INASP: Effective Network Management Workshops

Unit 4: Network Management

About these workshops

Authors:

- Dick Elleray, AfriConnect
    - delleray@africonnect.com
- Chris Wilson, Aptivate
    - chris + inaspbmo2013@aptivate.org

Date: 2013-04-29

Objectives

On completion of this session, we hope you will know about:

- Good practice in network management
- Identifying and solving problems
- Detecting and predicting problems
- Preparing for disasters
- The role of policy in guiding/changing behaviour

What is network management?

ISO network management model:

- Faults
- Configurations
- Performance
- Security
- Accounting

See Cisco's Network Management System: Best Practices White Paper.

What is Fault Management?

ISO says:

> Detect, isolate, notify, and correct faults encountered in the network.

Cisco says:

> The goal of fault management is to detect, log, notify users of, and (to the extent possible) automatically fix network problems to keep the network running effectively. Because faults can cause downtime or unacceptable network degradation, fault management is perhaps the **most widely implemented** of the ISO network management elements.

Why is it the *most widely implemented*? Because the first task of a network administrator is to *keep the network running*.

Some network admins never get beyond that part. Too busy fighting fires. Our aim is to get beyond there, but how?

Reduce downtime by reducing, eliminating and preventing faults.

Why Fault Management?

- The first task of a network administrator is to *keep the network running*.
- Effectiveness measured by *uptime* and *downtime* of services.
- Faults cause immediate, severe problems for users (and hence for us).

Feel free to challenge these assumptions!

If they are correct, then how do we maximise our effectiveness at this task, so that we reduce our stress, and have more time for better network management by implementing the other processes identified by the ISO?

Better Fault Management

How can we manage faults best, and prevent them from becoming disasters?

- Reduce incidences of faults
- Reduce effects of faults * faster diagnosis * redundant systems
- Detect and correct faults quickly (before users do)
- Keep users informed

It's important to avoid firefighting as much as possible, because:

- some system is not working, so users can't work;
- users are complaining and losing confidence;
- repair will have to be done quickly, under extreme time pressure;
- we tend to make mistakes when under pressure: more haste, less speed;
- we forget to clean up (documenting, fixing temporary patches);

In this unit we'll cover some principles of fault detection. Later on, in Unit 6, we'll cover network troubleshooting in more detail.

Fire fighting analogy

How do these components:

- Reduce incidences of faults?

- Speed up detection and diagnosis?
- Aid rapid action?
- Keep users informed?

These principles are well known and apply to all kinds of systems, not just networks!

- Eliminate dangerous items from the network, or contain them where they cannot cause further damage.
- Use early warning systems to detect faults quickly, before too much damage has occurred.
- Give users a way to report problems, and notify them when a problem is occuring.

What problems can we deflect?

- Users report faults all the time.
- What faults do you NOT want to fix?
- Can you set boundaries on your responsibilities?

We need boundaries to be relaxed. Otherwise, the amount of work you might be asked to do is infinite:

- “Fix my printer!”
- “Fix my light switch!”
- “Fix my telephone!”

Please spend a few minutes making a list of common problems that you’re expected to solve.

To be a relaxed sysadmin, we must spend less time solving urgent problems like these. We can do that by:

- Limiting our domain (job description), e.g. “no electrics”. This makes it someone else’s problem.
- Understanding user problems and solving them before they occur

Limiting our domain is a policy issue, needs management support:

- Decide what you want to support
- Negotiate with management to get it in writing (policy)
- Post it on the door and walls of the IT centre and online helpdesk

For example, you might be able to:

- Make it someone else’s job to fill up the printer?
- Get a support contract for your Internet connection?
- Shut down the internal phone system and use mobiles instead?

We can also help users to help each other, by setting up a forum for self-help within the organisation, and helping people to use online resources to solve problems themselves (for example ServerFault.com). Could you allow users to sign for and collect more paper or toner themselves?

Group work: try brainstorming out a list of 10-15 problems. Include some which you consider to be your job, and some which you’d like to refuse.

Look at your list of faults, and cross off the ones that can realistically be made someone else’s problem, or where users could help themselves.

How do we diagnose faults?

Users report faults all the time.

We need an objective standard to verify and understand them, quickly.

What could cause these faults?

- “The internet is slow”
- “The printer is broken”

How would you identify or eliminate possible causes?

Users aren't usually lying, but they often don't understand the problem:

- **The internet is slow** could mean "Internet Explorer is running slowly" or "I have too many programs open" or "I stepped on my network cable and now it's broken".
- **The printer is broken** could mean "the cable is broken" or "my nephew uninstalled the printer driver from my computer" or "I changed my default printer and don't know how to change it back."

We have a lot of practice at diagnosing problems, often by hand. There can be better ways to do it.

How can we learn and improve?

Whenever you solve a problem, take five minutes at the end to think about:

- how could you have solved it more quickly?
- how could you prevent it from happening again?

Apply this to some of the problems on your list, in a group.

Note the value of group work. Building a support community for yourself can have massive benefits. Help others, and they will help you when you need it.

Discuss together:

- How would you check the printer status?
- How would you tell if the user's computer is running slowly?
- How would you tell if the user's computer is connected to the network?
- How would you tell if the Internet connection is OK?

Now discuss the problems on your own lists in your groups. If you are working alone, try to find solutions online, for example using communities like the Server Fault website. Don't be afraid to ask new questions on these sites if you can't find a solution!

Do it remotely

If possible, diagnose and repair faults remotely:

- Saves you travelling time.
- Allows you to check many potential causes quickly.
- First step towards automation.

But sometimes, face-to-face contact is more important! (When?)

Faster diagnosis of faults means a higher level of user satisfaction and quality of service. The whole university network runs better.

Discuss in your groups:

- How would you check the printer status **remotely**?
- How would you tell if the user's computer is running slowly **remotely**?
- How would you tell if the user's computer is connected to the network **remotely**?
- How would you tell if the Internet connection is OK **remotely**?

Remote diagnosis tools

What problems could these tools help you with?

- Printer's built-in web server
- Switch's built-in web server
- Windows remote destop (mstsc)
- Windows performance counters
- `ping`

- `traceroute`
- A network diagram

There are many useful command-line and graphical tools that can help you to diagnose problems remotely:

- Most printers have a web server that tells you their status, if you connect them to the network.
- You can use Windows Remote Desktop or VNC to log into a computer and Task Manager to check its performance.
- You can enable Windows' Performance Counters and access these remotely over the network. (Memory, swap, disk and CPU usage would be particularly useful).
- You can use the `ping` command to check the network connection of a computer or the Internet.
- You can use the `traceroute` command to identify routing hops between you and a target computer, locally or on the Internet (layer three).
- Managed switches have web interfaces that you can use to check port link status (you did write down what's connected to each port? :-) and connected MAC addresses (in case you didn't).
- You can use Netdot to automatically draw network diagrams at layer two, to help you diagnose layer two faults (between switches).

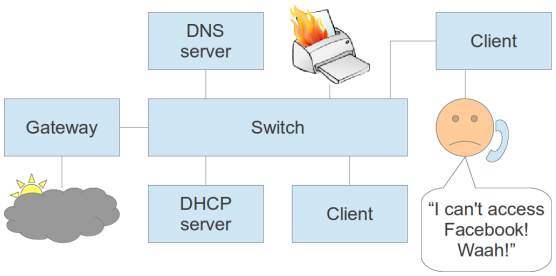
Look at your list of faults, and identify which tools you could use to diagnose them. Use the Internet or brainstorm in a group to generate ideas.

Practical session

- A fault has been reported
- Don't just dive in and fix it!
- Brainstorm ideas for what to check
- The facilitator will answer your questions

You may have been given a real, faulty piece of equipment, or if all our equipment is working perfectly, the facilitator may simulate a fault for you. If it's a real fault, you may be able to use network tools to investigate it yourself.

Your facilitator might ask you to refer to this network diagram:



If you are the facilitator, you may want to set up a test network like this for practice purposes. Early detection ———

If we can detect the fault early, we have more time to fix it, which means less stress and more flexibility:

> A stitch in time saves nine

How could you detect early warning signs of the problems on your list?

We can sometimes predict a problems before it occurs, and take evasive action:

Printer running out of toner.

> We can monitor the printer automatically with Nagios, which will send us email when the toner is running low.

Or we can detect them before users notice and complain about them:

Internet connection slowing down.

> If causes by one individual's excessive traffic, we can identify them, talk to them or take other measures to reduce their impact before other users complain.

Fewer noticed faults means a higher level of user satisfaction and quality of service. The whole university network runs better.

The more problems we can detect, the less firefighting we'll have to do, the calmer and less stressed we will be, and the better we can do our jobs to our own satisfaction.

Thinking Required

Warning: **hard work required!**

> Rarely do we find men who willingly engage in hard, solid thinking. There is an almost universal quest for easy answers and half-baked solutions. Nothing pains some people more than having to think. — Martin Luther King, Jr.

We all need time to relax and concentrate!

Adding to the List

Think about detection:

- After each problem is solved
- Make a list, or ticketing system
- Set aside time (meeting or thinking)
- Imagine future problems
- When creating or modifying a system
- Generalise from other systems

This process requires reflection. At least reflect on each problem after you solve it. It could be difficult to

you can make a list of problems. Analyse your day and see what you spend time on. Even better, track problems through a ticketing system, including how long it took to resolve them.

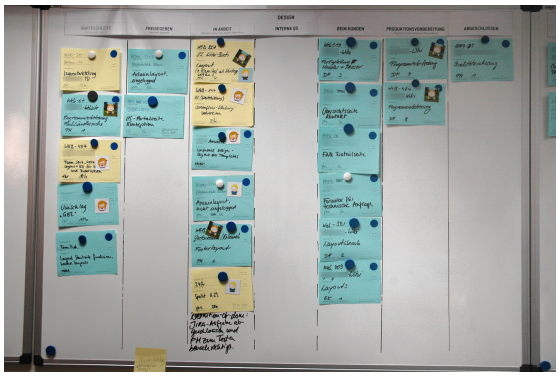
The more problems we can detect, the less firefighting we'll have to do.

After each problem is solved

Will you have enough time to do it properly? When you have multiple emergencies, it's difficult to relax and concentrate on defensive measures for the future.

Make a list

A list of some kind is your **most important tool** in tackling the problem. Even a paper list can help, but it's hard to rearrange and add information, and it becomes messy. Kanban cards or an electronic ticketing system can help:



- Kanban cards are easier to visualise and move around.
- Tickets are easier to sort, search, integrate with other systems and access remotely.

Set aside time

Ideally set aside half a day every month, or an hour a week, to start with, and increase if necessary. Arrange cover or close the IT office. Switch your phone off, so you can concentrate.

If you work alone, then spend the time thinking, preferably out of the office but with access to your list. If you work in a group, then have a meeting.

Run through your list, or at least the urgent and high priority items. Make a plan for dealing with it, allocate time and block that time out in your diary.

Running through the list shouldn't take more than an hour or two, so you can use the rest of the time time work on the highest priority tasks.

Time Boxing

This is a time management technique that helps us to achieve everything that we plan to. Time boxing is **strict scheduling**, where we do not allow ourselves to overrun the allocated time. If we are about to overrun, we can use the last few minutes to leave the task in a state where we can pick it up later, for example documenting what we did and what we were about to do, estimate how much more time is required, and schedule that time in our diaries to continue working on the task then.

Imagine future problems

Reacting to problems **after** they happen is a start, but we can do much better.

Make a "monitoring" list of all the systems that you can think of, which are your responsibility to manage. For each one, think about:

- how it might fail,
- how you could detect that,

- how you could automate monitoring of it,
- how you could prevent it from failing.

Put actions onto your task list to deal with these failures in the best way possible.

When creating or modifying a system

Whenever you create a new system, for example a web server, a ticketing system or install a printer, add it to your monitoring list and apply the same process as above.

Generalise

Try to apply the same solutions to multiple problems where possible. For example, monitoring all web servers, printers or IP phones using the same tool. Automated monitoring systems like `Nagios` allow you to create `host groups` that apply the same checks to any number of devices, by placing them into the group. For example, you could make a group of `web servers` and add everything that has a web interface to the group, whether it's a printer, a department website or Google.

Saving time with Automation

There are infinite possible problems:

- Checking for them takes time
- Risk = Hazard x Vulnerability – Capacity
- Prioritise your time (tickets or cards)
- Automate everything you can

If you have a tool that gives you a *yes or no* answer, *is there a problem or not*, then you can automate the use of that tool to generate alerts automatically as soon as a problem occurs.

Automated fault detection is difficult - but not impossible! Tools already exist to detect most faults in an automated way, if you can work out how to use them.

Look at your list of faults, and identify how you could get a *yes/no answer* about whether the fault exists. Use the Internet or brainstorm in a group to generate ideas.

How do we mitigate faults?

Make faults less likely:

- redundant systems
- automatic failover
- automatic updates (some risk)
- more reliable technology
- failure prediction and prevention
- test your disaster plans!

Make faults less severe or urgent:

- how can users continue to work without this system?

Reducing severity of faults means a higher level of user satisfaction and quality of service. The whole university network runs better.

Solutions to consider

How can users continue to work (maybe more slowly or less conveniently) without some system:

- a computer lab they could use instead of their desktop?

- a spare computer they could borrow?
- a wireless network they could use instead of the LAN?
- some way to send mail even if the main mail server is down?
- a backup (slower) internet connection/provider?

Look at your list of faults, and identify how you could mitigate each one, making it less likely or less severe. Use the Internet or brainstorm in a group to generate ideas.

Automatic Failover

In some cases you can make the fallback system automatic. This is known as **automatic failover**. This usually requires the most work to set up and keep running. For example:

- Most servers can use redundant power supplies and disks and error-correcting (ECC) RAM to reduce the risk that hardware failure will cause an outage. Sometimes a failed part can be replaced without bringing the system down for maintenance (hot swap).
- Web, database, file and mailbox servers can be configured in a cluster with automatic replication, so that the backup server has a full copy of all the latest data, changes and configuration, and will be activated automatically if the primary fails.
- Virtualised servers can often be moved to a new host without downtime (VMWare ESXi, OpenVZ and Proxmox support this).
- Many routers allow **hot failover** of the IP address from a primary to a backup router, but this requires you to keep the configurations manually synchronised between the two. Alternatively you could have different configurations using different Internet connections, and still have the router's internal IP address switched from the master to the secondary automatically, using `VRRP` or `CARP`. Open TCP/IP connections will be disrupted, so this should only be done when really necessary.
- Some routers can automatically switch from a primary to a secondary
- Many IP phone systems allow calls to automatically be forwarded to another number, such as the user's mobile phone, if their desk phone is faulty (switched off or unplugged).
- Some print servers allow multiple printers to service the same queue, and the next available printer will handle each job.
- Email (SMTP) can be handled using backup mail exchangers (MXes) which hold and deliver outbound mail, or queue incoming mail to the server holding the user's mailbox.

Some redundant systems protect against hardware failure, but not against user accidents such as files being deleted or corrupted. Therefore it's important to have other protection mechanisms too, such as backups.

Backups

Backups fall into three broad categories:

Full system images

: High space requirements, easy to restore whole systems, difficult to restore individual files, may require downtime for complete consistency, slow to backup so usually run daily or weekly.

Continous file protection

: Lower space requirements, not designed to restore whole systems, easy to restore individual files, not good for databases or email, usually not consistent, back up changed files almost immediately throughout the day and keep many versions.

Incremental backups

: Fall somewhere between these two extremes depending on the technology.

It's usually a good idea to have **both** full system images (for quick restore in case of a complete failure) **and** continuous file protection (for quickly restoring individual corrupted

files, and the latest changes after a complete system failure).

Hot Spares

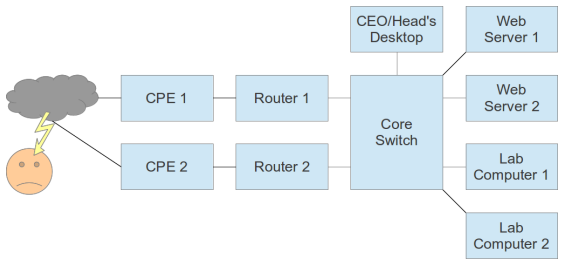
Sometimes it's too difficult to keep the backup server exactly synchronised with the main one (especially mailbox and file servers). In these cases you might have a physical spare server that's kept running (a **hot spare**) and synchronized every hour, or every day, with the master server. If you need to fail over to the hot spare, users will still have some service, but they may lose their most recent work.

Cold Spares

You can also keep spare hardware available but switched off (a **cold spare**) ready to be put into service by having a system image backup restored to it. Virtualisation alternatively allows you to temporarily overload some host servers in case of a failure, which will make systems slower but they will continue to function.

Redundancy Example

Study this network diagram and identify the single points of failure, and possible solutions including fallbacks (lower capability):



How do we keep users informed?

- Why does it matter to users?
- When do you notify them?
- What do you tell them?
- How do you tell them?

By notifying users when a fault has occurred, we allow them to plan their time, to avoid frustration and wasting time. This results in a higher level of user satisfaction and quality of service.

The network does not run better, but people are happier if their time is not being wasted.

We can notify users in advance, as soon as we know that a fault or reduced service will occur. The more advance warning they have, the better they can plan, and the less effect the downtime will have on them. But it will never be zero, so it's still better to avoid or mitigate downtime if possible.

If we did not notify them in advance, we can notify them as soon as possible when a fault has occurred.

In both cases, we can tell them when it will be fixed by. If we can't do that, we should make it clear why we don't know, what we are doing to find out, and when we expect to have more information by.

We can keep users informed of current and future status, for example by having a network status and planned outages web page. We can encourage users to check this page before contacting IT support or submitting fault reports. We can make the page easy to find and to update.

How do we get better?

The University Of Sheffield.

Student Services Information Desk

How did we do today?

Please use this form to provide feedback about our services

Your name / email address

Date of visit

Would you describe your feedback as (tick where appropriate)

Comment		Complaint		Compliment		Suggestion	

Your feedback

Please put completed forms into the SSiD suggestion box or hand into the SSiD front desk counter. **Thank you for your feedback.**

- Desire to improve
- Reflect on what we did
- Inquire into other possibilities
- Share knowledge with peers
- Ask users for their opinion

From Reflective Practice on Wikipedia:

> Reflective practice is “the capacity to reflect on action so as to engage in a process of continuous learning”, which, according to the originator of the term, is “one of the defining characteristics of professional practice”.
>
> It involves “paying critical attention to the practical values and theories which inform everyday actions, by examining practice reflectively and reflexively. This leads to developmental insight”.
>
> Reflective practice can be an important tool in … individuals learning from their own professional experiences, rather than from formal teaching or knowledge transfer. It may be the most important source of personal professional development and improvement. As such the notion has achieved wide take-up, particularly in professional development for practitioners in the areas of education and healthcare.
>
> The question of how best to learn from experience has wider relevance however, to any organizational learning environment. In particular, people in leadership positions have a tremendous development opportunity if they engage in reflective practice.

One great value of a support ticket system is in examining closed tickets. Regularly take a sample and ask yourself:

- How do you know it was resolved?
- Has it happened again since?
- How many times has this happened?

- How much frustration or difficulty does it cause?
- Could you reduce the risk or severity of it happening?
- How long did it take to resolve? Could it be faster?
- Could you add monitoring to detect if it happens again?

Also, look at the users of the system. How many people will come back next time they have a problem? If not, why not? Does the system reward or discourage them?

What fraction of support tickets are duplicates? How much extra work do they create? Can the system do anything to reduce them, such as searching for similar tickets and offering them to the user? Can you provide self-help information, such as instructions on resolving common problems?

How long does it take to resolve a support request/ticket? Is the time reasonable? Could it be reduced?

If some requests are denied, because they fall outside the scope of the IT department's responsibility or contradict policy:

- How many requests are successfully resolved and how many denied?
- Do successful and denied requests fall into categories?
- Could you help users to know in advance whether their request is likely to be accepted or denied?

It's a good idea to schedule time regularly to work on these issues. Preferably, leave a minimal support staff at the office, and work from a quiet place with phones off, where you can think clearly without distraction.

Personally I like to reflect, immediately after a problem is resolved, to find two sensible ways to prevent that problem from happening again, or detect it more quickly; and implement both of them.

If you don't have time to implement them immediately:

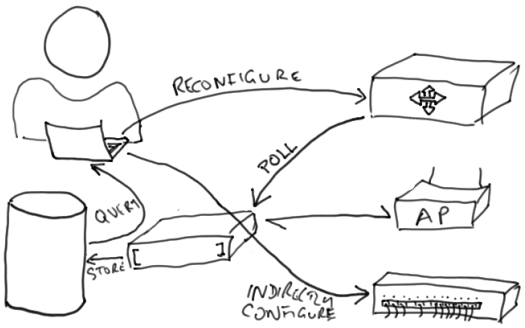
- Make a list of things to do.
- Schedule a reasonable amount of time each month to work on them.

Recognise and reward staff who provide excellent service to users.

Tell people about what you do. Write an organisational blog about the changes you've made to improve service. If users believe that you can change, they are likely to try to change you. *This is an excellent thing.*

Your users know what the biggest problems are. Who are your most important users? Do you talk to them?

What is Configuration Management?



Recording changes to configuration of network devices, for example:

- configuration file management,
- inventory management,

- software management.
- package management.

Why use Configuration Management?

Configuration Management helps us to answer the following questions:

- What changed?
- When did it change?
- Why did it change?
- Can we put it back?
- Can we solve the problem a better way?

Thus, it helps us to solve two important problems:

- We often need to understand what caused a problem in terms of changes to a system, in order to reverse them or find a better solution.
- We often want to repeat the changes, for example setting up a new router or server in minimal time, or reconfiguring all routers at the same time.

Configuration management is a form of *version control*, applied to system configurations: configuration files, installed packages and updates, registry changes, connection maps and diagrams.

Benefits of Configuration Management

- Always have an up-to-date network map for troubleshooting.
- Replace hardware more quickly.
- Faster testing of new configurations, with quick reversion.
- Easier and more reliable creation of test labs.
- Inventory for insurance, updates, patching and impact assessment.
- Diagnosing and repairing a fault caused by a configuration change.
- Better communication in a network management team.
- Better license compliance.

Configuration management is hard

Includes the entire state of every device:

- The partition layout, entire filesystem and registry of servers;
- The physical connections between machines and network devices;
- The configuration files of devices.

That's a lot of data! Where will you put it?

Configuration management of servers

- Another reason to have backups!
- Can you quickly identify file and registry changes?
- How quickly can you restore a server?

Configuration management of the network

Tools like `Netdisco` and `Netdot` can automatically draw network maps for you.

`RANCID` can collect

A lot of data! Do you have space?

How would you do it?

Some problems with this scope:

- Do you have space to back up your entire servers?
- Do you have the software and time to run those backups?
- How quickly can you identify a one-file change in a 400GB full-system backup?
- How do you monitor or backup physical connections between devices?
- How do you backup devices that don't have an automated interface?
- How do you cope with encrypted, binary or unreadable configuration files?
- How do you deal with dynamic configuration, such as IP addresses?
- What happens if you forget to document a change, or don't have time?

Some possible solutions:

- Can automatically monitor some switches and routers configurations with RANCID: Cisco routers, Juniper routers, Catalyst switches, Foundry switches, Alteon switches, and HP Procurve switches among others.
- Netdot can automatically generate network maps for you.
- Can configure Linux and FreeBSD servers using Puppet or Chef to make initial configuration easier.
- Can standardise and automate installation of servers.
- Can use full-system backups that support listing the changes between two abitrary increments, and restoring just those files (e.g. `rdiff-backup` or `duplicity` plus shell scripting).
- Can reduce the number of people allowed to make configuration changes.
- Can set policy to require documentation for changes.
- Can monitor servers and routers for unexpected or unauthorised changes, for example using `tripwire` or backup systems.

What is Performance Management?

> Monitor and measure various aspects of performance so that overall performance can be maintained at an acceptable level.

How would you do it?

- What can you measure?
- How would you measure it?
- How would you store it?
- How would you report it?

Why use Performance Management?

- Forensic analysis of failures
- Predicting future failures

We most often use these graphs when a failure has occurred, or a problem is detected by a monitoring system, such as:

- a server crashed
- a network connection is slow/congested/lossy
- Nagios detected a server running out of disk space or memory
- a server is/was running slowly (out of spec)

Nagios checks (yes/no answers) are our most important performance spec. We set thresholds for things like:

- "too slow" page loading
- "too slow" email delivery
- "too much" packet loss
- "too little" disk space free

All these thresholds are arbitrary, *but* they often detect problems before they occur. They also often create noise. Sometimes we have to change the thresholds to reduce noise and keep the system useful.

By looking at the performance graphs we can establish correlations:

- The disk filled up suddenly, overnight: someone left a job running?
- The network connection has been getting more congested for a while,
- Available bandwidth suddenly dropped two weeks ago when the ISP sent an engineer out to resolve another issue.
- The server's memory is nearly full every night: a scheduled cron job?

Usually these correlations either help us to create new hypotheses, or rule out some hypotheses, to help us identify the cause. They rarely answer the question by themselves.

Graphs are really useful for quickly analysing large quantities of performance data that would otherwise be useless, to pick out correlations and trends by engaging the brain's visual circuitry.

Case study: Diagnosis using baseline data

Useful questions for diagnosing performance-related problems:

- What is "normal"?
- Are we within a "normal" range?
- When did it change?
- What happened at the same time?

All of these require collecting and storing historical data (a baseline).

Careful with Performance Management!

- It's possible to measure everything;
- Everything has a cost;
- The benefits are limited:
    - post-mortem analysis,
    - early warning of problems.

Performance data can require:

- a lot of CPU time and bandwidth to collect and aggregate;
- a lot of disk space to store;
- a lot of CPU to produce graphs.

For example, our monitoring system with 55 hosts, 650 services in Nagios and 31 hosts, 1085 services in Munin, uses:

- 93% of one virtual CPU
- 5.7 GB disk space
- 1.8 GB per day bandwidth

Performance management tools

Some software tools return numerical information about services, or even archive and graph it:

Smokeping

collect and graph low-level packet loss and latency data from wireless and modulated Layer 2 links (quality of service/QoS).

Munin

collect and graphing information from Unix servers, such as disk, memory and CPU usage, mail queue size, etc.

Windows Performance Counters

collect and graphing information from multiple (newer) Windows servers across a network.

Nagios

collect performance metrics as well as up/down decision data and from each service, can derive service uptime, can be graphed using add-ons.

Zenoss and Cacti

collect performance metrics from devices with SNMP support and graph them.

pmacct, pmgraph, argus and nfsen

collect network traffic information broken down by flow for analysis.

squid cache manager, webalizer and Google Analytics

collect web traffic logs for analysis.

rrdtool

stores time-sequence data with high performance, automatic history purging, and decent graphs. Used by smokeping, munin and cacti.

What is Security Management?

Provide access to network devices and corporate resources to authorized individuals.

Why use Security Management?

- detect intrusions
- identify culprits
- ease account/password management
- prevent access by untrusted individuals

Security stops people from doing things they want to do:

- plugging their computer into the network
- downloading large files, videos and pornography
- installing unauthorised software on your computers
- accessing the network without authorization

Therefore it's important to sensibly balance security and ease of use, and to be flexible (willing to change if necessary).

Sometimes good security makes life hard for legitimate users:

- having to remember many user accounts and passwords
- having to use long passwords which are hard to remember
- having to change their passwords regularly or if compromised
- not being able to access the network from the Internet
- not being able to access the network using any socket they want
- requiring the use of antivirus software that slows their computer down
- blocking legitimate traffic based on keywords
- installing security updates regularly creates downtime

These can be balanced by improved security technology:

having to remember many user accounts and passwords:

provide a directory service integrated with most or all services, such as RADIUS, LDAP, Active Directory, Kerberos, FreeIPA, Samba4 or GoSa.

having to use long passwords which are hard to remember:

Provide two-factor authentication such as hardware tokens to reduce password strength and change requirements while maintaining security.

having to change their passwords regularly or if compromised:

provide a directory service as above.

not being able to access the network from the Internet:

provide a VPN service such as OpenVPN, Cisco IPsec or IPsec/L2TP.

not being able to access the network using any socket they want:

use 802.1x and RADIUS for network authentication instead of MAC address locking, to maintain identity and access control.

requiring the use of antivirus software that slows their computer down:

choose antivirus software with lower performance impact, or use systems which are less vulnerable, such as tablets, phones and Linux machines.

blocking legitimate traffic based on keywords:

slow down traffic, or monitor and investigate, instead of blocking traffic outright if it appears to breach acceptable use policy.

installing security updates regularly creates downtime:

schedule and automate installation of security updates at quiet times such as during the night. Test security updates on some canary servers before applying to others, or using one of a redundant group.

What are good security practices?

Too many to list! Some examples are:

- strong passwords, regularly changed, or two-factor authentication
- single sign on
- every password has an expiry date
- monitor attempts to log in using expired accounts
- run an intrusion detection system and tune it
- practise penetration testing, security bypass and intrusion response at least annually
- monitor web access for unusual/banned sites and protocols
- encrypt where possible: passwords, disks, emails, backups
- restrict access to certain IP addresses where possible
- use VPNs to provide additional protection on public services where possible
- install security updates regularly, on a schedule
- monitor security and antivirus status and alert if not updated
- lock down computers and user accounts to prevent unauthorised software installation
- ensure that computers use is always associated with a user account
- ensure that security protocols are written down and staff trained in them
- protect yourselves against social engineering and dumpster diving
- monitor blacklists to alert you if your IP addresses appear on them
- store system logs, especially security logs, on a write-only server
- physically secure network equipment, servers, desktops and laptops
- keep backups of servers and configurations to restore quickly after an intrusion

What is Accounting Management?

Usage information of network resources, for:

- detecting and tracing excessive use
- predicting future capacity needs
- auditing and forensics
- billing for usage or enforcing quotas (in some cases)

Detecting, tracing, billing and quotas are covered in the Bandwidth Management unit.

Disaster Response

- What disasters might happen to your network?
- How would you cope with them?

Plan for destruction of all your equipment by fire or theft:

- How long would it take you to recover?
- What does your insurance cover you for?
- How much would it cost?
- How would you respond?
- Can you have a backup/alternate system in place quickly?

Keep backups of all servers and practise restoring a server every year.

From the e-book How to Accelerate your Internet:

Make regular backups

Over time your network configuration will grow and expand to suit your particular network. Remembering the intricate details will become impossible making it very difficult to reproduce the same configuration if it is lost.

Making regular backups ensures that you can rebuild your configuration from scratch if required. Having multiple backups means you can roll back to a previous known working state if a configuration change goes awry.

Disaster plan

Technology is not always as reliable as we hope, and it is a certainty that at some point major problems will strike your network. By planning for these, and having a procedure in place for dealing with them, you will be in a far better situation when the lights go off!

Fallback network mode

It is useful to prepare a basic network configuration state, which only allows a minimum set of services on a network. When a problem occurs which stops the network from functioning effectively you can implement this fallback mode, allowing others to use essential services whilst you are troubleshooting the problem.

Summary

Hopefully you now feel confident with:

- Good practice in network management
- Identifying and solving problems
- Detecting and predicting problems
- Preparing for disasters
- The role of policy in guiding/changing behaviour

Please take a moment to reflect, review and share your learnings and plans for action with the group.