YGN ETHICAL HACKER GROUP

Article

HUNTING FOR BACKDOOR SCRIPTS

By

d0ubl3_h3lix
http://yehg.org

March 27, 2008

Have you ever made use of free/open-source scripts or applications on your web sites? If yes, how many times have you analyzed those scripts for ensuring no backdoor or bad scripts exist?

Many free applications on the web lack (security-related) documentations. Recently we have found a serious vulnerability in an open-source Gmail-Lite application, which allows attacker can do anything he wants – replacing legitimate php files with backdoor ones which steal usernames and passwords and send them to him secretly. We contacted Gmail-Lite author and he took immediate fix. Most web masters don't even closer look or analyze the source codes. They just extract, upload and install. If everything works, it's OK. Then next morning attackers replace those files with backdoors. Too many applications on the web become zombies which let attackers attack other web sites or hosts.

Never trust any applications without probing. In last year, the famous blogging application "WordPress" was backdoored by attackers in its distributed files.

Well, let's move on. In this article, I show you how to scan backdoor functions in php scripts/applications that you get freely from the web. You can do it with any a little bit advanced text editors which allow us (regular expression) find our desired string in specific directories. The following functions can be implemented by bad guys to backdoor your applications:

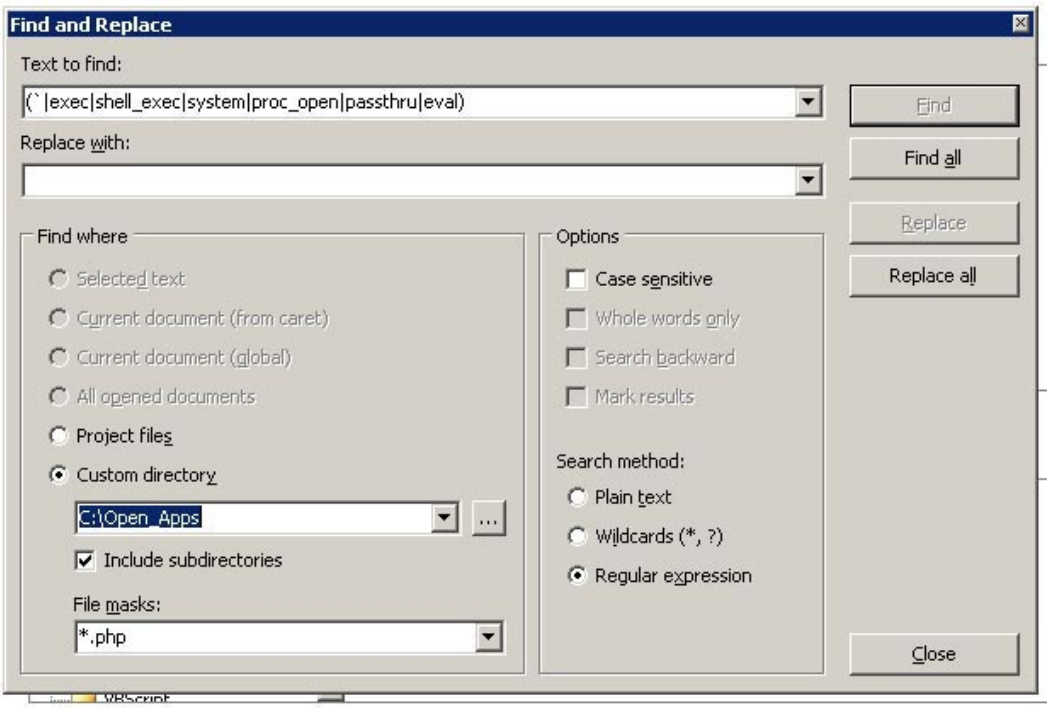
- exec -- Execute an external program
- passthru -- Execute an external program and display raw output
- shell_exec -- Execute command via shell and return the complete output as a string
- system -- Execute an external program and display the output
- proc_open -- Execute a command and open file pointers for input/output
- eval -- Evaluate a string as PHP code
- and an backtick operator (eg. echo " `ls –R` ")

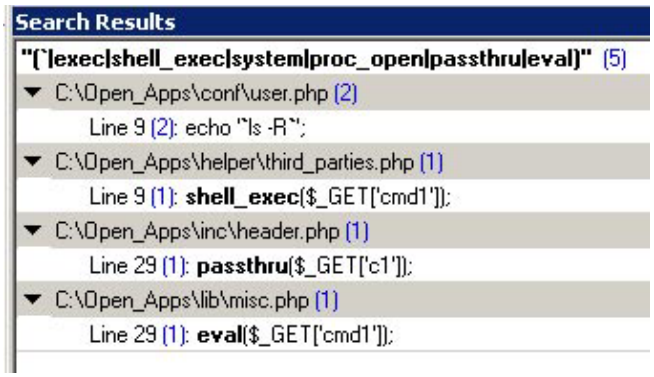
After the editor has found string that matches any above function names, take closer look at source codes and analyze them for any suspicious processing. Test calling such functions in a test page. Notice the intention of script author. In some cases, He doesn't notice his own flaw though he doesn't intend to make it backdoor. Then contact the author for fixes.

The regular expression format is:

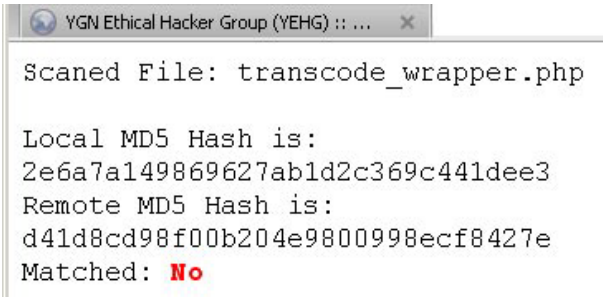
```
(`|exec|shell_exec|system|proc_open|passthru|eval)
```

Please turn over the next page for simple demonstration screenshots.





Then you rest assured to upload backdoor-free scripts and host them on your web sites for a long time. How could you know if an attacker hacked your servers and installed backdoors? Yes, you need to routinely do backdoor scanning process! The other alternative is to store and match[1] md5/sha1 hashes of your local php files and remote ones. If not matched, the file has been probably compromised!



[1] *Consult your php manual for* md5_file() /sha1_file() *functions.*