Secure Single Sign-On

Andrew Findlay

Head of Networking and Systems

Brunel University Computing Services

What is SSSO?

ONE username / password / token for all services

(Maybe one per role)

One Authentication per session

Why SSSO?

10 PCs sharing files
9 NT Domains
8 Remote database servers
7 Partners with extranets
6 Secure departmental webservers
5 Standalone Unix CAD stations
4 Active e-mail accounts
..... that's 49 passwords so far ...
And a Partridge in a Pear Tree

Steps to SSSO

Decide on structure of system / namespace

Separate Authentication from Authorisation

Give each person or role a single ID

Make all systems accept the new IDs

Manage authorisation

Install agents to handle onward authentication

Flat Namespace SSSO

Single authentication authority

Short IDs (e.g. lw98ikb)

Central point of control and update

Distribute appropriate ‘password files’

e.g. /etc/passwd, SAM, SYSUAF.DAT

Simple to implement (technically!)

SSSO by Central Authentication

Use flat namespace as before

Provide authentication servers

RADIUS

TACACS+

Modify login process to use server

Can handle authorisation and logging too

Open SSSO

Many authentication authorities
  Loose connections between them

Structured IDs to avoid clashes
  lw98ikb@brunel.ac.uk
  cn=I K Brunel, ou=Law, o=Brunel Uni, c=GB

Trust model must be clear

Apps must check strength of authentication
  as well as authorisation

Applications to cover

Desktop PC login

Server login

Web authentication

Client-server apps

Remote apps

Building Blocks

Public-Key Infrastructure (PKI)
- Certificates and Certification Authorities
- Directories
- Agents and security libraries

Hooks into existing systems
- Unix Pluggable Authentication Modules (PAM)
- NT GINA
- Bodge scripts (expect etc)

Standards

PKI: IETF PKIX group (RFC 2459 etc)

LDAP (RFC 2251, RFC1777), X500 (ISO 9594)

GSS-API (RFC1509)

SASL - Simple Authentication and Security Layer

SSL - Secure Sockets Layer

Lots more to choose from...

Summary

SSSO is good: less admin, easier for users

Better still if globally usable

Apps/libraries need changing

Plan for the large scale

Plan for flexibility