CMPS 261 Server Management - Module 3: Software Maintenance

Mark Voortman, Ph.D.

Course Modules

- *Module 0: Course Design*
- Module 1: Introduction to Servers and Server Operating Systems
- Module 2: Getting Started with FreeBSD Server
- **Module 3: Software Maintenance**
- Module 4: Tuning and Configuration
- Module 5: Storage Management
- Module 6: Networking
- Module 7: Shell Scripting
- Module 8: Building a WordPress Server

Module 3

- **Software Maintenance**
- Part A
  - Keeping the operating system current
  - Finding and installing software
  - Configuring the server with the software that it needs to do its job
  - Configuring the server with the tools that you need to do your job
  - Keeping that software up to date
- Part B
  - Using desktop environments
  - Using other software

Objectives

- Upon successful completion of this module, you should be able to:
  - Find and install software
  - Keep software current
  - Install a desktop environment

PART A

FreeBSD Software Overview

- FreeBSD comes with a large set of command line utilities
  - We have been using tools like ls, cat, ee, vi
- More advanced, higher level applications must be installed on top of the operating system
  - Some may come from the BSD Ports collection
  - Some may come from outside sources

VM: Backups, Clones and Snapshots

- The VM is just a set of files on disk
  - Can be backed up as any set of files would be
  - Files can be quite large
- A clone is a complete duplicate of a VM
  - Implication for Windows VMs if you are running simultaneously
  - Implication for network MAC address if you are running simultaneously
- A snapshot is a point in time
  - Disk space impact is low
  - Allows rollback to revert to earlier version
  - Useful if making substantial, risky changes to a system

FreeBSD Software Overview

- There are two ways to install software in FreeBSD
  - Binary packages
    - Installed using the pkg tool
  - Building from source code
- In both cases, the software needed to be ported to FreeBSD

FreeBSD Software Installation I

- Find and download the software
  - Might be distributed in source code format or as a binary
- Unpack the software from its distribution format
  - This is typically a compressed file
- Locate the documentation and read up on how to install the software
  - Or, do web research
- If the software was distributed in source format, compile it
  - This may involve editing a Makefile or running a configure script
- Test and install the software

FreeBSD Software Installation II

- A FreeBSD package contains
  - Pre-compiled copies of all the commands for an application
  - Configuration files and documentation
  - A package can be manipulated with the pkg commands, such as pkg install
- A FreeBSD port is a collection of files designed to automate the process of compiling an application from source code
  - The files that comprise a port contain all the necessary information to automatically download, extract, patch, compile, and install the application

FreeBSD Software Installation III

- Package Benefits
  - Download will be smaller
  - No compile time, which can be long for large applications like Mozilla
  - Packages do not require any understanding of the process involved in compiling software on FreeBSD

FreeBSD Software Installation IV

- Port Benefits
  - Compilation options can be changed to tailor to your needs
  - Some applications have compile-time options relating to which features are installed
  - In some cases, multiple packages will exist for the same application to specify certain settings
  - The licensing conditions of some software forbid binary distribution
  - Some people do not trust binary distributions
  - Source code is needed in order to apply custom patches

FreeBSD Software Vulnerabilities

- Best practices would suggest doing research to confirm stability and security of target applications
- See http://vuxml.freebsd.org/freebsd/index.html for a compilation of known software vulnerabilities
- See also CERT site for vulnerabilities.
- Homepage | CISA

Finding FreeBSD Software I

- For FreeBSD, this isn’t as easy as going to an app store or using a GUI package manager
- You will need to track down software, and know a bit about it, to get it

Finding FreeBSD Software II

- The FreeBSD web site itself
  - http://www.FreeBSD.org/ports/
  - The ports can be searched by application name or by software category
- FreshPorts.org
  - http://www.freshports.org/
  - Provides a search utility and also tracks changes to the applications
- Locations such as SourceForge.net or GitHub.com
  - Note that you will then need to find out if there is a port to FreeBSD

Finding FreeBSD Software III

- Search the binary package repository
  - The pkg command allows searches
  - Requires that the package management system be installed

Pkg Search I

```
$ pkg search subversion
git-subversion-1.9.2
java-subversion-1.8.8_2
p5-subversion-1.8.8_2
py27-hgsubversion-1.6
py27-subversion-1.8.8_2
ruby-subversion-1.8.8_2
subversion-1.8.8_2
subversion-book-4515
subversion-static-1.8.8_2
subversion16-1.6.23_4
subversion17-1.7.16_2
```

Search packages

Pkg Search II

```
$ pkg search -o subversion
devel/git-subversion
java/java-subversion
devel/p5-subversion
devel/py-hgsubversion
devel/py-subversion
devel/ruby-subversion
devel/subversion16
devel/subversion17
devel/subversion
devel/subversion-book
devel/subversion-static
```

Search ports

Finding FreeBSD Software

- The Ports Collection is installed on our systems, so it can be queried
- The whereis command, using a name
  - `$ whereis lsof`
  - `lsof: /usr/ports/sysutils/lsof`
- Alternately, an echo statement can be used
  - `$ echo /usr/ports/*/*lsof*`
  - `/usr/ports/sysutils/lsof`

Finding FreeBSD Software

- The Ports Collection's built-in search mechanism
  - To use the search feature, cd to /usr/ports then run make search name=program-name
  - Requires that an index tool be installed

Ports Search

```
$ cd /usr/ports
$ make search name=lsof
Port:   lsof-4.88.d,8
Path:   /usr/ports/sysutils/lsof
Info:   Lists information about open files (similar to fstat(1))
Maint:  ler@lerctr.org
Index:  sysutils
B-deps:
R-deps:
```

Ports search

Performing Installs

- We will now use both the package system and the ports collection to install software

Keeping Current

- FreeBSD, like all operating systems, needs to be patched
  - Fixes for bugs and security flaws
  - Performance enhancements
  - New versions
- Handbook reference
- Security advisories

Updating FreeBSD

- Process is usually painless.
  - We first fetch changes, and then install them
  - If no changes are detected, this will be reported
- `freebsd-update fetch`
- `freebsd-update install`

Updating FreeBSD

- You can automate the fetch process
  - Place daily cron entry into crontab file
  - Email will be sent to root user about available updates

Rolling Back FreeBSD updates

- If any issues arise, you can perform rollback
  - `freebsd-update rollback`

Minor Version Upgrades

- When a new minor version is released, run the following statement
  - Statement shows a hypothetical 12.9 release, which does not exist at this time
  - `freebsd-update -r 12.9-RELEASE upgrade`

Major Version Upgrades

- When a new major version is released, similar statement would be used
  - Statement shows a hypothetical 13.0 release, which does not exist at this time
  - More work is involved with a major upgrade because packages may no longer work
  - These statements will perform the upgrade, then force an update of installed packages
  - `freebsd-update -r 13.0-RELEASE upgrade`
  - `pkg-static upgrade –f`
  - `portmaster -af`

Updating Installed Packages

- pkg statement upgrade option will compare installed packages against latest version, and update
- `pkg upgrade`

Auditing Installed Packages

- pkg statement audit option will examine security status of installed packages
- `pkg audit -F`

Cleaning Installed Packages

- pkg statement autoremove option will delete orphan dependencies
- `pkg autoremove`

Software Update Summary

- FreeBSD provides easy to use tools to automatically keep system up to date
- Burden is still on system administrator to know what they are doing
  - Updates can have negative consequences

Exercise

- Building packages with Poudriere

PART B

Desktop Managers

- We will go examine the process of installing a GUI desktop manager
- Won’t perform this for two reasons:
  - It is complex
  - The value of a GUI for a server is open to debate
- Handbook Reference
- If you want to pursue, recommend that you work on a clone or a new instance of FreeBSD

Desktop Managers

- Step 1:
  - Install Xorg
    - The open source X Window System.
    - Provides base for graphical environment.
  - Perform install:
    - `pkg install xorg`
  - Add users who will run Xorg to the video group:
    - `pw groupmod video –m {users}`
  - Start Xorg:
    - `startx`
      - Starts a crude GUI.
      - Run `killall Xorg` in one of the terminals to exit
- See Handbook – Installing Xorg

Desktop Managers

- Step 2:
  - Install Desktop manager
  - Three choices:
    - GNOME
    - KDE
    - Xfce
  - Run `pkg install xfce` as root
  - Start Xfce as a regular user by running `startxfce4`
- See Handbook – Desktop Environments

Install Firefox

- `pkg install firefox`
- (Or `pkg install epiphany`)
- After installing you should be able to start Firefox from the Xfce menu

Install Nginx

- Follow the homepage setup tutorial
- Make adjustments wherever necessary
- Serve a webpage and open it in firefox
- Make sure you create the `www` directory in the home directory of a regular user

Install a Graphical Editor

- VS Code?
- Emacs?
- Other?
- Make a small update to your website using your new editor

Other Software

- Install a new programming language and write a little program
- Install LibreOffice by running `pkg install libreoffice`
- Suggestions?

THANK YOU!