

GNU Wget

GNU Wget es poderoso. Tremendamente poderoso, nos atrevemos a afirmar que es uno de los pilares que sostiene al internet (teniendo en cuenta que 90% de nuestro mundo digital conocido es el hipertexto, **http**) ya que viene con todas las distribuciones GNU Linux y casi todos los servidores web del mundo usan Linux. Otro comando que consideramos pilar fundamental es **cURL** el cual quedo absorbido -y eclipsado, la verdad- por el lenguaje PHP y al cual ambos [dedicamos un extenso articulo al respecto](#).

Historia

Su desarrollo comenzó en 1995 con el nombre de Geturl pero fue cambiado a Wget debido a que ya existía otro programa con ese nombre en la plataforma AmigaOS. Debido a su capacidad de manejar ambos HTTP y FTP aunado que provenía de ambiente UNIX hicieron la amalgama perfecta para la aceptación por parte de los usuarios.? El hardware y conexiones de la época eran lentas y con multitud de errores de transmisión por lo que se necesitaba una automatización del trabajo y Wget encajó muy bien y en el momento oportuno.

Introducción

Desde los años 1990 su objetivo ha sido siempre el mismo: descargar archivos de acuerdo a varios protocolos, incluso con diferentes formas y maneras de autenticación y transporte. Brilla por su reinicio de descargas y gran variedad de opciones mezcladas. Como todo en el ambiente GNU/Linux hace una sola cosa y la hace bien pero no solo eso, *responde bien también*, es decir, es una función también y devuelve valores rápidos para tomar decisiones en [guiones BASH](#) (también tiene una opción de registro de eventos que analizaremos).

Los valores devueltos al ejecutar **wget** en la línea de comandos o guion:

- 0 (cero): ningún problema.
- 1 (uno): código de error genérico.
- 2 (dos): cuando descargamos ciertos tipos de archivos, los que querramos específicamente, si hay algún error vendrá por este código.
- 3 (tres): error de entrada o salida de archivo (por ejemplo, no tenemos derecho de escritura en una carpeta).
- 4 (cuatro): error de red (wget es bueno pero si no tenemos conexión a internet, pues...)
- 5 (cinco): error en el manejo de conexiones seguras tales como SSL.
- 6 (seis): error de autenticación de usuario (error de contraseña, etc.).
- 7 (siete): error de protocolo (error muy poco común).

- 8 (ocho): error de respuesta del servidor (vamos, que nos salen con algo que no conocemos, también poco común).

Cada error menor priva sobre otro mayor, así que debemos tener siempre presente que *pueden ocurrir distintos tipos de errores al mismo tiempo y sin embargo **wget** siempre devolverá un solo valor (no hay suma de valores porque los números no son potencias de base dos para discernir).*

Usos prácticos

Bien, ya fue suficiente de teoría, ésta es buena y necesaria hasta cierto punto (totalmente necesaria si pertenecemos al equipo de desarrollo de **wget** pero por ahora no es así).

Descargando un archivo

Lo primero que haremos es crear una carpeta en nuestra cuenta de usuario y luego iremos a ese directorio a fin de tener todos nuestros ejemplos en un solo sitio y evitar así el mínimo de confusiones con nuestros otros archivos de datos cotidianos. Todos estos ejemplos los hemos probado en un ordenador cliente con Ubuntu 16.04 y nuestra página web con un subdominio que hicimos por allá en 2014 sobre un tutorial especial para instalar MySQL en Canaima 4.0, una distribución oficial GNU/Linux del Estado Venezolano.

Lo primero que haremos es descargar un fichero de imagen, la orden es, simplemente, **wget url_a_descargar**:

```
wget http://www.fundacite.ks7000.net.ve/img/01MySQL_en_Canaima4_mini.jpg
```

Y veremos *algo parecido a esto (haced click en la imagen para abrirla por completo)*:

Luego si hacemos un simple **ls** veremos el fichero en nuestra carpeta, y le aplicamos el comando para borrar **rm** ([revisad nuestro tutorial para trabajar con BASH](#) si necesitan recordar o repasar cualquier cosa).

Descargando varios ficheros por extensión de archivo

De manera predeterminada los servidores web están capacitados para servir siempre un archivo llamado "index.php" o "index.html" y cuando ninguno de dichos archivos esté presente (ni ningún otro archivo con esas extensiones) en un directorio entonces se procede a enviarnos un listado de dichos ficheros. En este caso particular podremos descargar los archivos de un mismo tipo con un caracter comodín (eso se conoce como "globbling") con el siguiente comando:

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

```
wget http://www.fundacite.ks7000.net.ve/img/*.png
```

Pero resulta que esto no funciona todo el tiempo, en nuestro caso el servidor web tiene instalado WordPress y lo que nos devuelve es una página de error de página no encontrada (la famosa 404) debido a las órdenes grabadas en el archivo.htaccess (tenemos un [tutorial publicado sobre esto](#)), en ese caso trataremos de usar el modo avanzado de **wget**:

```
wget -r -l1 --no-parent -A.gif http://www.ejemplo.com/directorio/
```

- "-r" significa recursivo
- "-l1" significa nivel uno en esa recursividad.
- "--no-parent", por así decirlo, circunscribe la descarga solo al directorio solicitado.
- "-A.png" equivale a "*.png".

Sin embargo se debe verificar la configuración global del servidor web, en nuestro caso tratamos de deshabilitar el fichero .htaccess sin embargo no nos fue posible evadir ese sistema de seguridad. Ojalá funcione de igual manera para cuando de verdad vengan los atacantes maliciosos a nuestro sitio web.

Descargando con diferentes protocolos

Descargando desde un listado de enlaces guardados en un fichero

Descargas fallidas y su recuperación

Detalles operacionales de una descarga

Limitar la velocidad de descarga

Credenciales de usuario al descargar

Descargar una página web, su código HTML

Descargar una página web completa

Descargar un sitio web completo

Convertir los enlaces web a locales

Convertir todas las extensiones a .html

Argumentos y parámetros

Fuentes consultadas:

En idioma castellano:

- [GNU wget en Wikipedia.](#)

En idioma inglés:

- [GNU wget at GNU org.](#)
- «[The curl and wget war](#)» by Daniel Stenberg.