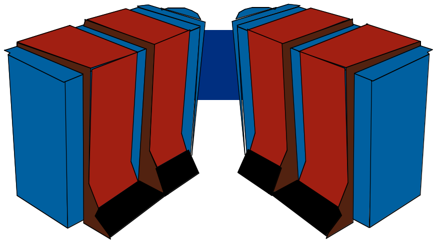
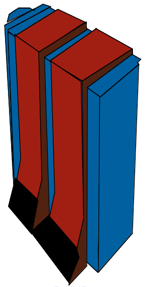
Cray Security Administration:
Tricks of the Trade



YMP8E



T3D128

Bonnie Hall

Senior System Specialist

Exxon Upstream Technical Computing

Overview

General discussion of controls

- What are they?
- Why do I need them?
- How do I determine what makes sense in my environment?

Controls in the UNICOS Environment

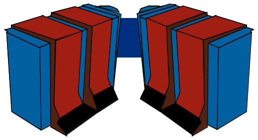
- Managing Data
- Managing Users
- Managing Privilege

What Are Controls?

“Security and Controls”

Control refers to the policies and procedures aspect of security.

Organization is a key piece of a secure system.



Why Do I Need Controls?

Minimize Risk:

- Information Disclosure
- Information Loss
- Disruption of computing services
- Financial Loss
- Legal or Ethical Violations

Appropriate Level of Control

- Determine probability of exposure
- Determine severity of exposure
- Evaluate cost of potential controls
- Implement controls which make sense:
  - Cost of controls **<** Potential loss
  - Reduce risk (severity x probability)

Managing Data

Data Owners:

- Approve and Review Access Permissions
- Approve and Review Group Membership

Management is a good choice. May appoint a representative to handle to approval process

Managing Data

- All data needs an owner
- All data in UNIX has a group
- Map each group to an owner

  and every file and structure on the machine has an owner.

Managing Data: Reports for Data Owners

```
#group:        owner:         org:           mailstop:
research:      Jane Doe:      R&D:           123 Bldg. A:
testing:       John Smith:    Systems:       999 Bldg. Q:
production:    Jill Smith:    Systems:       111 Bldg Q:
operations:    A. Jones:      Operations:    100 Bldg. Z:
```

Create list of groups in an organization: call   getgrps $org

```
getgrps ()
{ cat $grpown |\
  awk -F: -v org=$1 '{
    if ($3==org)
     { grprpt[i]=$1
       i++}}
  END{ for ( i in grprpt ) print grprpt[i]}' }
```

Managing Data: Access Violation Reporting

```
Apr 21 08:03:29 1997 Discretionary o_lvl: 0 s_lvl: 0 jid:57163 pid:42017
r_ids:[user1(12345),group1(100)] e_ids:[user1(12345),group1(100)]*******
Login uid: user1(12345)
Function: open (5)     Violation: Permission denied (13)
System call :  access (33)
Subject: Compartments :  none
Permissions :  none
Class :  0
Categories :  none
Access Mode :  read
Object: Level: 0 uid:user2(321) gid:group2(200) device:34,73 inode: 6157
Pathname :  /users/data/private
Compartments :  none
Class :  0
Categories :  none
Mode :  100551
```

Managing Data: Access Violation Reporting

```
/etc/reduce -g $grp -f $logfile -t desc -p | awk '{
  /Discretionary/  { month=\$1
                     day=\$2 }
  /e_ids/          { id=\$2 }
  /Access Mode/    { mode=\$4 }
  /Object/         { own=\$5
                     owng=\$7 }
  /Pathname/       { path=\$3 }

printf( "%s %s %s ATTEMPT %s TO %s OWN:%s,%s\n",\
            month, day, id, mode, path, owng, ownu } }'
```

Producing a one line per record format:

```
Apr 21 user1 ATTEMPT read TO /users/data/file OWN:group2(200),user2(321)
```

grep "OWN:group2" to produce a data access violation report for group2.

Managing Data: NFS ID Mapping

1. A file must exist in /etc/uidmaps/users called domain.passwd for each domain where user IDs will be mapped to Cray user IDs.

2. A file must exist in /etc/uidmaps/groups called domain.group for each domain with group IDs mapped to Cray Group IDs.

3. Exception files are created to map names and groups for each domain.

4. Create a list of hosts in each domain which will be using NFS. Call these files /etc/uidmaps/hosts/domain.

5. Create a list of domains to be mapped and use the nfsmerge command to create maps

6. Create a list of hosts in each domain

7. Use the nfsaddhost and the nfsaddmap commands to load the maps into the kernel

Managing Data: NFS ID Mapping

Script used to install the maps into the system kernel:

```
USRS=/etc/uidmaps/users
GRPS=/etc/uidmaps/groups
LOGS=/etc/uidmaps/log
MAPS=/etc/uidmaps/maps
HOST=/etc/uidmaps/hosts
DOMAINS=`cat /etc/uidmaps/domains`
/etc/uidmaps/nfsidmap -d                        # Disable ID mapping
/etc/uidmaps/nfsclear                           # Clear kernel NFS Map table
/etc/uidmaps/nfsaddhost -l localhost            # Add loopback entry
for domain in $DOMAINS
  do
  /etc/uidmaps/nfsaddmap -u $MAPS/u.cray.$domain -g $MAPS/g.cray.$domain  $domain
  for host in $HOST/$domain
    do /etc/uidmaps/nfsaddhost -d $domain -c -s -l $host
done ; done
/etc/uidmaps/nfsidmap -e                        #enable ID Mapping
```

Managing Users

UNIX Groups to identify user organization:

- Primary group
- Keyword group names

Group Owner is the responsible party:

- Approves user access
- Receives reports on user activity

Managing Users: Reports for Data Owners

```
#group:     owner:          org:          mailstop:       primary?
research:   Jane Doe:       R&D:          123 Bldg. A:    1:
testing:    John Smith:     Systems:      999 Bldg. Q:    0:
production:Jill Smith:      Systems:      111 Bldg Q:     1:
operations:A. Jones:        Operations:   100 Bldg. Z:    0:
```

A awk command like “getgrps” can be used to get primary groups. Report for each organization:

```
for org in *cut -d: -f3 $grp.own | sort -u*
  do for group in *getgrps $org*
       do
       processing for each group: data issues
  done
  for primary in *getpri $org*
     do
     processing for members of the primary group: user issues
  done
done > $organization.rpt
```

Managing Users

UNIX users have the ability to:

- set file permissions
- create .rhosts and .netrc files
- become inactive
- switch user to other users
- set easy to crack passwords

Managing Users: Login Aging

- Threshold one: suspend login: UDB Disabled bit.
- Threshold two: remove login: Data disposition??
Careful with home dirs of /
- Accounting system: /usr/adm/acct/sum/loginlog
- No record of system usage: date is 00-00-00
- Exempt system logins in the aging routine
- Include login expiration dates on other systems
- Create periodic reports for owning organizations

Managing Users: spcheck Command

List users who have administrative categories.

List all setuid and setgid files in the /bin, /usr/bin, /usr/lib, and /etc.

Report users in groups root, adm, bin, and sys.

Report infrequently used login IDs

List .profile and .cshrc files that are writable by anyone.

Report users with duplicate user IDs

List users who cannot change their passwords

List users with passwords that do not expire

Report users with numerous switch user failures

Report files that have the world read or write permission set

Report programs with the setuid or setgid bits set.

Report block/character special files not in /dev.

Checks /etc/passwd and /etc/group files for consistency

Distributed User Administration

UNIX users have many needs:

- ID Maintenance
- Group Maintenance
- Data Maintenance
- Resource Maintenance

Distributed User Management

Admin performs repetitive changes:

- Obtain owner approval
- Perform change
- Keep record of change
- Report changes to owners

Distributed User Management

Admin can automate user requests:

- Create scripts containing actions
- Prompt for & check input
- Add call to logging routine
- Create menu and secure shell appl.

See CUG 1995 Spring Proceedings; p349-354

Privilege Management

In UNIX: Superuser = Privilege

No privileged groups: just data access

No privileged accounts: just a login name.

Root is all powerful and must be trusted.

Privilege Management

There are five ways to get to root:

1) Single user mode
2) Log in as root
3) Switch user to root
4) SUID to root programs
5) Programs executed by root’s crontab

Privilege Management: Single User Mode

System is down for maintenance

- No logging
- No users
- No batch jobs
- No network activity

Change control

Management approval

Multiple people present

Privilege Management: Log in as root

Multiple administrators

Shared root password

Direct root login means no audit trail

Don’t do it! Let root users use su

Privilege Management: Switch User to root

su log provides audit trail of root access

but no record of actions

- Policy: one action per root access
- Front end su program to allow comments
- Reports and review

Privilege Management: SUID to root Programs

Run as root on behalf of a user

- designed to be run by all users
- designed to be run by subset of users

Problem: Group Membership controls access to privilege

Solution: Use Access Control Lists

Privilege Management: Entries in root’s crontab

Crontab: List of programs to execute as root by the scheduler (cron)

Root’s crontab can only be updated by root

Control over programs called by cron:

- File Access Permissions
- Change Control

Not the whole picture....

Audit Trails

Network Issues

Password Management

Change Control

Segregation of Responsibility

etc, etc, etc.

In Summary

Access Risk

Minimize Exposures

Distribute Responsibility

Distribute User Requests

Distribute Review

Questions??

This presentation is based on a paper of the same name which is available from the SecMIG page off the Cray User Group's Web Page:

http:\\www.cug.org

Bonnie.L.Hall@EXXON.sprint.com

(713) 966-6031