

# Cómo convertirse en hacker. Nat Friedman.

Traducido por Albert Abril.

Lunes en una entrevista en Beijing, alguien me preguntó como ser un hacker. (Este que está visitando tu web desde un artículo de la People Magazine. Constar que el término *hacker* se refiere a programador, no cómo delincuente de internet ).

Mi interlocutor quiso saber como contribuir en un proyecto de código abierto, qué herramientas debería usar, que libros debería leer, donde debería publicarse, donde debería empezar...

Se me ha preguntado algunas veces sobre esto y he pensado que debería repetir mi respuesta aquí. Miguel (de Icaza) me ha dicho que a él se lo han preguntado varias veces también, y tiene la misma respuesta que yo.

Así, voy a dejarte conocer el secreto. Aquí están los pasos de como convertirse en hacker:

1. Descarga el código fuente del programa que quieras cambiar.
2. Descomprímelo en tu disco duro.
3. Construyelo y hazlo funcionar.
4. Abre el código fuente en el editor.
5. Encuentra la parte del código que necesite el cambio para hacer que el programa haga lo que tu quieras que haga.
6. Realiza los cambios necesarios para construir el código, pruébalo y comprueba que funcione.
7. Ejecuta el `diff -u comm` y envía la salida a la lista de correo.

Esto es, sigue las instrucciones y te garantizo que te convertirás en hacker.

Si no existen programas que te gustase cambiar, entonces no querrás ser un hacker después de todo. O tal vez, no hayas usado suficiente software; cómo quieres ser un usuario de software en 2005 y no saber que cambios quieres hacer?

Los pasos 1-4 suenan estúpidos y obvios, pero la verdad es que mucha gente de queda pegado en el punto 1. Cómo puedes ser un hacker si no tienes código fuente en tu computador? Puede ser posible, pero yo todavía no lo he visto.

Si haces sangrar tus dedos en el paso 3 unas pocas veces, no te desanimes. Es ridículo y humillante pero algunas veces este paso es el más largo y el más complicado.

Si tienes suerte, el paso 5 es tan fácil como usar `grep` en el directorio del código en busca de alguna cadena revelante del GUI del programa o de la salida. Necesitas pasar bastante tiempo imaginándote el diseño del código, desglosando archivos fuente con ***printf's*** hasta que encuentres las líneas correctas. Usar un *debugger* también ayudará a avanzar.

El paso 5 va resultando más fácil según la experiencia que vas adquiriendo. Cuando más código hayas leído, más aspectos de programación conoces. Reconocer lenguajes de programación hace más fácil imaginar en qué estaba pensando quien escribía el código que estás intentando cambiar. Por supuesto, el paso 5 es más fácil si el código sobre el que estás trabajando fue escrito por un programador con mucha experiencia, quien intentó hacer un sobreesfuerzo para escribir código fácil de entender. Los programadores con experiencia escriben código fácil de leer porque alguna vez han tenido la frustración de tener que arreglar un *bug* (fallo) en el código que ellos escribieron un año antes y más tarde no han entendido nada.

El paso 6 es comúnmente reconocido como "*hacking*" pero no siempre es la parte más larga. Si estás intentando de hacer un cambio (hacer un *hack*) en algo grande y bastante complejo,

debes saber que el paso 5 va a eclipsar mucho tiempo del paso 6. Hace poco, uno de los mejores hackers de Novell estuvo dos meses trabajando en un *hack* en el proyecto Wine en el que acabó siendo un cambio de dos líneas. Entonces, prepárate mentalmente en usar mucho tiempo del paso 5 antes de adentrarte en el paso 6, y algunas veces volver al paso 5 desde el 6 varias veces.

Pero la mayoría de la gente no alcanza este punto, así que si estás en el paso 6 puedes llamarte con seguridad *hacker*. Hay libros enteros que explican cómo hacer un buen trabajo en el punto 6, no voy a elaborar demasiado aquí, excepto decir que probablemente no seas bueno escribiendo código hasta que no hayas escrito una gran cantidad de éste.

La verdadera clave para ser un hacker es centrarte cuando estes hackeando. Sin código, un trabajado conocimiento compilando y un trabajado conocimiento de diseño del código, no serás capaz de empezar a hacer *hacking*. Pero una vez conozcas el funcionamiento y te hayas adentrado escribiendo código y viendo como el programa va cogiendo forma, bien, esta es la parte divertida.

**Nat Friedman.**

Traducido por Albert Abril ( albert.abril@gmail.com ).