Tools for Open Source Patch and Package Management

NebraskaCERT Conference

August 3 2004

Mathew Caughron, CISSP
PHP Consulting

Public Service Announcement

Please silence all pagers

and cell phones now.

Thank you!

Conference Theme: Practical Security

- Theoretical comparison is necessary when the differences are great.

In theory, theory and practice are the same.
In practice, they are not.

- Topic for this talk was chosen because it accomplishes one of the most common tasks, the day to day work of security - namely, keeping changes current on production systems.

Outline of this Talk:
Survey of Tools

- Application-Level Change Control Overview
- Patch Management versus Package Management
- Overview of Package Management Tools for
  - Major Linux distributions
  - *BSD and MacOSX
  - 32 bit Microsoft Windows installer/updaters
- Compare and contrast the approaches

Why Patch or Upgrade?

In business context, software is deployed for a purpose. That purpose can be threatened by a vulnerability anywhere in the system.

Weigh risk of vulnerability with risk of inaction.

Systems on networks tend to have greater need for patches for two reasons: network software is complex and the risk presented by remote access is typically greater.

Software Development Theory Points to Need for Specialized Patch Management Tools

- Low coupling - principle of dependence.
  - Keep update tasks and kernel separate. Can updates fail without affecting uptime?

- High cohesion - principle of modularity.
  - Operating system and change control systems are at cross purposes. OS needs to be stable, relatively unchanging

(Constantine and Yourdon)

What This Talk Isn't

- Everything you ever wanted to know about change control or patch management - just the highlights.
- Debate between open source versus proprietary approaches.
- Debate of which BSD or Linux distribution takes the best approach. The goal here is an executive summary or survey of tools across various platforms.

Release Engineering

- Rarely talked about in software development circles but is a necessary part of any successful project, especially anything that is widely deployed.
- Plays a part in the Nebraska economy - Mindvision founded by Steve Kiene.
- Defined as the body of knowledge necessary for the deployment and maintenance of software across distributed systems.

Generic Application-Level Change Control Tasks

- Install
- Uninstall or Remove
- Update
  - Incremental
  - Replacement

Goals of Package Management Systems

- Integrity -
  - OS and kernel space
  - document and data space
  - application space
    - binaries
    - components and libraries
- Availability -
  - minimize downtime
  - eliminate downtime

Analogy: Highway Development

- Take the road offline and detour traffic?
- Funnel traffic through one lane and patch the other lane, then switch lanes and patch the other.
- Binaries are like concrete roads. Every so often they have to be totally redone, unlike gravel which can be refreshed. Think detour.
- Package management gives the accomplished sysadmin the tools to fix the “roads”. Without hands a tool is useless.

Patch Management versus Package Management

- Patches are incremental and are either binary or text-based.
- Packages are the entire thing. They are larger and often contain common components or shared libraries or at least have pointers to other packages for these.

Text Patching

```
diff -c orig/document document > diff-set
```

```
patch < diff-set
```

diff-set is a human readable file.
Use diff -i for case insensitive
-b and -w ignore white space
-c gives context output
-r will recurse all files in directory
Diff can be used with directories as well as files.

More on Diff

- bdiff provides a binary safe method and produces non-human-readable binary diff patch files

- W3C has a specification document for generic diff files, rsync can use gdiff:
  http://www.w3.org/TR/NOTE-gdiff-19970901

Challenges of Package Management Tool Comparison

- Tendency toward complexity.
  - Integral to the OS
  - Some can manage themselves
- Different languages.
- Tendency toward OS dependence.

Major Unix Package Management Tools

- make-based FreeBSD ports system
- Perl-based apt-get
- RedHat Package Manager (+SuSE,Mandrake)
- Darwinports in TCL
- XML Metapkg
- Language-specific tools: CPAN

Microsoft versus Unix/GNU

- Access control approach affects software management approach.

- Permissions-based file systems allow for software that can update itself.

- Should an update system have its own access control or share the access control of the operating system?

The Package Management Rosetta Stone -1

	list packages	install	update
BSD pkg	`pkg_info`	`pkg_add`	`pkg_add`
gentoo portage	`emerge -s`	`emerge [world]`	`emerge sync`

dpkg / apt-get	`apt-get list;`
	`cat /etc/sources.list;`
	`apt-get show package`
		`apt-get install`	`apt-get update ;`
			`apt-get dist-upgrade`

rpm	`rpm -qa`	`rpm -i`	`rpm -uvh`
darwinports	`port list`	`port install`	`port install`

FreeBSD ports	`cd /usr/ports;`
	`ls -R`
		`make install`	`make [un]install`

Solaris	`pkg-get describe`	`pkg-get install`	`pkg-get install`

The Package Management Rosetta Stone -2

	create	remove	help
BSD pkg	`edit make file`	`pkg_delete`	`man ports`
gentoo portage	`edit ebuild scripts`	`emerge unmerge`	`man emerge`

dpkg / apt-get	`debian-binary`
	`control.tar.gz`
	`data.tar.gz`
		`apt-get remove`	`apt-get help`

rpm	`edit spec file, use setup macro`	`rpm -evh --force --nodeps`	`rpm --help`
darwinports	`edit portfile with tcl format`	`port uninstall`	`port -h`
FreeBSD ports	`write make file`	`make uninstall`	`man ports`
Solaris pkg	`pkgmk -o -r / -d /tmp -f Prototype`	`pkgrm`	`pkg-add --help`

The Package Management Rosetta Stone -3

	More Information	#
BSD pkg	man page for ports http://www.openbsd.org/faq/faq8.html	about a thousand
gentoo portage	www.setuplinux.com www.gentoo.org	similar to FreeBSD ports
dpkg / apt-get	http://www.debian.org/doc/manuals/apt-howto/index.en.html http://xtronics.com/reference/rpm2apt-dpkg.htm	2,000 to 6,000
rpm	Maximum RPM, third edition www.rpm.org	distro specific
darwinports	darwinports.opendarwin.org also darwinports.org O'Reilly MacDevCenter.com Usage: port [-vdqfo] [-D portdir] target [portname] [options] [variants]	1400
*BSD ports	netbsd pkgsrc page http://www.freebsd.org/ports	11,000

MacOSX

Fink project - direct port of dpkg/apt-get

Darwinports - about 1 year old, headed by Jordan Hubbard and others at Apple but is open source, uses tcl-syntax for parsingportfiles

MetaPKG - a project to unify various efforts towards package managers on Mac OS X

RPM for MacOSX - works but is not being used widely

pax and the Apple Installer utility - there are tools to automate the process of building point-and-click installers, or use shell scripts to accomplish the same

Linux

- Package management is becoming the primary distinguishing features between the major distributions.
- RedHat - rpm was simultaneous with company formation and integral to its strategy
- Mandrake and SuSE both have tools that make use of rpm (yast) and this requires complete sets of many basic libraries
- Debian - the premiere distribution for an all-Perl approach. dpkg and apt-get used to maintain the entire system and distributed mirrors.
- Gentoo is ports system based (bash scripts and make)

Q. What do these systems have in common?

- Fink project on Mac OS X
- Knoppix, a bootable Linux CD
- Debian, popular Linux distribution

A. apt-get

All of the above use apt-get.

Because apt-get is perl, it can readily follow perl to other platforms.

The *BSD ports system is essentially as portable as the make command.

Major Differences in Package Tools

- local packages mirror or remote mirror?
- centralized repository or distributed repository?
- allow to modify config files or not?
- compile from source, binaries or both?
- how are source level patches applied?
- rollback to get system back to previous state?
- how are dependencies handled?

More Package Basics

- Rollback apt-get can do both binary and source packages. apt-get -b lets you build.
- In BSD-land, pkg is for binaries, ports is for building from scratch. Make files can be used to move binaries around though.
- RPM handles source and binaries with rpms and rpm files, separate packages.
- Configuration file changes allowed? (/etc/)

Common Issues

- Modification of shared libraries breaks other apps - rpm, apt-get,ports all handle this case with package-specific prevention.
- Permissions issues: root permission is a typical requirement, if not for the package itself, for the package manager to write to its log/database.
- Availability of mirrored files - can be out of date, compromised, DoS'ed, etc.
- Do you really want to have a C compiler on a production system? How best to organize mirror of binaries? Can config files be modified?

SmartFriends® Wisdom

```
>> We can all deal with the occasional mistake.

> I find that dealing with mistakes is almost

> my entire job.

There's a lot of job security in that.
```

Automated Sysadmin?

- RedHat provides this service via up2date
- Mac OS X updates are scripted to be downloaded but not installed weekly by default.
- FreeBSD and Debian admins script the synchronization of ports and packages but not typically their installation.

Commercial Installation Tools for Win32

- **Installer VISE** from Mindvision is used heavily by Adobe. Has live update capabilities.
- **Installshield** just purchased by Macrovision claims to be robustly cross-platform.
- **Wise for Windows** installer.
- **AppDeploy** useful for scripting installs.
- **Windows Installer** from Microsoft. Serious about deployment on Win32? Get MSDN.
- Long standing need for hash or other cryptographic-level protection. Very few Win32 download sites will give PGP signatures or MD5 sums for files.

Open Source Win32 Installer Tool: InnoSetup

- no cost, uses standard wizard-style interface
- offers complete uninstall
- updates typically handled by overwriting newer version on top of older version
- creates registry and .INI entries, shortcuts
- has a Pascal scripting engine
- multilingual, can run silently
- does not require service packs (pro/con)
- written in Borland Delphi 2/5

More about InnoSetup

- ISTool helps you to create iss files
- iss files describe everything necessary to build an executable installer with InnoSetup
- Open source project GIMP on Windows has had hundreds of thousands of installs with InnoSetup installer.

Gone But Not Forgotten

- SGI had .sw files which you could launch from a browser (MIME types like .books were also recognizzed)
- pax had archive files (analog to tar)
- others?

What’s Next?

- Maintenance of current package management applications is ongoing. Observe developer activity on package project mailing lists.
- For package management, stability typically outweighs new features.
- Subscription-based business models is driving innovation, e.g., RedHat’s ES/AS.
- Integration with PKI to establish link between trusted developers and corporate or individual users.

Integration Efforts

- **Gentoo** - crosses easily between chips. Achieves one ideal of open source: normality is your compiler.
- **ipkg** - itsy package management system.
- **MetaPKG** - efforts to integrate ports, pax/bom files, dpkg/apt-get and rpms on MacOSX.
- XML seems to be a trend, use the flexibility of XML as a bandaid for maintaining packages across disparate systems.

Thanks!

Mathew Caughron, CISSP
mat@phpconsulting.com
(402) 968-1332

PHP Consulting
www.phpconsulting.com

Final Thoughts

Seen on T-shirts:

There is no patch for human stupidity.

Actually, there is. It is called *education*.

Aristotle on Trust

Friendship requires time and familiarity; as the proverb says, men cannot know each other till they have 'eaten salt together.'

Bad men do not enjoy each other’s company unless some advantage is to be had.

Nicomachean Ethics - Book 8 Chapters 3,4