

Introduction

LISA 2001

Cfengine Workshop

Dynamic Cfengine

Christian Pearce

CommNav, Inc. <www.commnnav.com>

Introduction

- Christian Pearce
- Senior Software Engineer
- CommNav, Inc.
- Portal Framework Software

Goals

- Why discuss dynamic cfengine
- User Interface (Front End)
- Data storage (LDAP, Database)
- Other systems can make use of the centralized Data
- Templates for building configuration files both cfengine and application
- Scripting Language for gluing Data store to Templates
- Automated backend
- Security
- the building of an extensible framework

Portal

- Background on why cfengine
- Portal Framework has basic requirements
- Automatic deployment of third party apps (installs/upgrades)
- Automatic configuration for third party app configurations
- LDAP (DEN, DHCP, NIS, NFS, DNS)

Front End

- The first part to building a dynamic cfengine system is gather dynamic information
- Create a user interface (Andy Mayhew?)
- What information should be gathered?
hostname, ip, description, OS type, etc...
- Gather the information on how to configure the host

Data Storage - LDAP

- Storing information in LDAP
- An entry for each host
- Configuration information for apps stored as attributes
- This information could be stored anywhere LDAP or a Database

Front End - LDAP host entry example

```
dn: cn=abe.commnav.com, ou=hosts, o=commnav.com
owner: uid=administrator, ou=people, o=commnav.com
objectclass: top
objectclass: cnManageHost
cn: abe.commnav.com
ipHostNumber: 192.168.0.1
description: CommNav Development Server
cnManagedComponent: webalizer
cnManagedComponent: rpmcheck
cnPreference: webalizer.hosts....
```

Data Storage - LDAP

- Having a entry in LDAP for each host allows for cfengine scripts to be built dynamically.

Other uses for Data

- Storing the data in a central store gives the SysAdmin's the ability to build other systems around it.
- Existing data stores could make the process of creating a dynamic cfengine environment easy.

Dynamic Configuration Generation

- Create cfengine templates
- Templates contain code to create cfengine code
- Using a scripting language to parse the template files
- These dynamically generated scripts are then made available to the cfengine daemon.

Dynamic Scripting

- Scripting produces Dynamic cfengine code
- The PHP code reads the entries from LDAP
- Each entry is examined to see how cfengine code should be produced

Example of PHP code

control:

```
# Dynamic Host Definition for Webalizer
<? foreach ($hosts as $k => $v) {
    $p = $v->preference();
    foreach ($p as $po) {
        if ($po->appName == 'webalizer.hosts') {
            $j = str_replace('.', '-', $k);
            echo "$j::\n";
            echo "\twebalizer_hosts = ( \";
            $po_a = array_keys($po->value()); ...
```

Extending Script Glue

- Use the same Scripting system to build application configuration from templates
- Store application preferences in LDAP
- Building templates for application configuration extends the abilities of systems that build straight configs without templates
- Prevents tying a Sys Admin to the Front End

Backend

- Techniques for the backend
- Building an action for each host gives fine grunality for control
- Generate arrays for list interation
- Store class definitions in host entry
- Load scripts base on defined classes

Security

- Autogenerating the cfd.conf for complete host based security
- Comments?

The Framework

- Frontend gathers the data
- A set of templates generate cfengine code
- A script parse the templates
- The cfengine daemon makes the built templates available
- The cfengine cron runs and grabs the latest configurations
- The new policy is executed

Random Thoughts

- Create Perl Modules for generating dynamic cfengine
- Code snippets would be discrete functions
- Community would benefit from reuse of code and peer review (CCAN)
- Create a cfengine library for integration into other application
- Create an autoconf style system for generating applications
- LDAP enable cfengine?

Open Discussion

- Security
- Editable Templates (granularity)
- Dynamic Host information
- Leveraging LDAP for NIS, DNS, DHCP creating a DEN
- Comprehensive Cfengine Archive Network
- Better List Iteration
- Variables for host
- Dyanimic cfengine scripts add to complexity
- Random Thoughts

Thank You

Christian Pearce

pearcec@commnav.com

Comment, and Questions welcomed