

# BSD HACKS

*100 Industrial-Strength  
Tips & Tools*



O'REILLY®

*Dru Lavigne*

HACK  
#67

## Automate Security Patches

Keep up-to-date with security patches.

We all know that keeping up-to-date with security patches is important. The trick is coming up with a workable plan that ensures you're aware of new patches as they're released, as well as the steps required to apply those patches correctly.

Michael Vince created `quickpatch` to assist in this process. It allows you to automate the portions of the patching process you'd like to automate and manually perform the steps you prefer to do yourself.

### Preparing the Script

`quickpatch` requires a few dependencies: `perl`, `cvsup`, and `wget`. Use `which` to determine if you already have these installed on your system:

```
% which perl cvsup wget
/usr/bin/perl
/usr/local/bin/cvsup
wget: Command not found.
```

Install any missing dependencies via the appropriate port (`/usr/ports/lang/perl5`, `/usr/ports/net/cvsup-without-gui`, and `/usr/ports/ftp/wget`, respectively).

Once you have the dependencies, download the script from <http://roq.com/projects/quickpatch> and untar it:

```
% tar xzvf quickpatch.tar.gz
```

This will produce an executable Perl script named `quickpatch.pl`. Open this script in your favorite editor and review the first two screens of comments, up to the `#Stuff` you probably don't want to change line.

Make sure that the `$release` line matches the tag you're using in your `cvsupfile` [Hack #80]:

```
# The release plus security patches branch for FreeBSD that you are
# following in cvsup.
# It should always be a long the lines of RELENG_X_X , example RELENG_4_9
$release='RELENG_4_9';
```

The next few paths are fine as they are, unless you have a particular reason to change them:

```
# Ftp server mirror from where to fetch FreeBSD security advisories
$ftpserver="ftp.freebsd.org";
# Path to store patcher program files
$patchdir="/usr/src/";
# Path to store FreeBSD security advisories
$admdir="/var/db/advisories/";
$advdirtmp="$admdir"."tmp/";
```

If you're planning on applying the patches manually and, when required, rebuilding your kernel yourself, leave the next section as is. If you're brave enough to automate the works, make sure that the following paths accurately reflect your kernel configuration file and build directories:

```
# Path to your kernel rebuild script for source patches that require kernel
#rebuild
$kernelbuild="/usr/src/buildkernel";
#$kernelbuild="cd /usr/src ; make buildkernel KERNCONF=GENERIC && make
#installkernel KERNCONF=GENERIC ; reboot";
# Path to your system recompile script for patches that require full
# operating system recompile
$buildworld="/usr/src/buildworld";
#$buildworld="cd /usr/src/ ; make buildworld && make installworld ; reboot";
#Run patch command after creation, default no
$runpatchfile="0";
# Minimum advisory age in hours. This is to make sure you don't patch
# before your local cvsup server has had a
# chance to receive the source change update to your branch, in hours
$advisory_age="24";
```

Review the email accounts so the appropriate account receives notifications:

```
# Notify email accounts, eg: qw(billg@microsoft.com root@localhost);
@emails = qw(root);
```

## Running the Hack

Run the script without any arguments to see the available options:

```
# ./quickpatch.pl
# Directory /var/db/advisories/ does not exist, creating
# Directory /var/db/advisories/tmp/ does not exist, creating
Quickpatch - Easy source based security update system
"./quickpatch.pl updateadv" to download / update advisories db
"./quickpatch.pl patch" or "./quickpatch.pl patch > big_patch_file" to
create patch files
"./quickpatch.pl notify" does not do anything but email you commands of what
it would do
"./quickpatch.pl pgpcheck" to PGP check advisories
```

Before applying any patches, it needs to know which patches exist. Start by downloading the advisories:

```
# ./quickpatch.pl updateadv
```

This will connect to <ftp://ftp.freebsd.org/pub/FreeBSD/CERT/advisories> and download all of the advisories to `/var/db/advisories`. The first time you use this command, it will take a while. However, once you have a copy of the advisories, it takes only a second or so to compare your copies with the FTP site and, if necessary, download any new advisories.

After downloading the advisories, see if your system needs patching:

```
# ./quickpatch.pl notify
#
```

If the system is fully patched, you'll receive your prompt back. However, if the system is behind in patches, you'll see output similar to this:

```
# ./quickpatch.pl notify
#####
##### FreeBSD-SA-04%3A02.shmat.asc
##### Stored in file /var/db/advisories/tmp/FreeBSD-SA-04%3A02.shmat
##### Topic: shmat reference counting bug
##### Hostname: genesis - 20/2/2004 11:57:30
##### Date Corrected: 2004-02-04 18:01:10
##### Hours past since corrected: 382
##### Patch Commands
cd /usr/src
# patch < /path/to/patch
### c) Recompile your kernel as described in
<URL:http://www.freebsd.org/handbook/kernelconfig.html> and reboot the
system.
/usr/src/buildkernel
## Emailed root
```

It looks like this system needs to be patched against the “shmat reference counting bug.” While running in notify mode, quickpatch emails this information to the configured address but neither creates nor installs the patch.

To create the patch, use:

```
# ./quickpatch.pl patch
#####
##### FreeBSD-SA-04%3A02.shmat.asc
##### Stored in file /usr/src/FreeBSD-SA-04%3A02.shmat
##### Topic: shmat reference counting bug
##### Hostname: genesis - 21/2/2004 10:41:54
##### Date Corrected: 2004-02-04 18:01:10
##### Hours past since corrected: 405
##### Patch Commands
cd /usr/src
# patch < /path/to/patch
### c) Recompile your kernel as described in
#<URL:http://www.freebsd.org/handbook/kernelconfig.html> and reboot the
#system.
/usr/src/buildkernel

# file /usr/src/FreeBSD-SA-04%3A02.shmat
/usr/src/FreeBSD-SA-04%3A02.shmat: Bourne shell script text executable
```

This mode creates the patch as a Bourne script and stores it in `/usr/src`. However, it is up to you to apply the patch manually. This may suit your purposes if you intend to review the patch and read any notes or caveats associated with the actual advisory.

## Automating the Process

One of the advantages of having a script is that you can schedule its execution with `cron`. Here is an example of a typical `cron` configuration for `quickpatch.pl`; modify to suit your own purposes. Remember to create your logging directories and touch your log files before the first run.

```
# Every Mon, Wed, and Fri at 3:05 do an advisory check and download any
# newly released security advisories
5 3 * * 1,3,5 root /etc/scripts/quickpatch.pl updateadv > \
  /var/log/quickpatch/update.log 2>1

# 20 minutes later, check to see if any new advisories are ready for use
# and email the patch commands to the configured email address
25 3 * * 1,3,5 root /etc/scripts/quickpatch.pl notify >> \
  /var/log/quickpatch/notify.log 2>&1

# 24 hours later patch mode is run which will run the patch commands if
# no one has decided to interfere.
25 3 * * 2,4,6 root /etc/scripts/quickpatch.pl patch >> \
  /var/log/quickpatch/patch.log 2>&1
```

## See Also

- The `quickpatch.pl` web site (<http://roq.com/projects/quickpatch>)
- The FreeBSD Security Advisories page (<http://www.freebsd.org/security/index.html#adv>)