

Sec Maniac.com

Wordpress Security

A guide on how to not get hacked
when using wordpress....

David Kennedy (ReL1K)

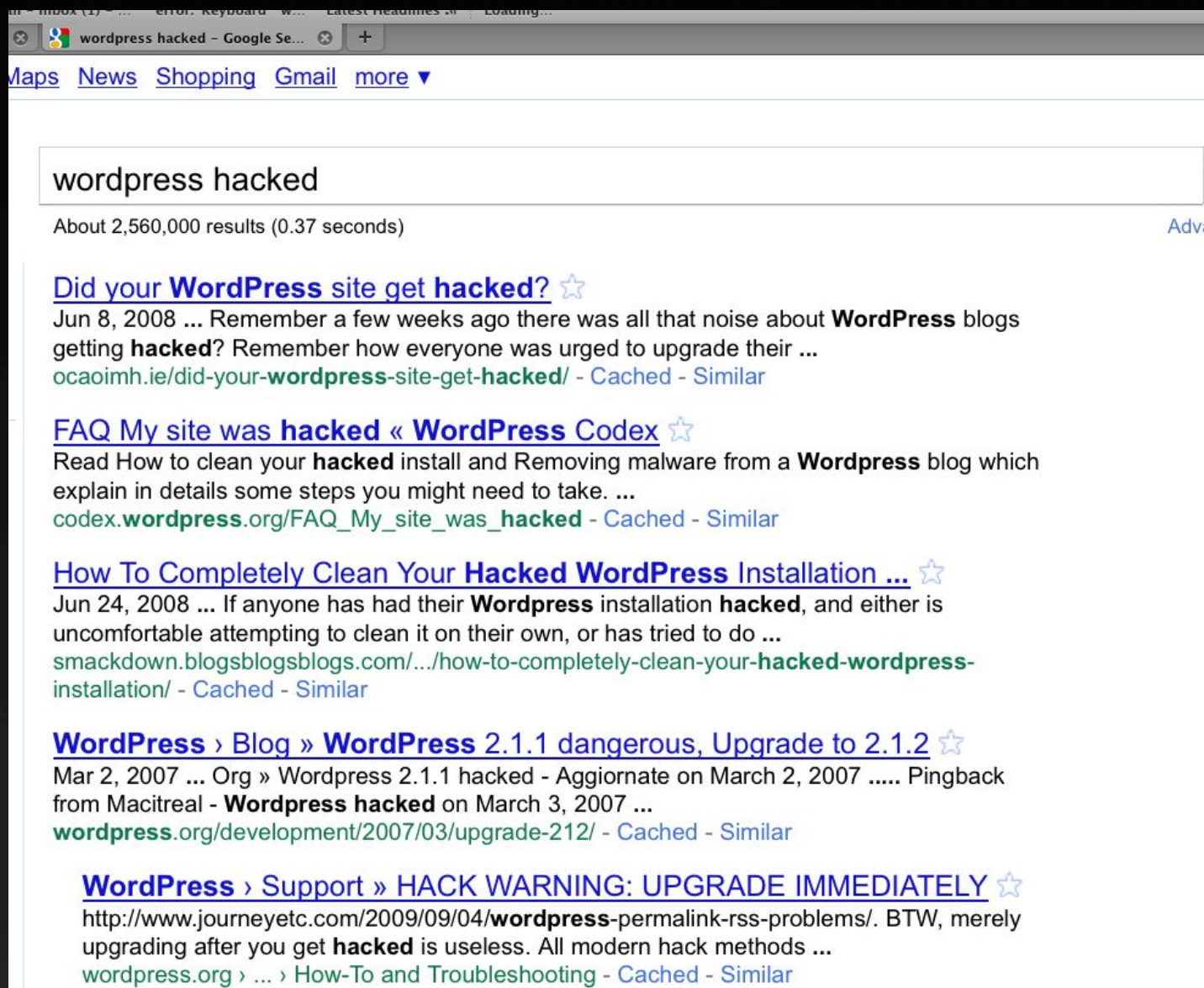
<http://www.secmaniac.com>

Twitter: Dave_ReL1K

So about wordpress....

- The number one website and blogging software out there to date.
- A free and open-source based project with heavy backing for creating entire websites and creating dynamic blog/information sharing.
- PHP + MySql based sites has been hacked a number of times, and probably will be hacked again... ☺

Wordpress Hacked?!



wordpress hacked

About 2,560,000 results (0.37 seconds)

[Did your WordPress site get hacked?](#) ☆
Jun 8, 2008 ... Remember a few weeks ago there was all that noise about **WordPress** blogs getting **hacked**? Remember how everyone was urged to upgrade their ...
ocaoimh.ie/did-your-wordpress-site-get-hacked/ - Cached - Similar

[FAQ My site was hacked « WordPress Codex](#) ☆
Read How to clean your **hacked** install and Removing malware from a **Wordpress** blog which explain in details some steps you might need to take. ...
codex.wordpress.org/FAQ_My_site_was_hacked - Cached - Similar

[How To Completely Clean Your Hacked WordPress Installation ...](#) ☆
Jun 24, 2008 ... If anyone has had their **Wordpress** installation **hacked**, and either is uncomfortable attempting to clean it on their own, or has tried to do ...
smackdown.blogsblogsblogs.com/.../how-to-completely-clean-your-hacked-wordpress-installation/ - Cached - Similar

[WordPress » Blog » WordPress 2.1.1 dangerous, Upgrade to 2.1.2](#) ☆
Mar 2, 2007 ... Org » Wordpress 2.1.1 hacked - Aggiornate on March 2, 2007 Pingback from Macitreal - **Wordpress hacked** on March 3, 2007 ...
wordpress.org/development/2007/03/upgrade-212/ - Cached - Similar

[WordPress » Support » HACK WARNING: UPGRADE IMMEDIATELY](#) ☆
<http://www.journeyetc.com/2009/09/04/wordpress-permalink-rss-problems/>. BTW, merely upgrading after you get **hacked** is useless. All modern hack methods ...
wordpress.org > ... > How-To and Troubleshooting - Cached - Similar

Some of the major risks...

- Plugins are third party tools that can be added to wordpress to enhance functionality or security.
- Plugins are third party tools that can be added to wordpress to decrease functionality or security.
- Plugins are vulnerable and are often the brunt for most of the attacks. Ensure you know what your using when you install your plugins..

What risk do you have?

- Injection flaws (including OS), XSS, full system compromise, etc.
- As with any type of software, as it becomes more popular the more it's attacked.
- Wordpress is a major attack vector right now as many websites use it.

My setup – Pick and choose

- Reverse Proxy running application firewall (you can use `mod_security` for free).
- Heavily stripped down kernel
- No access other than port 80, host integrity monitoring through OSSEC and deny hosts. Virtualized environment that is snapshotted and backed up on a daily basis.
- Remove all dynamic content until I need it (will explain).

Reverse Proxy

- The reverse proxy tunnels HTTP based traffic on a non-standard port in my back-end network. Multi-tiered DMZ that separates the MySQL database from the front-end web application.
- Egress filtering is heavily performed to not allow outbound connections except to Fedora updates and Wordpress.
- Mod-Security, free open source rules for Apache, a signature based WAF.

Stripped down OS

- Only install what you need.
- Remove any kernel level packages you don't need.

Non Standard Ports

- Move SSH off of its default port and lock it down with Denyhosts if you use it.
- DenyHosts is a brute force detection tool that after X amount of attempts will block the host IP through IPTables
- On the reverse proxy, recommend moving your back-end HTTP server and MySQL ports to non-standard ports.

The BIG One .htaccess .htpasswd

- In most cases the wp-includes folder will allow directory browsing, make sure you remove that and lock down access to only what you need:
 - Order Allow,Deny
 - Deny from all
 - `<Files ~ "(css|jpe?g|png|gif|js)$">`
 - Allow from all
 - `</Files>`

.htaccess continued

- Lock down the wp-admin interface in wp-admin/

```
# Auth File
AuthUserFile /var/www/html/wp-admin/.htpasswd
AuthGroupFile /dev/null
AuthName "Not Found"
AuthType Basic
<Limit GET>
require valid-user
</Limit>
<Files .htaccess>
deny from all
</Files>
```

Only allow by IP Address

- Lock down the wp-admin interface in wp-admin/

AuthName "OMG"

AuthType Basic

<Limit GET POST>

order deny,allow

deny from all

allow from X.X.X.X

</Limit>

Blacklist IP Addresses

- Quick script I wrote if you want to block via IP Tables, could also use `.htaccess`

```
#!/usr/bin/python
import subprocess
choice=raw_input("Enter IP to blacklist: ")
subprocess.Popen("iptables -I INPUT -s %s -j DROP" %
    (choice), shell=True).wait()
print "Done.."
```

OSSEC HIDS

- Open-Source platform for monitoring platform integrity and active responses against attack.
- Be careful in the reverse proxy model, it can blacklist the reverse proxy on the back-end web server.
- Anything modified on the OS, email notifications are sent.
- Fine tune alerts and ensure email is working properly.

Other useful pointers

- User different passwords and ports on each *nix system.
- Rename the wordpress default MySQL tables (wp_blah). Also remove the xmlrpc.php file, seems to get targeted a lot.
- Anything modified on the OS, email notifications are sent.
- Fine tune alerts and ensure email is working properly.

Useful pointers cont.

- Make the main wordpress directory only writable by root when not using it.
- Ensure the wp-directories are not set for directory listings.
- If you don't care about reboots, etc. schedule auto cron jobs for updating the systems automatically and forcing a reboot afterwards

ihazwordpress.py

```
#!/usr/bin/python
# Quick down and dirty for removing dynamic modifications to Wordpress
#
# By Dave Kennedy
import subprocess
choice1=raw_input("""

What do you want to do:

1. Lockdown Wordpress
2. Enable Editing of Wordpress

Enter choice : """)
filewrite=file("/var/www/html/htaccess", "w")
if choice1== '1':

    print "Enabling ModSecurity...."
    subprocess.Popen("mv -f /root/base_rules /etc/httpd/modsecurity.d/base_rules", shell=True).wait()
    # Move wp-admin
    print "[-] Turning off Wordpress Admin Interface...."
    subprocess.Popen("mv -f /var/www/html/wp-admin /root", shell=True).wait()
    # chmod wordpress
    print "[-] Disallowing writes to wordpress...."
    subprocess.Popen("chown root:root -R /var/www/html" , shell=True).wait()
    # turn on DotDefender
    print "[-] Turning on DotDefender...."
    subprocess.Popen("/etc/init.d/dotDefender_auditd start", shell=True).wait()

if choice1 == '2':

    print "Disabling ModSecurity...."
    subprocess.Popen("mv -f /etc/httpd/modsecurity.d/base_rules /root", shell=True).wait()
    subprocess.Popen("mkdir /etc/httpd/modsecurity.d/base_rules", shell=True).wait()
    # Enable wp-admin
    print "[-] Enabling Wordpress Admin Interface...."
    subprocess.Popen("mv -f /root/wp-admin /var/www/html/", shell=True).wait()
    # chown wordpress
    print "[-] Changing permissions to allow editing...."
    subprocess.Popen("chown apache:apache -R /var/www/html", shell=True).wait()
    # turn off DotDefender
    print "[-] Turning off DotDefender...."
    subprocess.Popen("/etc/init.d/dotDefender_auditd stop", shell=True).wait()

print "Finished...."
█
```

Some useful Security Plugins

- Login Lockdown – Looks for failed attempts from the same source address and blocks if threshold hit.
- Secure Wordpress – Removes standard attack patterns, hides certain information leakage, renames files, etc.
- Don't install just any plugins, these are usually what causes you to get owned.

Sec Maniac.com

Questions? 😊

<http://www.secmaniac.com>
davek@social-engineer.org
Twitter: Dave_ReL1K