzxmQZIMQccom06STl9IRd
dLCdO5vzR7JyC7UcspR6DpWHjCgpMBYWT7Lkv1ejwL1531S3QcPboSuAOw88Llzs
wa+FYKvhAeRY3ZvLH51Xl9ox4gFPuQENBE8k1gkBCADIJ83zxAX+vkhcQ2bNuq8X
4RG6pV6bTyjLq1dED1taqv4S43TKkyB2lTHgTKY+BALuhRcZtfryGK1NXO+6W5Ek
lgHmuRyyDrQcwYTq4Jg/aFn7ZyZaPrpzPb0HjEKm3G49396zOXntiNnxc6j7f/3R
R3ysPT891nZbaY0yDozLRjXjIVwiGgjrLmbDtb/p3GBgqDio9rSRc1Ls1vgDUof3
uldT/l7zlZlr/cAm5FChlTtBSF9phPDJVt2KGstULWFGOKOgr3RWjp/fio3eoac5
BMFuYfDJpamh2pYhF+E+291jfl6KRdBAbwDXfJFvT132rIE+FC7BFiRMPb0vwq9b
ABEBAAGJAR8EGAECAAkFAk8k1gkCGwwACgkQENkDUiMgzk+07AgAtQBdqCsHrwsl
vVieIXaUvEoZY30+g1Mk9+qKe0Mu1bXkxj7QxcR9xmw3nu+wnK2eUkcjLbTkCqOA
Mwq40KrkkXoesKw5YthBNYNUuScA/SQga/EBngUa6qjKNOM0a/yyFZmR3I/HnLt2
v9UjEo7eKQrWK532ugMpkv5WySiC97nLKRdfH33Snqj5XST1+JvP50HwcQ3NsAt+
kKqw+M5KxK96FSI1y5x3oleumaKONvvf2FC1NICNunim/mOmZ/PNI4vg4jjqP1zD
o1CZxbeAHMdp4FGTJS3WMl1ron2fuf3uw3OyQuhflfjAnfE1sVB3px87O4whGTsD
9T2qSQCLsw==
=vg5V
-----END PGP PUBLIC KEY BLOCK-----
[mockbuild@oc6127656113 ~]$
```

Para exportar la clave privada se haría poniendo gpg –armor –output fichoeDeSalida –export-secret-key ClaveID

```
[mockbuild@oc6127656113 ~]$ gpg --armor --output educacionit-secret-
key.asc --export-secret-keys juan@educacionit.com.ar
[mockbuild@oc6127656113 ~]$ cat educacionit-secret-key.asc
-----BEGIN PGP PRIVATE KEY BLOCK-----
Version: GnuPG v1.4.11 (GNU/Linux)

lQO+BE8k1gkBCADao0UF4bRSWqUotYNHcvHxiDJBieiJZ6ArgQxtchdhEsxmhZ48
vFmxeLkqlx9ARW11iTs1wB//7DF2IuEokTZslHEIDilE1EjgXVJA2eHCT5YwgMvA
X+dHPH4tCZ0wmJyY2p4fW/mtcZ0KnehUy0gZXyOfrHIYM80GDKpYtkC2+U/UwZ3Y
katRXkkh/PDELnfaQ+bsZgY9h04cmBFF4XVuQhjF4ALo+g73C8CE9S0S4K9yXrwe
lpF+naeMQpLXq82OILXZvQj6CI+03UWY6QaxikLG6nfZbF/UXnorw1uFkoumAZDE
ID4l5GcjOW0Wnm2AB7WsPpZ/WEk5VFwRH1OjABEBAAH+AwMCPKAJmZidKXhgtPbl
LLtxv/mO+uZbct7FQ15SOe/VqrwjESLfSXj+fn40lo+dhSEIvGZZHRiA6g5vCljN
OD7Ier5ZucfzCCZhtLwi6qyTAzhswzYuJcvdyPSR1sRwVfors4VsQ/lmsMGtkgzF
DxU1vaTekJA3Bkc7Fhlwbj8YGfihgo8cznCYgZoMN4rGXTGfX5Mw69brDVsQLDyD
rVHySsp4vyOx2KiLIrzgVdk1jLi58htKaLKnXLAL5MDQnjdcWyQ8RbvbdkW5Wxhk
/P+mhhukxlSk59RMOsa/d1rIRM0DqM/paZe/fHvG/RBqQ8i4Hxjva3fSWZKffrYl
x3Jbs1H8hn3UO9K/NOw3mRSx6SIaxZkrcHO/RSFuAo80eXU8lo1eKx3qTuK5D96J
tW0rL9VhDWcXR4D+ljudTTVBSpIULkjOWM0GsE+foV3hAbgDpmDCJqSdFT6Gj15C
2F/R4mAP2pHOR/fhYf8vlMVucAxVuDEipE3tlrxWSomUG7QdknUfRDin/cT0LVIp
```

```
UXXcKoP47031K7bckkvB9UW6SRsFWVjac5kOLLZmPD2q34HdeE6jO0cdnBqhsILR
xyb617sQPjhHdzb1r3UqTpsHlCJXVVPxs+9A5qOHtKSjYR0GdrRgt7tHEonpa3m7
CcUA10B8nB+N6w+uo9ou75xDn+ioc8b08snPlSLCUldVhtjvYh8ZuVI/9jPSuQA3
KT2u2a5gKkytornAjDs5tcJ9HAoIFe9JB3KermABOPQEePXRQW2PoA1+/JgvMq29
/1TlA9dT4fpcmJ53EzSJ1p7NOZC//m5OWU/5FjrPKMIG7XQrkeCY7sgOyVmPzcRF
yMKcx2wGWBuo8Lj7fd6DcXhIZgpH9QstaoVKYw/4WL5rmsDCCAgyFBMpIoaZfShQ
wrQ4SnVhbiBFZHVjYWNpb25pdCAoRWR1Y2FjaW9uSXQpIDxqdWFuQGVkdWNhY2lv
bml0LmNvbS5hcj6JATgEEwECACIFAk8k1gkCGwMGCwkIBwMCBhUIAgkKCwQWAgMB
Ah4BAheAAAoJEBDZA1IjIM5P5+sH/A1hmA+O/bof+vxrlEP5I5pDzzPdEkMEbzOO
ozn/IOqY5yTx8ct1pwRPebIzZMWg3RFpg72JPiwT4VWLhTurgXb03NcJLMlUlbVM
dItJmitwv7KmMX5wa8XX5qAPTxy9j8lv6NMyf5PMNxuxbwaa+1QPu6qDXGOqFqx9
JEgKwaBOD0LRgnzAbqBOM5Q589hqNVyMz9L+Ic3L0/ThTRJSu2Uj5AcUZaMnDFwy
9bPGZBkgxBxyibTpJOX0hF10sJ07m/NHsnILtRyylHoOlYeMKCkwFhZPsuS/V6PA
vXnfVLdBw9uhK4A7DzwuXOzBr4Vgq+EB5Fjdm8sfnVeX2jHiAU+dA74ETyTWCQEI
AMgnzfPEBf6+SFxDZs26rxfhEbqlXptPKMurV0QPW1qq/hLjdMqTIHaVMeBMpj4E
Au6FFxm1+vIYrU1c77pbkSSWAea5HLIOtBzBhOrgmD9oWftnJlo+unM9vQeMQqbc
bj3f3rM5ee2I2fFzqPt//dFHfKw9Pz3WdltpjTIOjMtGNeMhXCIaCOsuZsO1v+nc
YGCoOKj2tJFzUuzW+ANSh/e6V1P+XvOVmWv9wCbkUKGVO0FIX2mE8MlW3Yoay1Qt
YUY4o6CvdFaOn9+Kjd6hpzkEwW5h8MmlqaHaliEX4T7b3WN+XopF0EBvANd8kW9P
XfasgT4ULsEWJEw9vS/Cr1sAEQEAAf4DAwI8oAmZmJ0peGBpSCBTz7LYyH3BwRFY
8JWOBMREhM2/RHGtBB6LjEcqZTfq6aNw06q+bnA16Lq6K7V0eBmfqOZeVhTSlEhY
7f15wR8GWsriFO3vQn3c4htGFng8sQtlnlmKzDlm0a5wfJ6an9wwshPY1Y/c10RH
Z7U+NzAoaMyD7RZaax19uf3WDpskDGZUE1VB1pbONjvWrqhQq6lCrm+FWBd1rCJi
NrqSNTErjhE3ndrWknGIwXgRxfNKxKNFLZEBCcxZ5JD/lrdFByLJT7b2MvWkdOaU
AQF70oUuJ0lcTGdHYzfm7TWBZNBxBjjRnKiLyJ/nc9iC3cRJgJbARfVmUxUarg2C
ILV/UzN9HfDlh6VyLQb6v4uzKyGdgNwfAn3nghmWpyNVeCjfmPToGSI0UA+shJhX
yZI/uLMlMmlBh1M08qglnKXd61EQzzBY2B3PYfejmUL45ru3hpmiTJu64ILiYIrc
6BYJZgrYqC3IDs9qGp1jDDd7dGg1Ef09ldgJzh7+kcuXqFCt358FeAg77B/iYa0h
tSzndQd6CQ5ic0KbS01REb7Y4syzYZIeWB340JSHHwJZ9Bz6cF12NFguPKXmruFX
tulw3Cx2/j2BGDueTHFOyw0wU1Hogwx0VxuyOhuO6jy8G5PbHdLNNmdSIOfPi1Ht
BZBf5pwkLsUaUi7Hrdrg9bUQh5jsRlH6Rwx50IsRDgZSRUfTBj51xZq9LUzpj52+
z5m5gP6LNKkoZREJ11PtimFAXp7k0KLr9mK1681ndSgqRBApI6WbgymzCm2rwilc
CrwUGiDlqD0A28P5FfpuL+Ux6KVHH6cXMUASJs0t854lpivETMogawfGAkLgUU9m
of1JZ4KorNQtrcqEX5xoLOZnYnh9R+ZTlecpC/nVkhFsUArbhG0jiQEfBBgBAgAJ
BQJPJNYJAhsMAAoJEBDZA1IjIM5PtOwIALUAXagrB68LJb1YniF2lLxKGWN9PoNT
JPfqintDLtW15MY+0MXEfcZsN57vsJytnlJHIy205AqjgDMKuNCq5JF6HrCsOWLY
QTWDVLknAP0kIGvxAZ4FGuqoyjTjNGv8shWZkdyPx5y7dr/VIxKO3ikK1iud9roD
KZL+Vskogve5yykXXx990p6o+V0k9fibz+dB8HENzbALfpCqsPjOSsSvehUiNcuc
d6JXrpmijjb739hQpTSAjbp4pv5jpmfzzSOL4OI46j9cw6NQmcW3gBzHaeBRkyUt
1jJda6J9n7n97sNzskLoX5X4wJ3xNbFQd6cfOzuMIRk7A/U9qkkAi7M=
=Eoxr
-----END PGP PRIVATE KEY BLOCK-----
[mockbuild@oc6127656113 ~]$
```

Si quisiéramos salvar todas las claves que tenemos valdría con copiar los archivos pubring.gpg y secring.gpg y luego cuando vayamos al nuevo equipo ponerlas en el directorio de GnuPG.

Importar claves

Si se quiere importar claves nuevas porque por ejemplo hemos formateado el equipo y queremos volver a tener nuestras claves las importamos con el comando gpg –import ClaveID. En el apartado anterior se han salvado las claves pública y privada pues ahora vamos a importarlas. Primero importamos la pública y luego la privada.

```
[mockbuild@oc6127656113 ~]$ gpg --import educacionit-public-key.asc
gpg: clave 2320CE4F: clave pública "Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>" importada
gpg: Cantidad total procesada: 1
gpg:               importadas: 1  (RSA: 1)
[mockbuild@oc6127656113 ~]$ gpg --import educacionit-secret-key.asc
gpg: clave 2320CE4F: clave secreta importada
gpg: clave 2320CE4F: "Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>" sin cambios
gpg: Cantidad total procesada: 1
gpg:              sin cambios: 1
gpg:      claves secretas leídas: 1
gpg:  claves secretas importadas: 1
[mockbuild@oc6127656113 ~]$ gpg --list-keys
gpg: comprobando base de datos de confianza
gpg: no se encuentran claves absolutamente fiables
/home/mockbuild/.gnupg/pubring.gpg
---------------------------------
pub   2048R/2320CE4F 2012-01-29
uid                  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>
sub   2048R/211B0879 2012-01-29

[mockbuild@oc6127656113 ~]$
```

Ahora si queremos importar la clave de una amigo pues se haría igual.

```
[mockbuild@oc6127656113 ~]$ gpg --import /tmp/rino-public-key.asc
gpg: clave E7E4769F: clave pública "Rino Rondan (Coordinador)
<rrondan@educacionit.com.ar>" importada
gpg: Cantidad total procesada: 1
gpg:               importadas: 1  (RSA: 1)
gpg: no se encuentran claves absolutamente fiables
[mockbuild@oc6127656113 ~]$ gpg --list-keys
/home/mockbuild/.gnupg/pubring.gpg
---------------------------------
pub   2048R/2320CE4F 2012-01-29
uid                  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>
sub   2048R/211B0879 2012-01-29

pub   2048R/E7E4769F 2011-02-14 [caduca: 2012-02-14]
uid                  Rino Rondan (Coordinador)
<rrondan@educacionit.com.ar>
uid                  Rino Rondan (Restaurador)
<villadalmine@xxxxxxxx.com>
uid                  Rino Rondan (Interno)
<rino@restauradordeleyes.com.ar>
sub   2048R/A598CCC8 2011-02-14 [caduca: 2012-02-14]

[mockbuild@oc6127656113 ~]$
```

Encriptar mensajes

Si se quiere encriptar mensajes se puede hacer poniendo gpg –armor –recipient ClaveID –encrypt mensaje. Si por ejemplo queremos encriptar el archivo a.txt habría que poner

gpg --armor --recipient prueba@prueba.com --encrypt a.txt

```
[mockbuild@oc6127656113 ~]$ gpg --armor --recipient
rrondan@educacionit.com.ar  --encrypt archivo.txt
gpg: A598CCC8: No hay seguridad de que esta clave pertenezca realmente
al usuario que se nombra

pub  2048R/A598CCC8 2011-02-14 Rino Rondan (Coordinador)
<rrondan@educacionit.com.ar>
 Huella de clave primaria: 79D0 8D73 3243 6DF6 BA97  45A3 FABE 4A75
E7E4 769F
      Huella de subclave: 95FA 2266 BD66 7BFF 2F92  E629 DB0F 4C3A
A598 CCC8

No es seguro que la clave pertenezca a la persona que se nombra en el
identificador de usuario. Si *realmente* sabe lo que está haciendo,
puede contestar sí a la siguiente pregunta.

¿Usar esta clave de todas formas? (s/N) s
[mockbuild@oc6127656113 ~]$ more archivo.txt.asc
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.11 (GNU/Linux)
```

```
hQEMA9sPTDqlmMzIAQf/bYcaOd7XzKPHNpqISc/D67HlL6gvzHQ3mXXOq6Z7dWV7
OQm5LDNFE6j7Mdm2Ya46CEqjiTgEhxXL70CXEokgVmaTagKr2bFr1wQcwVlZ7ObC
fIKCi4lexfJ/9Hd4lTu2msg/bZq908Vitz9FX636zlowOlIsxaOHQMkHwoD1ZXOq
pWVw3O8BAPizHrUseb2YlQ7ajL7rm7x1EueWqmmLxDDLyp4BdLWMpI/lAcqD9zU2
wWd0gClf44sShCnmpqal4yD5XkX8IR7KGuaoIoDgufuYWhU0PKAz03B4r3oBMJCQ
ZdaMW6dfDYMdwP8sqDrcZKjA6/V5237oHS3EYcBbLtLqAVjE55QI39ksAxtShMhr
XSbC+NrmFsIlvvqlus9gdcl6153h7w6K7sfRahsN+jdykqAWcXuYSHNSkx7+E+Ov
RB9G9wyW/YJhPakHIyQsKqwWjxZQmLsKs8Oyf+b+4Y+QVkohg/m52K9EA3SzW+b+
EJOYm5GD+7HxftqBxAZXWS8YH+pEszh5ogVvyOVQqzOdm6hYuMv/ISUca9fr0oLE
E+3ngH7HDaBTmAumZK8CG0g2JE0T2MjqXke0r3uYzLc7SpGKSBzzip5xN3YSuwHf
XkdPapZWeb645HGlMirf/2AMVe+3+uolNUqRbCY4Ch5E368wBQF4K1aqC04iHt8Y
6sCclKklDrsmZi84Bqoc/f0wQIL0RdLj6KxPxgx3Q3Tiv+4sUqzRqHldbtZiUW+K
/qO9If6tXzuQipYMnZYSFwT1g0i3hto4TOXhbIgUCA176AJIPb69+4yPC4e2Aqch
oNDlWro7c2y7WQzoRiGE0NgKe6aKLZ0bzyhYPRs0CdpPvjCDZgm+g9iyb9MaMti8
ZXNQX8VBr1DApVJEcnryllQEa+sXdwB4GJ14KjTlDJ7/goiIhhmYl4gMFjz4wz0f
y2Aa6ScxXFEZIH/HKGXBZc+2Ac2tB4PPMU7lqHoFg/HglaWjpCTYW33jifMJIomB
ZWlugC2EWX0/ealEbFHpo6ia6uwlKcuQ+rBnBwzZivXUjEDdm2/0GzGc8D1/JHAl
RnjXINFIcREa8pssbMKKmsxTu/mBDs8Uv7UhIwQAJyp1UWq3elIkEs558iLLttFH
T9/tuAFAcOPjuQ8/RDv6vVAPfrZVtA6pHDCkPOjpFFPwOOcDZvYciNfmOisnOggU
4J2OuhI7hEzFyOz3Dh/YdHmrb2mjgZTPiYAy9jF7PUBVSMTUnseTZApn5j+WIPWQ
UXDcC4z7gk0l1yT7ji6og+rJry2fNwNQF1TIxQbWHEOqHzRO+RD6h4LDUZL2Uqgc
h0NJqWvThbator0SeA8XuPPQmqMo8mahrZFFp6oeuiIt3KYw35OcWhJjfccA1KgO
gNTkJuDrIiZewD78rKhTW4u3rY3g7d1/cC5POaoNP40oAuttEguArJ1Z8DLVOrHw
zfwZn0rUMqNi6/m+f7b6ld7A6Pq06XYBgjXKaCPz5TH9ZATYZ4frPh1zs0WhEKnK
KYsJWotwOCKmtLG5kL/kfqjEWrDfgCSYVQYyXPdsdEJJkOkYZZ7sJNdf5Kcs9B+s
MDthAHI51B3qh5a4kg7o2fONBiyoYkNZZWv/gsNmMAsu4hyoJ50ExCbvCccV/Vls
zZJf2eDDlBRpTMkWL+wWN+Y+Hss/8zGaaSggB7ZBrJyfkuEjknJm+1zX7JotnyH3
s4G8cGB5cHv/MecP6WXy8bq30l80r+f4nFd6yc12LB87Y0LFmpyUS2qSi+WmiMEB
Ptlzt1GoXmlMD7oXMZnDMftuy7OSwA4M65UXN7PGIWWTMsXo9QMwxOZfgZK6dP7Y
utk/8V9T3qCd2BYuSMauuD94GUAzT/c47DTqE263KtCuOm8=
=RfPU
-----END PGP MESSAGE-----
[mockbuild@oc6127656113 ~]$
```

También se puede encriptar a un fichero en concreto con la opción –output nombreFichero

```
^[[5~[mockbuild@oc6127656113 ~]$ gpg --armor --output salida.txt --
recipient rrondan@educacionit.com.ar --encrypt archivo.txt
gpg: A598CCC8: No hay seguridad de que esta clave pertenezca realmente
al usuario que se nombra

pub  2048R/A598CCC8 2011-02-14 Rino Rondan (Coordinador)
<rrondan@educacionit.com.ar>
 Huella de clave primaria: 79D0 8D73 3243 6DF6 BA97  45A3 FABE 4A75
E7E4 769F
      Huella de subclave: 95FA 2266 BD66 7BFF 2F92  E629 DB0F 4C3A
A598 CCC8

No es seguro que la clave pertenezca a la persona que se nombra en el
identificador de usuario. Si *realmente* sabe lo que está haciendo,
puede contestar sí a la siguiente pregunta.

¿Usar esta clave de todas formas? (s/N) s
[mockbuild@oc6127656113 ~]$ cat salida.txt
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.11 (GNU/Linux)
```

```
hQEMA9sPTDqlmMzIAQf/dKV1Vf4hmjKU1CK9zSN2yotVtmQ/VMyefZfJKjR3q+Js
CqHKjyEm0JxQKfTZcTv78jBMvaVHOD9PFxwX4gJjnWvE89MVufPBcKBYn2m0o934
juZETZ/+0Ss7aDWw3SoLDbHq7GurbC85cgb9w/mIB5QsM6GsefZa9UTvwcA80aqg
bK+eikOdtm8DFayvCHLsMjrqYSAoUEGbg7vJGsfGhfVDcognaM5mEq1yaXmY0si9
5eQ1bCZX50QML+VRjlMybT6l0HTEX1mzZYUjfDtZSS/n7TBC50BrhLOn2Uo8kCun
d74Pqjf12sxxWqKz0eJcFf7kQcjMfZsrv/y1t1KL4NLqAeqAPfG2WUTqMpi79g0k
AujosP05NE7WdW2BhUmpfH+0PktDdeH6D4EXqxBtuYJjIIpryx3WFRCamk0RuZa6
Do8mqJuYXPTfthAo4ZHS/CGBdNDM9//GQLtJzH7O1n4zlouAfleIR5zZDhHIB9BA
g0ZVcbITLRRH25FJjUxM1QxDFm3fectGDkgU7L0cnBV8F2iEqzmsNxKBxfn/vLNS
IE2MFWf7rQS4C6l17qbeDXSskuK4vATqhqZwXjWXeYCvjEeE0Wb2TRlJSivRjWCC
lTmd65gYTERE02TsEPnaMYJFa8hFRj71Gv96wxxiuj+H22WSGAthMyh3d9V6J645
tSJdPBCihkUMssLCcTX6ymq6KZ10O4sKBkKZ+YJbzXhZ2VQ54Oqe4uw0krM54czm
rNcNl4y9SJ8h9Mp/EK9a3V7lvhUlZt+4rrCz1nYyU6D+k/46S20NhI1kq+E3+hg2
8Cfv5B/M2OzyvFtoiido+BCRUavVu1grbmblx1123t4XAxpXc2HINQCatIKXjHha
pzVUYssW6gz6znzPUhTWSU3Lki9d7NmshT2qyu9+YtQE1WzgVQehMEFxVCuQz3kC
ObyALK/qNEDFY/GRCNIfhiH+fx+6C+v9mup12tvp91MZNY0EDTdKe40HMs/h45Qu
V2Apwke46sL6+GmkjBfTP9VKWo5mAzYTW88K371FG58m3gw9DTybZ9wt2ivnkTYq
sNT6wgJmvwIbkAVwjbuf1RATYRQT0v8fojgGrCPp6QZ44BTAInnhR8pWPOW1u57B
qvXcuE0pN+Hc1LavhDQuTXCwewsSAYDtjdNBPqBdtdWmRHDH6C7DOWocBUfwAA5p
DrbsHuJMbeZhiPR8vR2aPDj0C4nwfD+0EbvyxR1cdCxNLBbw/KyC2v7oxNZesFqV
qg5eVKeqndmdTXquY/B2JmPOtaDrqWCjxF/WnJprXjCcCzul/Ic4wC3Hh36Y0jlO
LQuSryj5JCYhT8aAaybKZmJGSaobH03VnhNBqB1r+5xsDkBgJ/TcGKdYIrEr2mL8
Gk8w86DOqMttNeplXg6pCR5NWpMd65jhu8dlTcfoSD59kwwCMksniCv7zbL1nRU1
xbFDWxzHnZzI4og7HDWgNyaXsKXRDjwfzdor8bn8dIKkYH1Mmrsedb+uKAeJlkdP
Mg9RFcp1zCn5ddCoMfEagHebJTo3vs2/r3+ANnppoGHtqdKDagsqWIcuMInE8DwS
hO7nAQRii5/sQypzKq50jCsDd4LomwmFzy5zSZtHdqj2QHeJSEEu9mh3wqfnIOZS
0wiVaL+5adMLgMojSDx0VUqEZjilOPccoIobYLKytv9oXHM3m5fRiY7ajtNrBuJb
z4EFZuqrYqkSYMIb9nhLl1o9vl8KmWBghh8Q0BP/jR2s0pIvmKwJ5Jt1klanBuWL
9TjlbKpm3ojNGyJuklCiGE9+sRpstHDwzzVBsjtO91rKV0BkjAAnJhXiO6SzS7Cq
EocX9VUkdm5xJTN89aauTNlfsePD1u1zXGYI+zrTJwzfdAQ=
=Kf/l
-----END PGP MESSAGE-----
[mockbuild@oc6127656113 ~]$
Si en las opciones no se le pasa el parámetro --armor lo que se
encripta lo deja en un archivo de tipo binario. Al poner la opción --
```

```
armor transforma lo que se encripta en texto ASCII con el mensaje
encriptado.
```

Desencriptar mensajes

Para desencriptar el mensaje que hemos encriptado antes hay que poner gpg –decrypt archivo. Para el caso anterior sería

gpg --decrypt a.txt.asc.

```
[crond1@oc6127656113 ~]$ gpg --decrypt /tmp/archivo.txt.asc

Necesita una frase contraseña para desbloquear la clave secreta
del usuario: "Rino Rondan (Coordinador) <rrondan@educacionit.com.ar>"
clave RSA de 2048 bits, ID A598CCC8, creada el 2011-02-14(ID de clave
primaria E7E4769F)

gpg: cifrado con clave RSA de 2048 bits, ID A598CCC8, creada el 2011-
02-14
      "Rino Rondan (Coordinador) <rrondan@educacionit.com.ar>"
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
..
..
redis:x:490:482:Redis Server:/var/lib/redis:/sbin/nologin
mockbuild:x:502:502::/home/mockbuild:/bin/bash
[crond1@oc6127656113 ~]$
```

Cuando desencriptamos algo se pide la password de nuestra clave para poder desencriptarlo. Para nuestro caso tenemos el archivo a.txt.asc encriptado al desencriptarlo nos deja el archivo a.txt y nos muestra su contenido.

Firmar mensajes

Firmar mensajes sirve para que cuando a alguien le llegue un mensaje que hemos firmado la persona que lo ha recibido verifique con GnuPG que la firma es buena y que entonces hemos sido nosotros quien le ha enviado el mensaje. Por ejemplo vamos a firmar el archivo a.txt para ello se pondría **gpg --clearsign a.txt**. Esto nos creará el archivo a.txt.asc con el contenido que se ve en la imagen.

```
[mockbuild@oc6127656113 ~]$ cat a.txt
Hola
[mockbuild@oc6127656113 ~]$ gpg --clearsign a.txt

Necesita una frase contraseña para desbloquear la clave secreta
del usuario: "Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>"
clave RSA de 2048 bits, ID 2320CE4F, creada el 2012-01-29

[mockbuild@oc6127656113 ~]$ cat a.txt.asc
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Hola
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.11 (GNU/Linux)

iQEcBAEBAgAGBQJPJPnBAAoJEBDZA1IjIM5P6ooH/i5JhBqb1QlWoS/dZ3G5v5x4
pEfdFlh/tYGYFAPIaV6e/BOLEW2Re40J/ImdLK87HK+citD7q6YBZws6Z6dpYIv1
```

```
Om09+x7Vr1UkYi0QfcQnrqKOVvPq1Uk99LVMkmWb50GtQUxk0eC1loVLGATooVhE
hLNMWok0hOJtmsWxtKWk4b8qXf+jyDiQbVJsAe144OMYaFb+M5J8aMepcZwKpUYa
9TQEFSnMzyBm+IxzeDsPYBfnTa74SW0PbyINGpTbsXvFkGHeyzZBBTWciAOZWIoj
VKgpnBe3Bue/99nlOQ5iG2ppHjyhb+olcNtEHcxkgZ+tmt41mlymaoZ5dh89+Oo=
=jAXY
-----END PGP SIGNATURE-----
[mockbuild@oc6127656113 ~]$
```

Para firmar algo se pide la contraseña para poder firmarlo. Como se ve en la imagen lo que se ha hecho en el fichero firmado es añadir unas líneas que contienen la firma.

A la hora de firmar si se firma con el parámetro –sign en lugar de –clearsign nos generará un fichero de salida en binario con extensión .gpg. Para validar la firma y ver el contenido hay desencriptarlo con la opción --decrypt.

```
[mockbuild@oc6127656113 ~]$ cat educacionit.txt
EducacionIT con Linux y LPI
[mockbuild@oc6127656113 ~]$ gpg --sign educacionit.txt

Necesita una frase contraseña para desbloquear la clave secreta
del usuario: "Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>"
clave RSA de 2048 bits, ID 2320CE4F, creada el 2012-01-29

[mockbuild@oc6127656113 ~]$ ls -la educacionit.*
-rw-rw-r-- 1 mockbuild mockbuild  28 ene 29 04:49 educacionit.txt
-rw-rw-r-- 1 mockbuild mockbuild 358 ene 29 04:50 educacionit.txt.gpg
[mockbuild@oc6127656113 ~]$ cat educacionit.txt.gpg
□□□□□□□□(p□9HY□□?□□□$□□□□□□□□□□□□□□□□□□□□_□□0!□□□□□<□
7r□]□□u@□□□v□□V;3□□□□□□O□□□W??□□;□5□□5□ëk□ □□\|hs?cc□□□o_|□□□>□□□□k□□{□,
Of□Z+){□M□□□□□□O□E4R=sfe□12□└□q□□□k□f¯□o□□N□□□□□q¸□□>□>{□□□p□q□□□□□□□6)
□[□{□□□43Y□r□`□□□□     □[$□□^□b□{□n□□□V□□u□□□□□□l□f□□c□□□c*M□□□□N□□□□-
□□TL□□+[mockbuild@oc6127656113 ~]$
[mockbuild@oc6127656113 ~]$ gpg --decrypt educacionit.txt.gpg
EducacionIT con Linux y LPI
gpg: Firmado el dom 29 ene 2012 04:49:57 ART usando clave RSA ID
2320CE4F
gpg: Firma correcta de "Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>"
gpg: ATENCIÓN: ¡Esta clave no está certificada por una firma de
confianza!
gpg:          No hay indicios de que la firma pertenezca al
propietario.
Huellas dactilares de la clave primaria: 8391 070D B54A 2D40 8D80
7325 10D9 0352 2320 CE4F
[mockbuild@oc6127656113 ~]$
```

La firma también se puede hacer que se muestre en un fichero aparte con la opción -b. Esta opción se suele usar para firmar archivos

binarios.

```
[mockbuild@oc6127656113 ~]$ file ls
ls: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically
linked (uses shared libs), for GNU/Linux 2.6.32, stripped
[mockbuild@oc6127656113 ~]$ file educacionit.txt
educacionit.txt: ASCII text
[mockbuild@oc6127656113 ~]$ gpg -b ls

Necesita una frase contraseña para desbloquear la clave secreta
del usuario: "Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>"
clave RSA de 2048 bits, ID 2320CE4F, creada el 2012-01-29

[mockbuild@oc6127656113 ~]$ cat ls
ls      ls.sig
[mockbuild@oc6127656113 ~]$ cat ls.sig
□□□□□□□□□O$□□
j□□□□□□G□L□□□□{S□□□Fy□□2s<□Z%□□rf□;OB-□x□□□□□□□□□□□□□¢3I}□□□&□*(□□□□k□□
\□'□□□□□ŭ·s‡Ęç0□□□□Cwm□r

□□□~□□2!□□□3>X♯□~□□x□□X□wE□y□ʲBXI□□r□□y□J□k□□?□)Ÿ□□□□n□□□jb"BP□□□p3p=□K
                                □V[mockbuild@oc6127656113 ~]$
[mockbuild@oc6127656113 ~]$ gpg --verify ls.sig
gpg: Firmado el dom 29 ene 2012 04:56:02 ART usando clave RSA ID
2320CE4F
gpg: Firma correcta de "Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>"
gpg: ATENCIÓN: ¡Esta clave no está certificada por una firma de
confianza!
gpg:          No hay indicios de que la firma pertenezca al
propietario.
Huellas dactilares de la clave primaria: 8391 070D B54A 2D40 8D80
7325 10D9 0352 2320 CE4F
[mockbuild@oc6127656113 ~]$
```

Verificar mensajes firmados

Para verificar mensajes firmados se hace poniendo **gpg --verify mensaje**. Para el caso anterior sería poner gpg –verify a.txt.asc

```
[mockbuild@oc6127656113 ~]$ gpg --verify archivo.txt.asc
gpg: Firmado el dom 29 ene 2012 04:47:02 ART usando clave RSA ID
2320CE4F
gpg: Firma correcta de "Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>"
gpg: ATENCIÓN: ¡Esta clave no está certificada por una firma de
confianza!
gpg:          No hay indicios de que la firma pertenezca al
propietario.
Huellas dactilares de la clave primaria: 8391 070D B54A 2D40 8D80
7325 10D9 0352 2320 CE4F
[mockbuild@oc6127656113 ~]$
```

Si la firma no fuese correcta podríamos ver un mensaje como el siguiente:

```
[mockbuild@oc6127656113 ~]$ gpg --verify archivo.txt.asc
gpg: Firmado el dom 29 ene 2012 04:47:02 ART usando clave RSA ID
2320CE4F
gpg: CRC error; 74D39D 14E33E
Huellas dactilares de la clave primaria: 8391 070D B54A 2D40 8D80
7325 10D9 0352 2320 CE4F
[mockbuild@oc6127656113 ~]$
```

Trabajar con claves en servidores

Podemos buscar claves públicas de gente a la que queremos enviar mensajes cifrados en servidores de claves. Para hacer una búsqueda se hace poniendo los parámetros --keyserver NombreDelServidor --search-keys ClaveID. Si encuentra claves que coinciden con esa ID nos las muestras, luego tenemos tres opciones mostrar los siguientes si es que se han encontrado muchos, poner el número del registro que nos interesa (en ese caso nos importa la clave al anillo de claves públicas) o salir.

Buscar claves públicas en servidores

Si queremos importar una clave en concreto se hace con los parámetros --keyserver NombreDelServidor --recv-keys ClaveID .

Importar una clave desde un servidor

Si queremos subir una clave a un servidor para que esté disponible para la gente se hace con los parámetros --keyserver NombreDelServidor --send-keys ClaveID

Exportar una clave a un servidor

GnuPG subshell

GnuPG viene con una especie de shell que nos da multitud de opciones para trabajar con la clave, Nos permite firmar la clave, cambiar la contraseña, cambiar la fecha de expiración de la llave …

Para acceder a esta shell hay que poner el parámetro --edit-key ClaveID.

Clave de revocación

La clave de revocación es una clave que lo que hace es que cuando la importemos a nuestro anillo de claves invalide esa clave. Para generarla se hace con la opción --gen-revoke. Esta clave se puede crear nada más generar las llaves o bien cuando se haya comprometido. Hay gente que lo crea nada más crear las claves porque si por ejemplo ha olvidado la contraseña no podrá generar la clave de revocación ya que al final del proceso de generación se pide la contraseña. Esta clave ha de guardarse en un lugar seguro ya que si alguien la obtuviese podría revocar nuestras claves y dejarnos las claves inutilizadas.

```
[mockbuild@oc6127656113 ~]$ gpg --verify archivo.txt.asc
gpg: Firmado el dom 29 ene 2012 04:47:02 ART usando clave RSA ID
2320CE4F
gpg: Firma correcta de "Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>"
gpg: ATENCIÓN: ¡Esta clave no está certificada por una firma de
confianza!
gpg:          No hay indicios de que la firma pertenezca al
propietario.
```

```
Huellas dactilares de la clave primaria: 8391 070D B54A 2D40 8D80
7325 10D9 0352 2320 CE4F
[mockbuild@oc6127656113 ~]$ gpg --output JuanRevoke.asc --gen-revoke
Juan

sec  2048R/2320CE4F 2012-01-29 Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>

¿Crear un certificado de revocación para esta clave? (s/N) s
Por favor elija una razón para la revocación:
  0 = No se dio ninguna razón
  1 = La clave ha sido comprometida
  2 = La clave ha sido reemplazada.
  3 = La clave ya no está en uso
  Q = Cancelar
(Probablemente quería seleccionar 1 aquí)
¿Su decisión? 1
Introduzca una descripción opcional; acábela con una línea vacía:
> La clave ya no es segura
>
Razón para la revocación: La clave ha sido comprometida
La clave ya no es segura
¿Es correcto? (s/N) y

Necesita una frase contraseña para desbloquear la clave secreta
del usuario: "Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>"
clave RSA de 2048 bits, ID 2320CE4F, creada el 2012-01-29

se fuerza salida con armadura ASCII.
Certificado de revocación creado.

Por favor consérvelo en un medio que pueda esconder; si alguien
consigue
acceso a este certificado puede usarlo para inutilizar su clave.
Es inteligente imprimir este certificado y guardarlo en otro lugar,
por
si acaso su medio resulta imposible de leer. Pero precaución: ¡el
sistema
de impresión de su máquina podría almacenar los datos y hacerlos
accesibles
a otras personas!
[mockbuild@oc6127656113 ~]$
```

Si queremos revocar la clave hay que importar el fichero que tiene la clave de revocación y ya está. Una vez revocada la clave ya no podemos cifrar mensajes aunque sí se pueden desencriptar, aunque al desencriptar se avisa de que la clave ha sido revocada..

```
[mockbuild@oc6127656113 ~]$ gpg --import Juan5Revoke.asc
gpg: no se puede abrir `Juan5Revoke.asc': No existe el fichero o el
directorio
gpg: Cantidad total procesada: 0
[mockbuild@oc6127656113 ~]$ gpg --import JuanRevoke.asc
gpg: clave 2320CE4F: "Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>" certificado de revocación importado
gpg: Cantidad total procesada: 1
gpg:         nuevas revocaciones de claves: 1
```

```
gpg: no se encuentran claves absolutamente fiables
[mockbuild@oc6127656113 ~]$ gpg --decrypt archivo.txt.asc
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
avahi-autoipd:x:170:170:Avahi IPv4LL Stack:/var/lib/avahi-
autoipd:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-
daemon:/sbin/nologin
openvpn:x:499:498:OpenVPN:/etc/openvpn:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
saslauth:x:498:497:"Saslauthd user":/var/empty/saslauth:/sbin/nologin
smolt:x:497:495:Smolt:/usr/share/smolt:/sbin/nologin
pulse:x:496:494:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
gdm:x:42:42:gdm system account:/var/lib/gdm:/sbin/nologin
mailnull:x:47:47::/var/spool/mqueue:/sbin/nologin
smmsp:x:51:51::/var/spool/mqueue:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:4294967294:4294967294:Anonymous NFS
User:/var/lib/nfs:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
crond1:x:500:500:crond1:/home/crond1:/bin/bash
sabayon:x:86:86:Sabayon user:/home/sabayon:/sbin/nologin
```

```
mockbuild:x:502:502::/home/mockbuild:/bin/bash
gpg: Firmado el dom 29 ene 2012 04:47:02 ART usando clave RSA ID
2320CE4F
gpg: Firma correcta de "Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>"
gpg: ATENCIÓN: ¡Esta clave ha sido revocada por su propietario!
gpg:          Esto puede significar que la firma está falsificada.
gpg: razón para la revocación: La clave ha sido comprometida
gpg: comentario a la revocación: La clave ya no es segura
gpg: ATENCIÓN: ¡Esta clave no está certificada por una firma de
confianza!
gpg:           No hay indicios de que la firma pertenezca al
propietario.
Huellas dactilares de la clave primaria: 8391 070D B54A 2D40 8D80
7325 10D9 0352 2320 CE4F
[mockbuild@oc6127656113 ~]$ gpg --armor -r Juan --encrypt archivo.txt
gpg: Juan: omitido: clave pública inutilizable
gpg: archivo.txt: encryption failed: clave pública inutilizable
[mockbuild@oc6127656113 ~]$
```

Al invalidar la clave tampoco se pueden firmar mensajes.

```
[mockbuild@oc6127656113 ~]$ gpg --default-key Juan --sign archivo.txt
gpg: no default secret key: clave secreta inutilizable
gpg: signing failed: clave secreta inutilizable
[mockbuild@oc6127656113 ~]$
```

Anillo de confianza

Crear un anillo de confianza consiste en tener claves de gente firmada por otra gente que la han firmado y que con su firma aseguran que esa clave es realmente de quien dice ser y no ha sido alterada.

Si por ejemplo tenemos las persona A y B. Las personas A y B son amigas y se intercambian entre ellas las claves públicas, verifican sus fingerprints para ver que las claves son las correctas y quedan para ver las claves que se han pasado son correctas. Entonces una vez verificado que todo es correcto cada uno firma la clave de su amigo. Ahora si por ejemplo yo obtengo la clave de B y veo que está firmada por A (que es una persona que conozco y en la que confió) entonces me fio de que esa clave es la clave correcta de B y la puedo usar. Si por un casual quedaría firmar la clave de B con mi firma para avalar que su clave es buena sería bueno que me pusiese en contacto con él y verificáramos la clave.

```
[mockbuild@oc6127656113 ~]$ gpg --sign-key Rino

pub  2048R/E7E4769F  creado: 2011-02-14  caduca: 2012-02-14  uso: SC
                     confianza: desconocido   validez: desconocido
sub  2048R/A598CCC8  creado: 2011-02-14  caduca: 2012-02-14  uso: E
[desconocida] (1). Rino Rondan (Coordinador)
<rrondan@educacionit.com.ar>
[desconocida] (2)  Rino Rondan (Restaurador) <villadalmine@gmail.com>
[desconocida] (3)  Rino Rondan (Interno)
<rino@restauradordeleyes.com.ar>

¿Firmar realmente todos los IDs de usuario? (s/N) s
```

```
pub  2048R/E7E4769F  creado: 2011-02-14  caduca: 2012-02-14  uso: SC
                     confianza: desconocido   validez: desconocido
 Huella de clave primaria: 79D0 8D73 3243 6DF6 BA97  45A3 FABE 4A75
E7E4 769F

     Rino Rondan (Coordinador) <rrondan@educacionit.com.ar>
     Rino Rondan (Restaurador) <villadalmine@gmail.com>
     Rino Rondan (Interno) <rino@restauradordeleyes.com.ar>

Esta clave expirará el 2012-02-14.
¿Está realmente seguro de querer firmar esta clave
con su clave: "Esteban Linux (Interno) <esteban@educacionit.com.ar>"
(CF8EACBF)?

¿Firmar de verdad? (s/N) s

[mockbuild@oc6127656113 ~]$
[mockbuild@oc6127656113 ~]$ gpg --list-sigs
gpg: comprobando base de datos de confianza
gpg: 3 dudosa(s) necesarias, 1 completa(s) necesarias,
modelo de confianza PGP
gpg: nivel: 0  validez:   1  firmada:   1  confianza: 0-, 0q, 0n, 0m,
0f, 1u
gpg: nivel: 1  validez:   1  firmada:   0  confianza: 1-, 0q, 0n, 0m,
0f, 0u
gpg: siguiente comprobación de base de datos de confianza el: 2012-02-
14
/home/mockbuild/.gnupg/pubring.gpg
---------------------------------
```

```
pub   2048R/2320CE4F 2012-01-29 [revocada: 2012-01-29]
rev          2320CE4F 2012-01-29  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>
uid                  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>
sig 3        2320CE4F 2012-01-29  Juan Educacionit (EducacionIt)
<juan@educacionit.com.ar>

pub   2048R/E7E4769F 2011-02-14 [caduca: 2012-02-14]
uid                  Rino Rondan (Coordinador)
<rrondan@educacionit.com.ar>
sig 3        E7E4769F 2011-02-15  Rino Rondan (Coordinador)
<rrondan@educacionit.com.ar>
sig          CF8EACBF 2012-01-29  Esteban Linux (Interno)
<esteban@educacionit.com.ar>
uid                  Rino Rondan (Restaurador)
<villadalmine@gmail.com>
sig 3        E7E4769F 2011-02-14  Rino Rondan (Coordinador)
<rrondan@educacionit.com.ar>
sig          EF9AAD20 2011-02-23  [ID de usuario no encontrado]
sig          CF8EACBF 2012-01-29  Esteban Linux (Interno)
<esteban@educacionit.com.ar>
uid                  Rino Rondan (Interno)
<rino@restauradordeleyes.com.ar>
sig 3        E7E4769F 2011-02-15  Rino Rondan (Coordinador)
<rrondan@educacionit.com.ar>
sig          CF8EACBF 2012-01-29  Esteban Linux (Interno)
<esteban@educacionit.com.ar>
sub   2048R/A598CCC8 2011-02-14 [caduca: 2012-02-14]
sig          E7E4769F 2011-02-14  Rino Rondan (Coordinador)
<rrondan@educacionit.com.ar>
```

Programas que soportan GnuPG

Para utilizar GnuPG de forma gráfica y no tener que estar escribiendo los comandos podemos encontrar para KDE el programa KGPG y para Gnome el programa SeaHorse.

Hay algunos clientes de correo que también soportan GnuPG como Evolution o Mozilla Thunderbird al que hay que ponerle el plugin Enigmail para que tenga soporte de GnuPG.

BIBLIOGRAFÍA

Libros:

LPI Linux Certification in a Nutshell, Third Edition, June 2010

LPIC-1: Linux Professional Institute Certification Study Guide: (Exams 101 and 102), 2nd Edition, February 2009

Páginas:

[1] GnuPG

[2] Wikipedia GNUPG

[3] GnuPG

[4] Ejemplos at

[5] Checksum

[6] Teoria Encriptacion

OPERACIONES DE BACKUPS

Peso	3
Tópico Cubierto	206.2 Operaciones de backup
Descripción	Los candidatos deberían ser capaces de usar las herramientas para realizar copias de respaldo de los datos del sistema importantes.
Temas	Este tópico pertenece al examen LPIC-2 201 • Saber que directorios tienen que ser incluidos en las copias de seguridad • Estar al tanto de las soluciones de copias de respaldo tales como Amanda, Bacula y BackupPC • Saber los beneficios y desventajas de las cintas, CDR, discos u otro medio de copia de seguridad • Realizar copias de respaldo parciales y manuales • Verificar la integridad de las copias de respaldo • Restaurar parcial o totalmente copias de seguridad
Ejemplos	• /bin/sh • dd • tar • /dev/st* y /dev/nst* • mt • rsync

Peso: Indica el valor de importancia que tiene este tópico en la certificación.

Tópico Cubierto: Indica según el programa de certificacion LPI que topico le corresponde a este tema.

Descripción: Un resumen de lo que se verá.

Temas: Un resumen de los conceptos primordiales que están cubiertos.

Ejemplos: Palabras claves que se tienen que tener en cuenta.

Qué es una copia de seguridad

Tanto los archivos del sistema como los del usuario son susceptibles a ser borrados, eliminados o corrompidos por:

- Fallas de software
- Fallas de hardware
- Errores humanos
- Ataques informáticos
- Accidentes y desastres climáticos

Atendiendo estos riesgos es importante realizar copias de los datos tanto del sistema como documentos del usuario de una manera segura, esto es lo que comúnmente conocemos como backup.

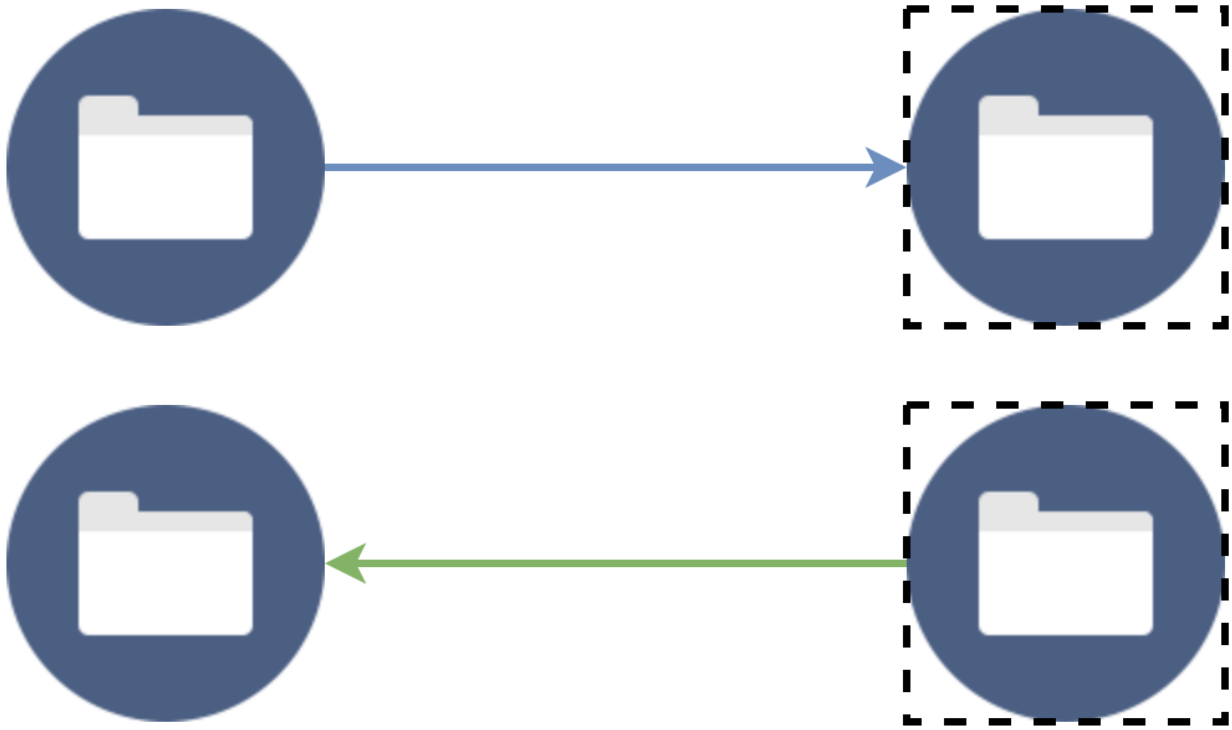
Niveles de backups

Las copias de seguridad tienen al menos tres niveles dependiendo de lo que se respalde.

Nivel Completo o Full

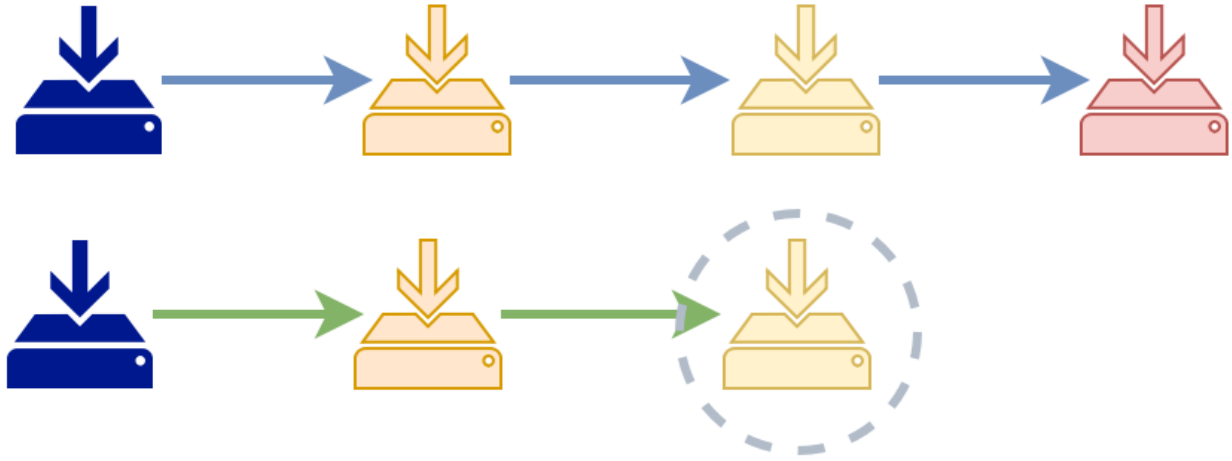
En este nivel se realiza un una copia de la totalidad de los datos que se desean respaldar.

Siempre es útil realizar un backup completo, no obstante. realizar copias de respaldo todas las veces, sería imposible o al menos costoso en cuanto a dispositivos de almacenamiento. Para recuperar un backup full se debe hacer la operación inversa. Muchas veces no se recuperan los datos sobre la ubicación original, sino en otro lado de manera de poder verificar los archivos recuperados si es necesario.



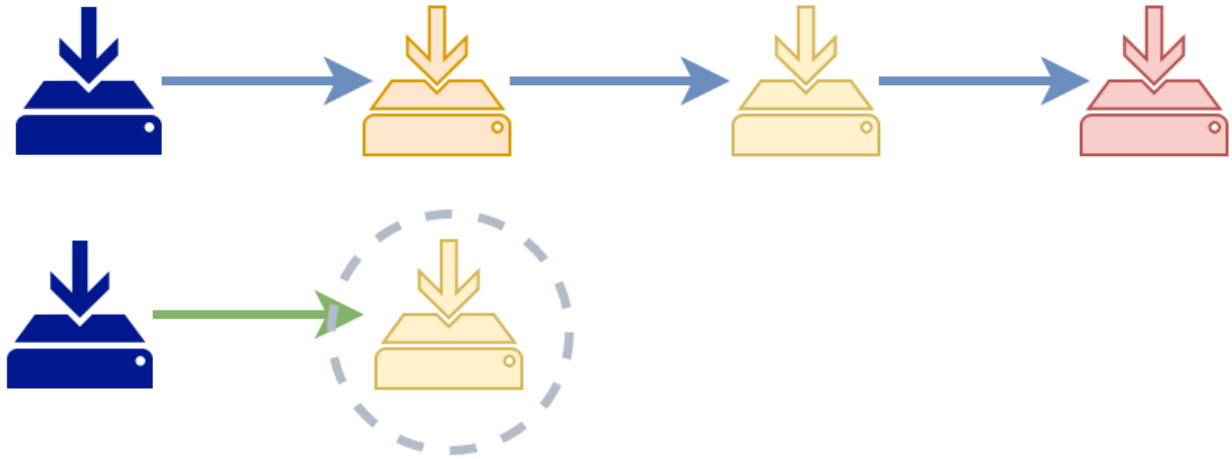
Nivel Incremental

El nivel incremental toma como punto de referencia el backup inmediatamente anterior (que puede ser Full o Incremental). Suponiendo que luego del backup completo (B0) se hicieron tres backups incrementales (B1, B2 y B3) si se desea volver al estado de los datos al realizar B2, es necesario aplicarle al B0, los backups B1 y B2.



Nivel Diferencial

El nivel incremental toma como punto de referencia el backup full. Suponiendo que luego del backup completo (B0) se hicieron tres backups diferenciales (B1, B2 y B3) si se desea volver al estado de los datos al realizar B2, es necesario aplicarle al B0 solamente el B2.



¿Qué tipo de backup es mejor? ¿Cuál deberíamos aplicar? Si podemos, lo ideal es usar los dos. El backup incremental ahorra más espacio de almacenamiento, pero hace que las restauración de datos sea algo engorrosa.

Directorios a respaldar

Es más fácil determinar qué directorios **no** hay que respaldar. Conviene copiarlo máximo que se pueda excluyendo aquellos directorios que son volátiles:

- /dev
- /proc
- /run
- /sys
- /tmp (Dependiendo de la distribución y de las aplicacin es usadas)

Si podemos elegir solamente entre unos pocos directorios, lo recomendable sería hacer backup de:

- /etc (Configuraciones)
- /root (Configuraciones del superusuario)
- /usr/local (Aplicaciones compiladas o que no están en la distribución)
- /home (Si es una estación de trabajo, o si en lugar de usar root se emplea un usuario con sudo para administrar)

- /var (Muy usado en servidores)

Y obviamente cualquier otro directorio que se haya creado a mano que esté directamente dentro del directorio raíz.

Medios de almacenamiento

Este es un análisis de los distintos medios de almacenamiento disponibles:

Medios ópticos: CDR, DVD, etc.	• Económicos	• Poco espacio
Cintas	• Larga duración	• Costosas • Necesitan hardware especial
Discos rígidos/sólidos	• Alta capacidad • Facilidad de uso	• Costosos (en particular los de estado sólido)
Nube	• Aislado de accidentes locales	• Interrupción del servicio no controlada (si se hace en equipos de un tercero)

Herramientas comunes en Linux

Comando dd

Esta herramienta nos sirve para crear imágenes de particiones de de discos. Su uso es sencillo:

dd if=/dev/sda of=/dev/sdb status=progress

Donde /dev/sda es el disco a respaldar y /dev/sdb es el disco en el que se copiará.

Para recuperar la copia deben invertirse los archivos:

dd if=/dev/sdb of=/dev/sda status=progress

También se puede realizar backups de particiones a un archivo regular:

dd if=/dev/sda1 of=/media/discoext/sda1.img status=progress

Atención: la opción ***status=progress*** está solamente en versiones recientes del comando.

Para verificar la integridad del backup se puede usar una herramienta como sha256sum para calcular el hash de ambos archivos:

 # sha256sum /dev/sda1 /media/discoext/sda1.img

El resultado tiene que ser el mismo para los dos archivos

Comando tar

Como ya sabemos este comando permite crear a partir de uno o más archivos paquetes (generalmente comprimidos). Pero además, sirve para crear backups incrementales de manera sencilla. Para hacerlo debemos indicarle un archivo en el cual guardará los datos del backup incremental:

 # tar -g FILE0 -cvzf /media/disco/etc0.tar.gz /etc

Para crear un backup incremental, hacemos lo siguiente:

cp FILE0 FILE1 && tar -g FILE1 -cvzf /media/disco/etc1.tar.gz /etc

Backups en unidades de cinta

Hay dos tipos de archivos que son nodos a unidades de cinta, ellos son:

- /dev/st+([0-9])
- /dev/nst+([0-9])

La diferencia entre los dos tipos de dispositivos a los que apuntan consiste en que los primeros se rebobinan antes de cerrar la cinta, mientras que los segundos no lo hacen de manera automática. El segundo tipo de cintas es preferible ya que no tiene que volver a buscar la posición en la que estaba.

Por ejemplo para hacer un backup del directorio /etc y el directorio /var en una cinta:

tar cvzf /dev/nst0 /etc /var

En Linux, en la gran mayoría de los casos las cintas son controladas en el espacio por el módulo **st**. Mientras tanto, en el espacio de usuario existe la herramienta **mt**.

Para rebobinar una cinta:

mt -f /dev/nst0 rewind

Para escribir una marca de fin de archivo:

Para recuperar todo:

tar xvzf /dev/nst0

Para recuperar solamente el directorio etc:

tar xvzf /dev/nst0 /etc

*Nota: En lugar de "**x**" se puede usar "**t**" para listar los archivos, sería una manera básica de verificar el backup.*

Para reciclar una cinta:

mt -f /dev/nst0 rewind && mt -f /dev/nst0 weof

Herramienta rsync

Rsync es una poderosa herramienta para copiar archivos tanto de manera local como remota. Tiene un algoritmo que transfiere solamente las diferencias entre el origen y el destino. Además de datos puede

transferir atributos de archivos. Estas características hacen de rsync un excelente utilitario para hacer backups.

Opciones principales del comando rsync

-a	Copia: • Recursivamente • Preservando los permisos • Preservando propietarios • Preservando archivos especiales • Preservando enlaces simbólicos • Preservando fecha de modificación
-v	Muestra más detalles
--progress	Muestra el progreso de la transferencia
--link-dest=DIR	Crea enlaces duros hacia los archivos del directorio DIR que no han sido modificados
-n (o **--dry-run**)	Hace una simulación, pero no copia nada realmente
--log-file=ARCHIVO	Deja un log de la transferencia en el archivo ARCHIVO
--delete	Borra los archivos que ya no están más en el origen
-b (o **--backup**)	Hace backups de los archivos borrados
--backup-dir=DIRECTORIO	Se usa junto a **-b** para guardar los archivos borrados en DIRECTORIO
--delete-before	Idem **--delete**: pero los borra antes de la transferencia
--delete-during	Id. ant.: pero los borra durante la transferencia
--delete-delay	id.ant: pero busca los archivos que ya no están en el origen para borrarlos al finalizar la transferencia
--delete-after	Borra los archivos que no existen en el origen en el destino al final de la transferencia
--bwlimit	Limita el ancho de banda utilizado en la conexión
--suffix	Agrega un sufijo a los archivos copiados con --backup
--checksum	En lugar de comparar los archivos por tamaño o por fecha de modificación, salteará archivos con el mismo hash (suma de comprobación) en origen y destino

Copiar archivos

Para transferir archivos de un host a otro podemos hacer lo siguiente:

rsync -av --progress --delete /home 192.168.1.10:/backup

En el ejemplo de arriba el directorio /home y todo su contenido es copiado (de manera predeterminada usando ssh) al directorio /backup del host 192.168.1.10.

Copiar archivos y crear backups

En el ejemplo siguiente el contenido del directorio /etc se copia a /media/disco/etc.copia. Los archivos que no existen más se copian en el directorio /media/disco/etc.borrados

#rsync -av --backup --backup-dir=/media/disco/etc.borrados /etc/ /media/disco/etc.copia

Verificación del backup

Si bien no hay una manera directa de verificar un backup con rsync se puede utilizar la herramienta md5deep para calcular hashes de manera recursiva:

md5deep -r /home > /tmp/home.sums

md5deep -r /backup > /tmp/backup.sums

Además, se puede usar la opción **--checksum** que da un resultado más exacto a costa de menor rendimiento:

 # rsync -av --checksum --progress --delete /home 192.168.1.10:/backup

Crear copias espejadas

rsync -av --progress --delete 192.168.1.10://home/ /home.viernes

rsync -av --progress --delete --link-dest=/home/viernes 192.168.1.10://home/ /home.sabado

Recuperación total de un backup

Si queremos recuperar todo el backup por pérdida de archivos en /home, sencillamente invertimos el orden, sin la opción --link-dest:

rsync -av --progress --delete /home.sabado 192.168.1.111://home/

Recuperación parcial de un backup

También puede ser que sea necesario recuperar solamente un archivo o directorio, en cuyo caso, por ejemplo haríamos:

 # rsync -av --progress --delete /home.sabado/juan 192.168.1.111://home/

En el caso de arriba el contenido del directorio /home remoto se copia en /backup.viernes. Al otro día se vuelve a ejecutar con rsync, pero esta vez le decimos que todo el contenido que no haya cambiado se replique /home.sabado mediante enlaces duros a /home.viernes. De esta manera solamente se copiarán los incrementos.

Usar un script para hacer copias espejadas durante una semana

Podemos escribir un script en un archivo llamado **backup.sh** como se muestra a continuación:

Copiar a Clipboard

#!/bin/bash

mount /dev/sdb1 /media/disco

Borramos el backup más viejo

rm -rf /media/disco/home.6

cd /media/disco

Rotamos los “espejos”

mv home.5 home.6

mv home.4 home.5

mv home.3 home.4

mv home.2 home.3

mv home.1 home.2

Rotamos el backup más reciente

mv home.0 home.1

rsync -av --delete --link-dest=home.1 /home/ home.0

Le asignamos permisos de ejecución:

chmod 755 backup.sh

Herramienta rdiff-backup

rdiff-backup es un script de python para respaldar directorios, pero a diferencia de rsync, guarda en el directorio de destino los incrementos que se van realizando, de modo que se puede restaurar un backup de una fecha en particular. Se vale de ssh para realizar copias de manera remota.

Realizar un backup

rdiff-backup -v5 /home /media/backup

Obtener estadístias del backup

rdiff-backup-statistics /home /media/backup

Verificar el estado de un backup

rdiff-backup --verify /media/backup

Obtener la cantidad de incrementos del backup

rdiff-backup --list-increments /home /media/backup

Mostrar los archivos que cambiaron en las últimas 24 hs

rdiff-backup --list-changed-since 1D /home /media/backup

Recuperar el backup de las últimas 24 hs

rdiff-backup -v5 -r 1D /media/backup /home.old

La opción -v indica el nivel de detalle que puede ir desde 0 hasta 9.

Herramienta duplicity

Duplicity utiliza un algoritmo que permite transferir los deltas de archivos binarios, y además:

- Crea un tarball a partir de archivos y carpetas cifrado con GnuPG
- El backup puede ser creado en el sistema local o en uno remoto
- Soporta archivos borrados, permisos de Unix completos, uid/gid, directorios, enlaces simbólicos, tuberías, etc.

Realizar un backup

duplicity /home file:///media/backup

Este comando nos pide una contraseña para cifrar el backup con una clave simétrica GnuPG. Si se vuelve a ejecutar el mismo comando duplicity realizará un backup incremental.

Verificar un backup

duplicity verify file:///media/backup /home

Restaurar el backup

 # duplicity file:///media/backup /home.restored

Sistemas de backup

En entornos profesionales y/o corporativos se utilizan soluciones más sofisticadas, describiremos tres de ellas.

BackupPC

BackupPC es un sistema para hacer copias de seguridad de Linux, Windows, y MacOS en discos creado por Craig H. Barrat. Extrae copias de respaldo usando SAMBA, rsync o tar por por ssh/rsh/nfs.

- Algunas de sus características son:
- Los archivos idénticos a través de múltiples backups de la misma o diferentes computadoras se guardan una sola vez.
- No hace falta un software especial en los clientes.
- Interfaz web para ver archivos de logs, configuración, estado actual, arrancar y cancelar backups, navegar y recuperar archivos de las copias de respaldo.
- Soporta entornos móviles con IP dinámica.
- Se envía a los usuarios recordatorios por e-mails si sus computadoras no han sido respaldadas.

Amanda (Advanced Maryland Automatic Network Disk Archiver)

Fue desarrollado originalmente por James Da Silva mientras estaba en el Departamento de Ciencias de la computación en la Universidad de Maryland. Algunas de sus características son:

- Está construido sobre herramientas estándares de backup, tales como dump/restore, GNU TAR, etc.
- Mantiene un catálogo de los archivos que están siendo respaldados y su ubicación el medio
- Al restaurar, dice qué cintas se necesitan, y encuentra la imagen del backup apropiada en la cinta
- Soporta intercambiadores de cintas por medio de una interfaz genérica.
- Soporta Disk Staging, es decir el uso de discos adicionales para luego transferir el backup a una cinta o a la nube
- Soporta comunicaciones seguras entre servidor y cliente usando OpenSSH

Bacula

¿Cómo es?

- Funciona bajo una arquitectura cliente/servidor
- Fácil de usar
- Eficiente
- Diseño modular
- Escalable

Algunas de sus prestaciones

- Niveles de backups: Full, Diferencial, Incremental, y otros más avanzados.
- Soporte para autochangers
- Backup a tape, disco y DVD
- Copiado y migración de backups
- Catálogo de SQL
- Interfaces gráficas de administración y/o monitoreo
- Multiplataforma
- Notificaciones
- Transmisión de datos por red cifrada mediante TLS
- Deduplicación a nivel archivo

Componentes

Bacula Director

Bacula Director se encarga de manejar los clientes (es decir los hosts que se van a respaldar), administrar los trabajos de backups y recuperación y conectarse con el dispositivo de almacenamiento.

Bacula File Daemon

Es el agente de Bacula que se ejecuta en el host que se va a respaldar. El cliente obtiene acceso al Director mediante autenticación.

Bacula Storage Daemon

Es el servicio que se encarga de controlar el dispositivo de almacenamiento de backups.

Bacula Catalog

Es la base de datos que almacena toda información acerca de los trabajos, clientes, volúmenes, etc.

Bacula Console

Bacula Console (bconsole) es el programa de línea de comandos mediante el cual se puede interactuar con Bacula Director y ejecutar distintos comandos: listado de trabajos de backup, de recuperación, ver el estado de servicios y clientes, etc. En definitiva, es la interfaz en modo texto de Bacula.

Bacula Director, Bacula Storage Daemon y el servidor de base de datos podrían ejecutarse en una sola máquina.

Bareos

Bareos es un derivado de Bacula, algunas de sus funcionalidades son:

- Backup de doble vía NMDP (transfiere directamente los archivos de un storage a una librería de cintas por la SAN).
- Tiene un plugin para realizar backups de instantáneas de máquinas virtuales de VMWare.
- Interfaz web

Webliografía y fuentes de consulta

- [Backup - Wikipedia](https://en.wikipedia.org/wiki/Backup)
- [dd(1) - Linux manual page](http://man7.org/linux/man-pages/man1/dd.1.html)
- [rsync(1) - Linux manual page](http://man7.org/linux/man-pages/man1/rsync.1.html)
- [tar(1) - Linux manual page](http://man7.org/linux/man-pages/man1/tar.1.html)
- [BackupPC: Open Source Backup to disk](https://backuppc.github.io/backuppc/BackupPC.html)
- [The Open Source Backup Wiki (Amanda, MySQL Backup, BackupPC)] (http://wiki.zmanda.com/index.php/Main_Page)
- [Bacula Main Reference](http://www.bacula.org/7.4.x-manuals/en/main/Main_Reference.html)
- [Bareos Open Source Data Protection - bareos.org](https://www.bareos.org/en/)
- [Incremental Backups on Linux » ADMIN Magazine](http://www.admin-magazine.com/Articles/Using-rsync-for-Backups)
- [Incremental Backups on Linux » ADMIN Magazine](http://www.admin-magazine.com/Articles/Using-rsync-for-Backups/(offset)/2)
- [Easy Automated Snapshot-Style Backups with Rsync] (http://www.mikerubel.org/computers/rsync_snapshots/)
- [Bare-metal server restore using tar | Linux.com | The source for Linux information] (https://www.linux.com/news/bare-metal-server-restore-using-tar)