**Configuration and Maintenance**

Policies, practices and procedures which are used to make changes to or to maintain the system as a whole, i.e. 'what humans do as part of the system administration process.

**Principle 32 (Policy) *A clear expression of goals and responses prepares a site for future*** *trouble and documents intent and procedure.*

**A system policy can include some or all of the following:**

*Organization: what responsibility will the organization take for its users' actions? What* responsibility will the organization take for the users' safety.

• *Users: allowing and forbidding certain types of software.*

*Network: will the network be segmented, with different access policies on different*

subnets? Firewall ? Ports?

• *Mail: limit the size of incoming and outgoing mail. Spam filtering.*

• *Printing: how many pages can be printed. Is printing of personal documents allowed.*

• *Security: physical security of hosts. Backup schedule. Who is allowed to be master of* their own hosts? Can arbitrary users mount other users' home directories or mailboxes with NFS on their private PCs (this means that they have automatic access to everyone's personal files)?

*Privacy: is encryption allowed?*

**Synchronizing Clocks**

Many security and maintenance issues depend upon clocks being synchronized

correctly.

A more reliable way of keeping clocks synchronized, which works both for Unix and for NT, is the use the NTP protocol, or Network Time Protocol. A time-server is used for this purpose.

The network time protocol daemon xntpd is used to synchronize clocks from a reliable time server.

**Executing Jobs at Regular Times**

**The Unix cron Service**

Unix has a time daemon called cr on: it's chronometer. Cron reads a configuration file called a cron tab file which contains a list of shell-commands to execute at regular time intervals.

On modern Unix-like systems, every user may create and edit a crontab file using the command

**crontab –e**

The format of a crontab file is a number of lines of the form

***minutes 0-59 hours 0-23 day 1-31 month 1-12 weekday***

***Mon-Sun Shellcommand***

An asterisk or star \* may be used as a wildcard, indicating 'any'. For example:

**# Run script every weekday morning Mon-Fri at 3 :15 am:**

**15 3 \* \* Mon-Fri /usr/local/bin/script**

**NT Schedule service**

**at** command provides a user interface for managing the queue of tasks to execute. The scheduling service is coordinated for all hosts in a domain by the domain server, so the host name on which a batch job is to run can be an argument to the scheduling command.

To schedule a new job to be executed either once or many times in the future, we write:

**at *host time command***

For example:

**at 3 : 00pm /next Friday, 13 C : \crsite\host \local \myscript**

**Tools for Automation**

Tivoli is probably the most advanced and wide-ranging product available. It is a Local Area Network (LAN) management tool based on CORBA and X/Open standards; it is a commercial product, advertised as a complete management system to aid in both the logistics of network management and an array of configuration issues.

Tivoli's strength is in its comprehensive approach to management. It relies on encrypted communications and client-server inter-relationships to provide functionality including software distribution and script execution.

Tivoli provides a variety of ways for activating scripts, rather like cfengine:

• Execute by hand when required.

• Schedule tasks with a cron-like feature.

• Execute an action (run a task on a set of hosts, copy a package out) in response to an event.

Sun's Solstice [180] system is a series of shell scripts with a graphical user interface which assists the administrator of a centralized LAN, consisting of Solaris machines, to initially configure the sharing of printers, disks and other network resources.

Host Factory [83] is a third party software system, using a database combined with a revision control system [261] which keeps master versions of files for the purpose of distribution across a LAN.

Host Factory attempts to keep track of changes in individual systems using a method of revision control

**Policy Decisions**

The policy must be digested, understood and, not least, obeyed by users and system staff alike:

Determine System Policy:

A sensible policy will allow for sufficient flexibility (fault tolerance). A rigid policy is more likely to fail.

**Sysadm team ag***reement: the team of system administrators need to work together,* not against one another. That means that everyone must agree on the policy and enforce it.

• ***Expect the worst****: be prepared for system failure and for rules to be broken.*

• ***Educate users*** *in good and bad practice, ignorance is our worst enemy.*

• ***Special users****: do some users require special attention, extra resources, or special assistance?*

**Diagnostic Principle:**

**Principle (Diagnostics) *When you hear the sound of distant hooves, think horses not*** *zebras, i.e. always eliminate the obvious first.*

**Establishing Cause and Effect**

Our task is to determine the source of that change, and identify a chain of events which resulted in the unfortunate effect

Once a cause has been found, a cure can be simple, but finding the problem itself often requires experience, a large knowledge base and an active imagination.

There is a three stage process:

• Gather evidence from users and from other tests.

• Make an informed guess as to probable cause.

• Try to reproduce (or perhaps just fix) the error.