Recientemente algún hater de hacks4geeks (o hater personal mío), ha estado atacando mis webs. El ataque dio resultado, pero no causó destrozos porque estoy más o menos preparado para ciertas acciones dado que tengo copias de seguridad de todo. De hecho, no creo que como usuarios hayan notado nada.
Mayoritariamente el ataque consistió en meter código malicioso dentro de los archivos php de wordpress y hacer otros cambios en el sistema.
¿Cómo me di cuenta de esto? Porque tengo configurado un sistema que va revisando cada 5 minutos, algunos archivos importantes que tengo configurados en un XML en casa, y luego mira los tamaños correctos de esos archivos en una carpeta predefinida en mi ordenador y los compara con los archivos que hay en la carpeta de mi hosting. Si alguno de esos archivos cambia de tamaño, el sistema me envía un mail y activa una luz especial que tengo configurada en mi blink(1) mk2 y que no se apaga hasta que le de la orden.
Evidentemente, en cada actualización de WordPress, me salta la luz y me llega el mail, pero fuera de eso, el sistema sólo salta cada vez que todo el theme. En realidad, ahora, lo que hago es desactivar el sistema, actualizar los archivos del theme y luego reactivarlo. Lo que pasa es que, una gran cantidad de veces me olvido de reactivarlo cuando acabo de modificar los archivos.
¿Qué ocurre entonces? Pues evidentemente, que cuando lo reactivo saltan las alarmas y me avisa. Lo que hago es simplemente reemplazar y fuera. No me enloquezco en ver qué código metieron ni nada. Es una costumbre que me creé cuando tenía montado mi servidor en casa. En vez de enloquecerme aprendiendo cosas de seguridad que nunca voy a usar, aplicaba el reemplazo automático de los archivos y fuera. Como mucho, los archivos modificados duraban así sólo 5 minutos.
Si, muchos estaréis pensando que podía cambiar la frecuencia con la que ejecutaba el script y el ataque duraría menos. Pero por alguna razón parecia que el proceso se tardaba unos 4 minutos y pico en revisar todos los archivos. Y no importaba que le pusiera menos archivos a las carpetas. Nunca quise indagar más en eso tampoco. Le puse 5 minutos y fuera.

El caso es que este ataque hizo algo que no me había pasado antes: no sólamente cambió los archivos sino que luego le cambió los permisos, entonces no pude volver a reemplazarlos. Además, cuando WordPress se quiso actualizar, que fue más tarde ese mismo día (deben haber parcheado la vulnerabilidad), tampoco pudo.
Pensé entonces: Bueno, me conecto por SSH, cambio los permisos y fuera. Pero resulta que mi hosting contratado me permite conectarme por SSH pero no me permite modificar permisos porque es un servidor compartido. chown 777 daba permission denied.
Fuck, pensé. Esto debe tener solución desde una cuenta propia, porque estoy haciendo chown a una carpeta y archivos a los que se supone que tengo acceso. Independientemente que no sea root.

Así que, después de un rato de darle di con una solución que seguramente ayudará a más de uno:

Loquéate mediante SSH. Cambia todas las carpetas a 755 con:

```
find /ruta/a/tu/carpeta/de/wordpress/en/el/servidor -type d -exec chmod 755 {} ;
```

Cambia todos los archivos a 644 con:

```
find /ruta/a/tu/carpeta/de/wordpress/en/el/servidor -type f -exec chmod 644 {} ;
```

Y finalmente cambia a 777 con:

```
chmod 777 /ruta/a/tu/carpeta/de/wordpress/en/el/servidor
```

Eso reparó mis permisos sin ser root. Evidentemente, ahora, lo que haré será agregar esas líneas de código a un script aparte, y en script de advertencia de cambios, agregaré una parte que copie el script de los permisos al servidor mediante SSH, lo ejecute y después realice los cambios.

Moraleja: No hay mal que por bien no venga. Ahora mi script es más potente y prácticamente no tendré ni que mirar el funcionamiento. Todo se hará automáticamente.

PD: Si estás pensando en que comparta el **script de comprobación**: Ningún problema. Tengo programado un post para dentro de 120 años. Acuérdate de agregar un recordatorio.