OS Tools for OpenBSD

Overview Presentation

Team Fugu

CS 486 - Capstone

Team Fugu

- Ben Atkin
- Thad Boyd
- Nauman Qureshi
- Erik Wilson

Fugu: A poisonous blowfish.
The blowfish is the OpenBSD mascot.

Team Roles

- Ben Atkin
  - Communicator
  - Researcher
- Thad Boyd
  - Facilitator
  - Webmaster

- Nauman Qureshi
  - Recorder
  - Documenter
- Erik Wilson
  - Team Leader
  - Organizer

Client

- US Geological Survey (USGS)
  - Astrogeology Team
  - Map Landscape of Planets
    - Custom software for image processing
    - Using high-end UNIX workstations
  - Information Technology Division
    - Multiple Servers (Mail, FTP, Web)
    - Multiple Architectures (x86, Sparc)

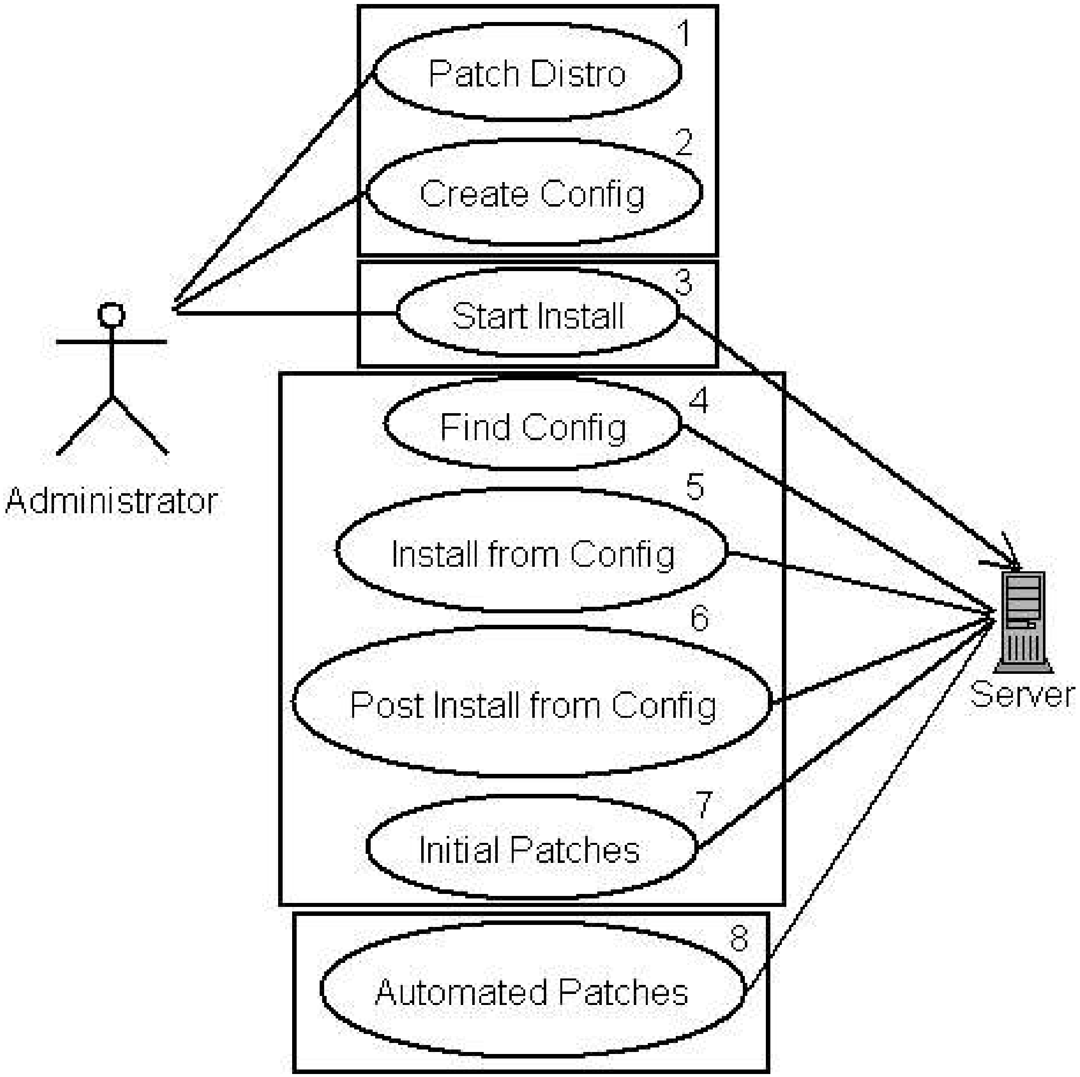
Problem

- Time-consuming to install OpenBSD on many systems
  - Interactive portion to installation required
  - Partitioning of disks is cumbersome
- Patches for OpenBSD require manual installation on each system
  - Administrator must check frequently for patches
  - Uninstall for patches is difficult
- **20 machines x (1 hour install + 1 hour patches) = *40 hours total***

Needs

- Need for OpenBSD Auto-Installer
  - Need a non-interactive system
  - *Similar Products:* Solaris Jumpstart, Redhat Kickstart
- Need for OpenBSD Auto-Patcher
  - Auto download and install of patches
  - Ability to “roll back” or uninstall patches
  - *Similar Product: Tepatche*
- *Both must run on i386, Sparc64 platforms*

Use Case Diagram



Nauman Qureshi

Installer Requirements

- Must be future-version compatible
- Must handle install *or* upgrade
- Install configuration file must be read from:
  - CDROM / Floppy
  - FTP / HTTP
  - Local hard drive
- Must handle partitioning of disks
- Must seek out and back up important files (eg SSH keys)

Patcher Requirements

- Must handle source or pre-compiled patches
- Must track what patches have been installed, and what patches have failed to be installed

Resources

- OpenBSD installer
  - Open-source (Revised BSD License) installation system
  - Based on shell scripting
  - Is not currently automated
- Tepatche
  - Open-source (Revised BSD License) patching system
  - Based on Perl scripting
  - Does not currently handle binary patches

Revised BSD License

- Code may be reused, rewritten, or redistributed in original or compiled form with or without charge
- Authors' names may not be used to endorse derivative works without express written consent
- Standard legal disclaimer
- License must be included with all source or binaries of software

Architecture - Installer

- Derived from existing shell scripts
- Performs similarly to existing installer
  - Clean install or upgrade
  - Partitions disk
- Adds functionality
  - Reads from configuration file
  - Backs up required files as needed

Architecture - Patcher

- Derived from existing Perl scripts
- Performs similarly to Tepatche
  - Will patch from source
  - Can remove patches after installation
  - Runs on a schedule
- Adds functionality: Will patch from binaries

Schedule: Requirements

- 02.18 (today!): Milestone 1.
  - Requirements completed
  - Functional specifications approaching completion
- 02.25: Final design presentation within team
- 02.28: Final specification draft submitted to sponsor

Schedule: Implementation

- Week of 02.23: All team members have running versions of OpenBSD and Tepatche.
- 03.01: Team coding standards determined. Begin preliminary coding.
- Week of 03.01: Design review
- 03.05: Software architecture defined
  - Individual coding assignments determined
  - Coding begins in full force

Thad Boyd

Schedule: Implementation, cont.

- 03.23: Milestone 2, implementation 50% complete
- 04.13: Design review
- 04.17: Testing plan
  - Code functional, integrated
  - Test, work out bugs
- 04.28: Capstone presentation prepared
- 04.30: Capstone presentation

Schedule: Final Analysis

- 05.03: Notebook, website finalized
- 05.05: Final report

Risks: Critical Severity

- Tools do not perform to standards
  - Low probability
  - Avoid through thorough testing
- Interface unusable
  - Low probability
  - Avoid through constant testing, communication with sponsor

Risks: High Severity

- Cannot back up existing files
  - Medium-low probability
  - Avoid through early planning, communication with sponsor
- Programs not fully automated
  - Low probability
  - Avoid by coding to read from configuration file

Summary and Progress

- Requirements understood
- Examining existing software
  - Test machines acquired
  - Have installed and run OpenBSD and Tepatche
- Early planning for software design
- Good communication in team, with sponsor
- *Enthusiastic*