8

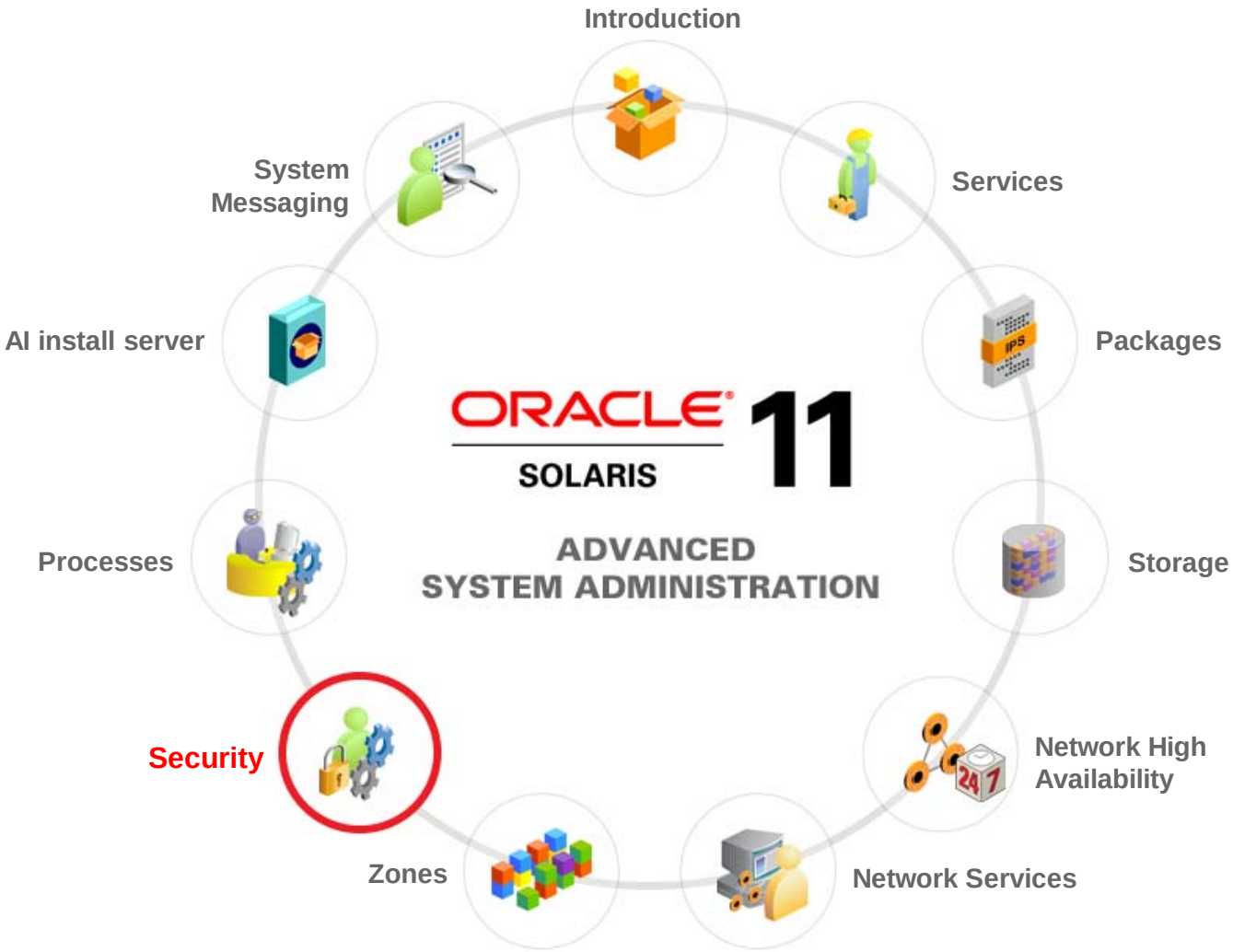
Securing the Oracle Solaris 11 OS

Objectives

After completing this lesson, you should be able to:

- Implement user privileges and roles
- Manage privileges
- Manage user rights
- Verify file integrity by using BART
- Monitor the audit service
- Assess the compliance of an Oracle Solaris system

Workflow Orientation



Agenda

- Implementing privileges, rights, and roles
- Managing privileges
- Managing user rights
- Verifying file integrity by using BART
- Monitoring the audit service
- Assessing the compliance of an Oracle Solaris system

Importance of Assigning User Privileges and Roles

It is important to assign user privilege and roles to ensure that:

- Processes and users have the appropriate level of access they need to perform their functions
- A company's requirements for process rights management and role-based access control are met

Process Rights Management and Privileges

- Process rights management is implemented by *privileges.*
- Privileges:
  - Enable processes to be restricted at the command, user, role, or system-specific resource level
  - Decrease the security risk associated with one user or one process having full superuser capabilities on a system
  - Enable gradation between user capabilities and root capabilities
  - Restrict programs and processes to only the capabilities that the program requires (the principle of “least privilege”)

Privilege Descriptions

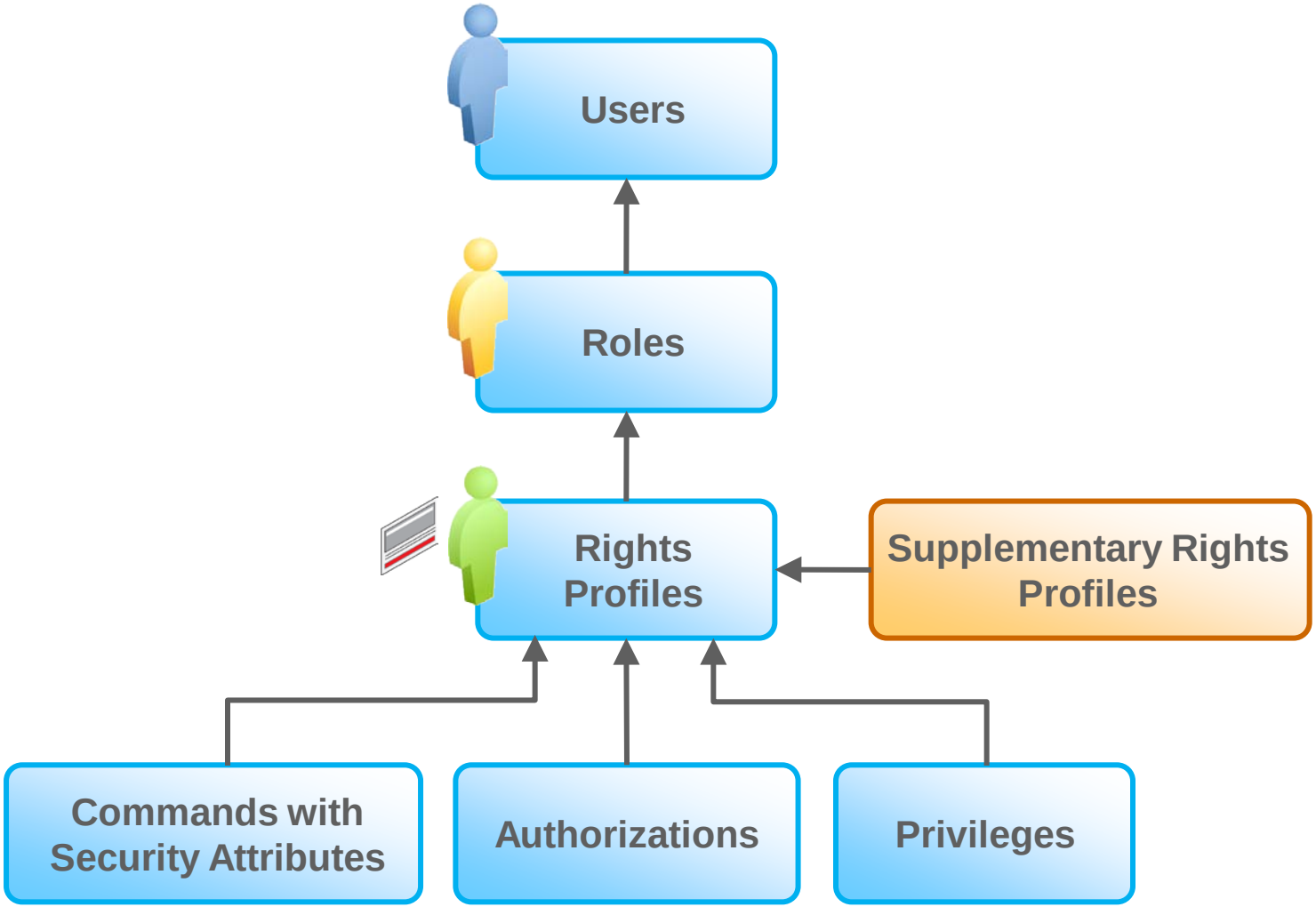
Area of Privilege	Description
`FILE` privileges	Privileges that begin with the string `file` operate on file system objects.
`IPC` privileges	Privileges that begin with the string `ipc` override IPC object access controls.
`NET` privileges	Privileges that begin with the string `net` give access to specific network functionality.
`PROC` privileges	Privileges that begin with the string `proc` allow processes to modify restricted properties of the process itself.
`SYS` privileges	Privileges that begin with the string `sys` give processes unrestricted access to system properties.

Implementing Privileges

Privilege Set	Description
Effective privilege set (E)	Set of privileges that are currently in effect
Inheritable privilege set (I)	Set of privileges that a process can inherit across a call to `exec()`
Permitted privilege set (P)	Set of privileges that are available for use
Limit privilege set (L)	Outside limit of the privileges that are available to a process and its children. By default, the limit set is `all` privileges

```
E (Effective): basic
I (Inheritable): basic
P (Permitted): basic
L (Limit): all
```

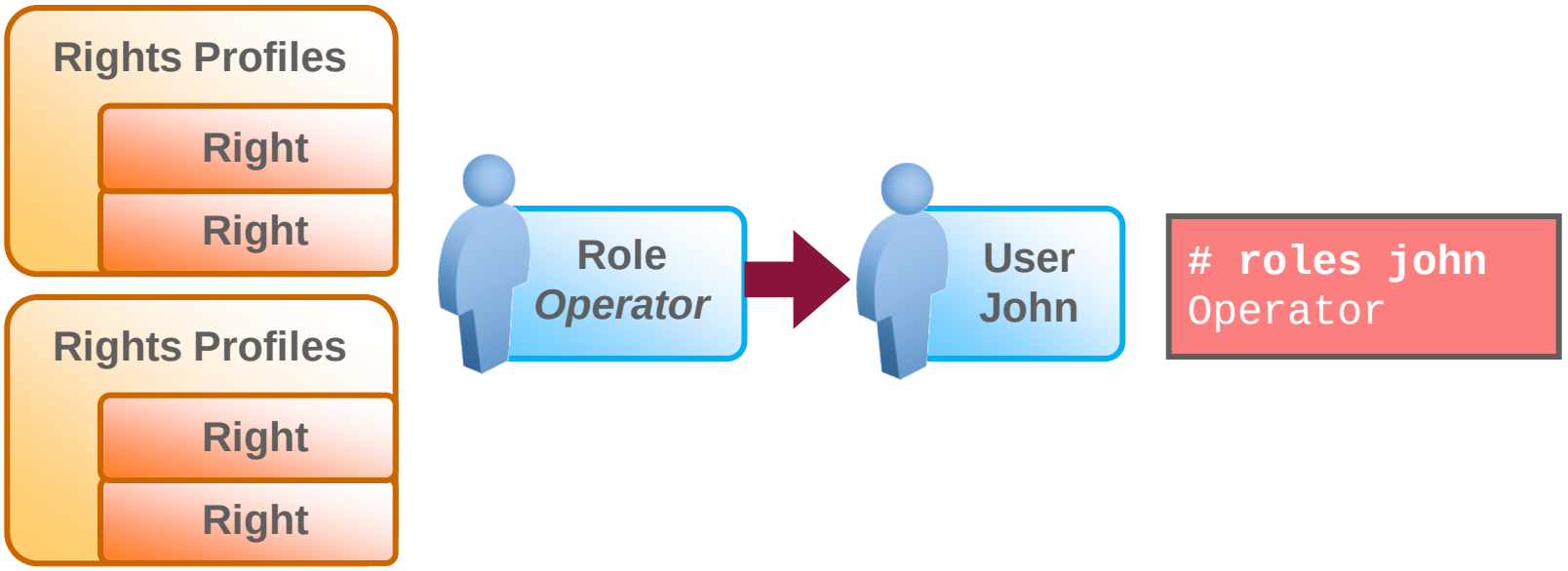
User Rights Management



Roles

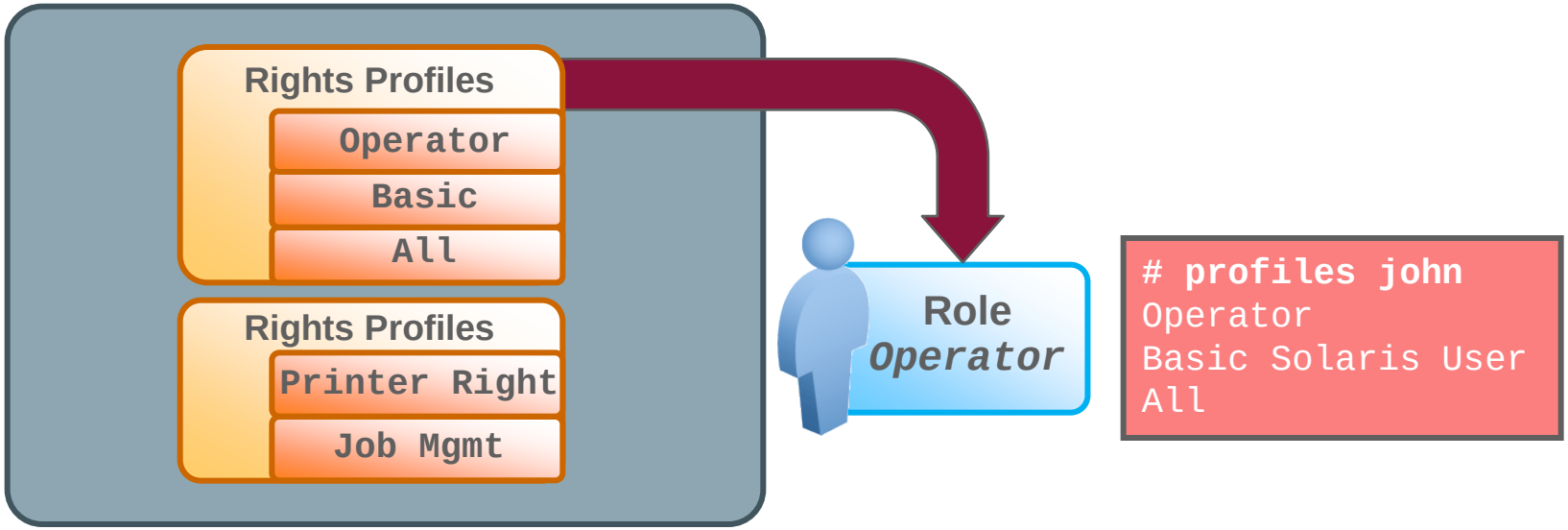
A role:

- Is a special type of user account that performs a set of administrative tasks
- Contains one or more rights profiles
- Provides access to restricted functionality



Rights Profile

- Is a collection of rights that can be assigned to a user or role
- Rights are commands or scripts that are run with special security attributes.



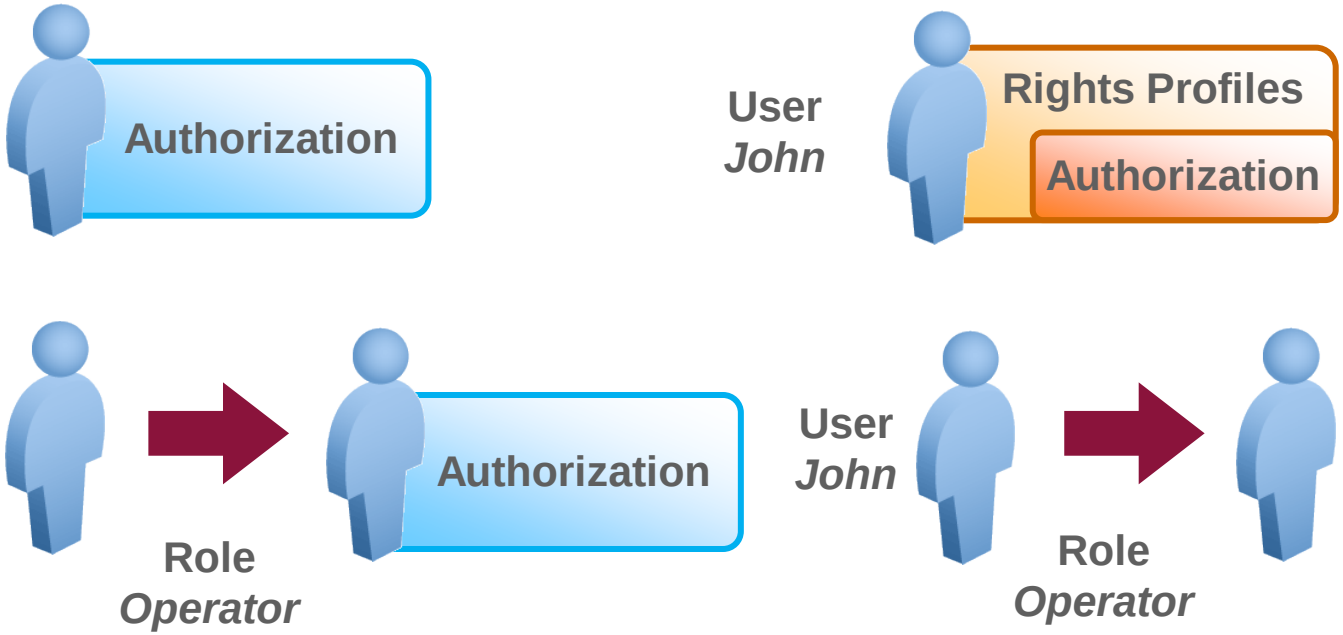
Basic Solaris User Rights Profile

```
# getent prof_attr | grep 'Basic Solaris User'
Basic Solaris User:RO::Automatically assigned
rights:auths=solaris.mail.mailq,solaris.network.autoconf.read,solaris.ad
min.wusb.read;profiles=All;help=RtDefault.html
```

Interpreting the /etc/security/policy.conf File

```
# cat /etc/security/policy.conf
<header and copyright output omitted>
#
AUTHS_GRANTED=
PROFS_GRANTED=Basic Solaris User
AUTH_PROFS_GRANTED=
CONSOLE_USER=Console User
#PAM_POLICY=
# crypt(3c) Algorithms Configuration
#
# CRYPT_ALGORITHMS_ALLOW specifies the algorithms that are allowed to
# be used for new passwords. This is enforced only in crypt_gensalt(3c).
#
CRYPT_ALGORITHMS_ALLOW=1,2a,md5,5,6
<output omitted>
#CRYPT_ALGORITHMS_DEPRECATE=__unix__
...
CRYPT_DEFAULT=5
#PRIV_DEFAULT=basic
#PRIV_LIMIT=all
#
# LOCK_AFTER_RETRIES specifies the default account locking policy for local
# user accounts (passwd(4)/shadow(4)). The default may be overridden by
# a user's user_attr(4) "lock_after_retries" value.
# YES enables local account locking, NO disables local account locking.
# The default value is NO.
#
#LOCK_AFTER_RETRIES=NO
```

Authorizations and Privileges

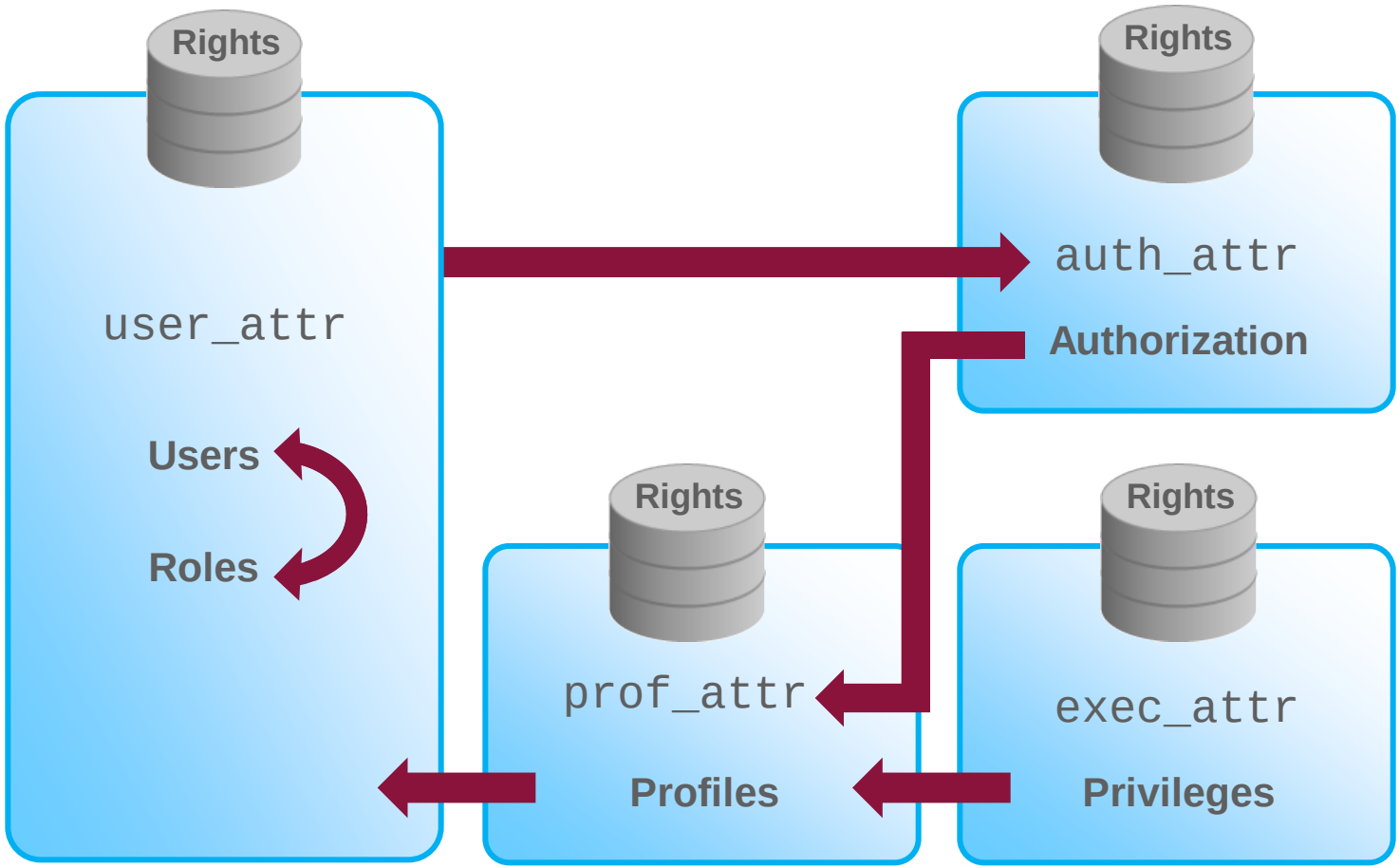



```
# auths john
solaris.admin.wusb.read,solaris.mail.mailq,sola
ris.network.autoconf.read
```

Security Attributes

- Enable a process to perform an operation that is otherwise forbidden to regular users
- Include authorizations, privileges, and `setuid` and `setgid` programs
- Can be assigned to a user

Key Rights Databases



Interpreting the `user_attr` Database

```
# getent user_attr | grep chris
chris::::profiles=Printer Management
```

Interpreting the `auth_attr` Database

```
# getent auth_attr | more
solaris.smf.read.ocm:::Read permissions for protected Oracle Configuration Manager
Service Properties::
solaris.smf.value.ocm:::Change Oracle Configuration Manager System Repository
Service values::
solaris.smf.manage.ocm:::Manage Oracle Configuration Manager System Repository
Service states::
solaris.smf.manage.dbus:::Manage D-BUS Service States::help=SmfDBUSStates.html
solaris.:RO::All Solaris Authorizations::help=AllSolAuthsHeader.html
solaris.account.:RO::Account Management::help=AccountHeader.html
solaris.account.setpolicy:RO::Change the account policy
settings::help=AccountSetPolicy.html
solaris.account.activate:RO::Sets the initial password for a newly created
account::help=AccountActivate.html
solaris.admin.edit:RO::Edit Administrative Files::help=AdminEdit.html
solaris.admin.idmap.rules:RO::Manage Identity Mapping Rules::help=IdmapRules.html
solaris.admin.wusb.:RO::Administer Wireless USB::help=WUSBHeader.html
solaris.admin.wusb.read:RO::Read Wireless USB Host and Device
Information::help=WUSBread.html
solaris.admin.wusb.modify:RO::Add or delete information of Wireless USB
Device::help=WUSBmodify.html
solaris.admin.wusb.host:RO::Manage Wireless USB Host::help=WUSBhost.html
solaris.audit.:RO::Audit System-wide Management::help=AuditHeader.html
<output omitted>
```

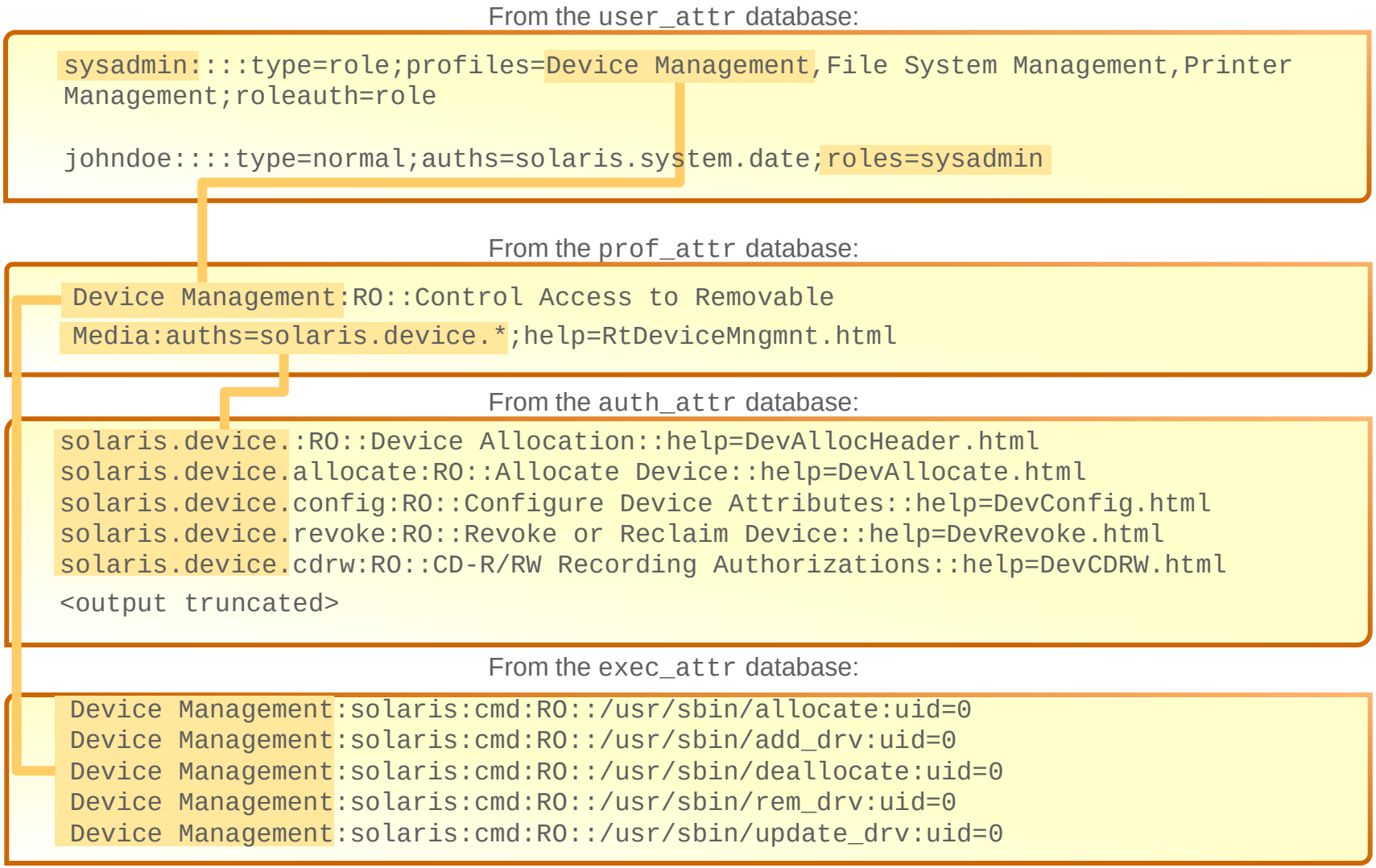
Interpreting the `exec_attr` Database

```
# getent exec_attr | grep 'Network Management'
Network
Management:solaris:cmd:RO::/usr/sbin/dladm:euid=dladm;egid=netadm;privs=sys_dl_conf
ig,net_rawaccess,proc_audit
Network Management:solaris:cmd:RO::/usr/sbin/dlstat:euid=dladm;egid=sys
Network
Management:solaris:cmd:RO::/usr/sbin/flowadm:euid=dladm;egid=sys;privs=sys_dl_confi
g,net_rawaccess,proc_audit
Network Management:solaris:cmd:RO::/usr/sbin/flowstat:euid=dladm;egid=sys
Network
Management:solaris:cmd:RO::/usr/sbin/ipadm:euid=netadm;egid=netadm;privs=sys_ip_con
fig,net_rawaccess
Network Management:solaris:cmd:RO::/usr/bin/ipstat:privs=dtrace_kernel
Network Management:solaris:cmd:RO::/usr/bin/netstat:uid=0
Network Management:solaris:cmd:RO::/usr/bin/rup:euid=0
Network Management:solaris:cmd:RO::/usr/bin/ruptime:euid=0
Network Management:solaris:cmd:RO::/usr/bin/setuname:euid=0
Network Management:solaris:cmd:RO::/usr/bin/tcpstat:privs=dtrace_kernel
Network Management:solaris:cmd:RO::/usr/sbin/asppp2pppd:euid=0
Network Management:solaris:cmd:RO::/usr/sbin/ifconfig:uid=0
Network Management:solaris:cmd:RO::/usr/sbin/ipaddrsel:euid=0
Network Management:solaris:cmd:RO::/usr/sbin/ipqosconf:euid=0
Network Management:solaris:cmd:RO::/usr/sbin/rndc:privs=file_dac_read
Network Management:solaris:cmd:RO::/usr/sbin/route:privs=sys_ip_config
...
<output truncated>
```

Interpreting the `prof_attr` Database

```
# getent prof_attr | more
TPM Administration:RO::Administer Privileged TPM
Operations:auths=solaris.smf.manage.tcsd,solaris.smf.value.tcsd
Desktop Configuration:RO::Configure graphical desktop
software:auths=solaris.smf.manage.x11,solaris.smf.manage.font,solaris.smf.manage.op
engl,solaris.smf.manage.dt.login
D-BUS Management:RO::Manage D-
BUS:auths=solaris.smf.manage.dbus;help=RtDBUSMngmnt.html
CUPS Administration:RO:::auths=solaris.smf.manage.cups
DTrace Toolkit::::
Software Installation:RO::Add application software to the
system:auths=solaris.smf.manage.servicetags;profiles=ZFS File System
Management;help=RtSoftwareInstall.html
Device Security:RO::Manage devices and Volume
Manager:auths=solaris.smf.manage.dt.login,solaris.device.*,solaris.smf.manage.vt,so
laris.smf.manage.allocate,solaris.smf.value.keymap;help=RtDeviceSecurity.html
NTP Management:RO::Manage the NTP
service:auths=solaris.smf.manage.ntp,solaris.smf.value.ntp
...
...
Audit Configuration:RO::Configure Solaris
Audit:auths=solaris.smf.value.audit;help=RtAuditCfg.html
Audit Control:RO::Control Solaris
Audit:auths=solaris.smf.manage.audit;help=RtAuditCtrl.html
Audit Review:RO::Review Solaris Auditing logs:help=RtAuditReview.html
<output omitted>
```

Relationships Among the Four Rights Databases



Profile Shells

- Enable access to the privileged rights that are assigned to the rights profile
- Are assigned to a specific user as a login shell or through the `su` command to assume a role
- Users must be assigned one of the profile shells: `pfsh` for Bourne shell (`sh`), `pfcsh` for C shell (`csh`), or `pfksh` for Korn shell (`ksh`).
- When a user executes a command, the profile shell:
  1. Searches the role's rights profiles and associated rights
  2. Uses the first matching entry if the same command appears in more than one profile
  3. Executes the command with the attributes specified in the RBAC configuration files

Implementing the User Privileges and Roles Plan

In the next section, you learn how Oracle Solaris 11:

- Supports process rights management
- Supports user rights management to grant appropriate privileges to users

Quiz

Oracle Solaris implements process rights management through privileges.

a. True
b. False

Quiz

Which letter indicates a set of privileges being used during a process execution?

a. E

b. I

c. P

d. L

Quiz

Which of the following rights databases contains rights profiles?

a. user_attr

b. auth_attr

c. exec_attr

d. prof_attr

Quiz

A profile shell is a special type of shell that enables access to the privileged rights that are assigned to the rights profile.

a. True
b. False

Agenda

- Introducing privileges, rights, and roles
- Managing privileges
- Managing users rights
- Verifying file integrity by using BART
- Monitoring the audit service
- Assessing the compliance of an Oracle Solaris system

Configuring and Managing Privileges

This section covers the following topics:

- Examining process privileges
- Managing user privileges

Examining Process Privileges

You first cover the following topics:

- Determining the privileges available to the shell
- Determining the privileges available to a process
- Displaying the description of a privilege

Determining the Privileges Available to the Shell

To determine which privileges are available to your processes, run `ppriv $$` to list the process privileges that are available to your shell.

```
# ps
  PID TTY      TIME CMD
  990 pts/4    0:01 bash
  991 pts/4    0:00 su
  993 pts/4    0:00 ps
# ppriv $$
990:  -bash
flags = <none>
    E: all
    I: basic
    P: all
    L: all
```

Determining the Process Privileges Available to a Shell

To display the names of the privileges in each privilege set, use `ppriv -v $$`.

```
# ppriv -v $$
990:bash
flags = <none>
     E: contract_event,contract_identity,contract_observer,cpc_cpu,dtrace_kernel,
     dtrace_proc,dtrace_user,file_chown,file_chown_self,file_dac_execute,
<output omitted>
     I: file_link_any,file_read,file_write,net_access,proc_exec,proc_fork,
     proc_info, proc_session
     P: contract_event,contract_identity,contract_observer,cpc_cpu,dtrace_kernel,
     dtrace_proc,dtrace_user,file_chown,file_chown_self,file_dac_execute,
<output omitted>
     L: contract_event,contract_identity,contract_observer,cpc_cpu,dtrace_kernel,
     dtrace_proc,dtrace_user,file_chown,file_chown_self,file_dac_execute,
<output omitted>
```

Determining the Privileges Available to a Process

To determine which privileges are available to a process, use `ppriv -v pid`.

```
# ppriv -v 476
476:  /usr/sbin/cron
flags = <none>
 E: contract_event,contract_identity,contract_observer,cpc_cpu,
    dtrace_kernel,dtrace_proc,dtrace_user,file_chown,
<output omitted>
 I: file_link_any,file_read,file_write,net_access,proc_exec,
    proc_fork,proc_info,proc_session
 P: contract_event,contract_identity,contract_observer,cpc_cpu,
    dtrace_kernel,dtrace_proc,dtrace_user,file_chown,
<output omitted>
 L: contract_event,contract_identity,contract_observer,cpc_cpu,
    dtrace_kernel,dtrace_proc,dtrace_user,file_chown,
<output omitted>
```

Displaying the Description of a Privilege

To display a privilege definition, use `ppriv -vl privilege`.

```
# ppriv -vl contract_event
contract_event
      Allows a process to request critical events without
      limitation.
      Allows a process to request reliable delivery of all
      events on any event queue.
# ppriv -vl proc_exec
proc_exec
      Allows a process to call execve().
```

Managing User Privileges

- Determining the privileges directly assigned to you
- Determining the privileged commands you can use
- Assigning privileges to a user or role
- Limiting privileges of a user or role
- Determining the privileges needed by a program by using the `ppriv` debugging command
- Using the `ppriv` debugging command to examine privilege use in a profile shell
- Using the `truss` command to examine privilege use in a regular shell

Determining the Privileges Directly Assigned to You

To view the privileges that have been directly assigned to your user account, use `ppriv -v $$`.

```
$ ppriv -v $$
990:  bash
flags = <none>
    E: file_link_any,proc_clock_highres,proc_session
    I: file_link_any,proc_clock_highres,proc_session
    P: file_link_any,proc_clock_highres,proc_session
    L: cpc_cpu,dtrace_kernel,dtrace_proc,dtrace_user,sys_time
$ ppriv -vl proc_clock_highres
    Allows a process to use high resolution timers.
```

Determining the Privileged Commands That You Can Use

To determine which rights profiles you have been assigned, use `profiles`.

```
$ profiles
All
Basic Solaris User
$ profiles -l
All
          *
Basic Solaris User
     auths=solaris.mail.mailq,solaris.network.autoconf.read,solaris.admin.wusb.read
     profiles=All
       /usr/bin/cdrecord.bin
privs=file_dac_read,sys_devices,proc_lock_memory,proc_priocntl,net_privaddr
       /usr/bin/readcd.bin        privs=file_dac_read,sys_devices,net_privaddr
       /usr/bin/cdda2wav.bin
privs=file_dac_read,sys_devices,proc_priocntl,net_privaddr
```

Assigning Privileges to a User or Role

To assign privileges to a user, use `usermod -K key=value loginname`.

```
# usermod -K defaultpriv=basic,proc_clock_highres jjones
# getent user_attr | grep jjones
jjones::::defaultpriv=basic,proc_clock_highres
```

To assign privileges to a role, use `rolemod -K key=value rolename`.

```
# rolemod -K defaultpriv=basic,proc_clock_highres realtime
# getent user_attr | grep realtime
realtime::::defaultpriv=proc_clock_highres
```

Limiting Privileges of a User or Role

1. Determine the privileges in a user's (or role's) basic set and limit set.
2. Remove one of the privileges from the basic set or from the limit set.
3. Test that the user (or role) can still perform other assigned functions as required.

```
# usermod -K limitpriv=all,!sys_linkdir jjones
# getent user_attr | grep jjones
jjones::::defaultpriv=basic;limitpriv=all,!sys_linkdir
```

```
# rolemod -K limitpriv=all,!sys_linkdir realtime
# getent user_attr | grep realtime
realtime::::defaultpriv=basic;limitpriv=all,!sys_linkdir
```

Determining Privileges Needed by a Program by Using the `ppriv` Debugging Command

1. Enter the command that is failing as an argument to the `ppriv` debugging command.

```
$ ppriv -De touch /etc/acct/yearly
touch[1298]: missing privilege "file_dac_write"
   (euid = 60004, syscall = “openat64”) for “/etc/acct/yearly
    at zfs_zaccess+0x245
touch: cannot create /etc/acct/yearly: Permission denied
```

2. If only the `syscall` ID is displayed in the output, you can determine which system call is failing by finding the `syscall` ID in the `/etc/name_to_sysnum` file.

Using the ppriv Debugging Command to Examine Privilege Use in a Profile Shell

```
jjones:~$ ls -l useful.script
-rw-r--r-- 1 aloe staff 2303 Dec 15 10:10 useful.script
jjones:~$ chown objadmin useful.script
chown: useful.script: Not owner
jjones:~$ ppriv -eD chown objadmin useful.script
chown[11444]: missing privilege "file_chown"
      (euid = 130, syscall = 16) needed at zfs_zaccess+0x258
chown: useful.script: Not owner
```

Using the `truss` Command to Examine Privilege Use in a Regular Shell

```
$ truss touch /etc/acct/yearly
...
...
(output truncated)

open64("/etc/acct/yearly", O_WRONLY|O_CREAT|O_TRUNC, 0666) Err#13 EACCES
[file_dac_write]
open("/usr/lib/locale/en_US.UTF-8/LC_MESSAGES/SUNW_OST_OSCMD.mo",
O_RDONLY) Err#2 ENOENT
open("/usr/lib/locale/en_US.UTF-8/LC_MESSAGES/SUNW_OST_OSLIB.mo",
O_RDONLY) Err#2 ENOENT
fstat64(2, 0xF5165980)                                  = 0
touchwrite(2, " t o u c h", 5)                         = 5
: cannot create write(2, " :   c a n n o t   c r e".., 16)  = 16
/etc/acct/yearlywrite(2, " / e t c / a c c t / y e".., 16)  = 16
: write(2, " :  ", 2)                                  = 2
Permission deniedwrite(2, " P e r m i s s i o n   d".., 17) = 17

write(2, "\n", 1)                              = 1
_exit(1)
```

Practice 8-1 Overview: Delegating Privileges to Users and Processes

This practice covers the following topics:

- Examining process privileges
- Managing user privileges

Agenda

- Introducing privileges, rights, and roles
- Managing privileges
- Managing user rights
- Verifying file integrity by using BART
- Monitoring the audit service
- Assessing the compliance of an Oracle Solaris system

Configuring and Using Role-Based Access Control RBAC

This section covers the following topics:

- Creating a role
- Creating, cloning, or changing a rights profile
- Assigning a rights profile to a role
- Assigning a role to a user
- Assuming a role
- Restricting an administrator to explicitly assigned rights
- Assigning a rights profile to a user
- Delegating authorization to a user
- Assigning authorization to a role
- Modifying a systemwide RBAC policy

Creating a Role

To create a role, use `roleadd -m -d dir rolename`.

```
# roleadd -u 3000 -g 10 -m -d /export/home/level1 -c "Level 1 Support“ \ -P
"Printer Management,Media Backup,Media Restore" level1
64 blocks
# passwd level1
New Password: <Type role password>
Re-enter new Password: <Type role password>
passwd: password successfully changed for level1
# getent passwd | grep level1
level1:x:3000:10:Level 1 Support:/export/home/level1:/usr/bin/pfbash
# grep level1 /etc/shadow
level1:$5$3jauLOtl$YDVdoH6q03m3YrOGZloql/MSrVaw0U7UgdNbiYEVbj8:16043::::::416
# getent user_attr | grep level1
level1::::profiles=Printer Management,Media Backup,Media
Restore;roleauth=role
```

Creating a Rights Profile

1. Create a rights profile.
2. Set the profile properties:
   - Use the `set` subcommand for profile properties that have a single value, such as `set desc`.
   - Use the `add` subcommand for properties that have more than one value, such as `add cmd`.

Creating a Rights Profile: Example

```
# profiles -p -S LDAP “New Users"
profiles:New Users> set desc="For all users of LDAP"
profiles:New Users> add profiles=“New Basic User"
profiles:New Users> set defaultpriv="basic,!proc_info"
profiles:New Users> set limitpriv="basic,!proc_info"
profiles:New Users> end
profiles:New Users> exit
#
# profiles -p “New Users"
Found profile in LDAP repository.
profiles:New Users> info
        name=New Users
        desc=For all users of LDAP
        defaultpriv=basic,!proc_info,
        limitpriv=basic,!proc_info,
        profiles=New Basic User
```

Cloning and Modifying a Rights Profile

1. Create a new rights profile from an existing profile.

```
# profiles -p [-S repository] existing-profile-name
```

   - To enhance an existing rights profile:
     a. Create a new profile.
     b. Add the existing rights profile as a supplementary rights profile.
     c. Add the enhancements.
   - To remove content from an existing rights profile, clone the profile, rename it, and then modify it.
2. Continue to modify the new rights profile by adding or removing supplementary rights profiles, authorizations, and other security attributes.

Cloning and Modifying a Rights Profile: Example

```
# profiles -p "Network IPsec Management"
profiles:Network IPsec Management> add auths="solaris.admin.edit/etc/hosts"
Cannot add. Profile cannot be modified
#
# profiles -p "Total IPsec Mgt"
Total IPsec Mgt> set desc="Network IPsec Mgt plus edit authorization"
Total IPsec Mgt> add profiles="Network IPsec Management"
Total IPsec Mgt> add auths="solaris.admin.edit/etc/hosts"
Total IPsec Mgt> add auths="solaris.admin.edit/etc/inet/ipsecinit.conf"
Total IPsec Mgt> add auths="solaris.admin.edit/etc/inet/ike/config"
Total IPsec Mgt> add auths="solaris.admin.edit/etc/inet/secret/ipseckeys"
Total IPsec Mgt> exit
#
# profiles -p "Total IPsec Mgt" info
        name=Total IPsec Mgt
        desc=Network IPsec Mgt plus edit authorization
        auths=solaris.admin.edit/etc/hosts,
              solaris.admin.edit/etc/inet/ipsecinit.conf,
              solaris.admin.edit/etc/inet/ike/config,
              solaris.admin.edit/etc/inet/secret/ipseckeys
        profiles=Network IPsec Management
```

Assigning a Rights Profile to a Role

To assign a rights profile to a role, use `rolemod [-P profile][-s shell] rolename`.

```
# rolemod -P profile1,profile2 -s /usr/bin/pfksh level1
```

Assigning a Role to a User

1. Assign the role to the user by using `usermod -u uid -g gid -m -d dir -R role -c comment loginname`.
2. Assign a password to the role by using `passwd rolename`.
3. Verify that an entry has been made in the `user_attr` database.

Assigning a Role to a User: Example

```
# useradd -u 4009 -g 10 -m -d /export/home/paul \
-R level1 -c “Paul” paul
64 blocks
# passwd paul
New Password: <Type rolename password>
Re-enter new Password: <Type rolename password>
passwd: password successfully changed for paul
# getent user_attr | grep paul
paul::::roles=level1
# roles paul
level1
# usermod -R level1 paul
# passwd -r repository level1
Password: <Type rolename password>
Confirm Password: <Retype rolename password>
# usermod -R "" paul
```

Assuming a Role

1. In a terminal window, determine which roles you can assume by using `roles`.
2. Use the `su` command to assume a role by using `su - rolename`.
3. Verify that you are now in a role by using `/usr/ucb/whoami`.
4. View the capabilities of your role by using `ppriv $$`.

```
# roles
sysadmin,oper,primaryadm
# su - sysadmin
Password: <Type sysadmin password>
$ /usr/ucb/whoami
sysadmin
$ ppriv $$
950:   bash
flags = <none>
    E: basic
    I: basic
    P: basic
    L: all
```

Restricting the Superuser

It is your responsibility to control and monitor system activity by performing the following tasks:

- Setting limits on who can use various resources
- Logging resource use
- Monitoring who is using the resources

Note: The system tracks real and effective user and group ID logins. To determine the real UID, use `who am i`. To determine the effective UID, user `whoami`.

Restricting an Administrator to Explicitly Assigned Rights

You can restrict a role or user to a limited number of administrative actions in either of the following ways:

- Use the Stop rights profile.
- Modify the `policy.conf` file on a system and require the role or user to use that system for administrative tasks.

```
# rolemod -P "Profile_Name,All,Stop" rolename
```

Assigning the Rights Profile to a User

```
# profiles chris
Basic Solaris User
All
# usermod -P "Printer Management" chris
# profiles chris
chris:
Printer Management
Basic Solaris User
All
# getent user_attr | grep chris
chris::::profiles=Printer Management
# profiles -l chris
Printer Management:
/etc/init.d/lp euid=0, uid=0
/usr/bin/cancel euid=lp, uid=lp
/usr/bin/lpset egid=14
/usr/bin/lpstat euid=0
/usr/lib/lp/local/accept uid=lp
/usr/lib/lp/local/lpadmin uid=lp, gid=8
/usr/lib/lp/lpsched uid=0
<output omitted>
All:
*
```

Delegating an Authorization to a User

1. Delegate the authorization to the user by using `usermod -A` *`authorization loginname`*.
2. Verify that an entry has been made in the `user_attr` database for the user.
3. View the authorizations for the user by using the `auths` command.

Delegating an Authorization to a User: Example

```
# su - chris
Oracle Corporation     SunOS 5.11  11.3     September 2015
chris:~$ crontab -l root
crontab: you must be super-user to access another user's crontab file
chris:~$ exit
# usermod -A solaris.jobs.admin chris
# getent user_attr | grep chris
chris::::auths=solaris.jobs.admin;profiles=Printer Management
# auths chris
solaris.admin.wusb.read,solaris.jobs.admin,solaris.mail.mailq,solaris.network.autoc
onf.read,solaris.print.*
# su - chris
Oracle Corporation     SunOS 5.11  11.3     September 2015
chris:~$ crontab -l root
#ident  "%Z%%M%   %I% %E% SMI"
#
# The root crontab should be used to perform accounting data collection.
(output omitted)
chris:~$ exit
```

Assigning Authorization to a Role

1. Assign the authorization to a role by using `rolemod -A "authorization" rolename`.
2. Verify that an entry has been made in the `user_attr` database for the role.
3. View the authorizations for the role by using the `auths` command.

```
# rolemod -A "solaris.admin.usermgr.*" level1
# auths level1
solaris.admin.usermgr.*,solaris.admin.wusb.read,solaris.mail.ma
ilq,solaris.media.extract,solaris.network.autoconf.read,solaris
.print.*,solaris.smf.manage.ndmp,solaris.smf.read.ndmp,solaris.
smf.value.ndmp
```

Modifying a Systemwide RBAC Policy

1. Determine the privileges that you want to comment out for the basic user.
2. Using a text editor, modify the `PRIV_DEFAULT=basic` default entry and restart the system.
3. As a user, test the modification.

```
# pfedit /etc/security/policy.conf
# grep PRIV_DEFAULT /etc/security/policy.conf
# There are two different settings; PRIV_DEFAULT determines the default
# Similarly, PRIV_DEFAULT=basic,!file_link_any takes away only the
PRIV_DEFAULT=basic,!proc_info,!proc_session
# init 6
<log in to the system>
# su - jjones
Oracle Corporation    SunOS 5.11     11.3    September 2015
$ ps -A -o user -o pid -o comm | grep jjones
   USER  PID COMMAND
 jjones 1935 -bash
 jjones 1941 grep
 jjones 1942 ps
```

Practice 8-2 and Practice 8-3: Overview

Practice 8-2 covers the following topics:

- Managing roles and profiles
- Configuring a rights profile
- Working with individual authorizations
- Creating a systemwide RBAC policy

Practice 8-3 covers monitoring and restricting the superuser.

Agenda

- Implementing privileges, rights, and roles
- Managing privileges
- Managing user rights
- Verifying file integrity by using BART
- Monitoring the audit service
- Assessing the compliance of an Oracle Solaris system

BART: Overview

BART is a file verification and reporting tool that:

- Performs a file-level check of the software contents of a system
- Enables you to determine what file-level changes have occurred on a system
- Compares changes to a known baseline

BART: Example

```
# vi bartrules
IGNORE all
/export/home/oracle1
CHECK all

# bart create -r bartrules > bart-`hostname`-`date '+%d%m%Y-%H:%M:%S'`

# ls bart*
bart-s11-server1-21092015-01:11:08  bartrules

# touch /export/home/oracle1/newfile

# bart create -r bartrules > bart-`hostname`-`date '+%d%m%Y-%H:%M:%S'`

# ls bart*
bart-s11-server1-21092015-01:11:08  bartrules
bart-s11-server1-21092015-01:11:50

# bart compare -r bartrules bart-s11-server1-21092015-01\:11\:08     \
bart-s11-servr1-21092015-01\:11\:50
/export/home/oracle1:
  size  control:5  test:6
  dirmtime  control:55f55ee4  test:55ffc9c6
/export/home/oracle1/newfile:
  add
```

BART: Example

```
# vi /export/home/oracle1/newfile
This is a test.

# bart create -r bartrules > bart-`hostname`-`date '+%d%m%Y-%H:%M:%S'`

# ls bart*
bart-s11-server1-21092015-01:11:08  bart-s11-server1-21092015-01:15:42
bart-s11-server1-21092015-01:11:50  bartrules

# bart compare -r bartrules bart-s11-server1-21092015-01\:11\:50     \
bart-s11-servr1-21092015-01\:15\:42
/export/home/oracle1:
  dirmtime  control:55ffc9c6  test:55ffcaa9
/export/home/oracle1/newfile:
  size  control:0  test:16
  mtime  control:55ffc9c6  test:55ffcaa9
  contents  control:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
test:11586d2eb43b73e539caa3d158c883336c0e2c904b309c0c5ffe2c9b83d562a1
```

Practice 8: Overview

- 8-4: Verifying file integrity by using BART

Agenda

- Implementing privileges, rights, and roles
- Managing privileges
- Managing user rights
- Verifying file integrity by using BART
- Monitoring the audit service
- Assessing the compliance of an Oracle Solaris system

Auditing in Oracle Solaris

Auditing is the process of collecting data about the use of system resources. Auditing in Oracle Solaris 11:

- Is enabled by default
- Records security-related system events
- Records events in a network-wide audit trail
- Detects misuse or unauthorized activity
- Reviews patterns of access and the access histories of users and objects
- Discovers attempts to bypass protection mechanisms
- Discovers extended use of privileges

Oracle Solaris 11 Auditing Features

Following are the enhancements in the Oracle Solaris Auditing feature in Oracle Solaris 11:

- Oracle Solaris Auditing and Device Allocation replace “BSM.”
- `bsmconv` and `bsmunconv` commands, and the need to restart have been removed.
- `bsmrecord` is renamed `auditrecord`.
- `/etc/security/audit` directory and symlink no longer exist. `/var/audit` is now the official audit directory.
- By default, Oracle Solaris Auditing is configured and enabled. `lo` events are the default configured events.
- By using the Audit Configuration rights profile, all global configurations can be performed by `auditconfig`. `audit_startup` and `audit_control` no longer exist.

Oracle Solaris 11 Auditing Features

- By using the `useradd`, `roleadd`, `usermod`, and `rolemod` commands, you can place the per-user `audit_flags` security attribute in the `user_attr` database.
- You can use the `profiles` command to place the desired per-user audit flags for the rights profiles in the `prof_attr` database.
- By using the Audit Control rights profile, you can control the Oracle Solaris audit service with `audit` to refresh, change audit files, disable, and re-enable.
- Global zone auditing need not be enabled for per-zone auditing.

Default Configuration of the Audit Service

- The audit service has a default configuration and is immediately operational on the global zone after you install Oracle Solaris 11.
- No additional action is required to enable or configure the service to become usable.
- With its default configuration, the audit service records the following operations:
  - Login and logout operations
  - Use of the su command
  - Screen lock and screen unlock operations

Displaying Audit Service Defaults

- Displaying the default audit policy

```
# auditconfig -getpolicy
configured audit policies = cnt
active audit policies = cnt
```

- Displaying default class for attributable events

```
# auditconfig -getflags
active user default audit flags = lo(0x1000,0x1000)
configured user default audit flags = lo(0x1000,0x1000)
```

- Displaying the default class for non-attributable events

```
# auditconfig -getnaflags
active non-attributable audit flags = lo(0x1000,0x1000)
configured non-attributable audit flags = lo(0x1000,0x1000)
```

Displaying Audit Service Defaults

- Displaying the default audit plugins

```
# auditconfig –getplugin
Plugin: audit_binfile
Attributes: p_age=Oh;p_dir=/var/audit;p_fsize=4M;p_minfree=1;

Plugin: audit_syslog (inactive)
Attributes: p_flags=;

Plugin: audit_remote (inactive)
Attributes: p_hosts=;p_retries=3;p_timeout=5;
```

- Displaying the audit queue controls

```
# auditconfig -getqctrl
no configured audit queue hiwater mark
no configured audit queue lowater mark
no configured audit queue buffer size
no configured audit queue delay
active audit queue hiwater mark (records) = 100
active audit queue lowater mark (records) = 10
active audit queue buffer size (bytes) = 8192
active audit queue delay (ticks) = 20
```

Enabling and Disabling the Audit Service

- Use the `audit -s` command to enable the audit service if it is not running or to refresh the service if it is currently running.

  Note: Auditing is enabled by default.

```
# audit -s
```

- Verify that auditing is enabled.

```
# auditconfig -getcond
audit condition = auditing
```

- Use the `audit –t` command to disable the service.

```
# audit -t
```

Viewing Contents of Binary Audit Files

```
# cd /var/audit
# ls
...
20150901132110.20150901190523.s11-server1  20150915172913.20150915181644.s11-server1
20150906173827.20150906180719.s11-server1  20150915181840.20150921092344.s11-server1
20150913113111.20150913113321.s11-server1  20150921092353.not_terminated.s11-server1

# praudit 20150915181840.20150921092344.s11-server1 more
file,2015-09-15 10:18:40.951 -08:00,
header,52,2,system booted,na,s11-server1,2015-09-15 10:17:20.708 -08:00
text,booting kernel
header,42,2,init(1m),na,s11-server1,2015-09-15 10:18:51.640 -08:00
text,booted
return,success,0
header,32,2,su,na,s11-server1,2015-09-15 10:19:01.833 -08:00
return,success,0
header,32,2,su logout,na,s11-server1,2015-09-15 10:19:04.667 -08:00
return,success,0
header,69,2,login - ssh,,s11-server1,2015-09-15 10:21:53.234 -08:00
subject,oracle1,oracle1,staff,oracle1,staff,1059,1249775782,3696 136704 s11-desktop
return,success,0
header,69,2,role login,,s11-server1,2015-09-15 10:21:56.902 -08:00
subject,oracle1,root,root,root,root,1064,1249775782,3696 136704 s11-desktop
return,success,0
...
```

Agenda

- Implementing privileges, rights, and roles
- Managing privileges
- Managing user rights
- Verifying file integrity by using BART
- Monitoring the audit service
- Assessing the compliance of an Oracle Solaris system

Compliance in Oracle Solaris OS

- The Oracle Solaris 11 OS provides the `compliance` command to assess system compliance and generate reports against the following security benchmarks:
  - Solaris security policy benchmark
  - Payment Card Industry Data Security Standard (PCI DSS) security policy benchmark
- The compliance report indicates which tests failed and which tests passed, and provides remediation steps.
- You must examine the compliance report and then perform additional tasks to comply with the standard.
- Oracle Solaris also supports creating tailorings from existing security benchmarks.

Solaris Security Policy Benchmark

- The Solaris security policy benchmark is a standard based on the “secure by default” (SBD) installation of Oracle Solaris.
- The benchmark provides two profiles:
  - `Baseline`: Matches closely with the default SBD installation of Oracle Solaris
  - `Recommended`: Satisfies organizations with stricter security requirements than the `Baseline` profile

PCI DSS Security Policy Benchmark

- The PCI DSS security policy benchmark is a security standard for organizations that handle cardholder information for major debit and credit cards.
- The standard is defined by the PCI Security Standards Council to offer robust and secure payments by cards.
- The key intent of the PCI DSS security benchmark is to reduce credit card fraud.

Assessing the Security Compliance of an Oracle Solaris System

1. Install the `compliance` package.

```
# pkg install compliance
```

2. List the benchmarks and profiles that are available.

```
# compliance list –p
```

3. Create an assessment.

```
# compliance assess -p Solaris_PCI-DSS -b pci-dss -a pci-dss
```

4. List the assessments in the compliance directory.

```
# compliance list -v -a pci-dss
```

5. Locate the assessments in the compliance directory.

```
# compliance report -a pci-dss
# compliance report -f log -a pci-dss
# compliance report -f xccdf -a pci-dss
```

6. View the assessments.
7. Fix any failures that your security policy requires to pass.

Tailorings for Security Benchmarks

Tailorings customize the benchmark assessments to verify the security policy of a particular system.

- To create customized assessments, you include or exclude rules from an existing benchmark, profile, or tailoring.
- To use a tailoring to assess systems, you must install the source benchmark as well as the tailoring.

Creating Tailorings for Security Benchmarks

1. Open the compliance editor to create a tailoring.

```
# compliance tailor –t baselinecustom
*** compliance tailor: No existing tailoring 'baselinecustom', initializing
tailoring:baselinecustom> info
Properties:
        tailoring=baselinecustom
        benchmark: not set
        profile: not set
```

2. Set the benchmark and exclude all rules.

```
tailoring:baselinecustom> set benchmark=solaris
tailoring:baselinecustom> set profile=Baseline
tailoring:baselinecustom> info
Properties:
        tailoring=baselinecustom
        benchmark=solaris
        profile=Baseline
tailoring:baselinecustom> exclude -a
Discard existing rule selections (y/N)? y
```

Creating Tailorings for Security Benchmarks

3. Open the pick screen, navigate, and include particular rules.

```
tailoring:baselinecustom> pick
...
> _   OSC-53005  The OS version is current
> _   OSC-16005  All local filesystems are ZFS
> _   OSC-61510  root login by using ssh(1) is disabled
> _   OSC-46014  Passwords require at least 14 characters
> _   OSC-59000  root is a role
> _   OSC-04511  Booting the system should require a password
...
```

4. Commit the changes and exit the compliance editor.

```
tailoring:baselinecustom> commit
tailoring:baselinecustom> exit
```

5. List the tailoring.

```
# compliance tailor list
        baselinecustom
```

Creating Tailorings for Security Benchmarks

6. Test the tailoring on a system and evaluate the output.

```
# compliance assess -t baselinecustom
```

7. Locate the assessment.

```
# compliance report
/var/share/compliance/assessments/baselinecustom.2015-09-11,12:04/report.html
```

8. Display the assessment in a web browser.

Quiz

BART is a tool that performs a file-level check of the software contents of a system and enables you to determine what file-level changes have occurred on a system.

a. True
b. False

Practice 8: Overview

- 8-5: Assessing the compliance of an Oracle Solaris System

Summary

In this lesson, you should have learned how to:

- Describe privileges, rights, and roles
- Manage privileges
- Manage user rights
- Verify file integrity by using BART
- Monitor the audit service
- Assess the compliance of an Oracle Solaris system