# rsync es una herramienta open source para transferir archivos de forma incremental que se utiliza como herramienta de backup

comandos:

$ rsync -av source destination

(crea un backup de source en destination, “a” une las opciones habituales de rsync (-rlptgoD bajo un mismo flag, y “v” (verbose) añade output informativo al comando)

ejemplo:

rsync -av /home/usuario/Escritorio/Seguridad /var/tmp/Backups

$ rsync -av --link-dest=”full” “orig” “orig\_1”

(crea un backup incremental)

mas variaciones:

--compare-dest=DIR

This option instructs rsync to use DIR on the destination machine as an additional hierarchy to compare destination files against doing transfers (if the files are missing in the destinationdirectory). If a file is found in DIR that is identical to the sender's file, the file will NOT be transferred to the destination directory. This is useful for creating a sparse backup of just files that have changed from an earlier backup.

--copy-dest=DIR

This option behaves like --compare-dest, but rsync will also copy unchanged files found in DIR to the destination directory using a local copy. This is useful for doing transfers to a new destination while leaving existing files intact, and then doing a flash-cutover when all files have been successfully transferred.

--backup-dir=DIR

In combination with the --backup option, this tells rsync to store all backups in the specified directory on the receiving side. This can be used for incremental backups. You can additionally specify a backup suffix using the --suffix option (otherwise the files backed up in the specified directory will keep their original filenames).

Backups en mysql

$ mysqldump --user=TU\_USUARIO --password=TU\_CONTRASEÑA NOMBRE\_BASE\_DE\_DATOS > copia\_seguridad.sql

sirve para crear la copia de seguridad

$ mysql --user=TU\_USUARIO --password=TU\_CONTRASEÑA < copia\_seguridad.sql

sirve para recuperar los datos

AUTOMATIZAR

<https://www.cduser.com/como-automatizar-los-backup-de-mysql-con-un-script-y-crontab/>

john

john –single nombrefichero: Realiza el ataque usando las reglas más sencillas, usando las combinaciones más típicas, las palabras más usadas (en inglés), etc.

• john –wordlist:ficheropalabras nombrefichero: Realiza el ataque usando las palabras del ficheropalabras. En dicho fichero cada palabra debe estar en una línea.

• john –wordlist:ficheropalabras –rules nombrefichero: Realiza el ataque usando las palabras del ficheropalabras y además les aplica una serie de reglas para modificarlas (concatenarles números, por ejemplo) y generar más opciones. Aunque no vamos a entrar en ello, se pueden modificar las reglas o programar nuevas.

• john –incremental: modo nombrefichero: Realiza el ataque usando todas las combinaciones posibles de los caracteres que se indiquen en el modo. El modo puede ser digits para los dígitos, Alpha para las letras mayúsculas y minúsculas, Lower para las letras minúsculas, All para todos, etc. Por defecto viene configurado para que las combinaciones sean de hasta 8 caracteres.

• john –show nombrefichero: Muestra los password ya obtenidos en ese fichero. Sistemas de Gestión de Seguridad de la Información 2021/2022

• john nombrefichero: Realiza el ataque más completo intentando distintas técnicas (puede tardar horas, días, etc.).

• john –session=nombresesion restoparametros: Permite poner un nombre a la sesión y si se interrumpe, recuperarla en un futuro.

• john –restore=nombresesion: Recupera la sesión y continúa en el punto donde se dejó.