Free Software Infrastructure in Informatics

Simon Wilkinson <sxw@inf.ed.ac.uk>
Stephen Quinney <squinney@inf.ed.ac.uk>
Overview

- Whistlestop tour of Informatics’s computing environment
  - Contexts & philosophy
  - Infrastructure & Middleware
  - Authentication & Authorisation
  - Directory Services
  - Machine Configuration
  - File Services
  - Putting it all together

- Necessarily high level - ask for more detail if you are interested

- Questions taken at any point...
Contexts

- ~4600 user accounts (~4100 currently active)
  - ~800 staff
  - ~3200 students

- Approximately 2500 hosts (1400 managed)
  - ~ 800 desktops
  - ~ 400 public lab machines
  - ~ 160 servers

- 4 geographically distinct sites
  - Kings Buildings
  - Appleton Tower
  - Forest Hill
  - Buccleuch Place
Philosophical Issues

- Services should be accessible from any platform and from anywhere

- The network shouldn’t be trusted (both internally & externally)

- Machines should be resilient to both local and remote network disconnection

- There will always be non-web delivered services. Web-only solutions are not appropriate.

- All lab and desktop machines should have an identical configuration, unless there are specific requirements otherwise.

- Systems should be based around Open Standards
Infrastructure models

- Traditionally ....

```
  Network
   \   /  \
   //   //
  Services
```

```
  Clients
```

Diagram showing the relationship between Network, Services, and Clients.
Infrastructure models

- Now ....

- Middleware is the glue that binds services together
Networking

• Information Services provide networking between our sites

• UKERNA/JANET provide wider network access

• We do all of our own internal networking

• Switches automatically configured from text based configuration data

• Firewalls automatically configured by our configuration management system

• Linux on all firewalls and routers
Managed Machines

- Provide an operating environment named DICE
- Fedora Core (some 5, now mainly 6) based
- Still have some Solaris servers, but they’re a dying breed
- Looking at moving to Scientific Linux 5 for servers in the near future
- Predominantly use an ‘off the shelf’ distribution, so aren’t going to talk much about it.
- However, we have a large number of local packages ...
Middleware

- Middleware is last year’s buzz phrase
- Also called ‘soft-infrastructure’ and many other things by the marketing folks
- Layer that sits between the network and the services you run
- Typically includes
  - Authentication Services
  - Authorisation Services
  - Directory Services
  - Machine configuration systems
  - File service
- Most of this talk is going to be about these 5 areas
Buzzword bingo

- One of the perils of middleware is that everyone wants to sell you something

- Buzz words prevail

- No two vendors use the same word to mean the same thing!
Authentication

- Authentication is the act of proving your identity to the system.

- It’s not about determining what that identity can do.
Authentication systems

- Trivial systems store username/password lists for every service
  - Password synchronization is hard
  - User has to remember many different passwords
  - User has to repeatedly enter passwords on many different occasions
  - Lots of machines have password lists lying around
  - Compromise of a machine allows the compromise of every other machine that uses the same passwords.

- Single signon systems solve these problems
Single signon

- A single password for all services (within the same trust domain)
- Password only required once at the start of a session
- Servers never obtain the user’s password
- The compromise of one service must not compromise the whole system
Kerberos

• Informatics use Kerberos for all internal authentication

• Kerberos is a cryptographically secure protocol linking clients and servers through a trusted third party - the Key Distribution Centre.

• Designed for use between trusted systems, across an untrusted network - the client’s password never crosses the network.

• Clients receive ‘tickets’ from the KDC which allow them to prove their identity to services, without allowing that service to impersonate them elsewhere

• Clients may ‘delegate’ their tickets to a service if they wish to permit impersonation (for example, when accessing a login service)
Kerberos services

- Whilst Kerberos can simply be used as a central password source, that doesn’t provide the full power

- To get the full benefit from Kerberos, you need support in your client and server applications

- Applications with support include:
  - **AdiumX**, amanda, Apache httpd, AFP, Apple CalendarServer, alpine, Balsa, Coda, CUPS, curl, cvs, Cyrus, DB2, Dovecot, evolution, Emacs, Eudora, Evolution, Fedora-DS, fetch, **firefox**, GridEngine, iCal, iChat, Internet Explorer, **jabberd2**, kmail, konqueror, Kermit, LPRng, Mail.app, Mutt, Mulberry, NFSv4, Oracle, OpenAFS, OpenLDAP, **OpenSSH**, qpopper, pine, **pidgin**, postfix, postgresQL, Safari, Samba, sendmail, SQL Server, **SunSSH**, svn, **Thunderbird**, UW-IMAPD, Wildfire
Kerberos @ Informatics

• Use MIT Kerberos for servers - clients use whatever they ship with.

• Multiple deployed KDCs for redundancy, all running on Linux.

• DNS records to support zero configuration

• Locally written services deployed to automate key management as part of our machine configuration system

• Can authenticate to our machines from any compatible machine, anywhere on the Internet!
Kerberos demo
Kerberos key management

- We still use other key management schemes
  - X509 (for TLS/SSL protected services)
  - SSH public keys

- Have developed software to bootstrap all of these keys from a machine’s Kerberos key.

- Only need to install one piece of key material on a given machine
Kerberos & the WWW

- Traditionally Kerberos support within web browsers has been poor.

- Need to solve the internet cafe case - machine with no Kerberos support but still wants to access our services.

- We use **Cosign** to provide web based authentication that ties in with our internal authentication system.

- A web user without Kerberos credentials is prompted for their username and password.

- Cookie based authentication then allows the user to access local web sites from anywhere on the internet.
Kerberos & guest users

• We have many virtual visitors - people who need more than public access to our services, but have no physical affiliation to the School.

• Created ‘iFriend’, which allows anyone with an email address to create an account on our KDCs.

• Remember - authentication has nothing to do with access control!
Authentication futures

- Passwords are becoming increasingly fragile
- Smartcards look like being the way to go
  - US Federal Government leading the way with PIV
- Smartcard support in Kerberos is under active development
Authorisation Services

• If authentication establishes “Who you are”, then authorisation controls “What you can do”

• General purpose authorisation services are hard
  • User X can print to printer Y
  • User X can print to printer Y, but only between 9am-5pm
  • User X can print to printer Y, but only documents less than 100 sheets
  • User X can print to printer Y, but only if they’re within their print quota

• Need to decide what kind be solved generally, and what is best left to the individual application

• No good, general purpose, open source solutions available in this space!
Authorisation @ Informatics

• System built around our directory service

• Each user has a set of ‘roles’. Some of these are determined from personnel data (‘staff’, ‘student’) - others are added by support staff (‘beowulfuser’)

• Each role gives a user a set of entitlements. Entitlements are service specific tags (‘print/laserprinter2’, ‘login/duffus’) which are checked to perform access control decisions

• Services then check whether a particular user has a particular entitlement in order to determine access

• Anything more complex is up to the individual service
Authorisation Implementation

- All source information stored in our directory service
  - Role => entitlement mapping information stored in custom maps
  - User role membership stored in their account record, and managed by account provisioning system

- Each entitlement has a LDAP group containing a list of users

- Entitlements also represented in netgroup format for use by legacy clients

- Trigger script updates entitlement groups and netgroups whenever source information is changed

- Code freely available (GPL) upon request
Directory Services

- Saying “we have a directory service” is a lot like saying “we have a database”!

- Informatics use our Directory Service for a number of distinct purposes
  - White pages
  - Yellow pages (NIS) information for Unix machines (passwd, group &c.)
  - Host information
  - Authorisation information
  - Application specific data
    - printcaps
    - KDC principal information
    - automounter maps
Directory Service Implementation

- OpenLDAP based service

- Single master, with public data replicated to every client (using locally written replication technology)

- Massive redundancy, but wouldn’t recommend this configuration!

- Currently looking to completely redesign our replication strategy

- All directory access is Kerberos authenticated, and controlled via authorisation entitlements
Directory Service Demo
Identity

• Putting the last three topics together, you nearly have the makings of a complete Identity Management solution

• Haven’t really discussed account provisioning, as it is so site specific

• Our account management toolset pulls information from multiple local, and University-wide, data sources to create appropriate entries in LDAP and our KDC.
The view from here ...

- All of these services are designed to be usable from any client, and by any service.

- Clients can authenticate by just doing `kinit user@INF.ED.AC.UK`

- Any standards-compliant application can use our LDAP white pages system

- Any Unix client which supports LDAP access from the name service switch can use our LDAP yellow pages data (using nss_ldap, or similar)

- Any Unix service which supports PAM can accept our Kerberos passwords (using pam_krb5)

- Any Unix service which supports LDAP group authentication can use our authorisation system
Configuration Management Options

• System imaging

• Just hack the machines till they work!

• Store config files elsewhere and use a deployment tool.

• Generate templates for config files that can be filled in per-host and deployed.

• Put templates on the host and deliver some configuration data to build config files.
Requirements

• Reproducibility
• Scalability
• Ease/Efficiency of Management
• Validity
• Manage relationships
• Manage change
An example – Create a new web server

• **Infrastructure**
  - DNS entry
  - DHCP entry
  - Firewall hole
  - SSL certificates
  - Backups

• **Configure the machine:**
  - Configure disks, install software, etc..
  - Configure dns, network, ntp, apache, etc..
LCFG

• Each managed host has a source profile

• Source profile pulls in headers
  • allows sharing of config data between machines

• Central server compiles this into XML.

• Client downloads XML profile and configures itself accordingly
The Profile

• Completely describes the required state of a machine

• Consists of a set of components and, optionally, a list of packages.
A Component

• Set of resources – basically key/value pairs.

• Optional templates for config files.

• Optional control code
  • Manage daemons (stop/start).
  • LCFG provides a standard framework for Perl and shell scripts.
Spanning Maps

- A component in one profile can *publish* resources to which a component in the profile of another machine *subscribes*.

- Usage includes:
  - dhcp
  - ipfilter
  - nagios, system monitoring
  - inventory
Configuration Conclusions

• LCFG
  • Improves usability (common language, etc.).
  • Allows a complete description of the state.
  • Allows devolved management.
  • Makes relationship management easy.
  • Provides the ability to manage change.
File services

- Providing a reproducible environment requires every machine has the same view of the filesystem
- In particular user’s homedirectories, but also research data
- Historically, used to use NFS
- Security was not good
- Manageability was even worse
AFS

- In the process of moving to OpenAFS

- AFS is a truly global filesystem - any AFS client, anywhere in the world can access files on any AFS server

- Data can be moved around between fileservers without the user noticing

- Security provided through Kerberos

- Many staff, and all new students now have AFS homedirectories. Roll out to continue over the next 4 years.
Putting it all together

• A quick demo of ‘iFile’ - web based access to AFS

• This shows off nearly everything we’ve talked about ... 

  • We’re authenticating with cosign (which gets Kerberos tickets for us)
  • The application checks us against the authorisation service
  • The web interface uses our Kerberos ticket to authenticate us to AFS
  • And the whole thing is configured and managed through LCFG
iFile Demo
Further Information

- Informatics Infrastructure
  - [http://www.dice.inf.ed.ac.uk/publications/](http://www.dice.inf.ed.ac.uk/publications/)
    Papers and talks about our Kerberos, OpenAFS and LCFG deployments

- Kerberos
  - [http://web.mit.edu/kerberos/](http://web.mit.edu/kerberos/) - the MIT Kerberos site
  - [http://www.kerberos.org/](http://www.kerberos.org/) - the Kerberos consortium
  - [http://www.eyrie.org/~eagle/software/pam-krb5](http://www.eyrie.org/~eagle/software/pam-krb5) - PAM krb5 module

- Directory Services
  - [http://www.openldap.org/](http://www.openldap.org/)
  - [http://www.padl.com/OSS/nss_idap](http://www.padl.com/OSS/nss_idap) - LDAP Name Service integration
Further Information (II)

- LCFG
  - [http://www.lcfg.org/](http://www.lcfg.org/)
  - [mailto:lcfg-discuss@inf.ed.ac.uk](mailto:lcfg-discuss@inf.ed.ac.uk)

- OpenAFS
  - [http://www.openafs.org](http://www.openafs.org)
  - [http://www.filedrawers.org](http://www.filedrawers.org)
Questions?

Talk available online at http://www.sxw.org.uk/computing/talks/edlug1007.pdf