Global Information Assurance Certification Paper

GSEC Practical Assignment V1.4b, Option 1

Configuration Management in a Heterogeneous Environment

Submitted By:

Adam B Jurevicius

04 January 2004

ABSTRACT **3**

THE DEFINITION **3**

THE TOOLS **4**

Common threads *4*

LOOKING INSIDE CONFIGURATION MANAGEMENT SYSTEMS **4**

The many faces of CMS delivery agents *4*
Commercial delivery agent tools *4*
Reasons people use Open Source *6*
Reasons people do not use Open Source *7*
Positive CMS concept examples *7*
Negative (or lack of) CMS examples *8*
APPLYING VERSION CONTROL TO YOUR DATA & TOOLS (7, 3, 18) 8
Commercial versioning tools *8*
Open versioning tools *8*
When version control is absent *9*
STORING AND RETRIEVING YOUR DATA WITH BACKUPS 9
Commercial products used for backups *9*
Open tools used for backups *10*

CREATING A CMS FOR WINDOWS AND LINUX **10**

Reacting in an ever changing environment *19*

CONFIGURATION MANAGEMENT CHALLENGES **20**

Why do systems and projects fail? *20*
Can outages be tied to mis-configuration? *20*
ANY ONE SYSTEM IS ONLY AS STRONG AS ITS WEAKEST LINK 20
Alerting & Auditing change *20*
Open Monitoring Console systems *21*
Commercial Monitoring Console systems *21*
Open Syslog Tools *21*

WHY CMS TOOLS ARE NECESSARY **22**

THE FUTURE **22**

BIBLIOGRAPHY **22**

Abstract

The problem of maintaining and implementing a Configuration Management System (CMS) has long tortured system administrators (Sysadmins). It is an issue with more than one solution for any given company and when not done correctly, can introduce an array of unexpected results and failures to any environment. Controlling the primary configuration of any system from a central source is something that many software vendors have attempted and something that has become even more critical today.

This paper will define some of the CMS tools available for Windows & UNIX. Throughout the paper I will break down those tools into two groups, commercial and Open Source. Hopefully, each product's detailed description will help you define your CMS requirements and decide which tool to implement. By defining the functions within a CMS we can take a deeper look into the tools themselves, why people choose the products they do and how to configure your own CMS. Looking at a CMS from the Sysadmin perspective will allow us to define the different CMS methods available, how these tools are audited and finally, I'll take a look at the future of standards in our industry.

The Definition

Murphy's law: "Left to themselves, things tend to go from bad to worse" (6)

"The purpose of Configuration Management is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits."(17)

CMS tools are generally composed of five functions. The order of importance of these functions can be changed and layers may be added but to have any success with this challenging problem we must have at least these five things.(19, 15, 4)

1. The delivery agent (A tool or toolset that will deliver your files to the clients that request them)
2. The version control system (A system that will track changes to your files)
3. The backup system (A system or process that will archive everything from the version control and golden systems)
4. The golden server (A central place to store and manage your files)
5. The auditing system (A process or system that can verify your changes)

Later I will give a very basic instruction on installation of the above products. It should be enough to get any small to medium sized enterprise started with an Open Source CMS.

The Tools

There are two basic categories of CMS tools, which are Open Source and commercial also called open and not open products. While commercial tools have some advantages over Open Source, I'll be focusing on the open tools because of their price, extendibility and portability. With Linux becoming an enterprise operating system, the Open Source market has come alive with depots for open technology. Two major Open Source depots are Sourceforge (http://www.sf.net) and FreshMeat (http://freshmeat.net).

Common threads

Both categories of tools have common characteristics. Some are web based and some have a command line interface that allows easy access. It seems that every company requires a different set of tools but the main goals of stability, cost and security are the same. Most needs are being addressed by the commercial software space but there's one primary need that keeps users of commercial products wanting more and that is extendibility.

Because the source code is not open with commercial products, it does not allow an organization to extend that product to fill any special need it may have, leaving a gap in functionality. This situation can cause each company to end up with many different tools, from possibly different vendors, that will most likely perform overlapping functions and exist in different parts of the business. When ownership of these tools is spread throughout and organization, communication and management of the tools becomes complex and difficult.

Looking inside Configuration Management Systems

The many faces of CMS delivery agents

It seems that some form of CMS has been around since the birth of the computer age. Below is a short list of products available in the commercial and Open Source space which perform some function of a CMS solution. The last product listed under each Open Source tool section is what I feel offers the most complete solution.

Some of the tools listed are not strictly used for configuration management. They attack specific problems within the category itself. ISconf, for example, allows for the recording of a system configuration change and the ability to play it back in bulk while Cfengine allows for secure and controlled configuration distribution from server to client.

Commercial delivery agent tools (1)

- Tivoli http://www-306.ibm.com/software/tivoli/

Tivoli is a massive product that can control almost every type of system available. The configuration management features are tied directly to a tiered database structure that has advanced system management tools and reporting capabilities.

- Bmc Patrol
  http://www.bmc.com/products/products_services_detail/0,,0_0_0_27,00.html

Patrol is a direct competitor of Tivoli in the systems management space. They aim to provide not only configuration management but monitoring as well. The product is built for large organizations and requires a substantial financial investment to get started.

- Opsware http://www.opsware.com/

One of the newest commercial systems management tools, Opsware not only tries to tackle the remote script execution & patch management aspect of computer management but they also try to ease the installation process of UNIX and Windows systems.

- Microsoft
  - GPO (Group Policy Objects)
    - The tool allows Sysadmins to control almost every aspect of the operating system including its security and installed package list.
  - SUS (Server Update System)
    - The tool allows for a localized (on site) server that stores and distributes Microsoft software patches to domain members.

Open delivery agent tools (1)

- RDist http://www.magnicomp.com/rdist/

As part of the remote administration (unencrypted r tools) suite of products, its use has fallen off in recent years. Its primary focus is to easily replicate identical files to multiple systems. It does not allow for client site modifications or site specific customizations.

- ISconf http://www.isconf.org/

This tool is mostly used for recording and replaying the installation and removal of tools. It allows for the control of the many actions taken during a package installation and allows for the controlled reaction to questions that might need manual intervention. Its webpage describes it as, "… a framework for recording and playing back all Sysadmin work done to a network of UNIX machines." (21)

- CVSup http://www.cvsup.org/

CVSup is similar in functionality to RDist but is directly tied to a CVS repository. By accessing the data source directly the tool can update only changed sections of the files, increasing the speed of the overall process and labeling changed areas of the update.

- Red Hat Network (RHN) http://www.redhat.com/software/rhn/management/table/

The Red Hat Network represents an enterprise class package management tool. It's a subscription-based service that allows for the management of individual and groups of systems with multiple levels of security and running separate operating system versions.

- Cfengine http://www.iu.hio.no/Cfengine/

Cfengine allows the user to perform configuration file and binary package management on almost every operating system. It's a client server based application that allows the user to create a series of groups that a system or group of systems can belong to. My research found Cfengine's versatility to provide the best overall coverage for the delivery agent portion of our CMS. It solves many common problems for any Sysadmin. Here are a few:

1. Allows for secure file transfer
2. The ability to communicate with a client while being authenticated
    a. Ensuring time services are in sync
    b. Name services are correct
    c. Client/Server daemon architecture allows Sysadmins to disallow root access via the terminal and still be able to update systems.
3. The ability to write wrapper scripts to recover from a potential administrator error
4. The ability to archive a copy of changed data for rollback purposes.
5. The ability to cleanup after itself
6. It's free and actively developed

Reasons people use Open Source

The computing community as a whole is growing and evolving. Companies want to pay for stable products that fill a business need or replace an existing function with faster, better and cheaper products. They want a product that will help them innovate in their market place. The computing community has found its voice in Open Source. Independent research shows there are 4 main reasons why companies select Open Source products. (10)

1. Reliability
2. Scalability
3. Lower cost
4. Security

The popularity of Open Source is often highlighted in many articles, from technical magazines to websites. Here is what a few of these publications have to say about Open Source products and their place in the market. "The survey published by IBM claims that one in four companies with between 100 and 1,000 employees is currently using Linux, and half of them are expecting it to become their core operating system." (8) A second quote states, "In a November 2002 CIO survey of 375 information executives, 54 percent said that within five years Open Source would be their dominant server platform."(13)

Any organization that has concerns about license agreements or the security of their products can find their concerns addressed by Open products as well. One article that addresses industry concerns about the state of product security says, "A survey of 100 CIOs, (75 in the US and 25 in Europe) found that 58 percent were looking at Open

Source because of its better record on security." (14) This leads me to believe that the Open Source coding method of, "write once, review many times", meaning write the code and let the computing community review it, seems to be addressing the security concerns of decision makers.

Product deployment times and cost benefits are addressed in another article. "The ability of users to deploy the software without having to sign licenses, or make financial cases to your management, aids initial take up". (12)

In the end, the decision to implement Open Source products in any organization depends on the long term goals (both monetary and project based), risk levels, and technical support staff availability of the organization. In an article about Content Management Systems, the writer weighs the open vs. commercial question by building a, "Build vs. buy checklist/scorecard".(11) In the conclusion of this article the author states, "Deciding which way to go on your CMS deployment depends on a number of factors. But ultimately, you want the best ROI possible on the deployment. It really comes down to your requirements, your resources, and the demands of your particular situation." (11)

Reasons people do not use Open Source

In an article that defines pros and cons for Open Source products, the following list of cons is defined: (9)

1. The wild west of operating environments
2. Steep admin learning curve
3. Lack of support available
4. Documentation is overly technical

"Reasons for not deploying Linux centered around concerns regarding application availability and service and support". (10)

This seems to be a much smaller group each day. What was once a reason not to adopt an "emerging" technology has now become what seems more like an excuse to not engage the computing community and contribute to its overall development.

Positive CMS concept examples

- Immunology

The author of this concept describes it as, "…the need for a new paradigm leading to the construction of a bona fide computer immune system. With an immune system, a computer could detect problem conditions and mobilize resources to deal with them automatically, letting the machine do the work." (16) It's a goal that every system architect strives for and that every configuration management tool seems to briefly touch on. The idea of immunology is brilliant because of its simplicity; however its implementation is quite difficult. Immunology, despite its problems, is my first choice for a CMS concept.

- Reboot and configuration sync schedules.

Although I can find no references to this being an official procedure, many Sysadmins I know setup a weekly reboot schedule that includes syncing systems manually with home grown tools & rebooting the systems. This technique is time consuming and can quickly eat into any scheduled downtime numbers your organization might have, but it does tend to flush out changes that were implemented without following the proper procedure and application problems that result from memory leaks.

Negative (or lack of) CMS examples

- Hey Joe, did you make any changes to Production?

Without any process for tracking and change notification, your organization is completely tied to the memory of all involved parties. Trying to replay an outage or define a timeline of changes to a system becomes an impossible task.

- If it ain't broke.…don't fix it.

One common reason that systems are not updated comes from a leader or organization being adverse to change. This methodology will contribute to an overall insecure and technically inconsistent environment. As product lines become unsupported and newer products become more robust, it's in every organization's best interest to look into emerging products to see where things can be consolidated or updated to save money and time.

Applying Version Control to your data & tools (7, 3, 18)

Commercial versioning tools

-Visual SourceSafe (VSS) - http://msdn.microsoft.com/ssafe/

VSS performs software development version control functions and provides a version control toolset found in most other tools available. It operates with a database backend and a client interface is available for Microsoft desktops. The Open Source community has written plug-ins for VSS that allow code compilation products like Apache Ant (http://ant.apache.org) to pull files directly from the VSS SourceSafe Database. There are also ways to access the data from the command line from a Linux server with a Windows emulator such as Wine (http://www.winehq.com/). Here's one basic tutorial on how to use Wine to retrieve your VSS files.(http://www.kegel.com/Linux/vss-howto.html)

Open versioning tools

- Revision Control Systems (RCS) http://www.gnu.org/software/rcs/rcs.html

"…(RCS) manages multiple revisions of files. RCS automates the storing, retrieval, logging, identification, and merging of revisions." (7) It is the base of many other version control systems and is currently the most widely used system of its kind in the computing community.

-Subversion http://subversion.tigris.org/

Subversion is a product still in its beta release phase. Daily subversion snapshots of the dev tree can be downloaded but there seems to be no firm release date for 2004. "Subversion is positioned as "CVS done right", with the known design problems fully addressed, and in 2003 probably has the best near-term prospect of replacing *CVS*."(7) This project will be a huge jump for open versioning systems.

- Concurrent Version System (CVS) http://www.cvshome.org/
"CVS began life as a front end to *RCS* developed in the early 1990s, but the model of version control it uses was different enough that it immediately qualified as a new design. Modern implementations don't rely on *RCS*."(7) CVS is currently my recommended solution for a version control system. It contains almost every feature an organization could need for a CMS system including Windows support, remote file repository and available web tools. It's my choice because it's easy to use and setup.

When version control is absent

A company without version control is something that consists of:
a. Several Sysadmins editing lines of existing code and saving old code in a commented form as a backup.

Ex:
rsync –aop /tmp/foo.sh /home/user/do.sh
#scp –rp /tmp/foo.sh /home/user/do.sh
#cp –rp /tmp/foo.sh /home/user/do.sh

b. Different Sysadmins will copy the file to .bak or .save and edit the original file.

"The most primitive (but still very common) method is all hand-hacking. You snapshot the project periodically by manually copying everything in it to a backup. You include history comments in source files. You make verbal or email arrangements with other developers to keep their hands off certain files while you hack them."(7)

As you can see, not using a versioning system increases roll back complexity and decreases code cleanliness. The amount of time spent troubleshooting hacked together scripts during an outage is far greater than building a change management process for configuration files and sticking to it.

Storing and retrieving your data with backups (2)

Commercial products used for backups

-Veritas NetBackup
http://veritas.com/products/category/ProductDetail.jhtml?productId=nbux
Netbackup is a widely used commercial product that prides itself on ease of use. It supports most any tape device and runs on most UNIX or Windows based platforms. Its

reporting functions and extendable command line tools make it a great choice for any sized organization.

Open tools used for backups

-BackupPc http://backuppc.sourceforge.net/index.html
BackupPc is what seems to be the next generation of Open Source backup tools. With a web based management tool that includes an easy to use job status window with extended log viewing, it seems to be picking up were others left off. Used mostly for backups to a disk based repository, this tool can utilize a cheap disk based backup server that would enable fast restores.

-Amanda http://www.amanda.org/
Amanda is a robust Open Source tool born at University of Mayland in the mid 1990's. It supports many different types of drives and can run on almost any operating system. It's completely command line driven so, the user interface is not as "friendly" as some of the other products available.

-Rsync http://samba.anu.edu.au/rsync
Rsync is really just for syncing files between hosts. It can be run with its down daemon configuration and authentication or with other tools as a transport. Using SSH, for example gives Rsync the ability to encrypt data in transit. Rsync may not provide the level of backup granularity your site might require for a CMS system. Other tools available in the commercial and Open Source market handle point in time snapshots, image archiving and providing reporting options. If your site requires such options, I encourage you to look into some of the other backup tools listed. Rsync continues to be my choice because of its simplicity and availability on different platforms.

Creating a CMS for Windows and Linux

Here's a basic template on how to implement Cfengine, CVS, & Rsync on a Linux "golden server" to create a CMS that will work for both Windows and Linux clients. I recommend using a Linux golden server because of its cost, ability to run on almost any machine available in your environment and finally, the flexibility of its standard toolset. Here are some things to keep in mind when configuring and using Cfengine.

- ⇒ It is extremely sensitive to drifts in local system time between the golden server and each client that communicates with it.
- ⇒ It is extremely Domain Name Service (DNS) dependant
- ⇒ Cfengine has a variety of predefined configuration options. Always refer to the online documentation for its definition when a configuration option is not defined in the environment
- ⇒ The –v option is used in all functions of Cfengine for verbose mode and the –n option is used for "all talk and no action" mode which will execute a dry run of any Cfengine configuration file.

1. Load a server with a flavor of Linux you feel most comfortable with. Keep these things in mind when selecting your server:
    a. Find a server with enough space to store your entire configuration tree.
    b. This will be your configuration core so you'll want stable hardware and a redundant disk configuration. You might consider a High Availability (HA) solution if your environment requires very high uptime numbers.

2. Define a configuration storage directory tree:

    Ex: (for a site with multiple versions, sites, machines & application tiers)

```
/golden
        /scripts
        /<version> {an internal version number}
                    /users
                            /<user name>
                                        /scripts {user specific scripts}
                     /general {used for hosts files or issue files}
                    /packages
                                /Linux
                                /Windows
                     /config
                              /Linux
                                    /init.d
                                     /rc.d
                              /Windows
                      /<site>
                           /config
                                     /Linux
                                     /Windows
                                /general
                                /users
                               /<machine type> {infra or content server
                                                    /config
                                                    /<machine name> {actual machine
name}
                                                                        /config
                                                                        /users
```

3. Install and configure Cfengine on your golden server.

Cfengine has two main server configuration files, cfservd.conf and cfrun.hosts. Cfservd.conf is used to define all aspects of the Cfengine server. For example, you would define what hosts can connect to this server, what the time zone is or where file repositories are located in this file.

The somewhat confusing part of the server component is that there are two Cfengine server configuration files. One is used for the "golden server" instance of cfservd and the other is used for the client instance of cfservd. Here is an example of a cfservd.conf file for the "golden server". An example of the client version can be found in the next section.

Cfservd-golden.conf:

```
control:

  AllowConnectionsFrom = ( 10.0.0 10.0.1 )
  AllMultipleConnectionsFrom = ( "10.0.0 10.0.1" )
  AllowUsers     = ( root )
  ver            = ( 1.0 )
  site           = ( nyc )
  domain         = ( foo.com )
  timezone       = ( PST )
  smtpserver     = ( mailhost )  # used by cfexecd
  sysadm         = ( sysadmin@foo.com )       # where to mail output
  goldroot       = ( "/golden/$(ver)" )
  golddir        = ( "/golden/$(site)/$(host)" )
  goldhost       = ( "update.$(site).$(domain)" )
  scripts        = ( "$(goldroot)/cfengine/scripts" )
  configs        = ( "$(goldroot)/cfengine/etc" )
  LogDirectory   = ( /var/log )
  SecureInput    = ( on )
  Syslog         = ( on )
  IfElapsed = ( 1 )
  MaxConnections = ( 20 )

  # Number of seconds to wait for a stale network event.
  TimeOut        = ( 10 )
  TrustKeysFrom  = ( "$(site).$(domain)" )

 linux::
  cfrunCommand  = ( "/usr/sbin/cfagent" )

grant:
  /golden/$(ver)/$(site)                *.foo.com
  /usr/sbin/cfagent                     *.foo.com
```

The cfrun.hosts file is used by the cfrun command. It defines a list of machine names that can be called from the "golden server" to run the cfagent command on the local machine. In the next section I define what a client cfservd.conf file should look like. Inside that file you should see a variable that allows you to define any local script you wish to act as the cfagent command. The following is an example of the crun.hosts file:

```
cfrun.hosts:
domain=nyc.foo.com #this must match the domain of the client being called.
access=root
machine_name_1
machine_name_2
```

You can invoke cfrun from the "golden server" by running the following from your command line:

```
. /.myconfig
/usr/sbin/cfrun -v machine_name_1 -- -p (this will call only the one machine)

or

. /.myconfig
/usr/sbin/cfrun -v -- -p (this will call every machine listed in the cfrun.hosts file)
```

4. Install & configure Cfengine on your client(s).

Each client needs a very basic configuration file to define certain constant values. This configuration file can contain machine, locale, and application specific values to be used in all of your system scripts and machine specific options for Cfengine. Because Cfengine allows you to use shell variables during runtime of the cfagent utility, this file gives you the most flexibility when dealing with servers with multiple applications that live on different operating systems. Here's an example of such a file called /.myconfig:

```
export MYTYPE="content"
export MYENV="dev"
export MYSITE="nyc"
export MYNUMBER="01"
export MYDOMAIN="foo.com"
export CFINPUTS=/var/cfengine/inputs
export MYINFRAVER="1.0"
export DATE=`date +%Y%m%d`
export YESTDATE=`date --date "yesterday" "+%Y%m%d"`
export ID=`id -nu`
```

Cfengine has four main client configuration files that are used to pull system configuration files and binary packages from the golden server, cf.preconf, update.conf, cfagent.conf and cfservd.conf. Keep in mind that this is a very basic configuration of Cfengine and you should refer to its full documentation for implementation details and common gotchas. If you get into trouble, subscribe to as many mailing lists as you can and reach out for assistance. Below are examples of each file.

Cf.preconf:

```
#!/bin/sh
# cf.preconf is an emergency/bootstrap file to get things going
# in case cfengine is unable to parse its config file
# If these files don't exist, you might not be able to parse cfengine.conf

if [ ! -s /etc/resolv.conf ]; then

 echo Patching basics resolv.conf file
 cat > /etc/resolv.conf << XX
domain nyc.foo.com
nameserver 10.0.0.1
XX
fi
```

Update.conf:

```
control:
   actionsequence  = ( copy tidy )  # Keep this simple and constant
   site            = ( "$(MYSITE)" )
   domain          = ( "$(MYSITE).$(MYDOMAIN)" )  # Needed for remote copy
   ver             = ( "$(MYINFRAVER)" )

 # Which host/dir is the master for configuration roll-outs?
   policyhost      = ( "golden.$(MYSITE).$(MYDOMAIN)" )
   master_cfinput  = ( "/golden/$(ver)/Cfengine/etc" )
   master_cfscripts  = ( "/golden/$(ver)/Cfengine/scripts" )
   servers         = ( "<your golden server(s)>" )

 # Some convenient variables - shouldn't need to change these
   workdir         = ( /var/Cfengine )
   cf_install_dir  = ( /usr/sbin )
   debug_dir       = ( /var/tmp )
   TrustKeysFrom = ( "<your golden server(s)>" )     # trusted hosts
   DefaultCopyType = ( binary )

copy:
    $(master_cfscripts)/$(host).config dest=/.myconfig server=$(policyhost)
tidy:
       linux||cygwin::
       $(workdir)/outputs pattern=* age=7
```

Cfagent.conf:

```
control:
   actionsequence  = ( copy tidy )  # Keep this simple and constant
   access          = ( root )
```

```
  timezone      = ( PST )
  repository    = ( /var/Cfengine/save )
  golddir       = ( "/golden/$(MYINFRAVER)" )
  goldserver    = ( "golden.$(MYSITE).$(MYDOMAIN)" )
  TrustKeysFrom  = ( "<your gold server IP>" )    # trusted hosts
  DefaultCopyType = ( binary )
  Inform        = ( on )
  debug_dir     = ( /var/tmp )
  Warnings      = ( off )
  Verbose       = ( off )
  Syslog        = ( on )
  SyslogFacility = ( LOG_DAEMON )

required:
      linux||cygwin::
      /.myconfig
copy:
      linux::
      $(golddir)/$(MYSITE)/$(MYENV)/$(host)/config/passwd dest=/etc/passwd
server=$(goldserver)
      $(golddir)/$(MYSITE)/$(MYENV)/$(host)/config/shadow dest=/etc/shadow
server=$(goldserver) encrypt=true
      $(golddir)/$(MYSITE)/$(MYENV)/$(host)/config/group dest=/etc/group
server=$(goldserver)
      $(golddir)/$(MYSITE)/$(MYENV)/$(host)/config/hosts dest=/etc/hosts
server=$(goldserver)

      cygwin::
      $(golddir)/general/win/Lmhosts
dest=$(SystemRoot)/system32/drivers/etc/Lmhosts server=$(goldserver)
tidy:
$(repository) pattern=* age=30
```

The cfservd.conf file is only needed on the clients if you wish to invoke Cfengine from a remote system. If you do not require this functionality you can skip this portion.

```
cfservd.conf:
control:

  AllowUsers    = ( root administrator )
  smtpserver    = ( mailhost )  # used by cfexecd
  sysadm        = ( sysadmin@foo.com )       # where to mail output
  LogDirectory  = ( /var/log )
  SecureInput   = ( on )
  Syslog        = ( on )
```

```
   IfElapsed = ( 1 )
   MaxConnections = ( 20 )

   # Number of seconds to wait for a stale network event.
   TimeOut       = ( 10 )
   TrustKeysFrom  = ( "<the ip of your golden server>" )
   AllowConnectionsFrom = (<the ip of your golden server>)

  linux||cygwin::
   cfrunCommand  = ( "/root/scripts/cfwrap" )

#############################################################

grant:
   /root/scripts/cfwrap <the same ip's listed in the TrustKeysFrom section>
```

The cfwrap script is useful if you wish to use environment variables to define functions within your Cfengine configuration file. If you would like to have this configuration ability you must create the cfwrap script for each client to execute. Here's an example of a /root/scripts/cfwrap file:

```
#!/bin/sh
#this script allws us to source our config file then run the cfagent command.
. /.myconfig
/usr/sbin/cfagent –v –f update.conf
. /.myconfig
/usr/sbin/cfagent –v –f cfagent.conf
```

Once the cfwarp script and the local cfservd.conf files are in place you can start the local cfservd daemon by running the following the command line (the –F option will disable the daemon option):

```
. /.myconfig
/usr/sbin/cfservd –F –v –f  /path/to/cfservd.conf
```

a. Using Cfengine on Linux clients

After installing the proper binary packages on the client system, you can start a cfengine pull sequence by sourcing the .myconfig file into your shell environment and running cfagent utility by executing the following from the command line:

```
. /.myconfig
cfagent -v
```

By running Cfengine this way, the client will execute all three client configuration files in order. First the cf.preconf then update.conf and finally

the cfagent.conf file will be executed. This gives you the ability to ensure core files are available from within your cf.preconf files, then to copy core system configuration files from the "golden server" inside your update.conf file, and finally execute more specific file and package update functions inside your cfagent.conf file.

b. Using Cfengine on Windows
    i. Install Cygwin (http://cygwin.com/) to enable a working UNIX like environment on your Windows system. You should install, SSHD, all of the user shells you prefer, Rsync and any other tools you commonly use on your Linux systems.
    ii. Install Cfengine inside the Cygwin environment. If a Cygwin package is not available for Cfengine you can download the source code and compile it for yourself. See the install documentation inside the product tarball for instructions on how to do that.
    iii. The above configuration files for Cfengine will work seamlessly with a Windows client. Within each configuration file any section with cygwin:: will be executed on a Windows client. To execute a pull session you can run the same command sequence defined in section a above.

5. Using CVS
    a. The creation of a CVS repository requires installation of the CVS server files on your golden server. If you wish to access the CVS repository from a remote machine you'll need to make it available via the pserver function by adding the following to your xinetd configuration as a file called pserver:

```
service cvspserver
{
        disable         = no
        socket_type     = stream
        wait            = no
        protocol        = tcp
        user            = root
        server          = /usr/bin/cvs
        server_args     = --allow-root=/data/cvs pserver
        env             = HOME=/data/cvs
        passenv         =
}
```

Once this file is installed recycle the xinetd daemon to activate it. In order for any remote client to communicate with and download CVS files you'll need to install the CVS package on that client and insert the following in your .bashrc file:

export CVSROOT=:pserver:<user id>@golden.<site>.foo.com:/data/cvs

b. How your golden tree interacts with your CVS repository is beyond the scope of this paper but here are some rough ideas. Your tree will have a series of scripts. There's really no way to get around it. Here are three main script ideas:
   i. Create/refresh your machine configuration. This script will create an entry for your server or group of servers. If a configuration tree for that server already exists, the script should refresh the configuration files within the tree.
   ii. Create and manage your user account files. This script will build your password, shadow & group files from templates. One way is to have a passwd.base & passwd.add file. The .base file will define users that each system must have. The .add file will define users to be added to the system, the Sysadmins for instance. The script will then pull two files together to create the main password file.
   iii. Update/pull files from your CVS repository. This script will allow you to manage each file or tree of files and their relationship to the CVS repository. This script might tie into the one defined in the first script suggestion.

6. Using Rsync to backup your data:

The beauty of the CMS defined above is that it is all flat file based. There are no proprietary databases to worry about stopping/starting or special file handling options to be considered. Rsync is a flexible command line tool that allows you to setup secure file transfers using SSH as a transport or a simple user authenticated session using its own communications daemon. A basic Rsync server setup using the Rsyncd daemon would have your backup server configured with the following rsync.conf changes:

```
[backups]
    uid     = <backup user>
    gid     = <backup user>
    path    = /data/my_backups
    comment = My Backups Depot
    use chroot = yes
    refuse options = checksum
    dont compress = *
    ignore nonreadable = yes
    read only = false
    hosts allow = <your golden server ip>
```

The golden server itself would use a command similar to the following to make sure data is, at the very least, backed up to a second location.

```
rsync -rulogp --delete --bwlimit=500 /data/cvs backup_server_name::backups
```

This setup would dump your entire CVS repository /data/csv to the backup server, overwriting the existing copy of things (if necessary) and limiting its overall bandwidth usage to 500kbs.

Reacting in an ever changing environment

With Microsoft & Linux products being consistently tested for bugs like buffer overflows and general protocol flaws, it's becoming more difficult to stay secure and consistent. Configuration management, which includes things like applying software patches, needs to be a top priority for every organization. Without a process to implement changes and track their progress, various parts of the infrastructure can be vulnerable to bugs and be left vulnerable to attack.

Many organizations and resources have been created to track bugs, potential virus out breaks and much more. Mailing lists are still the number one source of information. The BugTraq and Incidents mailing lists from security focus (http://securityfocus.com/archive) are two such lists where you can find information quickly about a new bug found in software or a discussion about new network traffic found from an undefined source.

The SANS organization has also setup weekly newsletters that bring you the top vulnerabilities and news from in and around the security industry. After signing up for a portal account on the SANS website you have access to these resources under the subscriptions section. They have also worked with the Federal Bureau of Investigation (FBI) to create a top 20 list of the most common vulnerabilities (http://www.sans.org/top20/).

Tracking vulnerabilities in software products is an essential part of assessing risk in your organization. The CERT organization has tackled this very important task and provides the most up to date information about product vulnerabilities available. There are two sections to their site. The Advisories (http://www.cert.org/advisories/) section is a listing of the top vulnerabilities found in the most commonly used software. The Vulnerability Notes Database section (http://www.kb.cert.org/vuls) is a listing of all software bugs available. By creating a ranking system, the CERT team is able to display the highest impact bugs first.

As you can see, there are many tools and early warning systems available to help every type of organization be proactive in their security response. With new government regulations in place there's an increasing focus on the protection of PII. It is an individual's private data, be it customer data or employee data, which needs to be held at the highest security levels. Applying defense in depth, using data encryption and defining and enforcing solid change control and configuration management methods are the best defense against anyone trying to gain unauthorized access.

Configuration Management challenges

Why do systems and projects fail?

The answer to this very broad question is fairly simple; communication. In a survey named "The Bull Survey" published by IT Cortex, we find that, "…the major causes of project failure during the lifecycle of the project are a breakdown in communications (57%), a lack of planning (39%), and poor quality control (35%)." (20) This seems to be a common theme for companies of all sizes and has a direct impact on the security of each company.

Can outages be tied to mis-configuration?

During my research I could find no data directly tying system outages and security breaches to configuration management. This was fairly expected due to the sensitive nature of outage information within each company. Technical credibility is a huge priority among clients and potential damaging effects of this knowledge in the marketplace prevents its release.

I do, however, believe that some percentage of security related outages are caused by or are directly related to a breakdown in configuration management systems.

Any one system is only as strong as its weakest link

Systems and configuration components rely on each other. Some systems are stand alone, like printers, and others are used in interconnecting groups of systems into communities. Configuration Management Systems are no different. The systems themselves are bundled functions that create a unified product that can save you much time and money.

System failures will occur. It's inevitable. However, the severity of the outage can be controlled or kept to a minimum by having an effective CMS in place. If the groups that manage configurations do not have, at the very least, a good change control tool, there can be widespread outages and communication disconnects. Communication is essential when system reliability needs to be above 99.995%.

Alerting & Auditing change (1)

The most common way I've seen configuration auditing done is by email. Others include Syslog and Simple Network Management Protocol (SNMP) traps. A script that wraps around any number of tools needed to perform specific functions will test the failure or success of those tools while logging centrally, by sending email or executing an SNMP trap to a central alerting console.

Using an existing alerting console is always the correct choice. The preferred method for controlling unified messaging in any organization would be to integrate any independent scripts and tools into an existing monitoring system.

Auditing change is the most difficult portion of a CMS. Auditing change is not a direct function of a common CMS system but is so important to the process I had to make mention of it here. Because a CMS is a living toolset that changes with the requirements of the business, your Quality Assurance (QA) team must be aware of new things to verify when your requirements shift. Both the operations and QA teams can use monitoring systems to verify, change and track application availability. Below is a list of commonly used monitoring systems.

Each tool has its strengths and weaknesses and during the evaluation process it is likely you will find one that performs at least 90% of the functions you require. I recommend a combination of tools to give you the best overall monitoring coverage. When both Nagios and Opennms are configured together, they provide all of the scanning, reporting and extendibility functions necessary to monitor a complete system and its applications.

Open Monitoring Console systems

-Nagios http://www.nagios.org/
-Opennms http://www.opennms.org/
-Bigbrother http://bb4.com/
-Bigsister http://bigsister.graeff.com/
-Mon http://www.kernel.org/software/mon/
-Esm http://esm.sourceforge.net/
-Pikt http://pikt.org/ (crossover tool, both monitoring and management)

Commercial Monitoring Console systems

-Sitescope http://www.mercuryinteractive.com/products/sitescope/

Open Syslog Tools

-Swatch http://swatch.sourceforge.net/
Swatch is a program that can watch log files for specific error conditions or messages. By applying a series of Regular Expressions to any log file, Swatch can extract the alerts any operations team would commonly look for and perform basic alert functions on those conditions.

-Syslog-ng http://www.balabit.com/products/syslog_ng/
Syslog-ng is a syslog tool that replaces the integrated syslog daemon of UNIX systems. Its configuration file is one that allows for the highest level of message flexibility and its biggest feature is the ability to insert all log entries into a database real time.

Why CMS tools are necessary

Sysadmins are one of many tools in the tool chest of any business. Without the proper leadership from all levels of management, configuration tools can lose focus and become a desperate toolset managed very differently by several groups within an organization. Once there is a break in consistency and tools start performing the same functions as their counterparts, the potential for error goes up rapidly.

Hopefully this paper has outlined some common issues that hinder configuration management system implementations and has given you some ideas on how to think about this particular problem. Almost every organization I've worked in has been challenged with configuration management problems. I hope I have provided more insight into the many great, cost effective open tools available.

The future

The computing community as a whole is beginning to realize the benefits of Open Source products and the value standards within the computing community. New standards for practice and implementations are being published every day. This can only mean good things for everyone that wishes to have some level of control over systems and their configurations. In my opinion the concept of immunology will be one that is continually followed as bulk-computing models become more prevalent. Below are some examples of standards and policies that can be found on the web.

Standards:
http://www.nist.gov/public_affairs/standards.htm#Information
http://www.sans.org/score/
http://www.cve.mitre.org/about/introduction.html

Policies:
http://www.sans.org/resources/policies/

Bibliography

1. Stokely Consulting. "Automated system Management Products." 1 December 2003. URL: http://www.stokely.com/UNIX.sysadm.resources/autosysmgm.backup.html (January 4, 2004)

2. Stokely Consulting. "Backup and Archival Software for Unix." 5 September 2003. URL: http://www.stokely.com/UNIX.sysadm.resources/backup.html (January 4, 2004)

3. Stokely Consulting. "Source Code Version Control / Software Configuration Management & Unix." 15 September 2003. URL: http://www.stokely.com/UNIX.sysadm.resources/vrsctrl.html (January 4, 2004)

4. Traugott, Steve., Huddleston, Joel., Traugott, Joyce Cao. "Best Practices in Automated Systems Administration and Infrastructure Architecture." URL: http://www.infrastructures.org/ (January 4, 2004)

5. Schneier, Bruce. URL: http://www.schneier.com (January 4, 2004)

6. Murphy Laws Site. "Murphy's Laws." 04 June 2003. URL: http://www.murphys-laws.com/murphy/murphy-laws.html (January 4, 2004)

7. FAQS.ORG. "Version Control Systems Chapter 15. Tools." URL: http://www.faqs.org/docs/artu/ch15s05.html (January 4, 2004)

8. Kotadia, Munir. "Half of small firms want to ditch Windows for Linux." 08 December 2003. URL: http://www.silicon.com/software/os/0,39024651,39117247,00.htm?rolling=1 (January 4, 2004)

9. Bowness, Stuart. "Issues Surrounding Linux and Implications for IT Managers." 11 December 2003. URL: http://www.osnews.com/story.php?news_id=5367&page=2 (January 4, 2004)

10. Mears, Jennifer. "Linux to Gain Ground in 2004." 12 December 2003. URL: http://www.pcworld.com/news/article/0,aid,113864,00.asp (January 4, 2004)

11. Kerner, Sean Michael. Choose between a commercial, open source, or customized CMS." 18 July 2003. URL: http://builder.com.com/5100-6374-5054863-2.html (January 4, 2004)

12. Kingpin Linux. "Why Choose Open Source Software?" URL: http://www.kingpinLinux.co.uk/OSS1.html (January 4, 2004)

13. Koch, Christopher. "Your Open Source Plan." 15 March 2003. URL: http://www.cio.com/archive/031503/opensource.html (January 4, 2004)

14. Thomson, Iain. "Security fears push users to open source." 12 May 2003. URL: http://www.vnunet.com/News/1151313 (January 4, 2004)

15. Wallace, Simon. "Configuration Management." June 2002. URL: http://www.epmbook.com/configuration.htm (January 4, 2004)

16. Burgess, Mark. "Computer immunology." 28 October 1998. URL: http://www.iu.hio.no/~mark/research/immune/AIdrift/AIdrift.html (January 4, 2004)

17. CMMI Product Team. "7.1.7 Configuration Management." 19 April 2003.URL: http://chrguibert.free.fr/cmmi/text/176411b-681.php (January 4, 2004)

18. Vepstas, Linas. “Linux Version Control & Configuration Management Tools” March 2001. URL: http://linas.org/Linux/cmvc.html (January 4, 2004)

19. CMMI Product Team. “7.1.7.1.2. Establish a Configuration Management System.” 19 April 2003. URL: http://chrguibert.free.fr/cmmi/text/176411b-684.php (January 4, 2004)

20. IT Cortex. “Failure Causes.” The Bull Survey (1998). URL: http://www.it-cortex.com/Stat_Failure_Cause.htm (January 4, 2004)

21. Traugott, Steve., Huddleston, Joel., Traugott, Joyce Cao. “ISconf.org.” URL: http://www.isconf.org/ (January 4, 2004)