Oracle Solaris 11 Advanced System Administration

Student Guide - Volume II

D72965GC30

Edition 3.0

March 2013

D81024

ORACLE®

Author

Vijetha M Malkai

**Technical Contributors
and Reviewers**

Tammy Shannon

Anies Rahman

Rosemary Martinak

Editors

Malavika Jinka

Aju Kumar

Smita Kommini

Graphic Designer

Seema Bopaiah

Publishers

Jayanthy Keshavamurthy

Veena Narasimhan

Contents

v

ix

7   Managing Services and Service Properties

8   Configuring Privileges and Role-Based Access Control

12   Monitoring and Troubleshooting Software Failures

7

Managing Services and Service Properties

Objectives

After completing this lesson, you should be able to:

- Implement a plan to configure services
- Configure SMF services
- Recover a service from a snapshot
- Troubleshoot SMF services

Workflow Orientation

Before you begin the lesson, take a moment to orient yourself in your job workflow. You have successfully installed the operating system and have updated it. You have configured the data storage environment as well as the physical and virtual networks. In this lesson you manage the SMF services and the service properties. As a system administrator, it is your responsibility to ensure that the system and business processes that are running on the system continue uninterrupted. To do this, you need to know which services are controlling which functions so that you can take down or bring up a service as required.

Lesson Agenda

- **Planning Services Configuration**
- Configuring SMF Services
- Troubleshooting SMF Services

ORACLE

Planning for Services Configuration

Services configuration planning ensures that:

- The right services are enabled and running
- Existing services can be easily modified
- Downed services can be recovered and restored quickly
- New services can be created to meet emerging business needs
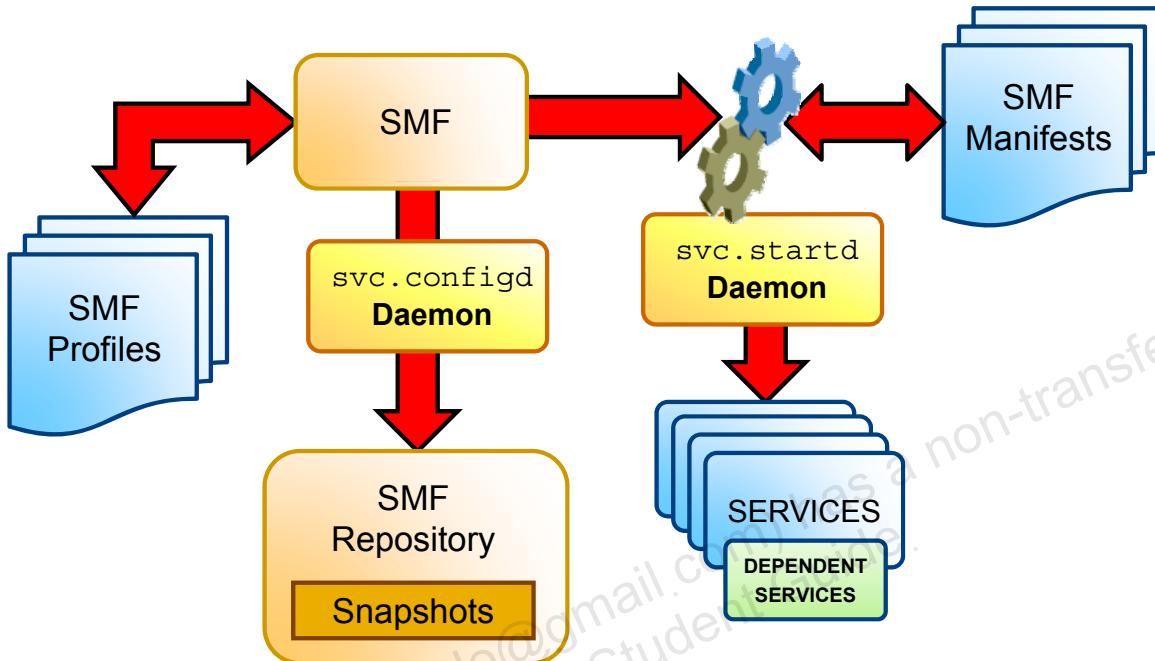
Your company recognizes the importance of ensuring that the right services are enabled and running on the system and that these services can be easily and quickly modified, recovered, and restored. Moreover, the company is interested in being able to have new services created and supported by the SMF to meet emerging business needs.

In this section, you are introduced to the more advanced features of the SMF: manifests, profiles, the service configuration repository, and repository backups using snapshots. You are also introduced to service script creation.

SMF Advanced Features

When a system is booted, the SMF consults the SMF profiles to determine which services should be enabled. The SMF then starts the `svc.startd` daemon, which in turn consults the SMF manifests to gather property and instance information about each service before starting each service and its associated dependents. The SMF uses the Service Configuration Repository (also known as the SMF Repository) to store state and configuration information about each service instance in addition to per-service snapshots that are taken at the time each service is successfully started and used as backups. The SMF repository is managed by the `svc.configd` daemon.

You are to look at each feature in more detail next, beginning with the SMF profiles.

SMF Profiles

- An SMF profile is an XML file that allows customization of services and instances delivered by the system.
- Profiles delivered with the operating system include:
  - `/etc/svc/profile/generic_open.xml`: Enables standard services
  - `/etc/svc/profile/generic_limited_net.xml`: Disables many Internet services
  - `/etc/svc/profile/ns_*.xml`: Enables services associated with the name service that is configured to run on the system
  - `/etc/svc/profile/platform_*.xml`: Enables services associated with particular hardware platforms

An SMF profile is an XML file that allows customization of services and instances delivered by the system. Profiles are available for configuration customization using a file rather than a set of scripts, or to customize configuration at deployment or installation time. All configurations may be customized by using a profile, including adding instances for system-supplied services.

Some profiles that are delivered with the operating system release include:

- `/etc/svc/profile/generic_open.xml`: This profile enables the standard services that have been started by default in earlier releases.
- `/etc/svc/profile/generic_limited_net.xml`: This profile disables many of the Internet services that have been started by default in earlier releases. The `network/ssh` service is enabled to provide network connectivity.
- `/etc/svc/profile/ns_*.xml`: This profile enables services associated with the name service that is configured to run on the system.
- `/etc/svc/profile/platform_*.xml`: This profile enables services associated with particular hardware platforms.

SMF Profile: Example

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM
     '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<!--
<header content omitted>
<service_bundle type='profile' name='generic_open'
 xmlns:xi='http://www.w3.org/2003/XInclude' >
  <!--
      Include name service profile, as set by system id tools.
  -->
  <xi:include href='file:/etc/svc/profile/name_service.xml' />

  <!--
      svc.startd(1M) services
  -->
  <service name='system/coreadm' version='1' type='service'>
    <instance name='default' enabled='true'/>
  </service>
  <service name='system/cron' version='1' type='service'>
    <instance name='default' enabled='true'/>
  </service>
```

This example presents an excerpt from the `/etc/svc/profile/generic_open.xml` file. As was discussed, this profile enables the standard services that have been started by default in earlier releases. Each service is listed in the same basic format:

```
<service name='system/coreadm' version='1' type='service'>
    <instance name='default' enabled='true'/>
</service>
```

You learn how to create and apply your own profile in the next topic.

When SMF Profiles Are Applied

- `/etc/svc/profile/generic.xml` profile:
  - Applied during the first boot after a new installation or an upgrade
  - Symbolically linked to `generic_open.xml` or `generic_limited_net.xml`
- The contents of `site.xml` in `/etc/svc/profile`:
  - Applied during first boot
  - Added between boots
- Profiles in `/etc/svc/profile` are applied during early manifest import.
- Profiles in `/var/svc/profile` are applied during later manifest import.

During the first boot after a new installation or an upgrade, the `/etc/svc/profile/generic.xml` profile is applied. This file is usually symbolically linked to `generic_open.xml` or `generic_limited_net.xml`. Also, if a profile called `site.xml` is in `/etc/svc/profile` during the first boot or is added between boots, the contents of this profile are applied.

Note: By using the `site.xml` profile, the initial set of enabled services may be customized by the administrator.

Similar to manifests, profiles in `/etc/svc/profile` are applied during the early manifest import. Profiles in `/var/svc/profile` are applied during the later manifest import.

Note: The `generic_xxx` profiles are mutually exclusive. Any conflicting definitions between files in `/etc/svc/profile/site` are treated as conflicts, and the affected service instances are put into the maintenance state.

SMF Manifests

- An SMF manifest is an XML file that describes a service and a set of instances.
- Manifests are imported to load the properties of that service and its instances into the repository.
- The preferred location for manifests is `/lib/svc/manifest`.
- Manifests are imported and upgraded during the boot process before any services start.
- Site subdirectory is reserved for site-specific use.
- Manifests in the site directory can be modified directly.

An SMF manifest is an XML file that describes a service and a set of instances. Manifests are imported to load the properties of that service and its instances into the repository.

The preferred location for manifests is `/lib/svc/manifest`. Manifests stored there will be imported and upgraded during the boot process before any services start. Running the import process early ensures that the repository will contain information from the latest manifests before the services are started. At other times, you can import information from these manifests by running this command: `svcadm restart manifest-import`. `/var/svc/manifest` remains available for compatibility purposes, but manifests located there will not be imported or upgraded until the `svc:/system/manifest-import:default` service runs, which is significantly later in the boot process.

The site subdirectory of `/lib/svc/manifest` and `/var/svc/manifest` is reserved for site-specific use. Manifests in the site directory may be modified directly. Other manifests included in the software release should not be modified because those modifications will be lost during software upgrades. If you need to make changes to the set of properties included in the generic manifests, you should either create a profile or use the `svccfg` command. You learn how to create a profile in the next topic.

With the introduction of `svcbundle` in Oracle Solaris 11.1, the creation of manifests and profiles is easier. `svcbundle` enables you to take advantage of the benefits of automatic application restart without requiring you to have full knowledge of the XML file format that is used when integrating with the Service Management Facility (SMF). You can use the `svcbundle` command to generate SMF manifests and get the manifest validated using the `svccfg` command. The `svcbundle` command allows you to create and, optionally, install a manifest or system profile. For more information, refer to
http://www.oracle.com/technetwork/articles/servers-storage-admin/howto-svcbundle-manifest-profile-1866525.html, http://docs.oracle.com/cd/E26502_01/html/E29003/eqbrs.html#smft-5
and `svcbundle` (1M).

SMF Manifest: Example

```xml
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM
    "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<!--
<header and copyright content omitted>
<service_bundle type='manifest' name='SUNWcsr:rbac'>

<service
name='system/rbac'
type='service'
version='1'>

<create_default_instance enabled='true' />

<single_instance />

---
---
```

This example presents an excerpt from `rbac.xml manifest`. A manifest file consists of the following basic entries:

- **`<service_bundle type`:** Identifies the name of the service. The type (manifest) indicates a simple service rather than a milestone, the package providing the service, and the service name.
- **`<service`:** Identifies service category, type, name, and version
- **`<create_default_instance`:** Creates the default instance
- **`<single_instance/>`:** Identifies whether multiple instances of the service will run
- **`<dependency`:** Identifies dependencies for this service
- **`<dependent`:** Identifies what service has this service as a dependent
- **`<exec_method`:** Defines how the service is started and stopped
- **`<property _group name`:** Identifies the service model to use
- **`<template>`:** Creates information to describe the service

The `rbac.xml` manifest is displayed as follows (minus the header and comment content) for you to familiarize yourself with a manifest's layout:

```
<service_bundle type='manifest' name='SUNWcsr:rbac'>

<service
    name='system/rbac'
    type='service'
    version='1'>

    <create_default_instance enabled='true' />

    <single_instance />

        <dependency
                name='usr'
                type='service'
                grouping='require_all'
                restart_on='none'>
                <service_fmri value='svc:/system/filesystem/minimal' />
        </dependency>

    <dependent
            name='manifest'
            grouping='optional_all'
            restart_on='none'>
                <service_fmri value='svc:/system/manifest-import' />
    </dependent>

    <dependent
            name='name-service-cache'
            grouping='optional_all'
            restart_on='none'>
            <service_fmri value='svc:/system/name-service-cache' />
    </dependent>
```

```
<exec_method
     type='method'
     name='start'
     exec='/lib/svc/method/svc-rbac start'
     timeout_seconds='300'>
</exec_method>

<exec_method
     type='method'
     name='refresh'
     exec='/lib/svc/method/svc-rbac refresh'
     timeout_seconds='300'>
</exec_method>

<exec_method
     type='method'
     name='stop'
     exec=':true'
     timeout_seconds='300'>
</exec_method>

<property_group name='startd' type='framework'>
     <propval name='duration' type='astring'
          value='transient' />
</property_group>

<property_group name='options' type='application'>
</property_group>

<stability value='Unstable' />
```

```
            <template>
                    <common_name>
                            <loctext xml:lang='C'>
                            Assemble the RBAC *attr files.
                            </loctext>
                    </common_name>
            </template>
    </service>

</service_bundle>
```

You create your own service manifest in Practice 7.

Service Configuration Repository

- Stores state and configuration information about each service instance
- Is located in `/etc/svc/repository.db`
- Is managed by the `svc.configd` daemon
- Provides a consistent and persistent way to enable or disable a service
- Provides a consistent view of service state
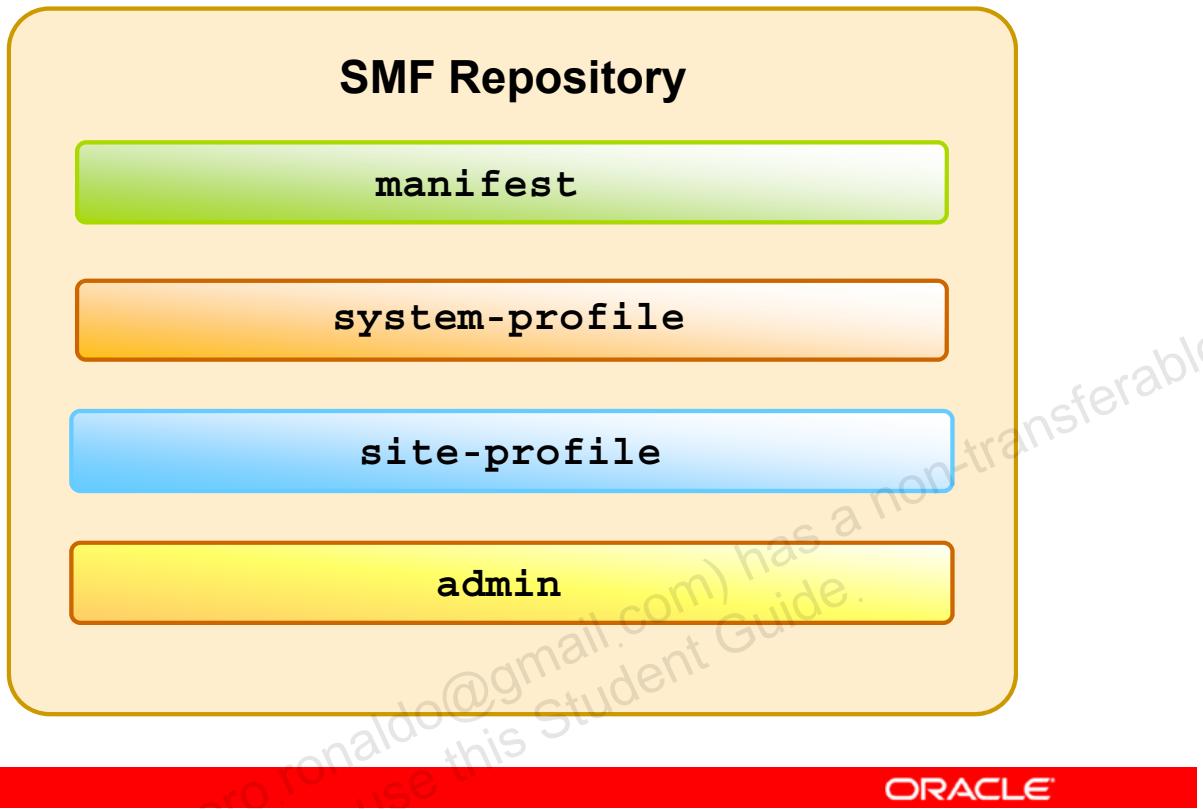
SMF stores state and configuration information about each service instance in the service configuration repository. The repository is distributed among local memory and local disk-based files and is stored in `/etc/svc/repository.db`.

The repository is managed by the `svc.configd` daemon. This daemon is the interface between the repository and the user and ensures that a consistent picture of the repository is presented to the user.

In turn, the repository provides a consistent and persistent way to enable or disable a service, as well as a consistent view of the service state. This capability helps you debug service configuration problems.

SMF Administrative Layers

SMF Repository

manifest

system-profile

site-profile

admin

The SMF repository consists of four layers that can be used to help determine which settings have been customized by an administrator and which settings are delivered by the software. The four layers are as follows:

- **manifest:** Imported full manifest files that completely define a service or an instance, that is located in a standard location: `/lib/svc/manifest` or `/var/svc/manifest`

- **system-profile:** Specifically named profiles (`/etc/svc/profile/generic.xml` or `/etc/svc/profile/platform.xml`) that are applied to the system and delivered by the Solaris consolidations

- **site-profile:** Profiles that are site specific and are either applied from the `/etc/svc/profile/site` directory or from the `/etc/svc/profile/site.xml` or `/var/svc/profile/site.xml` file

- **admin:** Administrative customizations to the system done with `svccfg add/set/del` subcommands as well as through enabling/disabling services through the command line. Manifests and profiles imported and applied from nonstandard locations (that is, outside of `/lib/svc/manifest` or `/var/svc/manifest`) are considered customizations and are brought in at the `admin` layer.

The layers are hierarchical, with the `admin` layer taking precedence. If a property has a value in the `admin` layer, that value will be used by the service. If not, the `site-profile` layer is consulted, and then the `system-profile` layer, and eventually the `manifest` layer. This behavior allows for local changes to take precedence over the default settings.

The system automatically manages these layers. Any direct changes that you as the system administrator make to the repository appear only in the `admin` layer. Other layers are changed only by placing or removing files in standard locations. When a property is put into the repository because of file contents, the information about that property includes the name of that file.

Note: You can use the `svccfg listprop` command to explore layers. You can use the `svccfg listcust` command only to list customizations.

Introducing SMF Repository Backups

- SMF automatically takes these backups:
  - Boot backup: Taken immediately before the first change to the repository is made during each system startup
  - `manifest_import` backups: Occur after `svc:/system/early-manifest-import:default` or `svc:/system/manifest-import:default` completes
- System maintains four copies of each type.
- Backups are stored as `/etc/svc/repository-type-YYYYMMDD_HHMMSWS` for the date and time when the backup was taken.
- Repository can be restored from these backups.

The SMF automatically takes the following backups of the repository:
- The boot backup is taken immediately before the first change to the repository is made during each system startup.
- The `manifest_import` backups occur after `svc:/system/early-manifest-import:default` or `svc:/system/manifest-import:default` completes, if the service imported any new manifests or ran any upgrade scripts.

Four backups of each type are maintained by the system. The system deletes the oldest backup when necessary. The backups are stored as `/etc/svc/repository-type-YYYYMMDD_HHMMSWS`, where `YYYYMMDD` (year, month, day) and `HHMMSS` (hour, minute, second) are the date and time when the backup was taken. Note that the hour format is based on a 24-hour clock.

You can restore the repository from these backups, if an error occurs. You learn how to do this later in the lesson.

Introducing SMF Repository Snapshots

- Snapshots are taken per service at the time when a service is successfully started.
- Standard snapshots include:
  - `initial`: Taken on the first import of the manifest
  - `running`: Used when the service methods are executed
  - `start`: Taken at the last successful start
- SMF service always executes with the `running` snapshot.
- Current property values for a service are incorporated into the `running` snapshot with the `svcadm refresh` command.
- Instance configurations can be viewed or reverted to in a previous snapshot by using the `svccfg` command.

The service configuration repository provides a per-service snapshot at the time each service is successfully started so that fallback is possible. The standard snapshots that are stored in the SMF repository are listed in the slide.

The SMF service always executes with the `running` snapshot. This snapshot is automatically created if it does not exist.

When you change the property values of a service, the changes are incorporated into the `running` snapshot when you execute the `svcadm refresh` command. You can use the `svccfg` command to view or revert to instance configurations in a previous snapshot. You learn how to revert to a previous snapshot later in this lesson.

Creating New Service Scripts

- Determine the process for starting and stopping your service.
- Establish a name for the service and the category that this service is in.
- Determine whether your service runs multiple instances.
- Identify any dependency relationships between this service and any other services.
- If a script is required to start and stop the process, create the script and place it in a local directory, such as `/usr/local/svc/method`.
- Create a service manifest file for your service.
- Incorporate the script into the SMF by using the `svccfg` utility.

You can create new scripts to start and stop additional processes or services to customize a system. For example, to eliminate the requirement for a manual start of a database server, you could create a script to start the database server automatically after the appropriate network services have started. You can then create another script to terminate this service and shut down the database server before the network services are stopped.

The procedure for service script creation is outlined in the steps in the slide. You learn how to complete these steps in the next topic.

Implementing the Services Administration Plan

Your assignment is to:

- Create a new service and incorporate it into the SMF
- Modify a service configuration
- Restore and recover a service

Quiz

The preferred location for manifests is `/lib/svc/manifest`.

   a. True

   b. False

Answer: a

Quiz

Which of the following profiles is used to enable standard services?

a. `/etc/svc/profile/generic_open.xml`

b. `/etc/svc/profile/generic_limited_net.xml`

c. `/etc/svc/profile/ns_*.xml`

d. `etc/svc/profile/platform_*.xml`

Answer: a

Quiz

Which daemon manages the service configuration repository?

a. `svc.ipfd`

b. `svc.configd`

c. `svc.startd`

Answer: b

Quiz

Which service configuration repository snapshot does the SMF service always execute with?

a. initial

b. running

c. start

Answer: b

Lesson Agenda

- Planning Services Configuration
- **Configuring SMF Services**
- Troubleshooting SMF Services

ORACLE

Configuring SMF Services

- Creating and exporting a service
- Modifying a service's manifest
- Changing an environment variable for a service
- Changing a property for an `inetd`-controlled service
- Creating and applying an SMF profile
- Changing services and their configurations by using the `netservices` command

Creating and Exporting a Service

1. Create the script by using the following command:
   `vi /usr/local/svc/method/servicename`

2. Grant the execute permission on the script so it can be executed by using the following command:
   `chmod 544 /usr/local/svc/method/servicename`

3. Change directories to `/lib/svc/manifest/site` and edit the manifest `.xml` file for your new service.

4. Import the new service into the SMF by using the following command: `svccfg import \`
   `/lib/svc/manifest/site/servicename.xml`

5. Verify that the new service is available by using the `svcs servicename` command.

Notes for step 3: An explanation of each of the entries in the file is provided on the following pages.

Notes for step 4: When using the default manifest, `/lib/svc/manifest`, use the `import` command as shown in this step; otherwise, use the `manifest-import` command.

Creating and Exporting a Service: Example

```
# vi /usr/local/svc/method/newservice
#!/sbin/sh
#
# ident "@(#)newservice 1.14 04/08/30 SMI"
case "$1" in
'start')
/usr/bin/newservice &
;;
'stop')
/usr/bin/pkill -x -u 0 newservice
;;
*)
echo "Usage: $0 { start | stop }"
;;
esac
exit 0
# chmod 544 /usr/local/svc/method/newservice
```

ORACLE

In the example shown in the slide, you are creating a new service called `newservice`. Here you see the steps for editing the new service script and granting execute permissions.

Creating and Exporting a Service: Example

```
# cd /var/svc/manifest/site
# vi newservice.xml
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE service_bundle SYSTEM
     '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='manifest' name='OPTnew:newservice'>
  <service name='site/newservice' type='service' version='1'>
    <create_default_instance enabled='true'/>
    <single_instance/>
    <exec_method name='start' type='method'
    exec='/usr/local/svc/method/newservice start'
    timeout_seconds='30'>
    </exec_method>
    <exec_method name='stop' type='method' exec=':true'
    timeout_seconds='30'>
    </exec_method>
    <property_group name='startd' type='framework'>
    <propval name='duration' type='astring' value='transient'/>
    </property_group>
  </service>
</service_bundle>
```

Here you are changing directories to `/var/svc/manifest/site` and editing the manifest `.xml` file entries for your new service. Take a closer look at each of the entries in the file. To begin, you have the standard header:

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE service_bundle SYSTEM
'/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
```

Just below the header is the name of the service. The type (manifest) indicates a simple service rather than a milestone, the package providing the service, and the service name.

```
<service_bundle type='manifest' name='OPTnew:newservice'>
```

Next you have the service category, type, name, and version.

```
<service name='site/newservice' type='service' version='1'>
```

The next entry creates the instance and the entry below that specifies whether multiple instances of the service will run.

```
        <create_default_instance enabled='true'/>
        <single_instance/>
```

The next entry is how the service is started and stopped.

```
<exec_method name='start' type='method'
exec='/usr/local/svc/method/newservice start'
timeout_seconds='30'>
</exec_method>
<exec_method name='stop' type='method' exec=':true'
timeout_seconds='30'>
</exec_method>
```

Next is the service model to use. The entry shows that the service will be started by `svc.startd`. Transient services are not continuously running services.

```
<property_group name='startd' type='framework'>
        <propval name='duration' type='astring'
        value='transient'/>
</property_group>
</service>
</service_bundle>
```

Note: If you need to define dependencies for the service, you can do so by using the following entry:

```
<dependent
        name='newservice'
        grouping='require_all'
        restart_on='none'>
        <service_fmri value='svc:/milestone/multi-user' />
</dependent>
```

In this example, you are ensuring that the service is associated with the multiuser milestone and that the multiuser milestone requires this service.

After you have completed editing the manifest file and have reviewed the file to make sure that you have not missed any XML tags or introduced typing errors, it is a good practice to validate the file by running the following command:

```
# svccfg validate /var/svc/manifest/site/newservice.xml
```

Creating and Importing a Service: Example

```
# svccfg import /var/svc/manifest/site/newservice.xml
svccfg: Taking "previous" snapshot for svc:/site/newservice:default.
svccfg: Upgrading properties of svc:/site/newservice according to
    instance "default".
svccfg: svc:/site/newservice: Deleting property
    "general/entity_stability".
svccfg: svc:/site/newservice: Upgrading property "stop/exec".
svccfg: svc:/site/newservice: Deleting property group "tm_common_name".
svccfg: svc:/site/newservice: Deleting property group "tm_man_utmpd1M".
svccfg: svc:/site/newservice: Deleting property group "tm_man_utmpx4".
svccfg: Taking "last-import" snapshot for svc:/site/newservice:default.
svccfg: Refreshed svc:/site/newservice:default.
svccfg: Successful import.
# svcs newservice
STATE STIME FMRI
online 8:43:45 svc:/site/newservice:default
```

Here you are importing the service into the SMF by using the `svccfg import` command.

Note: The SMF is creating a snapshot of this service to be stored in the service configuration repository.

After the service has been imported into the SMF, your final step is to verify that it is visible to the system by using the `svcs` command. Note that the service is online.

Creating and Exporting a Service: Example

```
# svcadm -v disable site/newservice
site/newservice disabled.
# svcs newservice
STATE STIME FMRI
disabled 9:11:38 svc:/site/newservice:default
# svcadm -v enable site/newservice
site/newservice enabled.
# svcs newservice
STATE STIME FMRI
online 9:11:54 svc:/site/newservice:default
#
```

It is also a good practice to verify that it is possible to disable and enable the service by using the svcadm command, as shown in the example in the slide.

Modifying a Service's Manifest

1. Modify the manifest.
2. Re-import the manifest with `svcadm restart manifest-import` if in the standard location. If not in the standard location, run `svccfg import<manifest>`.
3. Importing the service will refresh it; however, a restart may be required.

There might be times when you need to modify a service's manifest due to structural changes that impact the execution method. To change the configuration of a service that is not managed by the `inetd` service, you use the steps listed in the slide.

Notes for step 1: Many of the services have one or more configuration files that are used to define the startup or other configuration information. These files can be changed while the service is running. The contents of the files are checked only when the service is started.

Notes for step 2: The `svcadm` utility enables you to perform common service management tasks, such as enabling, disabling, or restarting service instances.

Modifying a Service's Manifest: Example

```
# vi crmsvc.xml
# grep monitor crmsvc.xml
    <exec_method name='start' type='method'
  exec='/export/home/sstudent/smf/monitor1.crm'
  timeout_seconds='60'/>
# svcadm restart manifest-import
# svcadm restart crmsvc
# svcadm enable crmsvc
# svcs crmsvc
online          10:27:25 svc:/site/crmsvc:default
```

In the example shown in the slide, the decision has been made to modify the crmsvc service's manifest. To make the modification to the service manifest, you use a text editor to change the crmsvc.xml to refer to monitor1.crm instead of monitor.crm. To accomplish the change, you use the svcadm restart manifest-import command. After the import, you restart and enable the service. Finally, you verify that the service is online.

Changing an Environment Variable for a Service

1. Verify that the service is running by using `svcs` *FMRI.*
2. Set environment variables by using `svccfg -s` *FMRI* `setenv` *envar value.*
3. Refresh the service by using `svcadm refresh` *FMRI*.
4. Restart the service by using `svcadm restart` *FMRI*.
5. Verify that the change has been made by using `pargs -e` `` `pgrep -f /usr/sbin/ ``*FMRI*`` ` ``.

Notes for step1: The `svcs` utility provides detailed views of the service state of all service instances in the service configuration repository.

Notes for step 2: The `-s` option selects the entity indicated by the fault management resource identifier (FMRI) before executing any subcommands. The modification subcommand `setenv` searches for the "start" property group in the currently selected entity and, if an instance is currently selected, its parent is also searched. After the property is located, all values that begin with *envvar* followed by a "=" are removed, and the value "*envvar=value*" is added.

Notes for step 3: The `svcadm` command is used to manipulate service instances. The command issues requests for actions on services executing within the SMF. Actions for a service are carried out by its assigned service restarter agent. The `refresh` subcommand requests that the assigned restarter update the service's running configuration snapshot with the values from the current configuration. Some of these values take effect immediately (for example, dependency changes). Other values do not take effect until the next service restart.

Notes for step 5: The `pargs -e` command prints the parameter arguments and environment variables that have been passed to the service.

Changing an Environment Variable
for a Service: Example

```
# svcs system/cron
STATE          STIME    FMRI
online         13:02:52 svc:/system/cron:default
# svccfg -s system/cron:default setenv UMEM_DEBUG default
# svccfg -s system/cron:default setenv LD_PRELOAD libumem.so
# svcadm refresh system/cron
# svcadm restart system/cron
# pargs -e `pgrep -f /usr/sbin/cron`
100657: /usr/sbin/cron
envp[0]: LOGNAME=root
envp[1]: LD_PRELOAD=libumem.so
envp[2]: PATH=/usr/sbin:/usr/bin
envp[3]: SMF_FMRI=svc:/network/ssh:default
envp[4]: SMF_METHOD=/lib/svc/method/svc-ssh
envp[5]: SMF_RESTARTER=svc:/network/svc/restarter:default
envp[6]: TZ=GB
envp[7]: UMEM_DEBUG=default
#
```

ORACLE

In the example shown in the slide, you are changing the environment variable for the `ssh` service to help with debugging. First, you verify that the service is up and running, and it is. Next, you set the `UMEM_DEBUG` and `LD_PRELOAD` environment variables by using the `svccfg -s` command with the `setenv` subcommand. To make the changes effective, you refresh and then restart the system by using the `svcadm refresh` and `svcadm restart` commands. Finally, you verify that the change has been made by using the `pargs -e` command. Here you can see that the two variables are present. The `LD_PRELOAD` environment variable is `envp[1]`, and the `UMEM_DEBUG` environment variable is `envp[7]`.

Changing a Property for an `inetd`-Controlled Service

1. List the properties for the specific service by using `inetadm -l` *FMRI.*

2. Change the property for the service by using `inetadm -m` *FMRI property-name=value.*

3. Verify that the property has changed by using `inetadm -l` *FMRI.*

4. Confirm that the change has taken effect.

If you have a need to impose more access controls on a particular Internet service, you can do so by modifying the service's property settings.

Notes for step 1: The `inetadm` command enables you to observe or configure services controlled by `inetd`, which is the delegated restarter for Internet services for the SMF. Its basic responsibilities are to manage service states in response to administrative requests, system failures, and service failures and, when appropriate, to listen for network requests for services.

The `inetadm -l` command displays all the properties for the service identified by the FMRI.

Notes for step 2: The `-m` option is used to change the values of the specified properties of the identified service instances. Each property for an `inetd`-controlled service is defined by a property name and an assigned value. Supplying the property name without a specified value resets the property to the default value.

Notes for step 3: You want to list the properties again to make sure that the appropriate change has occurred.

Changing a Property for an
`inetd`-Controlled Service: Example

```
# inetadm -l svc:/network/telnet
SCOPE        NAME=VALUE
             name="telnet"
             endpoint_type="stream"
             proto="tcp6"
             isrpc=FALSE
             wait=FALSE
             exec="/usr/sbin/in.telnetd"
             user="root"
default      bind_addr=""
default      bind_fail_max=-1
default      bind_fail_interval=-1
default      max_con_rate=-1
default      max_copies=-1
default      con_rate_offline=-1
default      failrate_cnt=40
default      failrate_interval=60
default      inherit_env=TRUE
default      tcp_trace=FALSE
default      tcp_wrappers=FALSEgrep inetd /etc/init.d/inetsvc
default      connection_backlog=10
default      tcp_keepalive=FALSE
```

ORACLE

In the example shown in the slide, you enable the `tcp_trace` property for the `telnet` service. As you can see, currently the `tcp_trace` property is set to `FALSE`.

Changing a Property for an
`inetd`-Controlled Service: Example

```
# inetadm -m telnet tcp_trace=TRUE
# inetadm -l telnet
SCOPE        NAME=VALUE
             name="telnet"
             endpoint_type="stream"
             proto="tcp6"
             isrpc=FALSE
             wait=FALSE
             exec="/usr/sbin/in.telnetd"
             user="root"
default      bind_addr=""
default      bind_fail_max=-1
default      bind_fail_interval=-1
default      max_con_rate=-1
default      max_copies=-1
default      con_rate_offline=-1
default      failrate_cnt=40
default      failrate_interval=60
default      inherit_env=TRUE
default      tcp_trace=TRUE
default      tcp_wrappers=FALSEgrep inetd /etc/init.d/inetsvc
default      connection_backlog=10
default      tcp_keepalive=FALSE
```

Here you can verify that the property has been changed.

Changing a Property for an
`inetd`-Controlled Service: Example

```
# tail -1 /var/adm/messages
Dec 15 08:04:39 client1 inetd[655]: [ID 317013 daemon.notice]
    telnet[2390] from 192.168.0.100 34098
# grep /var/adm/messages /etc/syslog.conf
*.err;kern.debug;daemon.notice;mail.crit   /var/adm/messages
```

The last step is to confirm that the change has taken effect. First, you telnet to your host from another host. You then check the `/var/adm/messages` file to see if the telnet connection was logged, which as you can see it was. You then confirm the entry in `/etc/syslog.conf`, which is configured to log this message. You have successfully changed the service property.

Creating and Applying an SMF Profile

1. Create a profile by using `svccfg extract> profile.xml`.
2. Edit the `profile.xml` file to make any required changes.
   a. Change the name of the profile in the `service_bundle` declaration.
   b. Remove any services that should not be managed by this profile.
   c. Add any services that should be managed by this profile.
   d. If necessary, change the enabled flag for selected services.
3. When necessary, apply the new profile by using `svccfg apply profile.xml`.

You can create an SMF profile that reflects which services you want enabled or disabled on the current system. Not all services need to be listed in a profile. Each profile needs to include only those services that need to be enabled or disabled to make the profile useful.

The steps for how to create an SMF profile are shown in the slide.

Notes for step 1: The `svccfg` utility enables you to display and manipulate the contents of the service configuration repository. The service profile subcommand `extract` prints a service profile that represents the enabled status of the service instances in the repository to standard output. You can redirect the output to a file by using `extract>` (as you are doing in step 1).

Notes for step 2b: For each service, remove the three lines that describe the service. Each service description starts with `<service` and ends with `</service`.

Notes for step 2c: Each service needs to be defined by using the three-line syntax shown here:

```
<service name='network/ldap/client' version='1' type='service'>
        <instance  name='default' enabled='true'/>
</service>
```

Notes for step 3: Applying the service profile subcommand takes the properties, including general/enabled, that are specified in the file and modifies them in the SMF repository.

Creating and Applying an SMF Profile: Example

```
# svccfg extract > profile.xml
# vi profile.xml
# cat profile.xml
  ...
<service_bundle type=`profile` name=`profile`
    xmlns::xi='http://www.w3.org/2003/XInclude`
  ...
 <service name='network/ldap/client' version='1'
   type='service'>
         <instance  name='default' enabled='true'/>
 </service>
  ...
<service  name='network/smtp' version='1' type='service'>
    <instance  name='sendmail' enabled='false'/>
  </service>
  ...
# svccfg apply profile.xml
```

In the example shown in the slide, you use the svccfg command to create a profile called profile.xml that reflects which services are enabled or disabled on the current system. The assumption is that you are in your own home directory while performing this task.

Note: It is a best practice to use profile as the default name for your profile. Also, you do have the option of making a copy of an existing profile to edit instead of creating a new profile.

In the first line of the profile.xml file, you change the name of the profile in the service_bundle declaration to profile. In the second line, you add the LDAP client service to the profile. In the third line, you disable the sendmail service. You then apply the profile.

Changing Services and Their Configurations by Using the `netservices` Command

Run the `netservices` command to select either open (traditional) or limited network exposure.

- For open or traditional network exposure, run `/usr/sbin/netservices open`.

- For limited network exposure, run `/usr/sbin/netservices limited`.

The `netservices` command switches system services between minimal network exposure and traditional network exposure. The switch is done with the `generic_limited.xml` and `generic_open.xml` profiles. In addition, some service properties are changed by the command to limit some services to a local-only mode or to the traditional mode, as appropriate.

Note: The `generic_limited_net` profile and the local-only-mode service properties are applied by default.

To have open or traditional network exposure, you use the `/usr/sbin/netservices open` command.

To have limited network exposure, you use the `/usr/sbin/netservices limited` command. This command changes properties to run some services in local mode, as well as restricts which services are enabled with the `generic_limited_net` profile. The command should be used only if the `generic_open.xml` profile is applied.

Practice 7-1 and Practice 7-2 Overview:
Configuring SMF Services
and Working with Service Profiles

These practices cover the following topics:

- Creating and exporting a service

- Modifying a service configuration

- Creating and applying an SMF profile

The practices for this lesson are designed to reinforce the concepts that have been presented in the lecture portion. These practices cover the following tasks:

- **Practice 7-1:** Configuring SMF services
- **Practice 7-2:** Working with service profiles
- **Practice 7-3:** Restoring and recovering a service

Practices 7-1 and 7-2 should take you a total of about 40 minutes to complete.

Lesson Agenda

- Planning Services Configuration
- Configuring SMF Services
- **Troubleshooting SMF Services**

ORACLE

Troubleshooting SMF Services

This section covers the following topics:

- Debugging a service that is not starting
- Restoring a service in maintenance state
- Reverting to an SMF snapshot
- Repairing a corrupt repository
- Debugging the services during a system boot
- Addressing `system/filesystem/local:default` service failures during boot

Debugging a Service That Is Not Starting

1. Request information about the hung service by using `svcs -xv servicename`.

2. Enable the service by using `svcadm enable serviceinstance`.

3. Verify that the service is online by using `svcs -a servicename`.

```
# svcs -xv
svc:/application/print/server:default (LP Print Service)
 State: disabled since Thu 15 Dec 2011 02:20:37 PM PDT
Reason: Disabled by an administrator.
   See: http://sun.com/msg/SMF-8000-05
   See: man -M /usr/share/man -s 1M lpsched
Impact: 2 services are not running:
        svc:/application/print/rfc1179:default
        svc:/application/print/ipp-listener:default
# svcadm enable application/print/server
# svcs printer
online         11:06:14 svc:/application/print/server:default
```

If you have a service that is disabled and not starting, you can debug it by using the steps shown in the slide.

Notes for step 1: The `-x` option provides additional information about the service instances that are affected.

In the example, the `print` service is disabled. To find out more about the problem, you run the `svcs -xv` command for the service. The output for the `svcs -xv` command provides the following information:

- **State:** The state of the service and the date and time stamp
- **Reason:** Why the service is disabled
- **See:** The URL to a knowledge article on the issue
- **See:** Man page references to help resolve the issue
- **Impact:** What services have been affected by the problem

Here you can see that the service was disabled by an administrator. You can also see that having the printer service disabled is impacting two other services. Because the issue is that an administrator disabled the service, you try to correct the problem by enabling the service. To verify that the service is back online, you can use the `svcs servicename` command.

Restoring a Service in Maintenance State

1. Determine if any processes that are dependent on the service have not stopped by using `svcs -p` *FRMI*.
2. Kill any remaining processes as required by using `pkill -9` *PID*.
3. If necessary, repair the service configuration by using `svcs -x` *FRMI*.
4. Restore the service by using `svcadm clear` *FMRI*.

SMF places a service in maintenance mode when it is unable to bring it up. As a system administrator, it is your job to figure out what has caused the problem. The steps for restoring a service in `maintenance` state are shown in the slide.

Notes for step 1: Normally, when a service instance is in `maintenance` state, all processes associated with that instance have stopped. However, you should make sure before you proceed. The `svcs -p` *FRMI* command lists all the processes that are associated with a service instance as well as the PIDs for those processes.

Notes for step 2: Repeat this step for all processes that are displayed by the `svcs` command.

Notes for step 3: The `-x` option provides you with details that you might find useful for debugging the issue. You can also examine the appropriate service log files in `/var/svc/log` for a list of errors.

Restoring a Service
in Maintenance State: Example

```
# svcs time-slider:default
STATE             STIME    FMRI
maintenance       8:22:10 svc:/application/time-slider:default

# svcs -p time-slider:default
STATE             STIME    FMRI
maintenance       8:23:06 svc:/application/time-slider:default
svcs -x time-slider:default
svc:/application/time-slider:default (GNOME Desktop Snapshot
   Management Service)
 State: maintenance since Dec 15, 2011 08:22:41 AM MDT
Reason: Start method exited with $SMF_EXIT_ERR_FATAL.
   See: http://sun.com/msg/SMF-8000-KS
   See: zfs(1M)
   See: /var/svc/log/application-time-slider:default.log
Impact: This service is not running.
# svccfg delete time-slider:default
```

In the example shown in the slide, the time-slider: default service is in the maintenance state. Your first step is to determine if any processes that are dependent on the service have not stopped using the svcs -p command. As you can see, no dependent processes are listed, so your next step is to repair the service by using the svcs -x command. The output from this command indicates that there is an issue with the start method.

Note: You can examine the log for further details.

Your next step is to determine what in the execution method configuration in the time-slide.xml manifest file is causing the problem. However, before you do that, you are going to delete the corrupted service by using the svccfg delete command.

Restoring a Service
in Maintenance State: Example

```
# svcadm refresh time-slider:default
# svcadm enable time-slider:default
# svcadm clear time-slider:default
# svcs time-slider:default
STATE          STIME    FMRI
online          9:37:52 svc:/application/time-slider:default
```

Assume you opened the `time-slider.xml` manifest file, found the problem with the start method, fixed it, and imported the file into SMF. You are now ready to bring the service back up. To do this, you first refresh the service by using the `svcadm refresh` command to make sure SMF is reading the new service manifest file, enable the service, and then restore the service by using the `svcadm clear` command. You then verify that the service is back online, and it is. You have successfully restored the service.

Reverting to an SMF Snapshot

1. Run the `svccfg` command.
   a. Select the service instance that you want to fix.
   b. Generate a list of available snapshots by using `listsnap`.
   c. Select to revert to the `start` snapshot by using `revert start`.
   d. Quit `svccfg` by using `quit`.
2. Update the information in the service configuration repository by using `svcadm refresh` *FMRI*.
3. Restart the service instance by using `svcadm restart` *FMRI*.

If the service's administrative customizations are wrong, you can fix the problem by reverting to the last snapshot that started successfully. The steps for how to revert to a previous SMF snapshot are shown in the slide.

Notes for step 1a: You must use an FMRI that fully defines the instance. No shortcuts are allowed.

Notes for step 1c: The start snapshot is the last snapshot in which the service successfully started.

Notes for step 2: This step updates the repository with the configuration information from the start snapshot.

Note: None of the file-backed properties (that is, properties delivered via manifests or profiles) from the snapshot are restored. Instead, all the administrative customizations in the current configuration are removed, and then all the administrative customizations from the selected snapshot are propagated forward.

Reverting to an SMF Snapshot: Example

```
# svccfg
svc:> select system/console-login:default
svc:/system/console-login:default> listsnap
initial
last-import
previous
running
start
svc:/system/console-login:default> revert start
svc:/system/console-login:default> quit
# svcadm refresh system/console-login:default
# svcadm restart system/console-login:default
# svcs console-login:default
online   18:15:32 svc:/system/console-login:default
```

In the example shown in the slide, it is assumed that the `console-login:default` service is in the `maintenance` state. To resolve the issue, you have decided to revert to a previous SMF snapshot to bring the service back online. You have selected the `start` snapshot.

Note: The version of the snapshot you choose to use is based on what you are trying to accomplish.

When you have selected the type of snapshot you want, you quit the service configuration. You then refresh and restart the service. Your final step is to verify that the service is back online.

Configuration Repository
Failed Integrity Check Process

Message is sent to console if integrity check fails:

```
<MESSAGE DISPLAYED BY SMF>
svc.configd: smf(5) database integrity check of:

   /etc/svc/repository.db

 failed.  The database might be damaged or a media error might have
 prevented it from being verified.  Additional information useful to
 your service provider is in:

   /etc/svc/volatile/db_errors

 The system will not be able to boot until you have restored a working
 database.  svc.startd(1M) will provide a sulogin(1M) prompt for
 recovery purposes. The command:

   /lib/svc/bin/restore_repository

 can be run to restore a backup version of your repository. See
 http://sun.com/msg/SMF-8000-MY for more information.
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When the repository daemon, `svc.configd`, is started, it does an integrity check of the configuration repository. If the integrity check fails, the `svc.configd` daemon writes a message to the console similar to the one shown in the slide. The `svc.startd` daemon then exits and starts `sulogin` to enable you to perform maintenance as shown on the next page.

Note: The repository can become corrupted due to one of the following reasons:

- Disk failure
- Hardware bug
- Software bug
- Accidental overwrite of the file

Oracle Solaris 11 Advanced System Administration 7 - 57

Unauthorized reproduction or distribution prohibited. Copyright© 2014, Oracle and/or its affiliates.

Repairing a Corrupt Repository

1. Enter the `root` password at the `sulogin` prompt.
2. Run the following command:
   `/lib/svc/bin/restore_repository`
3. Enter the appropriate response.
4. Enter `yes` to remedy the fault.

Notes for step 1: `sulogin` enables the `root` user to enter system `maintenance` mode to repair the system.

Notes for step 2: Running this command takes you through the necessary steps to restore a non-corrupt backup. SMF automatically takes backups of the repository at key system moments. When started, the `/lib/svc/bin/restore_repository` command displays a message similar to the following:

```
Repository Restore utility

See http://sun.com/msg/SMF-8000-MY for more information on the use of

this script to restore backup copies of the smf(5) repository.


If there are any problems which need human intervention, this script

will give instructions and then exit back to your shell.


Note that upon full completion of this script, the system will be

rebooted using reboot(1M), which will interrupt any active services.
```

If the system that you are recovering is not a local zone, the script explains how to remount the / and /usr file systems with read and write permissions to recover the databases. The script exits after printing these instructions. Follow the instructions, paying special attention to any errors that might occur.

After the root (/) file system is mounted with write permissions, or if the system is a local zone, you are prompted to select the repository backup to restore, as follows:

```
The following backups of /etc/svc/repository.db exists, from oldest to newest:

... list of backups ...
```

Backups are given names, based on type and the time the backup was taken. Backups beginning with boot are completed before the first change is made to the repository after system boot. Backups beginning with manifest_import are completed after svc:/system/manifest-import:default finishes its process. The time of the backup is given in YYYYMMDD_HHMMSS format.

Notes for step 3: Typically, you will select the most recent backup option. The list of options is as follows:

```
Please enter one of:
        1) boot, for the most recent post-boot backup
        2) manifest_import, for the most recent manifest_import backup.
        3) a specific backup repository from the above list
        4) -seed-, the initial starting repository. (All customizations
           will be lost.)
        5) -quit-, to cancel.

Enter response [boot]:
```

If you press Enter without specifying a backup to restore, the default response, enclosed in [ ] is selected. Selecting -quit- exits the restore_repository script, returning you to your shell prompt.

Selecting -seed- restores the seed repository. This repository is designed for use during initial installation and upgrades. Using the seed repository for recovery purposes should be a last resort.

After the backup to restore has been selected, it is validated and its integrity is checked. If there are any problems, the restore_repository command prints error messages and prompts you for another selection.

When a valid backup is selected, the following information is printed and you are prompted for final confirmation.

```
After confirmation, the following steps will be taken:

svc.startd(1M) and svc.configd(1M) will be quiesced, if running.
/etc/svc/repository.db
    -- renamed --> /etc/svc/repository.db_old_YYYYMMDD_HHMMSS
/etc/svc/volatile/db_errors
    -- copied --> /etc/svc/repository.db_old_YYYYMMDD_HHMMSS_errors
repository_to_restore
    -- copied --> /etc/svc/repository.db
and the system will be rebooted with reboot(1M).


Proceed [yes/no]?
```
Notes for step 4: The system reboots after the `restore_repository` command executes all the listed actions.

Repairing a Corrupt Repository: Example

```
# cd /lib/svc/bin
#:/lib/svc/bin# ./restore_repository

<output omitted>

The following backups of /etc/svc/repository.db exist, from oldest to newest:

manifest_import-20111215_035411
boot-20111214_124026
boot-20111215_150206

Please enter either a specific backup repository from the above list to restore it,
or one of the following choices:

CHOICE                ACTION
----------------      ---------------------------------------
boot                  restore the most recent post-boot backup
manifest_import       restore the most recent manifest_import backup
-seed-                restore the initial starting repository  (All
                      customizations will be lost, including those
                      made by the install/upgrade process.)
-quit-                cancel script and quit

Enter response [boot]: boot-20111215_150206
```

In the example shown in the slide, you are restoring the repository by using the most recent post-boot backup option. The confirmation for this selection is shown on the next page.

Repairing a Corrupt Repository: Example

```
<output continued from previous page>
…
…
After confirmation, the following steps will be taken:

svc.startd(1M) and svc.configd(1M) will be quiesced, if running.
/etc/svc/repository.db
    -- renamed --> /etc/svc/repository.db_old_20111215_060922
/etc/svc/repository-boot-20111215_150206
    -- copied --> /etc/svc/repository.db
and the system will be rebooted with reboot(1M).

Proceed [yes/no]? yes
```

Here you are prompted for the final confirmation. You enter yes to tell the system to remedy the fault. After the restore_repository command executes all the listed actions, the system reboots.

Debugging the Services During a System Boot

1. Log in to the system as `root`.
2. Enable all services by using `svcadm milestone all`.
3. Determine where the boot process is hanging:
   a. Run `svcs -a` to determine which services are not running.
   b. Look for error messages in the log files in `/var/svc/log`.
4. After fixing the problems, verify that all services have started.
   a. Verify that all needed services are online by using `svcs -x`.
   b. Verify that the `console-login` service dependencies are satisfied by using `svcs -l system/console-login:default`.
5. Continue the normal booting process.

If problems with starting services occur, sometimes a system will hang during the boot. You can use the steps shown in the slide to troubleshoot this problem.

Notes for step 2: There is an additional system state associated with the `all` milestone. With the `all` milestone, all the services with a defined dependency on the multiuser-server milestone are started, as well as any services that do not have a defined dependency. If you have added services, such as third-party products, they may not be started automatically unless you use the `boot -m milestone=all` command.

Notes for step 4b: This command verifies that the login process on the console will run.

Addressing `system/filesystem/local:default` Service Failures During Boot

1. Modify the `system/console-login` service as follows by using `svccfg -s svc:/system/console-login`:
   - svc:/system/console-login> **addpg site,filesystem-local dependency**
   - svc:/system/console-login> **setprop site,filesystem-local/entities = fmri: \ svc:/system/filesystem/local**
   - svc:/system/console-login> **setprop site,filesystem-local/grouping = astring: require_all**
   - svc:/system/console-login> **setprop site,filesystem-local/restart_on = astring: none**
   - svc:/system/console-login> **setprop site,filesystem-local/type = astring: service**
   - svc:/system/console-login> **end**
2. Refresh the service by using `svcadm refresh console-login`.

Local file systems that are not required to boot the system are mounted by the `svc:/system/filesystem/local:default` service. When any of those file systems are unable to be mounted, the service enters a maintenance state. System startup continues, and any services that do not depend on `filesystem/local` are started. Services that require `filesystem/local` to be online before starting through dependencies are not started. To change the configuration of the system so that a `sulogin` prompt appears immediately after the service fails instead of allowing system startup to continue, you can use the steps shown in the slide.

Note: When a failure occurs with the `system/filesystem/local:default` service, the `svcs -vx` command should be used to identify the failure. After the failure has been fixed, the following command clears the error state and allows the system boot to continue: `svcadm clear filesystem/local`.

Practice 7-3 Overview:
Restoring and Recovering a Service

This practice covers the following topics:

- Restoring a service in maintenance state
- Reverting to a previous SMF snapshot
- Repairing a corrupt repository
- Debugging a service that is not starting

This practice should take you about 20 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Implement a plan to configure services
- Configure SMF services
- Recover a service from a snapshot
- Troubleshoot SMF services

8

Configuring Privileges and Role-Based Access Control

ORACLE

Objectives

After completing this lesson, you should be able to:
- Implement a plan to configure privileges
- Implement a plan to configure role-based access control
- Configure privileges
- Manage privileges
- Configure role-based access control
- Use role-based access control

ORACLE

Workflow Orientation

Before you begin the lesson, take just a moment to orient yourself in your job workflow. You have successfully installed the operating system and have updated it. You have configured the data storage environment as well as the physical and virtual networks. You have also ensured that all the system services are up and running. In the Oracle Solaris 11 Operating System, the root, a process, and a non-root user need appropriate privileges to perform their functions. To protect the integrity of system resources, the system administrator is responsible for ensuring that both users and processes have been granted the appropriate level of privilege.

Lesson Agenda

- **Planning for User Privileges and Roles Assignments**
- Configuring and Managing Privileges
- Configuring and Using RBAC

Planning for User Privileges and Roles Assignments

User privilege and roles assignment planning is required to ensure that:

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

Your company is very security conscious and wants to ensure that processes and users have only the level of access or privilege to system resources they need to perform their required functions. The predeployment plan contains activities to investigate what features and functionality Oracle Solaris 11 offers that would support the company's security policy, specifically in the area of process rights management and role-based access control.

In this topic, you learn how Oracle Solaris 11 supports process rights management and role-based access control.

Process Rights Management and Privileges

- Process rights management
  - Enables processes to be restricted at the command, user, role, or system level
  - Is implemented through privileges
- Privileges
  - Decrease the security risk associated with one user or one process having full superuser capabilities on a system
  - Allow gradation between user capabilities and root capabilities
  - Restrict programs and processes to only the capabilities that the program requires (principle of least privilege)

ORACLE

Process rights management enables processes to be restricted at the command, user, role, or system level. The Oracle Solaris OS implements process rights management through privileges. Privileges decrease the security risk that is associated with one user or one process having full superuser capabilities on a system.

A system that enforces policy with privileges allows a gradation between user capabilities and root capabilities. A user can be granted privileges to perform activities that are beyond the capabilities of regular users, and `root` can be limited to fewer privileges than `root` currently possesses. With RBAC, a command that runs with privileges can be isolated in a rights profile and assigned to one user or role.

Privileges, then, can restrict programs and processes to just the capabilities that the program requires. This capability is called the principle of least privilege. On a system that implements least privilege, an intruder who captures a process can access only those privileges that the process has. The rest of the system cannot be compromised.

Displaying Privilege Descriptions

- **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 various system properties.

Privileges are logically grouped on the basis of the area of the privilege. The areas of privilege are as follows:

- **FILE privileges:** Privileges that begin with the string `file` operate on file system objects. For example, the `file_dac_write` privilege overrides discretionary access control when writing to files.
- **IPC privileges:** Privileges that begin with the string `ipc` override IPC object access controls. For example, the `ipc_dac_read` privilege enables a process to read remote shared memory that is protected by DAC.
- **NET privileges:** Privileges that begin with the string `net` give access to specific network functionality. For example, the `net_rawaccess` privilege enables a device to connect to the network.
- **PROC privileges:** Privileges that begin with the string `proc` enable processes to modify restricted properties of the process itself. PROC privileges include privileges that have a very limited effect. For example, the `proc_clock_highres` privilege enables a process to use high-resolution timers.
- **SYS privileges:** Privileges that begin with the string `sys` give processes unrestricted access to various system properties. For example, the `sys_linkdir` privilege enables a process to make and break hard links to directories.

Implementing Privileges

- **Effective privilege set, or E:** Set of privileges that are currently in effect

- **Inheritable privilege set, or I:** Set of privileges that a process can inherit across a call to exec

- **Permitted privilege set, or P:** Set of privileges that are available for use

- **Limit privilege set, or 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
```

Every process has four sets of privileges that determine whether a process can use a particular privilege. The kernel automatically calculates the effective set of privileges. You can modify the initial inheritable set of privileges. A program that is coded to use privileges can reduce the program's permitted set of privileges. You can shrink the limit set of privileges.
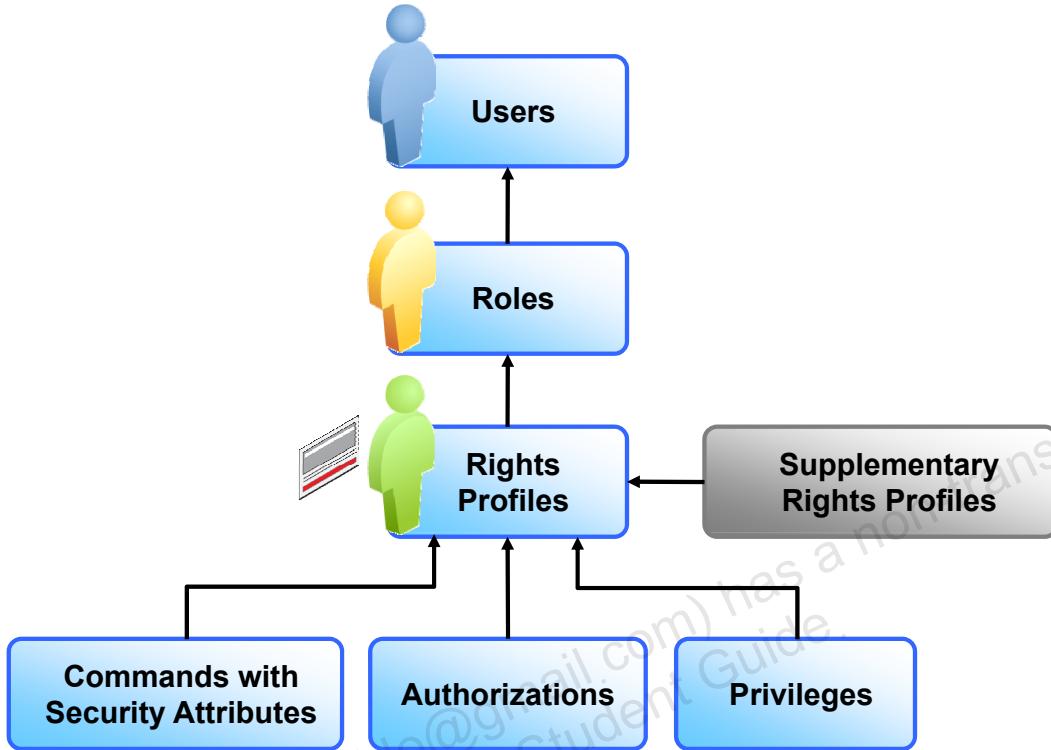
- **Effective privilege set, or E:** Is the set of privileges that are currently in effect. A process can add privileges that are in the permitted set to the effective set. A process can also remove privileges from E.

- **Inheritable privilege set, or I:** Is the set of privileges that a process can inherit across a call to exec. After the call to exec, the permitted and the effective sets are equal, except in the special case of a setuid program. For a setuid program, after the call to exec, the inheritable set is first restricted by the limit set. Then, the set of privileges that were inherited (I), minus any privileges that were in the limit set (L), are assigned to P and E for that process.

- **Permitted privilege set, or P:** Is the set of privileges that are available for use. Privileges can be available to a program from inheritance or through assignment. An execution profile is one way to assign privileges to a program. The `setuid` command assigns all privileges that root has to a program. Privileges can be removed from the permitted set, but privileges cannot be added to the set. Privileges that are removed from P are automatically removed from E.

  Note: A privilege-aware program removes the privileges that a program never uses from the program's permitted set. In this way, unnecessary privileges cannot be exploited by the program or a malicious process.

- **Limit privilege set, or L:** Is the outside limit of what privileges are available to a process and its children. By default, the limit set is all privileges. Processes can shrink the limit set but can never extend the limit set. L is used to restrict I. Consequently, L restricts P and E at the time of execution.

If a user is assigned a profile with a program that has been assigned privileges, the user can usually run that program. On an unmodified system, the program's assigned privileges are within the user's limit set. The privileges that have been assigned to the program become part of the user's permitted set. To run the program that has been assigned privileges, the user must run the program from a profile shell.

The kernel recognizes a basic privilege set. On an unmodified system, each user's initial inheritable set equals the basic set at login. Although you cannot modify the basic set, you can modify which privileges a user inherits from the basic set. On an unmodified system, a user's privilege sets at login would appear similar to the example shown in the slide. Therefore, at login, all users have the basic set in their inheritable set, their permitted set, and their effective set. A user's limit set is equivalent to the default limit set for the zone, global or non-global. To put more privileges in the user's effective set, you must assign a rights profile to the user. The rights profile would include commands to which you have added privileges.

Role-Based Access Control (RBAC)
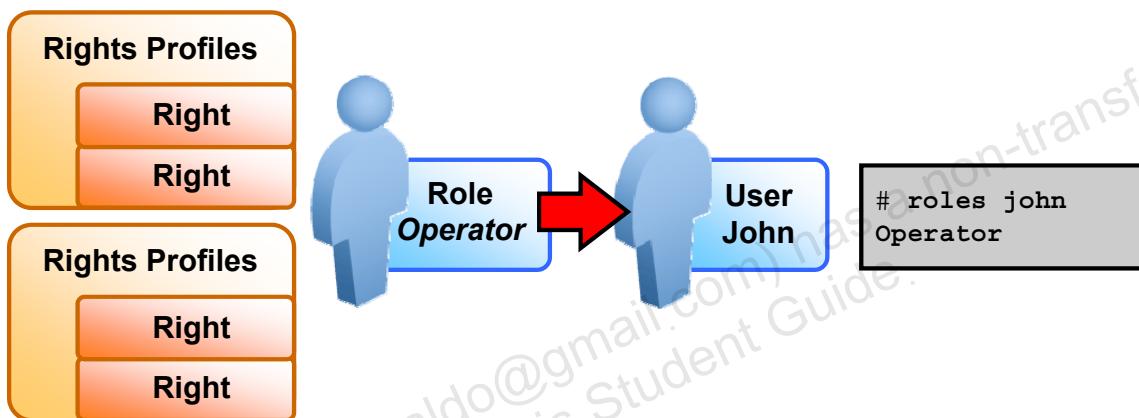
RBAC is a security feature for controlling user access to tasks that would normally be restricted to the superuser. RBAC collects superuser capabilities into rights profiles. Rights profiles can contain authorizations, privileges, privileged commands, and other supplementary rights profiles. Privileged commands are commands that execute with security attributes. Rights profiles are assigned to special user accounts that are called roles. A user can then assume a role to do a job that requires some of the superuser's capabilities.

Take a closer look at each of the key RBAC concepts, beginning with roles.

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

A role is a special type of user account that performs a set of administrative tasks. Usually, a role contains one or more rights profiles, and a user is associated with one or more roles to gain access to restricted functionality. A role can be shared among users. Because of this, roles are preferred in RBAC as they simplify the management of large numbers of users.
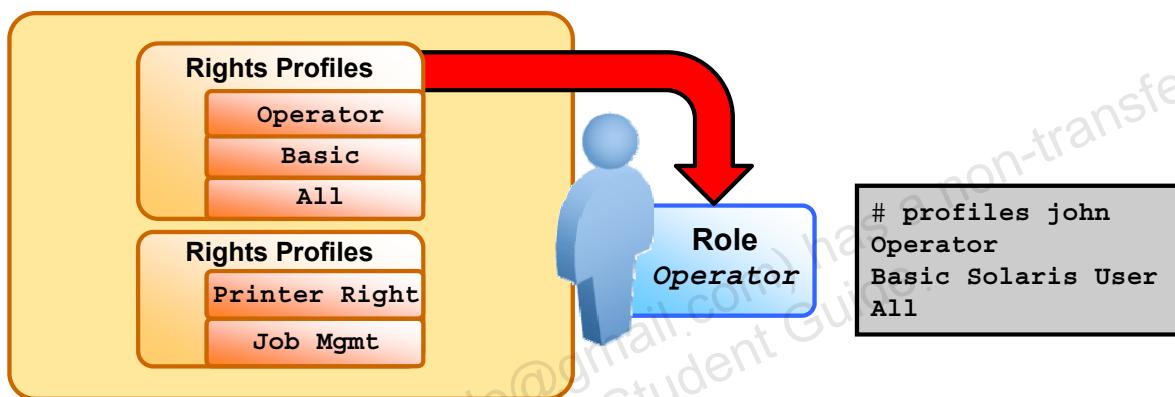
Note: A role cannot log in to the system. A user must be logged in to the system to assume a role.

The graphic illustrates how the user John is assigned the Operator role, which contains several rights profiles.

The roles assigned to a user can be displayed by using the `roles` command. In the code example, the roles assigned to the user `john` are displayed. `john` has one role assigned to him: the Operator role.

Rights Profile

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


```
# profiles john
Operator
Basic Solaris User
All
```

A rights profile can consist of authorizations, commands with security attributes, and other rights profiles. Rights profiles offer a convenient way to group security attributes.

New rights profiles can be created by editing this file. You are shown how to do this later in this lesson.

The graphic illustrates rights profiles being assigned to the user john.

The profiles assigned to a user can be displayed by using the profiles command. In the code example, the profiles assigned to the user john are displayed. john has three rights profiles assigned to him: Operator, Basic Solaris User, and ALL.

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
```

All users have the `Basic Solaris User` profile by default. This profile grants users access to all listed authorizations, as indicated by `auths=`.

Note: An authorization is divided into hierarchies, which are separated by periods. For example, in the `solaris.network.autoconf.read` authorization, the first level of the hierarchy is `solaris`, followed by the second level, which is network.autoconf (`automatic configuration of the network`), and the third level, which is `read`. Taken together, this entry is giving the basic user the authority to display the rights profile. The `solaris.mail.mailq` authorization enables the basic user to look at the mail queue, and so on.

Note: Other default authorizations for every user can be defined in the `/etc/security/policy.conf` file.

A semicolon in a rights profile means that a different type of information is being specified, an example of which can be seen just before `profiles=All`. In this case, the `Basic Solaris User` profile is being attached to the `All` profile. The last file is a `help` file.

Note: The `profiles=All` field grants unrestricted access to all Oracle Solaris OS commands that have not been restricted by a definition in a previously listed authorization.

Interpreting the
`/etc/security/policy.conf` File
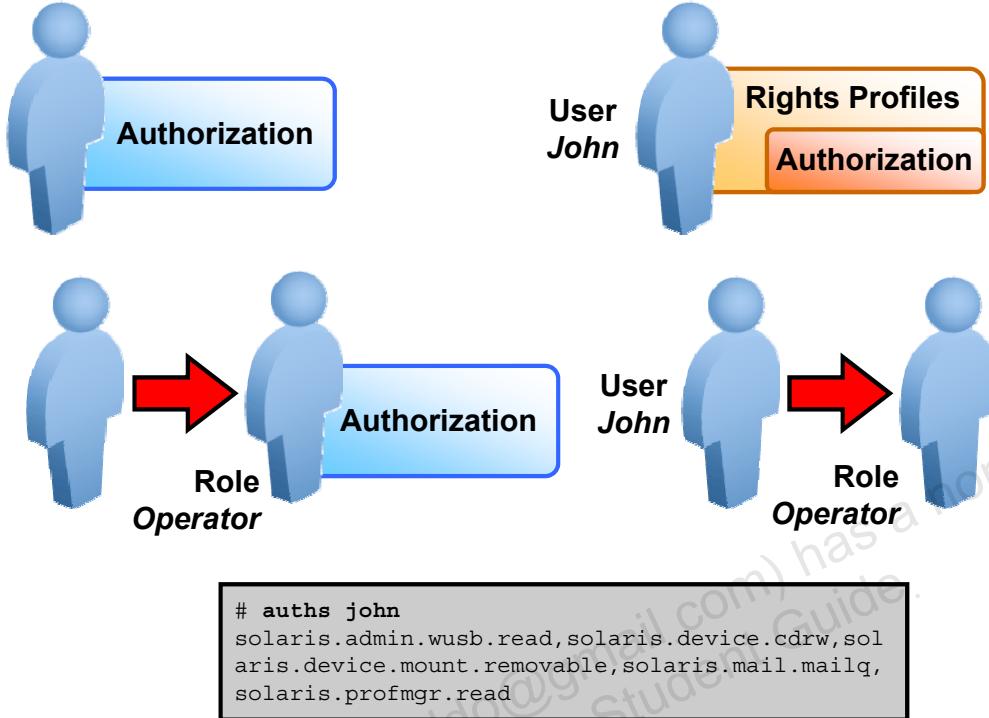
```
# cat /etc/security/policy.conf
<header and copyright output omitted>
#
AUTHS_GRANTED=solaris.device.cdrw
PROFS_GRANTED=Basic Solaris User
CONSOLE_USER=Console User
# crypt(3c) Algorithms Configuration
#
# CRYPT_ALGORITHMS_ALLOW specifies the algorithms that are allowed to
# be used for new passwords. This is enforced only in crypt_gensalt(3c).
#
CRYPT_ALGORITHMS_ALLOW=1,2a,md5,5,6
<output omitted>
#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
```

Rights profiles given to all new user accounts are defined in the
`/etc/security/policy.conf` file. The settings in this file determine the default privileges
that users have. If they are not set, the default privileges are taken from the inherited set.
There are two different settings: `PRIV_DEFAULT` determines the default set on login, and
`PRIV_LIMIT` defines the Limit set on login. Individual users can have privileges assigned or
taken away through `user_attr`.

Authorizations and Privileges


```
# auths john
solaris.admin.wusb.read,solaris.device.cdrw,sol
aris.device.mount.removable,solaris.mail.mailq,
solaris.profmgr.read
```

An authorization is a name associated with the right to access restricted functionality. Authorizations enforce policy at the user application level. Authorizations can be assigned directly to a role or to a user. Typically, authorizations are included in a rights profile. The rights profile is then included in a role, and the role is assigned to a user. For example, security policy at installation gives regular users the solaris.device.cdrw authorization. This authorization enables users to read and write to a CD-ROM device.

The graphic illustrates that authorizations can be assigned to user accounts, to roles, or embedded in a rights profile, which can be assigned to a user or a role.

The authorizations assigned to a user can be displayed by using the auths command. In the code example, the authorizations assigned to the user john are displayed. john has all Oracle Solaris authorizations assigned to him.

A privilege is a discrete right that can be granted to a command, a user, a role, or a system. Privileges enable a process to succeed. For example, the proc_exec privilege allows a process to call execve(). Regular users have basic privileges.

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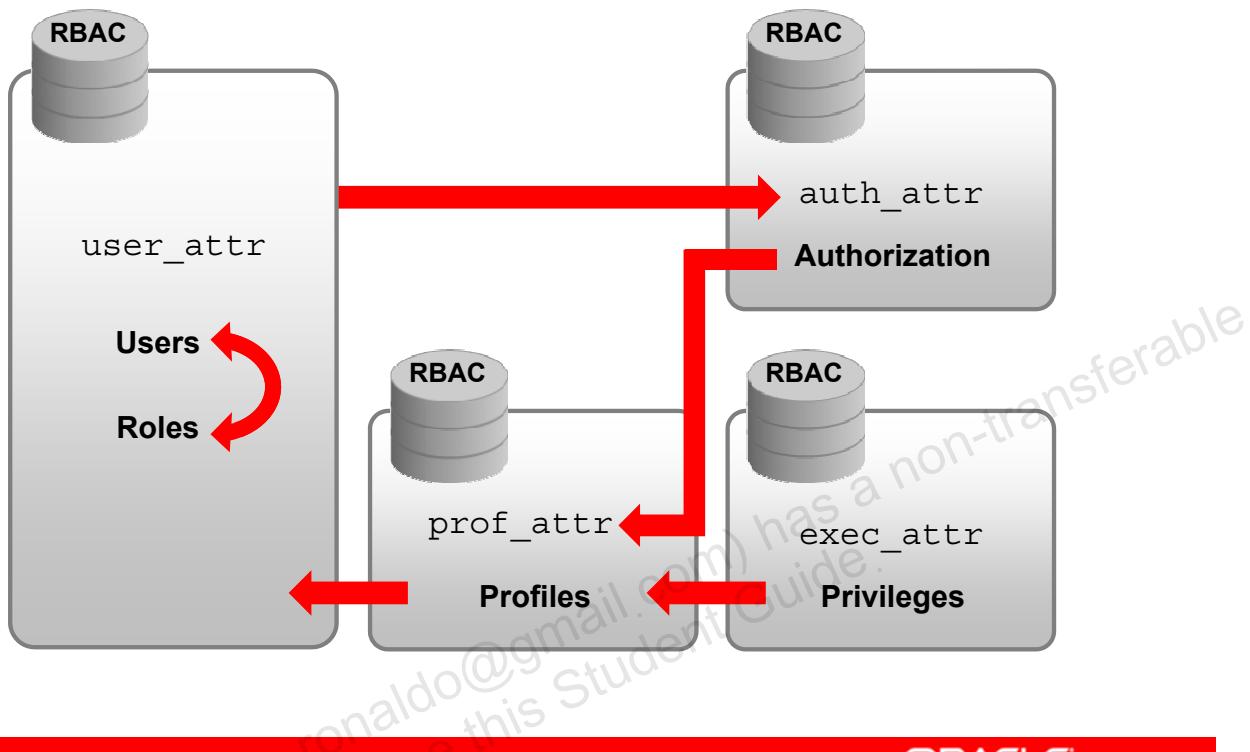
- Enable a process to perform an operation that is otherwise forbidden to regular users
- Include authorizations and privileges and `setuid` and `setgid` programs
- Can be assigned to a user

A security attribute is an attribute that enables a process to perform an operation. In a typical UNIX environment, a security attribute enables a process to perform an operation that is otherwise forbidden to regular users. For example, as seen in the lesson "Managing Services and Service Properties," the `setuid` and `setgid` programs have security attributes. In the RBAC model, authorizations and privileges are security attributes in addition to the `setuid` and `setgid` programs. These attributes can be assigned to a user. For example, a user with the `solaris.device.allocate` authorization can allocate a device for exclusive use. Privileges can be placed on a process. For example, a process with the `file_flag_set` privilege can set immutable, no-unlink, or append-only file attributes.

Key RBAC Files

The roles, rights profiles, authorizations, and privileges commands are defined in four files.

- **user_attr**: Contains the rights profiles and authorizations associated with users and roles that supplement the /etc/passwd and /etc/shadow files
- **auth_attr**: Contains authorization attributes
- **exec_attr**: Contains execution attributes
- **prof_attr**: Contains rights profiles

These files are interrelated as illustrated in the graphic.

Take a closer look at the contents of each file, beginning with the user_attr file.

Interpreting the `user_attr` File

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

The `user_attr` file uses colons (`:`) to separate the fields on each line. The first field is the username as it appears in the `/etc/passwd` and `/etc/shadow` files. The middle fields are reserved for future use, and the last field is a lsist of semicolon-separated (`;`) key-value pairs that describe the security attributes to be applied when the user runs commands.

Interpreting the `auth_attr` File

```
# getent auth_attr | more
solaris.smf.manage.cups:::Manage CUPS service
states::help=ManageCUPS.html
solaris.smf.manage.dt.:::Manage Desktop Service
States::help=ManageDtHeader.html
solaris.smf.manage.dt.login:::Manage Desktop Login Service
States::help=ManageDt
Login.html
solaris.smf.manage.dbus:::Manage D-BUS Service
States::help=SmfDBUSStates.html
solaris.smf.value.tcsd:::Change TPM Administation value properties::
solaris.smf.manage.tcsd:::Manage TPM Administration service states::
solaris.smf.manage.servicetags:::Manage Service Tags Service
States::help=StStat
es.html
solaris.smf.value.servicetags:::Change Service Tag Service Property
Values::help
=StValue.html
solaris.:RO::All Solaris Authorizations::help=AllSolAuthsHeader.html
solaris.account.:RO::Account Management::help=AccountHeader.html
<output omitted>
```

The predefined authorizations are listed in the configuration file for authorization attributes named `auth_attr`, an example of which is shown here. Each entry in the `auth_attr` database consists of one line of text containing six fields separated by colons (`:`). The format of each entry is:

name:*res1*:*res2*:*short_desc*:*long_desc*:*attr*

The description for each field is as follows:

- **`name`:** Name of the authorization. Authorization names are unique strings.
- **`res1`:** The characters RO in this field indicate it is read only and not modifiable by the tools that update this database.
- **`res2`:** Reserved for future use
- **`short_description`:** Short description or terse name for the authorization
- **`long_description`:** Reserved for future use
- **`attr`:** An optional list of semicolon-separated (`;`) key-value pairs that describe the attributes of an authorization. Zero or more keys may be specified. The keyword help, identifies a `help` file in HTML.

Note: Authorizations can end with various suffixes:

- **.read:** Provides read access to user configuration files.
  Example: `solaris.admin.usermgr.read`

- **.write:** Provides write access to user configuration files.
  Example: `solaris.admin.usermgr.write`

- **.pswd:** Provides password access to user configuration files.
  Example: `solaris.admin.usermgr.pswd`

- **.grant:** Permits a user to delegate any assigned authorizations that begin with the same prefix to other users. Example: `solaris.admin.usermgr.grant`

Interpreting the `exec_attr` File

```
# getent exec_attr | grep 'Network Management'
Network
Management:solaris:cmd:RO::/usr/sbin/dladm:euid=dladm;egid=netadm;privs=
sys_dl_config,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=s
ys_dl_config,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_config,net_rawaccess
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/sbin/asppp2pppd:euid=0
Network Management:solaris:cmd:RO::/usr/sbin/ifconfig:uid=0
...
<output truncated>
```

An execution attribute is associated with a rights profile name. An execution attribute can be a command with no options or a script that contains a command with options. Each entry in the `exec_attr` database consists of one line of text containing seven fields separated by colons (`:`). The basic format of each entry is:

   name:*policy*:*type*:*res1*:*res2*:*id*:*attr*

The description for each field is as follows:

- `name`: Name of the profile. Profile names are case-sensitive.
- `policy`: Security policy that is associated with the profile entry. The valid policies are `suser` (standard Solaris superuser) and `solaris`. The `solaris` policy recognizes privileges; the `suser` policy does not.
- `type`: Type of object defined in the profile. The `cmd` type specifies that the ID field is a command that would be executed by a shell.

- *res1*: The characters RO in this field indicate it is read only and not modifiable by the tools that update this database.
- *res2*: Reserved for future use
- *id*: A string that uniquely identifies the object described by the profile. For a profile of type cmd, the ID is either the full path to the command or the asterisk (*) symbol, which is used to allow all commands. An asterisk that replaces the file name component in a path name indicates all files in a particular directory.
- *attr*: An optional list of semicolon-separated (;) key-value pairs that describe the security attributes to apply to the object upon execution. Zero or more keys may be specified. The list of valid keywords depends on the policy enforced. The following keywords are valid: euid, uid, egid, gid, privs, and limitprivs.
    - **euid** and **uid**: Contain a single user name or a numeric user ID. Commands designated with euid run with the effective UID indicated, which is similar to setting the setuid bit on an executable file. Commands designated with uid run with both the real and effective UIDs.
    - **egid** and **gid**: Contain a single group name or a numeric group ID. Commands designated with egid run with the effective GID indicated, which is similar to setting the setgid bit on a file. Commands designated with gid run with both the real and effective GIDs.
    - **privs**: Contains a privilege set that will be added to the inheritable set before running the command
    - **Limitprivs**: Contains a privilege set that will be assigned to the limit set before running the command
    Note: privs and limitprivs are valid only for the solaris policy.

The example in the slide shows the commands and special security attributes for the Printer Management rights profile.

Interpreting the `prof_attr` File

```
# getent prof_attr | more
NTP Management:RO::Manage the NTP service:auths=solaris.smf.manage.ntp,solaris.s
mf.value.ntp
TPM Administration:RO::Administer Privileged TPM Operations:auths=solaris.smf.ma
nage.tcsd,solaris.smf.value.tcsd
D-BUS Management:RO::Manage D-BUS:auths=solaris.smf.manage.dbus;help=RtDBUSMngmn
t.html
DTrace Toolkit::::
Desktop Removable Media User:RO::Access removable media for desktop user:
Console User:RO::Manage System as the Console User:profiles=Desktop Removable Me
dia User,Suspend To RAM,Suspend To Disk,Brightness,CPU Power Management,Network
Autoconf User;auths=solaris.system.shutdown,solaris.device.cdrw,solaris.device.m
ount.removable,solaris.smf.manage.vbiosd,solaris.smf.value.vbiosd;help=RtConsUse
r.html
All:RO::Execute any command as the user or role:help=RtAll.html
Administrator Message Edit:RO::Update administrator message files:auths=solaris.
admin.edit/etc/issue,solaris.admin.edit/etc/motd;help=RtAdminMsg.html
Audit Configuration:RO::Configure Solaris Audit:auths=solaris.smf.value.audit;he
lp=RtAuditCfg.html
Audit Control:RO::Control Solaris Audit:auths=solaris.smf.manage.audit;help=RtAu
ditCtrl.html
Audit Review:RO::Review Solaris Auditing logs:help=RtAuditReview.html
Contract Observer:RO::Reliably observe any/all contract
events:help=RtContractObserver.html
<output omitted>
```

An execution profile is a mechanism that is used to bundle together the commands and authorizations needed to perform a specific function. Each entry in the `prof_attr` database consists of one line of text containing five fields separated by colons (`:`). The format of each entry is:

profname:*res1*:*res2*:*desc*:*attr*
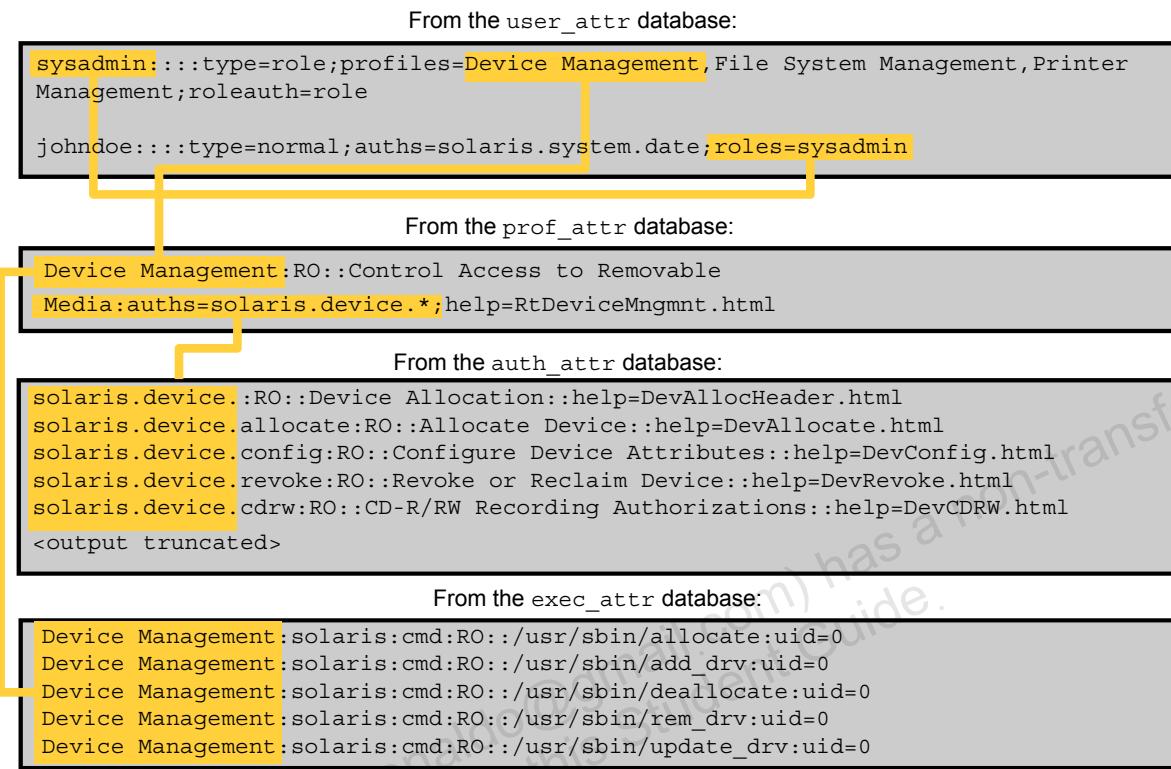
The description for each field is as follows:

- `name`: Name of the profile. Profile names are case-sensitive.
- `res1`: The characters RO in this field indicate it is read only and not modifiable by the tools that update this database.
- `res2`: Reserved for future use
- `desc`: A long description that explains the purpose of the profile, including what type of user would be interested in using it

- *attr:* An optional list of semicolon-separated (`;`) key-value pairs that describe the security attributes to apply to the object upon execution. There are four valid keys: `help`, `profiles`, `auths`, and `privs`.
    - **help:** Assigned the name of a file ending in `.htm` or `.html`
    - **auths:** Specifies a comma-separated list of authorization names chosen from those names defined in the `auth_attr` database. Authorization names can be specified using the asterisk (`*`) character as a wildcard. For example, `solaris.printer.*` would mean all of Oracle Solaris authorizations for printing.
    - **profiles:** Specifies a comma-separated list of profile names chosen from those names defined in the `prof_attr` database
    - **privs:** Specifies a comma-separated list of privileges names chosen from those names defined in the `priv_names` database

Take a look at the profiles for the Auditing feature. Following the profile description, you can see the list of profile attributes that specify what Auditing configurations you can perform.

Relationship Among the Four RBAC Files

From the `user_attr` database:

```
sysadmin::::type=role;profiles=Device Management,File System Management,Printer
Management;roleauth=role

johndoe::::type=normal;auths=solaris.system.date;roles=sysadmin
```

From the `prof_attr` database:

```
Device Management:RO::Control Access to Removable
 Media:auths=solaris.device.*;help=RtDeviceMngmnt.html
```

From the `auth_attr` database:

```
solaris.device.:RO::Device Allocation::help=DevAllocHeader.html
solaris.device.allocate:RO::Allocate Device::help=DevAllocate.html
solaris.device.config:RO::Configure Device Attributes::help=DevConfig.html
solaris.device.revoke:RO::Revoke or Reclaim Device::help=DevRevoke.html
solaris.device.cdrw:RO::CD-R/RW Recording Authorizations::help=DevCDRW.html

<output truncated>
```

From the `exec_attr` database:

```
Device Management:solaris:cmd:RO::/usr/sbin/allocate:uid=0
Device Management:solaris:cmd:RO::/usr/sbin/add_drv:uid=0
Device Management:solaris:cmd:RO::/usr/sbin/deallocate:uid=0
Device Management:solaris:cmd:RO::/usr/sbin/rem_drv:uid=0
Device Management:solaris:cmd:RO::/usr/sbin/update_drv:uid=0
```

ORACLE

Now that you are familiar with the contents of each of the four RBAC files, look at an example of how the fields in the files are related.

The first section of the graphic shows a portion of a `user_attr` file. The user `johndoe` is a normal user account. The user is given the role of `sysadmin`. The `sysadmin` role is a role account. When assuming the `sysadmin` role, `johndoe` has access to specific rights profiles, defined as Device Management, Filesystem Management, and Printer Management profiles.

From the `sysadmin` role entry in the first section to the next section, which is the `prof_attr` file, you can see one relationship between the `user_attr` file and the `prof_attr` file. The Device Management rights profile, which is defined in the `prof_attr` file, is assigned to the `sysadmin` role in the `user_attr` file.

From the second section containing the prof_attr file example, you can see the relationship between the prof_attr and the auth_attr file, a portion of which is displayed in the third section of the graphic. The Device Management profile is defined in the prof_attr file as having all authorizations, beginning with the solaris.device.* string, assigned to it. These authorizations are defined in the auth_attr file.

From the second section containing the prof_attr file example, you can see the relationship between the prof_attr and the exec_attr files, a portion of which is displayed in the fourth section. The Device Management profile is defined in the prof_attr file as having all authorizations, beginning with the solaris.device. string, assigned to it.

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:
    - Searches the role's rights profiles and associated rights
    - Uses the first matching entry if the same command appears in more than one profile
    - Executes the command with the attributes specified in the RBAC configuration files

A profile shell is a special type of shell that enables access to the privileged rights that are assigned to the rights profile. Standard UNIX shells cannot be used because they are not aware of the RBAC files and do not consult them.

Administrators can assign a profile shell to a specific user as a login shell, or the profile shell is started when that user runs the `su` command to assume a role. Users must be assigned one of the following profile shells: `pfsh` for Bourne shell (`sh`), `pfcsh` for C shell (`csh`), or `pfksh` for Korn shell (`ksh`). For the list of profile shells, see the `pfexec`(1) man page.

When the user executes a command, the profile shell searches the role's rights profiles and associated rights. If the same command appears in more than one profile, the profile shell uses the first matching entry. The profile shell executes the command with the attributes specified in the RBAC configuration files.

Implementing the Assigning User Privileges and Roles Plan

Your assignment is to investigate how Oracle Solaris 11:

- Supports process rights management
- Uses RBAC to grant appropriate privileges to users

In accordance with your company's predeployment testing plan, you have been given the task of investigating how Oracle Solaris 11 supports process rights management and uses RBAC to grant appropriate privileges to users.

Quiz

The Oracle Solaris OS implements process rights management through privileges.

a. True

b. False

Answer: a

Quiz

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

a. E

b. I

c. P

d. L

Answer: a

Quiz

Which of the following RBAC files contains rights profiles?

a. user_attr

b. auth_attr

c. exec_attr

d. prof_attr

ORACLE

Answer: d

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
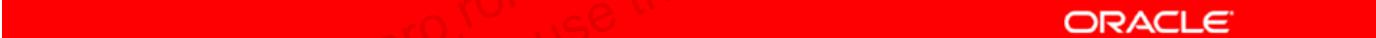
Answer: a

Lesson Agenda

- Planning for User Privileges and Roles Assignments
- **Configuring and Managing Privileges**
- Configuring and Using RBAC

ORACLE

Configuring and Managing Privileges

This section covers the following topics:

- Examining process privileges
- Managing user privileges

Examining Process Privileges

The following are first discussed:

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

ORACLE

Determining the Privileges Available to the Shell

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

```
# ps
 PID TTY       TIME CMD
 990 pts/1   0:00 bash
 993 pts/1   0:00 ps
# ppriv $$
990:  bash
flags = <none>
    E: all
    I: basic
    P: all
    L: all
```

The `ppriv` command is used to inspect or modify process privilege sets and attributes. The double dollar sign (`$$`) passes the process number of the parent shell to the command.

In the example, you run the `ps` command to see what processes are currently running and to verify what shell you are using. Here you can see that you are using the `bash` shell. Next, you run the `ppriv $$` command. Again, you see that the shell is bash. There are no flags set, the effective (`E`), permitted (`P`), and limit (`L`) privilege sets are all set to `all`, and the inherited (`I`) privilege set is set to basic.

Note: The flags field is associated with the getpflags() and setpflags() functions that are used to get or set process flags. The following values are supported.

- **PRIV_AWARE:** This one-bit flag takes the value of 0 (unset) or 1 (set). Only if this flag is set does the current process become privilege-aware. See privileges(5) for a discussion of this flag.
- **PRIV_DEBUG:** This one-bit flag takes the value of 0 (unset) or 1 (set). Only if this flag is set does the current process have privilege debugging enabled.
- **NET_MAC_AWARE** and **NET_MAC_AWARE_INHERIT:** These flags are available only if the system is configured with Trusted Extensions. These one-bit flags each take the value of 0 (unset) or 1 (set).

Determining the Process Privileges 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>
```

ORACLE

Alternatively, you can use the `-v` option with the `ppriv $$` command to display the names of the privileges by privilege set, as shown in this example that contains partial output.

Take a closer look at the privileges in the inheritable (`I`) privilege set. The privileges listed here indicate that you will be able to link to any file, read any file, and write any file. You will have access to the network, which means you will be able to perform network configuration tasks. In addition, you can execute any process, run a process in another subshell (`proc_fork`), display information about any processes, and look at any session in the process.

Determining the Privileges on 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>
```

ORACLE

Use the `ppriv -v` command with the process ID number (PID). The example presents the
partial output for the `cron` process.

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().
#
```

If you need to determine the definition of a privilege that is listed for a process, you can do so by using the `ppriv -vl` command followed by the privilege name. There are two examples: the first is for the `contract_event` privilege and the second is for the `proc_exec` privilege.

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

Now that you are more familiar with how to determine what privileges a process has, look at how to manage user privileges, including how to assign privileges, limit privileges, and debug privilege use.

The most secure way to manage privileges for users and roles is to confine the use of a privilege to commands in a rights profile. The rights profile is then included in a role. The role is assigned to a user. When the user assumes the assigned role, the privileged commands are available to be run in a profile 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.
```

Note: The privileges that are listed in the effective set are in effect throughout your session. If you have been directly assigned privileges in addition to the basic set, the privileges are listed in the effective set.

In this example, the user always has access to the `proc_clock_highres` privilege. This privilege allows a process to use high-resolution timers.

Note: To see the privileges that have been directly assigned to a role, you `su` to the role and then run the `ppriv -v $$` command just as you did for the user account.

Determining the Privileged Commands
That You Can Use

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

```
$ profiles
    Basic Solaris User
    All
$ profiles -l
      All
      *
      Basic Solaris User
    /usr/bin/cdda2wav.bin
  privs=file_dac_read,sys_devices,proc_priocntl,net_privaddr
    /usr/bin/cdrecord.bin
  privs=file_dac_read,sys_devices,proc_lock_memory,proc_priocntl,net_pri
  vaddr
    /usr/bin/readcd.bin    privs=file_dac_read,sys_devices,net_privaddr
```

When a user is not directly assigned privileges, the user obtains access to privileged commands through a rights profile. Commands in a rights profile must be executed in a profile shell. To determine which privilege commands you can use or run, you need to see which rights profiles have been assigned to you. To do this, you use the `profiles` command. To see more details about the privileges, you can use the `profiles -l` command.

Note: To see the details of a specific privilege, you use the `profiles -l` command with the privilege name, as in this example:

    $ profiles -l Basic Solaris User

To see what roles and authorization privileges you have, you use the `roles` and `auth` commands, respectively, as in this example:

    $ roles

    No roles

    $ auths

    solaris.admin.wusb.read,solaris.device.cdrw,solaris.device.mount
    .removable,solaris.mail.mailq,solaris.profmgr.read

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:::::type=normal;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:::::type=role;defaultpriv=proc_clock_highres
```

You might want to assign a user or role with a particular privilege all the time. Very specific privileges that affect a small part of the system are good candidates for assigning to a user or role. To assign privileges to a user, you use the `usermod -K` command followed by the `key=value` pair you want to assign and the user's login name.

Note: The `-K key=value` option is used to replace or add to a user's or role's `key=value` pair attributes. See `user_attr`(4) for a list of valid `key=value` pairs.

In the example, you enable user `jjones` to use high-resolution timers by assigning the `proc_clock_highres` privilege to his basic default privileges. The values for the `defaultpriv` keyword replace the existing values. Therefore, for the user to retain the `basic` privileges, the value `basic` must be specified. In the default configuration, all users have basic privileges. To verify that the privilege has been assigned, you look at the `user_attr` entry for `jjones`. Here you can see how the privileges have been modified.

To assign privileges to a role, the same logic applies. You use the `rolemod -K` command followed by `key=value` pair you want to assign and the role name. In the role example, you use the same example, changing the user to role as appropriate. The role name is `realtime`.

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:::::type=normal;defaultpriv=basic;limitpriv=all,!sys_linkdir
```

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

There may be circumstances in which you want to limit the privileges that are available to a user or role. You can do this by reducing the basic set or by reducing the limit set. However, you should have a very good reason why you want to limit the privileges, because such limitations can have unintended side effects. To limit the privileges of a user or role, follow the steps listed in the slide.

Caution for step 2: Do not remove the proc_fork or the proc_exec privilege. Without these privileges, the user cannot use the system. In fact, these two privileges are only reasonably removed from daemons that do not fork() or exec() other processes.

Notes for step 3: You must thoroughly test any user's or role's capabilities where you have modified the basic set or the limit set for a user or role. It is possible to prevent a user or role from being able to use the system when the basic set is less than the default. When you modify the limit set to be less than all privileges, it is possible for processes that need to run with an effective UID=0 to fail.

In the first example, all sessions that originate from jjone's initial login are prevented from using the sys_linkdir privilege. After this change is implemented, the user jjones will no longer be able to make hard links to directories or unlink directories even after he runs the su command. The same scenario is used in the second example for a role.

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

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

2. Determine which system call is failing by finding the `syscall` number in the `/etc/name_to_sysnum` file.

```
$ ppriv -eD touch /etc/acct/yearly
touch[5245]: missing privilege "file_dac_write"
   (euid = 130, syscall = 224) needed at zfs_zaccess+0x258
touch: cannot create /etc/acct/yearly: Permission denied
$ grep 224 /etc/name_to_sysnum
creat64        224
```

ORACLE

The Oracle Solaris OS provides two tools to debug privilege failure: the `ppriv` debugging command (`ppriv -eD`) and the `truss` command.

Note: The `-e` option with the `ppriv` command interprets the remainder of the arguments as a command line and runs the command line with specified privilege attributes and sets. The `-D` option turns on privilege debugging for the process or command supplied.

The steps for using the `ppriv` debugging command on a failed command or process are listed in the slide.

In the example, `ppriv -eD touch` is being used to determine why the command `/etc/acct/yearly` has failed. The output indicates that the missing privilege is `file_dac_write` and provides the `euid` and system call information. To determine which system call is failing, you take the syscall number from the debugging output and locate it in the `/etc/name_to_sysnum` file. Here you can see that the system call `create64` is failing.

When you know the missing privilege, you can assign it to the program as needed.

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
```

ORACLE

The `ppriv` command can debug privilege use in a profile shell. If you assign a rights profile to a user, and the rights profile includes commands with privileges, the commands must be entered in a profile shell. When the privileged commands are entered in a regular shell, the commands do not execute with privilege.

In this example, the `jjones` user can assume the `objadmin` role. The `objadmin` role includes the Object Access Management rights profile. This rights profile allows the `objadmin` role to change permissions on files that `objadmin` does not own. In the example, `jjones`'s attempt to change the permissions on the `useful.script` file fails. The user then runs the `ppriv` debugging command to determine why the command failed and is shown that the `file_chown` privilege is missing.

To fix this issue, you assign the `file_chown` privilege to the `jjones` user.

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

```
$ truss touch /etc/acct/yearly
execve("/usr/bin/touch", 0x08047E74, 0x08047E80) argc = 2
sysinfo(SI_MACHINE, "i86pc", 257)                    =6
mmap(0x00000000, 32, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANON, -1,0) =
0xFEFB0000
mmap(0x00000000, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANON, -1,0)
= 0xFEFA0000
mmap(0x00000000, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANON, -1,0)
= 0xFEF90000
mmap(0x00000000, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANON, -1,0)
= 0xFEF80000
memcntl(0xFEFB7000, 32184, MC_ADVISE, MADV_WILLNEED, 0, 0) = 0
memcntl(0x08050000, 4216, MC_ADVISE, MADV_WILLNEED, 0, 0) = 0
resolvepath("/usr/lib/ld.so.1", "/lib/ld.so.1", 1023) = 12
resolvepath("/usr/bin/touch", "/user/bin/touch", 1023) = 14
sysconfig(_CONFIG_PAGESIZE                    = 4096
stat64("/usr/bin/touc", 0x08047A10)          = 0
open("/var/ld/ld.config", )_RDONLY)          ERR#2 ENOENT
<output omitted>
close(3)
_exit(0)
```

The `truss` command can debug privilege use in a regular shell, as shown in the example, where you are using the `truss` command to debug the failing touch process.

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

This practice covers the following topics:

- Examining process privileges
- Managing user privileges

The practices for this lesson are designed to reinforce the concepts that have been presented in the lecture portion. These practices cover the following tasks:

- **Practice 8-1:** Delegating privileges to users and processes
- **Practice 8-2:** Configuring role-based access control

Practice 8-1 should take you about 30 minutes to complete.

Lesson Agenda

- Planning for User Privileges and Roles Assignments
- Configuring and Managing Privileges
- **Configuring and Using RBAC**

ORACLE

Configuring and Using 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 system-wide 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:102:1:Level One Support:/export/home/level1:/bin/pfsh
# grep level1 /etc/shadow
level1:CUs8aQ64vTrZ.:12713::::::
# getent user_attr | grep level1
level1::::type=role;profiles=Printer Management,Media Backup,Media
Restore
```

To create a role, you use the `roleadd` command combined with one or more options. The more common options are as follows:

- `-u uid`: Specifies the user ID of the new role
- `-g gid`: Specifies an existing group's integer ID or character-string name
- `-m`: Creates the new role's home directory if it does not already exist
- `-d dir`: Specifies the home directory of the new role
- `-c comment`: Text string that provides a short description of the role
- `-P profile`: Assigns rights profiles to the role. Use commas (`,`) to separate multiple rights profiles.
- `rolename`: Name of the new role. For restrictions on acceptable strings, see the `roleadd` (1M) man page.

Note: To create a role, you must be an administrator with the User Management rights profile. To assign a password to the role, you must be assigned the User Security rights profile.

The `roleadd` command creates a role entry in the `/etc/passwd`, `/etc/shadow`, and `user_attr` files. In this example, the `roleadd` command creates a new role called `level1`, builds the home directory, and assigns the role with rights profiles of `Printer Management`, `Media Backup`, and `Media Restore` to the user ID `3000` and group ID `10`. The role cannot be used until a password is applied to it.

Note: The installation of the Oracle Solaris 11 OS has the Printer Management, Media Backup, and Media Restore rights profiles already defined in the `exec_attr` and `prof_attr` files, so there is no need to add an entry for these profiles in these two files. However, if you had created a new rights profile, you would need to make a new entry in the `prof_attr` file. You will look at how to do that next.

The changes to the `/etc/passwd`, `/etc/shadow`, and `user_attr` files are shown in the example. The type of this account is `role` (`type=role`) and includes the rights profiles `Printer Management`, `Media Backup`, and `Media Restore`.

Creating a Rights Profile

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

Creating a Rights Profile: Example

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

In this example, the administrator creates a rights profile for Sun Ray users in the LDAP repository. The administrator has already created a Sun Ray version of the Basic Solaris User rights profile, and has removed all rights profiles from the policy.conf file on the Sun Ray server. The administrator verifies the contents.

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.

The rights profiles that Oracle Solaris provides are read-only. You can clone a provided rights profile for modification if its collection of security attributes is insufficient. For example, you might want to add the `solaris.admin.edit/path-to-system-file` authorization to a provided rights profile.

Creating or Changing 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> end
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
```

In this example, the administrator adds several solaris.admin.edit authorizations to a site IPsec Management rights profile. The administrator verifies that the Network IPsec Management rights profile cannot be modified. Then, the administrator creates a rights profile that includes the Network IPsec Management profile. The administrator verifies the contents.

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
```

To assign a rights profile to a role, use the `rolemod` command. The `rolemod` command changes the definition of the specified role and makes the appropriate login-related changes to the system file and file system.

Note: The `rolemod` command modifies the entry for the specified role in the `/etc/passwd`, `/etc/shadow`, and `user_attr` files.

You can use the following options with the `rolemod` command:

- `-e expire`: Date a role expires. Use this option to create temporary roles.
- `-l new_logname`: Specifies the new login name for the role
- `-P profile`: Specifies one or more comma-separated rights profiles, as defined in the `prof_attr` file
- `-s shell`: Login shell for `rolename`. This shell must be a profile shell.
- `Rolename`: Name of the role you are modifying

In the example, the `profile1` and `profile2` profiles and the `/usr/bin/pfksh` profile shell are assigned to the role named `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` file.

A user can have access to many roles. The `useradd` command can be used to define which roles a new user has access to. To add roles to an existing user account, you use the `usermod` command as shown in the steps displayed in the slide.

Notes for step 2: If you are assigned the User Security rights profile, you can create the password. Otherwise, a user who is assigned the role must create it by using the `su –` *rolename* command. Typically, because a role account is assigned to more than one user, the superuser creates a role password and provides the users with the password.

Note: To remove all role access from a user account, you use the `usermod` command with the `-R  ""` option followed by the user login name.

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::::type=normal;roles=level1
# roles paul
level1
# usermod -R level1 chris
# passwd -r repository level1
Password: <Type rolename password>
Confirm Password: <Retype rolename password>
# usermod -R "" chris
```

The example in the slide shows the useradd command being used with the -R option to define the level1 role for the user paul. To verify that the level1 role has been assigned to paul, you view the user_attr file for the user paul. Here you can see that the entry for paul has the level1 role. You can also use the roles command to see the roles that are assigned to the paul user.

Note: The association between the paul user account and the level1 role is defined automatically in the user_attr file.

Next, you are assigning the level1 role to the existing user account chris by using the usermod -R command. In the last line you are removing all role access from the chris account by using the usermod -R "" command.

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
```

You can use the steps listed in the slide to assume a role.

Notes for step 4: In contrast to the `root` role, the System Administrator role has the basic set of privileges in its effective (E) set.

In the example shown in the slide, you first determine which role you can assume. You then assume the role of System Administrator. You then verify that you have assumed the System Administrator role. Your final step is to view the capabilities for your role, which (as you can see) are all `basic` except for the limit (L) privilege set, which by default is `all`.

Note: The command prompts displayed might differ based on the shell you are using.

Restricting an Administrator to Explicitly Assigned Rights

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

- You can use the Stop rights profile.
- You can 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
```

You can restrict a role or user to a limited number of administrative actions in two ways.

- You can use the Stop rights profile. The Stop rights profile is the simplest way to create a restricted shell. The authorizations and rights profiles that are assigned in the `policy.conf` file are not consulted. In the default configuration, the role or user is not assigned the Basic Solaris User rights profile, the Console User rights profile, or the `solaris.device.cdrw` authorization.
- You can modify the `policy.conf` file on a system, and require the role or user to use that system for administrative tasks.

The `rolemod -P` command is used with the Stop rights profile, as shown in the example. This command is especially useful if you have many profiles assigned to a role and you want to limit the role to only a few profiles.

Assigning the Rights Profile to a User

```
# profiles chris
Basic Solaris User
All
# usermod -P "Printer Management" chris
# profiles chris
Printer Management
Basic Solaris User
All
# getent user_attr | grep chris
chris::::type=normal;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:
*
```

ORACLE

The rights profiles assigned to a user can be listed with the `profiles` command. Every account has the `All` rights profile. It allows any command to be executed but with special security attributes.

Note: Other rights profiles given to all new user accounts are defined in the `/etc/security/policy.conf` file.

To assign a rights profile to a user, you use the `usermod` command. This example shows the Printer Management rights profile being assigned to the `chris` user account. If you run the `profiles` command again for the user, you can see that the `Printer Management` rights profile has been added.

The `usermod` command automatically updates the `user_attr` file for the specified user, as shown in the example. The new line for the user `chris` shows the new profile assignment.

You can examine the contents of a rights profile with the `-l` option of the `profiles` command. The individual commands in the rights profile can be seen, along with the special security attributes with which they are executed. This example shows the user `chris` being able to enable and disable a printer.

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` file for the user.
3. View the authorizations for the user by using the `auths` command.

Authorizations can be assigned to user accounts. Authorizations can also be assigned to roles or embedded in a rights profile that can be assigned to a user or role.

To delegate an authorization to a user, you use the `usermod` command with the `-A` option, the authorization, and the user login name.

Note: Only a user or role who has grant rights to the authorization can assign it to an account. The `roleadd` command automatically updates the `user_attr` file.

To verify that the authorization has been assigned to the user, you can check the `user_attr` file. You can also use the `auths` command for the user to see if the authorization is listed in the entry.

Delegating an Authorization to a User: Example

```
# su - chris
Oracle Corporation    SunOS 5.11  11.0     November 2011
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:::::type=normal;auths=solaris.jobs.admin;profiles=Printer Management
# auths chris
solaris.admin.printer.read,solaris.admin.printer.modify,solaris.admin.pri
nter.delete,solaris.device.cdrw,solaris.profmgr.read,solaris.jobs.users,s
olaris.mail.mailq,solaris.admin.usermgr.read,solaris.admin.logsvc.read,so
laris.admin.fsmgr.read,solaris.admin.serialmgr.read,solaris.admin.diskmgr
.read,solaris.admin.procmgr.user,solaris.compsys.read,solaris.admin.prodr
eg.read,solaris.admin.dcmgr.read,solaris.snmp.read,solaris.project.read,s
olaris.admin.patchmgr.read,solaris.network.hosts.read,solaris.admin.volmg
r.read
# su - chris
Oracle Corporation     SunOS 5.11  11.0     November 2011
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
```

In this example, a regular user is not permitted to look at another user's `crontab` file. To grant the user authorization to manage the other user's `crontab` file you use the `usermod` command with the `-A` option to add an authorization. The `user_attr` file is automatically modified with this new information. The `chris` account is a normal user account (`type=normal`). You can see in the `user_attr` file that `chris` has had the `solaris.jobs.admin` authorization and the `Printer Management` rights profile added previously. Next, you use the `auths` command to see the authorizations assigned to `chris`. With this authorization, `chris` can now view or modify other users' `crontab` files.

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` file for the role.
3. View the authorizations for the role by using the `auths` command.

```
# rolemod -A "solaris.admin.usermgr.*" level2
# auths level2
solaris.admin.usermgr.*
```

If a large number of user accounts require the same configuration and management of authorizations, it can be easier to assign the authorizations to a role and give the users access to the role. You can assign the authorization to the role by using the `rolemod -A` command. The steps for completing this task are listed in the slide.

Note: The `rolemod` command automatically updates the `user_attr` file.

In the example, the `solaris.admin.usermgr.*` authorization is being assigned to the `level2` role. You use the `auths` command to verify that the authorization has been assigned to the role.

Modifying a System-wide RBAC Policy

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

```
# vi /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    11.0         November 2011
jjones:~$ ps -A -o user -o pid -o comm | more
   USER  PID COMMAND
 jjones 1941 ps
 jjones 1935 -bash
```

The `/etc/security/policy.conf` file establishes a system-wide RBAC policy. There are two different settings for the system-wide policy: `PRIV_DEFAULT`, which determines the default, and `PRIV_DEFAULT=basic,!file_link_any`, which can be used to modify the default. The default is set to `PRIV_DEFAULT=basic`. You can modify this file to deny non-administrative users specific privileges. The steps for performing this task are listed in the slide.

The example shows how to deny a non-administrative user the privilege to look at the processes of other users. You edit the `PRIV_DEFAULT=basic` entry as follows:

    PRIV_DEFAULT=basic, !proc_info, !proc_session

For the changes to the policy to take place, you reboot the system. After you log back in to the system, you `su` to the `jjones` user account and issue the command to access the processes. The only processes the user can display are the user's own processes.

Note: The `-A` and `-o` options used in the `ps` command are telling the system to write information for all processes in the specified format, which in the example is by `user`, `pid`, and `command`.

Practice 8-2 Overview:
Configuring Role-Based Access Control

This practice covers the following topics:

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

ORACLE

This practice should take you about 30 minutes to complete.

Summary

In this lesson, you should have learned how to:
- Implement a plan to configure privileges
- Implement a plan to configure role-based access control
- Configure privileges
- Manage privileges
- Configure role-based access control
- Use role-based access control

Securing System Resources by Using Oracle Solaris Auditing

ORACLE

Objectives

After completing this lesson, you should be able to:

- Implement a plan for Oracle Solaris auditing
- Configure Oracle Solaris auditing
- Administer the audit service
- Manage audit records

Workflow Orientation

Before you begin the lesson, take a moment to orient yourself in your job workflow. You have successfully installed the operating system and have updated it. You have configured the data storage environment as well as the physical and virtual networks. You have also ensured that all the system services are up and running and that both users and processes have been granted the appropriate level of privilege. In order to monitor proper use of business resources and assigned privileges, the Oracle Solaris 11 OS provides several security features, one of which is the Oracle Solaris audit service. It is the system administrator's responsibility to configure, administer, and manage this service.

Lesson Agenda

- **Planning for Oracle Solaris Auditing**
- Configuring Oracle Solaris Auditing
- Administering the Audit Service
- Managing Audit Records on Local Systems

ORACLE

Planning for Oracle Solaris Auditing

- Determine if you want a single-system image audit trail.
- Determine the audit policy.
- Determine if you want to modify event-to-class mappings.
- Determine which audit classes to preselect.
- Determine user exceptions to the system-wide preselections.
- Decide how to manage the audit_warn email alias.
- Decide in which format and where to collect audit records.
- Determine when to warn the administrator about shrinking disk space.
- Decide what action to take when all the audit directories are full.
- Determine how much storage space to allocate to auditing.

ORACLE

As with all companies, your company is concerned with ensuring that their system resources are kept secure. As part of investigating ways to keep the system resources secure, your company wants to evaluate the Oracle Solaris auditing service. By using the audit service, your company hopes to be able to monitor and record specific, security-related events. They also want to be able to detect suspicious activities by reviewing patterns of access and access histories as well as discover attempts to circumvent the protections that have been put in place to safeguard the system. In short, they want to keep a log of what was done, when it was done, by whom, and what was affected.

Your company recognizes that setting up auditing takes a considerable amount of planning and, as a result, they have put together a plan that addresses each of the requirements listed in the slide. As the system administrator responsible for configuring, administering, and managing the Oracle Solaris audit service, you will need this information to do your job.

In this topic you are introduced to Oracle Solaris auditing and shown how the audit service addresses each of these requirements.

Oracle Solaris Auditing

Oracle Solaris auditing is:

- A service controlled by the audit daemon, `auditd`

- Enabled by default

- Configured to provide the following defaults when first enabled:
  - All login events are audited.
  - All users are audited for login, logout, and role assumption events.
  - The `audit_binfile` plug-in is active.
  - The `cnt` audit policy is set.
  - These audit queue controls are set:
    - Maximum number of records before records lock: 100
    - Maximum number of records before blocked auditing process unblock: 10
    - Buffer size: 8192 bytes
    - Interval for writing records to the audit trail: 20 seconds
  - All zones are audited identically.

Oracle Solaris auditing is a service. The audit service is controlled by the audit daemon, `auditd`, and is enabled by default.

Note: The audit daemon controls the generation and location of audit trail files and the generation of `syslog` messages based on its configuration.

When the audit service is first enabled, the following defaults are provided:

- All login events are audited. Both successful and unsuccessful login attempts are audited.

  Note: An event is a security-related system action that is audited.

- All users are audited for login, logout, and role assumption events.

- The `audit_binfile` plug-in is active. The `/var/audit` directory stores audit records, the size of an audit file is not limited, and the queue size is 100 records.

  Note: An audit plug-in is a module that transfers the audit records in the audit queue to a specified location. The `audit_binfile` plug-in creates binary audit files (the audit trail). The audit trail is a collection of one or more audit files that store the audit data from all systems that run the audit service by using the default plug-in, `audit_binfile`. You will learn more about the audit plug-ins later in this lesson.

- The `cnt` audit policy is set. This policy has the following effect: When audit records fill the available disk space, the system keeps a count of the number of dropped audit records. No warning is issued.

  Note: The audit policy is a set of auditing options that you can enable or disable at your site. The `cnt` policy is one option. These options include whether to record certain kinds of audit data. The options also include whether to suspend auditable actions when the audit queue is full. You will take a closer look at the audit policy shortly.

- The following audit queue controls are set:
    - Maximum number of records in the audit queue before generating the records locks: 100
    - Minimum number of records in the audit queue before blocked auditing processes unblock: 10
    - Buffer size for the audit queue: 8192 bytes
    - Interval between writing audit records to the audit trail: 20 seconds
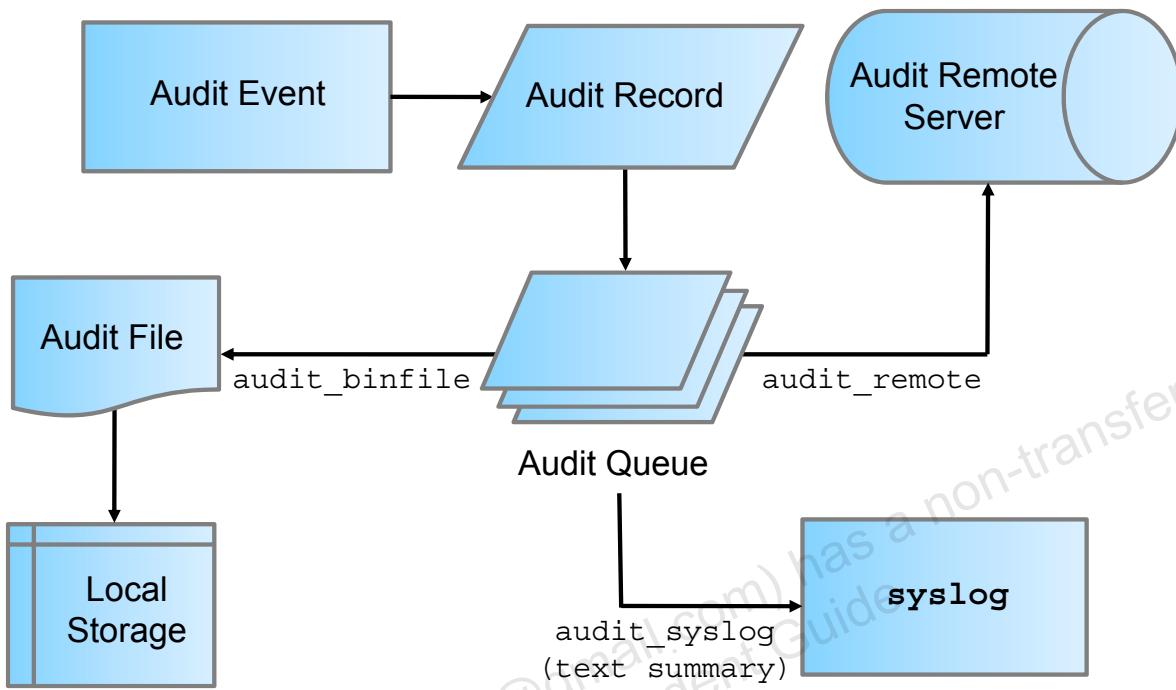
By default, all zones are audited identically.

Note: You will be shown how to configure zones for auditing identically and on a per-zone basis later in this lesson.

Rights profiles control who can administer the audit service. There are rights profiles for configuring the audit service, for enabling and disabling the service, and for analyzing the audit trail. The System Administration rights profile includes the Audit Review rights profile. A role with the System Administrator rights profile can analyze audit records.

Oracle Solaris Auditing

The auditing process begins when a specified, security-related audit event occurs that generates an audit record.

Note: The most common audit events are:

- System startup and system shutdown
- Login and logout
- Process creation or process destruction, or thread creation or thread destruction
- Opening, closing, creating, destroying, or renaming of objects
- Use of privilege capabilities or RBAC
- Identification actions and authentication actions
- Permission changes by a process or user
- Administrative actions, such as installing a package
- Site-specific applications

Each audit record contains information that identifies the event, what caused the event, the time of the event, and other relevant information. This record is then placed in an audit queue for the active plug-ins to retrieve. The active plug-ins can include the default plug-in, `audit_binfile`, the `audit_remote` plug-in, and the `audit_syslog` plug-in. The `audit_binfile` plug-in writes the records to audit files. These audit records are stored locally in binary format. The `audit_remote` plug-in sends these records to an audit remote server, and the `audit_syslog` plug-in sends text summaries to the `syslog` utility.

Now that you have a high-level understanding of how Oracle Solaris auditing works, take a closer look at each part of the process, beginning with audit events.

Interpreting the
`/etc/security/audit_event` File

```
number:name:description:flags
```

Each entry in the file contains four fields:

- *number*: Event number
- *name*: Event name
- *description*: Event description
- *flags*: Specify classes to which the event is mapped

Examples:

```
6153:AUE_logout:logout:lo
6161:AUE_reboot_solaris:reboot(1m):ss
6180:AUE_prof_cmd:profile command:ua,as
6207:AUE_create_user:create user:no
```

As discussed, audit events represent auditable actions on a system. Audit events are listed in the `/etc/security/audit_event` file. The `/etc/security/audit_event` file entry format is shown in the slide. Each entry in the file contains four fields, with a colon separating each field. Each event appears on its own line. The following is the format for an entry:

> `number:name:description:flags`

The description and requirement for each field are as follows:

- *number*: Event number. Event number ranges are assigned as follows:
    - 0: Reserved as an invalid event number
    - 1 – 2047: Reserved for the Solaris Kernel events
    - 2048 – 6143: Reserved for user-level audit events
    - 6000 – 7999: Allocated for Solaris user-level audit events, includes SMF-related, `ilbd`, `netcfgd`, TCSD, and `hotplugd` events
    - 9035 – 9201: Reserved for the Solaris Trusted Extensions events
- *name*: Event name

- *description*: Event description
- *Flags*: Flags specifying classes to which the event is mapped. Classes are comma-separated, without spaces.

Note: In addition to the audit events that are defined by the Oracle Solaris audit service, third-party applications can generate audit events.

Each of the examples is a Solaris user-level audit event. The first event example, AUE_logout, tracks when a user logs out of the system. lo is the audit_class designation for login or logout. The second event example, AUE_reboot_solaris, tracks when a user reboots the operating system. ss is the audit_class designation for a change in the system state. The third event example, AUE_prof_cmd, tracks when a user executes the profile command. ua and as are the audit_class designations for user administration and system-wide administration respectively. The last event example, AUE_create_user, tracks when a user executes the user create command. no audit_class designation indicates that this is an invalid class and any event mapped solely to this class will not be audited.

Event Types

- **Synchronous:** Events associated with a process in the system
- **Asynchronous:** Events not associated with any process, so no process is available to be blocked and later woken up
- **Attributable:** Events attributed to a user. All attributable events are synchronous events.
- **Non-attributable:** Events that occur at the kernel-interrupt level or before a user is authenticated. Most non-attributable events are asynchronous events.

ORACLE

Oracle Solaris auditing handles these types of events:

- **Synchronous:** Events that are associated with a process in the system. Synchronous events are the majority of system events.
- **Asynchronous:** Events that are not associated with any process, so no process is available to be blocked and later woken up. Initial system boot and PROM enter and exit events are examples of asynchronous events.
- **Attributable:** Events that can be attributed to a user. The `execve()` system call can be attributed to a user, so the call is considered an attributable event. All attributable events are synchronous events.
- **Non-attributable:** Events that occur at the kernel-interrupt level or before a user is authenticated. The `na` audit class handles audit events that are non-attributable. For example, booting the system is a non-attributable event. Most non-attributable events are asynchronous events. However, non-attributable events that have associated processes, such as failed login, are synchronous events.

Interpreting the
`/etc/security/audit_class` File

```
mask:name:description
```

Each entry in the file contains four fields:

- `mask`: Class mask
- `name`: Class name
- `description`: Class description

Examples:

```
0x00001000:lo:login or logout
0x00010000:ss:change system state
0x00040000:ua:user administration
0x00020000:as:system-wide administration
0x00000000:no:invalid class
0xffffffff:all:all classes (meta-class)
```

Each audit event belongs to an audit class or classes. Audit classes are convenient containers for large numbers of audit events. Audit classes are defined in the `/etc/security/audit_class` file. The `/etc/security/audit_class` file entry format is shown in the slide. Each entry in the file contains three fields, with a colon separating each field. The following is the format for an entry:

> `mask:name:description`

The description and requirement for each field are as follows:

- `mask`: Class mask
- `name`: Class name
- `description`: Class description

Each class is represented as a bit in the class mask, which is an unsigned integer. There are 32 different classes available. Meta-classes can also be defined. You can have supersets composed of multiple base classes, which will have more than 1 bit in the mask.

Two special meta-classes are also predefined: `all` and `no`.

- `all`: Represents a conjunction of all allowed classes and is provided as a shorthand method of specifying all classes

- `no`: Represents an invalid class. Any event mapped solely to this class will not be audited. Turning auditing on to the `all` meta-class will not cause events mapped solely to the `no` class to be written to the audit trail. This class is also used to map obsolete events that are no longer generated. Obsolete events are retained to process old audit trails files.

The examples show the audit classes that you saw associated with the previous audit event examples: login or logout (`lo`), change system state (`as`), user administration (`ua`), system-wide administration (`as`), and invalid class (`no`). An example of the `all` audit class is also included.

Displaying the
`/etc/security/audit_class` File

```
# cat /etc/security/audit_class
<header output omitted>
0x00000000:no:invalid class
0x00000001:fr:file read
0x00000002:fw:file write
0x00000004:fa:file attribute access
0x00000008:fm:file attribute modify
0x00000010:fc:file create
0x00000020:fd:file delete
0x00000040:cl:file close
0x00000100:nt:network
0x00000200:ip:ipc
0x00000400:na:non-attribute
0x00001000:lo:login or logout
0x00004000:ap:application
0x00008000:cy:cryptographic
0x00010000:ss:change system state
<continued on next page>
```

ORACLE

The default list of audit classes as they appear in the `/etc/security/audit_class` file is shown in the slide.

Displaying the
`/etc/security/audit_class` File

```
<continued from previous page>
0x00020000:as:system-wide administration
0x00040000:ua:user administration
0x00070000:am:administrative (meta-class)
0x00080000:aa:audit utilization
0x000f0000:ad:old administrative (meta-class)
0x00100000:ps:process start/stop
0x00200000:pm:process modify
0x00300000:pc:process (meta-class)
0x00400000:xp:X - privileged/administrative operations
0x00800000:xc:X - object create/destroy
0x01000000:xs:X - operations that always silently fail, if bad
0x01c00000:xx:X - all X events (meta-class)
0x20000000:io:ioctl
0x40000000:ex:exec
0x80000000:ot:other
0xffffffff:all:all classes (meta-class)
```

A continuation of the default audit classes is shown in the slide.

Audit Class Preselection

- Preselection is the choice of which audit classes to monitor.
- Preselected audit class events are collected in the audit queue.
- You can preselect events that specify:
  - System-wide auditing defaults (system-wide audit mask)
  - Exceptions for individual users (user-specific audit mask)
- When combined, these preselections constitute the process preselection mask.

Preselection is the choice of which audit classes to monitor. The audit events of preselected audit classes are collected in the audit queue. Audit classes that are not preselected are not audited, so their events do not appear in the queue. For example, when you preselect the `ps` and `na` audit classes, `execve()` system calls and system boot actions, among other events, are recorded.

You can specify system-wide auditing defaults (referred to as the system-wide audit mask) by preselecting events on a system, and you can specify exceptions to the system-wide auditing defaults for individual users by preselecting events initiated by a particular user (referred to as the user-specific audit mask). When combined, these preselections constitute the process preselection mask. When a user logs in, the login process combines the preselected classes to establish the process preselection mask for the user's processes. The process preselection mask specifies whether events in each audit class are to generate audit records.

Note: After the audit service is enabled, you can change the preselections.

You are shown how to modify the preselection mask later in this lesson.

Audit Records and Audit Tokens

- Audit record:
  - Records the occurrence of a single audited event
  - Includes the following information:
    - Who performed the action
    - Which files were affected
    - What action was attempted
    - Where and when the action occurred
- Audit token:
  - Defines the type of information saved for each audit event
  - Which tokens are recorded is determined by the type of event.

```
header,69,2,login - local,,example_system,2011-12-16 10:10:10.020 -07:00
subject,root,root,other,root,other,378,378,1234567891 2 example_system
return,success,0
```

ORACLE

Each audit record records the occurrence of a single audited event. The record includes information such as who did the action, which files were affected, what action was attempted, and where and when the action occurred.

The type of information that is saved for each audit event is defined by a set of audit tokens. Each time an audit record is created for an event, the record contains some or all of the tokens that are defined for the event. The nature of the event determines which tokens are recorded.

An audit record always begins with a `header` token. The `header` token indicates where the audit record begins in the audit trail. In the case of attributable events, the `subject` and the `process` tokens refer to the values of the process that caused the event. In the case of non-attributable events, the process token refers to the system. Each audit token has a token type identifier, which is followed by data that is specific to the token. Each token type has its own format.

Note: For a listing of the audit token formats, see the "Oracle Solaris Auditing (Reference)" chapter in *Oracle Solaris Administration: Security Services.*

To display the tokens that comprise an audit record, you use the `auditrecord -e event` command.

The example in the slide shows the login audit record, which comprises the following tokens:

- `Header`, which marks the beginning of the audit record and contains the following:
    - Record bite count (`69`)
    - Version number (`2`)
    - Audit event type (`login - local`)
    - System on which the event occurred (`example_system`)
    - Date/time stamp (`2011-12-16 10:10:10.020 -07:00`):
      `header,69,2,login - local,,example_system,2011-12-16 10:10:10.020 -07:00`

- `subject`, which describes a user who performs or attempts to perform an operation:
  `subject,root,root,other,root,other,378,378,1234567891 2 example_system`

- `Return`, which contains the return status of the system call (`u_error`) and the process return value (`u_rval1`): `return,success,0`

  Note: The `return` token is always returned as part of kernel-generated audit records for system calls. In application auditing, this token indicates exit status and other return values.

Audit Plug-in Modules

- An audit plug-in transfers an audit record from the audit queue to a specified location.
- The audit services provides these plug-ins:
  - `audit_binfile`: Delivers audit records from the audit queue to the binary audit files
  - `audit_remote`: Delivers audit records from the audit queue to a configured remote server
  - `audit_syslog`: Delivers selected records from the audit queue to the `syslog` log
- At least one plug-in must be active.
- By default, the `audit_binfile` plug-in is active.
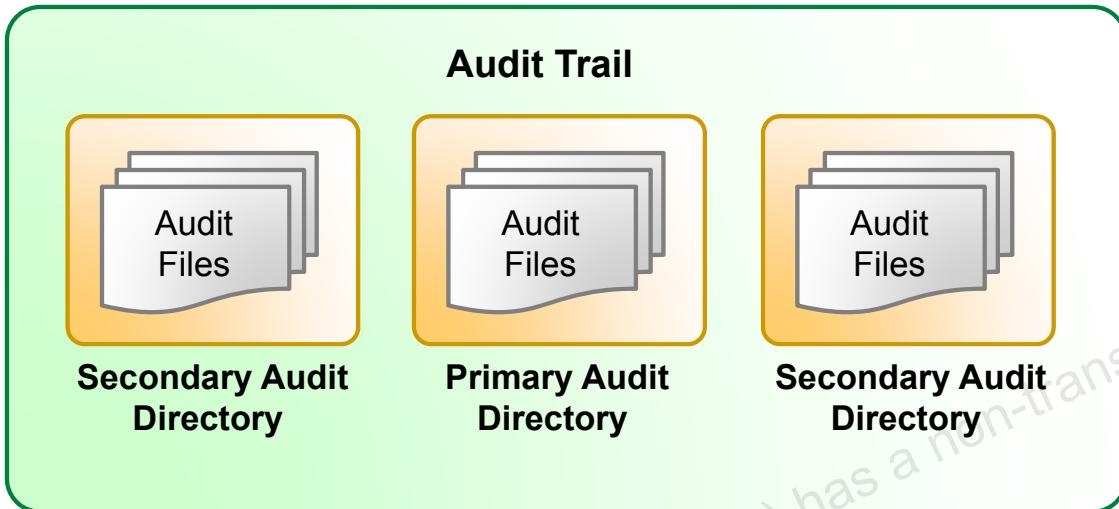
As discussed earlier, an audit plug-in is a module that transfers an audit record from the audit queue to a specified location. The Oracle Solaris audit service provides the following plug-ins:

- `audit_binfile`: Handles delivery of an audit record from the audit queue to the binary audit files.
- `audit_remote`: Handles secure delivery of binary audit records from the audit queue to a configured remote server. The `audit_remote` plug-in uses the `libgss()` library to authenticate the server. The transmission is protected for privacy and integrity.
- `audit_syslog`: Handles delivery of selected records from the audit queue to the `syslog` log.

You can configure the systems at your site to use binary mode locally, to send binary files to a remote repository, or to use `syslog` mode, or to use any combination of these modules. However, at least one plug-in must be active. By default, the `audit_binfile` plug-in is active.

Note: You are shown how to configure plug-ins in the next topic.

Storing and Managing the Audit Trail

Audit Trail

Audit Files	Audit Files	Audit Files
Secondary Audit Directory	**Primary Audit Directory**	**Secondary Audit Directory**

Audit records are stored in audit logs (also called audit files). In turn, audit files are stored in audit directories. The contents of all audit directories comprise the audit trail. The audit trail requires dedicated file space. This space must be available and secure. A best practice is to configure several audit directories for audit files.

Audit files are stored in audit directories in the following order:

- **Primary audit directory:** A directory where the audit files for a system are placed under normal conditions. The ZFS files are used for the primary audit directory. You will be shown how to set this up later in this lesson.

- **Secondary audit directories:** Directories where the audit files for a system are placed if the primary audit directory is full or not available

A directory is not used until a directory that is earlier in the list is full.

You are shown how to manage the audit files later in this lesson.

Audit Remote Server (ARS)

- ARS receives audit records over a secure link from audited systems and stores the records.
- ARS is delivered as a disabled Oracle Solaris audit component.
- To observe and configure ARS, use the `-setremote` and `-getremote` options of the `auditconfig` command.

Audit Remote Server (ARS) is the counterpart of the `audit_remote`(5) plug-in. Data sent by the `audit_remote` plug-in can be captured, processed, and stored by the server according to its configuration.

It is necessary to configure ARS before it can be used to process a remote audit trail. ARS configuration is two fold:

- The underlying security mechanisms used for secure audit data transport have to be configured (a Kerberos realm with specific audit principles and a GSS-API mechanism). See the `audit_remote` man page.
- The audit remote subsystem has to be configured.

The ARS configuration is divided between the configuration of `server` and `group`.
The `server` configuration allows changing common ARS parameters, while
the `group` keyword allows configuration of connection groups (sets of hosts sharing the same local storage parameters).

Audit Policies

An audit policy determines the characteristics of the audit records for the local system.

```
# auditconfig -lspolicy
policy string    description:
ahlt             halt machine if it can not record an async event
all              all policies
arge             include exec environment args in audit recs
argv             include exec command line args in audit recs
cnt              when no more space, drop recs and keep a cnt
group            include supplementary groups in audit recs
none             no policies
path             allow multiple paths per event
perzone          use a separate queue and auditd per zone
public           audit public files
seq              include a sequence number in audit recs
trail            include trailer token in audit recs
windata_down     include downgraded window information in audit recs
windata_up       include upgraded window information in audit recs
zonename         include zonename token in audit recs
```

By default, most audit policy options are disabled to minimize storage requirements and system processing demands. These options are stored as properties of the audit service and determine the policy options that are in effect at system boot or when the service is restarted.

You can display a list of available policy options by running the `auditconfig -lspolicy` command, as shown in this example.

The following policies add tokens to audit records: `arge`, `argv`, `group`, path, `seq`, trail, `windata_down`, `windata_up`, and `zonename`. The `windata_down` and `windata_up` policies are used by the Trusted Extensions feature of Oracle Solaris.

The remaining policies do not add tokens. The `ahlt` and `cnt` policies determine what happens when audit records cannot be delivered, the public policy limits auditing of public files, and the `perzone` policy establishes separate audit queues for non-global zones.

Note: For a description of each policy option and how each option affects the audit service, see the "Determining Audit Policy" section in *Oracle Solaris Administration: Security Services.*

Implementing the Oracle Solaris Auditing Plan

Your assignment is to:

- Configure the audit service
- Configure audit logs
- Configure the audit service in zones
- Administer the audit service
- Manage audit records on local systems

It is now time to implement the Oracle Solaris auditing plan. Your assignment is to configure the audit service and logs as well as set up the audit service in both the global zone and non-global zones. You will then administer the audit service. Your final task will be to manage the audit records.

Quiz

Oracle Solaris auditing is a service controlled by the audit daemon, `auditd`.

  a. True

  b. False

ORACLE

Answer: a

Quiz

Oracle Solaris auditing is enabled by default.
a. True
b. False

Answer: a

Quiz

Which audit plug-in module is active by default?

a. `audit_binfile`

b. `audit_remote`

c. `audit_syslog`

ORACLE

Answer: a

Quiz

Audit classes that are not preselected are not audited.

a. True

b. False

Answer: a

Quiz

Which Oracle Solaris auditing component determines the characteristics of the audit records for the local system?

a. Audit class

b. Audit event

c. Audit profile

d. Audit token

ORACLE

Answer: c

Quiz

Which audit policy is set by default?

a. all

b. cnt

c. none

d. zonename

ORACLE

Answer: b

Lesson Agenda

- Planning for Oracle Solaris Auditing
- **Configuring Oracle Solaris Auditing**
- Administering the Audit Service
- Managing Audit Records on Local Systems

ORACLE

Configuring Oracle Solaris Auditing

This section covers the following topics:

- Configuring the audit service
- Configuring audit logs
- Configuring the audit service in zones
- Administering the audit service
- Managing audit records on local systems

Configuring the Audit Service

This section covers the following topics:

- Determining audit service defaults
- Preselecting audit classes
- Configuring a user's audit characteristics
- Modifying the audit policy
- Specifying the audit warning destination email alias
- Adding an audit class
- Changing an audit event's class membership

Before you enable auditing on your network, you can modify the defaults to satisfy your site auditing requirements. Best practice is to customize your audit configuration as much as possible before the first users log in.

If you have implemented zones, you can choose to audit all zones from the global zone. Alternatively, to audit non-global zones individually, you can set the `perzone` policy in the global zone. In the `perzone` configuration, each non-global zone administrator manages auditing in their non-global zone.

Determining Audit Service Defaults

1. Display the preselected classes for attributable events by using `auditconfig -getflags`.
2. Display the preselected classes for non-attributable events by using `auditconfig –getnaflags`.
3. Display the audit policy by using `auditconfig – getpolicy`.
4. Display information about the audit plug-ins by using `auditconfig –getplugin`.
5. Display the audit queue controls by using `auditconfig –getqctrl`.
6. Display the `audit_flags` for existing users by using `userattr audit_flags` *loginname*.

Notes for step 6: By default, users are audited for the system-wide settings only.

Determining Audit Service Defaults: Example

```
# auditconfig -getflags
active user default audit flags = lo(0x1000,0x1000)
configured user default audit flags = lo(0x1000,0x1000)
# auditconfig -getnaflags
active non-attributable audit flags = lo(0x1000,0x1000)
configured non-attributable audit flags = lo(0x1000,0x1000)
# auditconfig -getpolicy
configured audit policies = cnt
active audit policies = cnt
# auditconfig -getplugin
Plugin: audit_binfile (active)
    Attributes: p_dir=/var/audit;p_fsize=0;p_minfree=0;

Plugin: audit_syslog (inactive)
    Attributes: p_flags=;

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

In the example shown in the slide, you are looking at the defaults on an unconfigured system with regards to the audit service configuration. The first thing you do is look at the preselected classes for attributable events.

Note: lo is the flag for the login/logout audit class. The format of the mask output is (*success,failure*).

Next, you are looking at the preselected classes for non-attributable events.

Note: To see which events are assigned to a class, and therefore which events are being recorded, you can run the auditrecord -c *class* command.

Your next step is to look at the default policy.

Note: The configured policy is a property of the audit service and is restored when you restart the audit service. The active policy is the policy that is currently used by the kernel, but is not a property of the audit service.

Next, you look at the default settings for the audit plug-ins. The audit_binfile plug-in is active by default.

Determining Audit Service Defaults: Example

```
$ 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
# who
jjones   pts/1          Dec 15 10:20     (:0.0)
jjones   pts/2          Dec 15 10:20     (:0.0)
tbone    pts/5          Dec 16 12:20     (:0.0)
tbone    pts/6          Dec 16 12:20     (:0.0)
...
# userattr audit_flags jjones
# userattr audit_flags tbone
```

Next, you look at the audit queue controls. The active policy is the policy that is currently used by the kernel. The string `no configured` indicates that the system is using the default settings.

The final default configuration you look at is the `audit_flag` settings for existing users. First, you run the `who` command to see who is on the system and then you run `userattr audit_flags` command for each user.

Preselecting Audit Classes

1. Determine the current preselected classes by using the `auditconfig` command's `-getflags` and `-getnaflags` options.

2. Set the new audit configuration as follows:
   a. Preselect the attributable classes by using `auditconfig -setflags lo,ps,fw`.
   b. Preselect the non-attributable classes by using `auditconfig -setnaflags lo,na`.

```
# auditconfig -setflags lo,ps,fw
user default audit flags = ps,lo,fw(0x101002,0x101002)
# auditconfig -setnaflags lo,na
non-attributable audit flags = lo,na(0x1400,0x1400)
```

To configure systemwide auditing for attributable and non-attributable events, you use the `auditconfig` command, as shown in the steps in the slide.

Notes for step 1: See steps 1 and 2 from the previous task for how to use these commands to view the current preselected classes.

Notes for step 2b: The `auditconfig -set*flags` commands do not add classes to the current kernel defaults. These commands replace the kernel defaults, so you must specify all classes that you want to preselect.

In the example in the slide, the events in the three classes are being audited for success and for failure. The second command in the example audits the events in the `na` class, and the `login` events that are not attributable. `lo` and `na` are the only legal arguments to the `-setnaflags` option.

Configuring a User's Audit Characteristics

1. To set audit flags for a user, use `usermod -K audit_flags=fw:no loginname`.

2. To set audit flags for a rights profile, use `profiles -K audit_flags=fw,as:no "Profile_Name"`.

```
# auditconfig -getflags
active user default audit flags = ss,lo(0x11000,0x11000)
configured user default audit flags = ss,lo(0x11000,0x11000)
# usermod -K audit_flags=pf:no jjones
# userattr audit_flags jjones
pf:no
```

Audit class preselections for each user are specified by the `audit_flags` keyword and are stored in the `user_attr` database and `prof_attr` database. These definitions, plus the preselected classes for the system, determine the user's audit mask. Follow the steps listed in the slide to configure the audit characteristics for a user.

Notes for step 1: The format of the `audit_flags` keyword is always *-audit:never-audit*, as follows:

- *always-audit*: Lists the audit classes that are exceptions for this user. Exceptions to the system-wide classes are prefixed by a caret (^). Added classes are not prefixed by a caret.
- *never-audit*: Lists the audit classes that are never audited for the user, even if these audit events are audited system-wide. Exceptions to the system-wide classes are prefixed by a caret (^).

To specify multiple audit classes, you separate the classes with commas.

Notes for step 2: When you assign the rights profile to a user or a role, that user or role is audited for those flags.

The example shows how to change the events that are audited for one user. You begin by displaying the audit preselection mask for all users. You then preselect the `pf` class for the `jjones` user. You run the `userattr` command to show the addition.

The audit preselection mask for `jjones` is a combination of the `audit_flags` settings with the system default settings.

Modifying the Audit Policy

1. View the current audit policy by using `auditconfig -getpolicy`.

2. View the available policy options by using `auditconfig -lspolicy`.

3. Enable or disable selected audit policy options by using `auditconfig [ -t ] -setpolicy [prefix]policy[,policy...]`.

The audit policy determines the characteristics of the audit records for the local host. You can inspect, change, and temporarily change audit policies with the `auditconfig` command. Follow the steps listed in the slide to modify the audit policy.

Notes for step 3: The options for the `auditconfig [ t ] -setpolicy` command are as follows:

- `-t`: Creates a temporary, or active, policy. The policy setting is not restored when you restart the audit service. This option is optional.
- `prefix`: A `prefix` value of `+` adds the list of policies to the current policy. A `prefix` value of `-` removes the list of policies from the current policy. Without a prefix, the audit policy is reset.
- `policy`: Selects the policy to be enabled or to be disabled.

A temporary policy is in effect until the audit service is refreshed, or until the policy is modified by the `auditconfig -setpolicy` command.

Modifying the Audit Policy: Example

```
$ auditconfig -lspolicy
policy string    description:
ahlt             halt machine if it can not record an async event
all              all policies for the zone
arge             include exec environment args in audit recs
argv             include exec command line args in audit recs
cnt              when no more space, drop recs and keep a cnt
group            include supplementary groups in audit recs
none             no policies
path             allow multiple paths per event
perzone          use a separate queue and auditd per zone
public           audit public files
seq              include a sequence number in audit recs
trail            include trailer token in audit recs
windata_down     include downgraded window information in audit recs
windata_up       include upgraded window information in audit recs
zonename         include zonename token in audit recs
# auditconfig -setpolicy -cnt
# auditconfig -setpolicy +ahlt
```

In the example shown in the slide, you are viewing the available policy options.

Note: The `perzone` and `ahlt` policy options can be set only in the global zone.

After reviewing the policy options, it is decided to disable the `cnt` policy and enable the `ahlt` policy. With these settings, system use is halted when the audit queues are full and an asynchronous event occurs. When a synchronous event occurs, the process that created the thread hangs. These settings are appropriate when security is more important than availability.

Specifying the Audit Warning Destination Email

To configure the `audit_warn` email alias, choose one of the following options:

- Option 1: Replace the `audit_warn` email alias with another email alias in the `audit_warn` script, as follows:

```
ADDRESS=audit_warn      # standard alias for audit alerts
```

- Option 2:
  - Redirect the `audit_warn` email to another mail account.
  - Run the `newaliases` command to rebuild the random access database for the aliases file.

```
audit_warn: root
# newaliases
/etc/mail/aliases: 14 aliases, longest 10 bytes, 156 bytes total
```

If you want to be notified if the audit directories are close to filling up or have already filled up, you can set up an email to warn you of this. To send this mail to a valid email address, you can follow one of the options shown in the slide. The `/etc/security/audit_warn` script generates mail to an email alias that is called `audit_warn`.

Note: If the `perzone` policy is set, the non-global zone administrator must configure the `audit_warn` alias in the non-global zone.

Adding an Audit Class

1. Save a backup copy of the `audit_class` file as follows:

   ```
   # cp /etc/security/audit_class \
   /etc/security/audit_class.orig
   ```

2. Add new entries to the `audit_class` file by using *0xnumber:flag:description*.

```
0x08000000:pf:profile command
```

When you create your own audit class, you can place it into just those audit events that you want to audit for your site. When you add the class on one system, you copy the change to all systems that are being audited. A best practice is to create audit classes before enabling the audit service.

Note: You must choose free bits. Your choice can be overwritten by a future release of the Oracle Solaris OS.

Notes for step 1: Although not required, it is a good practice to save a backup copy of the `audit_class` file before you modify it.

Notes for step 2: The entry must be unique in the file. Do not use existing audit class masks.

In the example in the slide, a class to hold administrative commands that are executed in a role is being created. The entry creates the new `pf` audit class.

Note: If you have customized the `audit_class` file, make sure that any user exceptions to the system audit preselection mask are consistent with the new audit classes. Errors occur when an `audit_flags` value is not a subset of the `audit_class` file.

Changing an Audit Event's Class Membership

1. Save a backup copy of the `audit_event` file as follows:

   ```
   # cp /etc/security/audit_event \
   /etc/security/audit_event.orig
   ```

2. Change the class membership for an audit event by changing the `class_list` field in the audit event entry.

3. Verify the change by using `auditconfig -setflags` *class_list*.

```
# grep pf /etc/security/audit_class
0x08000000 pf:profile command
# vi /etc/security/audit_event
116:AUE_PFEXEC:execve(2) with pfexec enabled pf
# auditconfig -setflags pf
user default audit flags = pf(0x8001000,0x8001000)
```

ORACLE

You might want to change an audit event's class membership to reduce the size of an existing audit class or to place the event in a class of its own. When you reconfigure audit `event-class` mappings on one system, you need to copy the change to all systems that are being audited. A best practice is to change event-class mappings before users log in.

In the example in the slide, an existing audit event is being mapped to the `pf` audit class. By default, the `AUE_PFEXEC` audit event is mapped to four classes: `ps`, `ex`, `ua`, and `as`. Using the `vi` text editor, you change the mapping for the event to the `pf` audit class. The new class replaces the existing classes. Replacement enables you to audit for events in the other classes while not generating the records of the `AUE_PFEXEC` event. With the final command, you verify that the change has been made successfully.

Configuring Audit Logs

This section covers the following topics:

- Creating ZFS file systems for audit files
- Allocating audit space for the audit trail
- Sending audit files to a remote repository
- Configuring the system log as the audit message destination

ORACLE

Creating ZFS File Systems for Audit Files

1. Determine the amount of disk space that is required.
2. Create a mirrored ZFS storage pool.
3. Create a ZFS file system and mount point for the audit files.
4. Create a ZFS file system for the audit files.
5. Protect the parent audit file system.
6. Compress the audit files in the pool.
7. Set quotas on the audit file system.
8. For a large pool, limit the size of the audit files.

Notes for step 2: Assign at least 200 MB of disk space per host. However, keep in mind that the amount of auditing you require will dictate the disk space requirements. You might find that your disk space requirements are far greater.

Notes for step 4: You might want to create additional file systems for the audit files. If so, repeat this step as many times as necessary.

Notes for step 5: To protect the parent audit file system, you set three ZFS properties to off for all file systems in the pool: `devices`, `exec`, and `setuid`.

Notes for step 6: Typically, compression is set on file systems. However, because all the file systems in this pool contain audit files, compression is set at the pool level.

Notes for step 7: These quotas are used by the `audit_warn` alias to notify you when the space is filling up.

Notes for step 8: By default, an audit file can grow to the size of the pool.

Allocating Audit Space for the Audit Trail

1. Determine the attributes to the `audit_binfile` plug-in by using `man audit_binfile`.
2. To add directories to the audit trail, specify the `p_dir` attribute by using the following command:
   ```
   # auditconfig -setplugin audit_binfile
   active \
   p_dir=/audit/example1/files,/var/audit
   ```
3. Refresh the audit service by using `audit -s`.

```
# auditconfig -setplugin audit_binfile active \
p_dir=/audit/client1/files,/var/audit
# audit -s
```

After you have created ZFS file systems for the audit files, the next task is to allocate audit space for the audit trail. By default, the `/var/audit` directory holds audit files for the `audit_binfile` plug-in.

Notes for step 2: The command presented for this step sets the `/audit/example1/files` directory as the primary directory for audit files, and the default `/var/audit` directory as the secondary directory.

Notes for step 3: The `auditconfig -setplugin` command sets the configured value. This value is a property of the audit service, so it is restored when the service is refreshed or restarted. The configured value becomes active when the audit service is refreshed or restarted.

In the example shown in the slide you are activating the `audit_binfile` plug-in and setting the storage for auditing. You are setting your ZFS file systems as the primary storage location with the `/var/audit` as the secondary audit file directory. You then refresh the audit service.

Sending Audit Files to a Remote Repository

1. Determine the attributes to the `audit_remote` plug-in by using `man audit_remote`.

2. To specify the remote hosts, use the `p_hosts` attribute as follows:

   ```
   # auditconfig -setplugin audit_remote active \
   p_hosts=rhost1:16088:kerberos_v5
   ```

3. To specify the number of retries, use the `p_retries` attribute as follows:

   ```
   # auditconfig -setplugin audit_remote active \
   p_retries=5
   ```

4. To specify the length of a connection timeout, use the `p_timeout` attribute as follows:

   ```
   # auditconfig -setplugin audit_remote active \
   p_timeout=3
   ```

5. Refresh the audit service by using `audit -s`.

ORACLE

Notes for step 1: Read the `OBJECT ATTRIBUTES` section. The default port is the `solaris_audit` IANA-assigned port, port `16162/tcp`. The default mechanism is `kerberos-v5`. The timeout default is 5 seconds. You can also specify a queue size for the plug-in.

Configuring the System Log as the Audit Message Destination

1. Select classes to be sent to the `audit_syslog` plug-in and make the plug-in active.
2. Add an `audit.notice` entry to the `syslog.conf` file.
3. Create the log file.
4. Refresh the configuration information for the `syslog` service.
5. Refresh the `audit service` by using `audit -s`.
6. Regularly archive the `syslog` log files.

```
# auditconfig -setplugin audit_syslog active p_flags=-lo,-ss,+pf
# vi /etc/syslog.conf
# grep audit.notice /etc/syslog.conf
audit.notice                    /var/log/auditlog
# touch /var/log/auditlog
# svcadm refresh system/system-log
# audit -s
```

Notes for step 1: These classes must be preselected as either system defaults, or in a user's `audit_flags` attribute. Records are not collected for a class that is not preselected.

Notes for step 3: The entry includes the location of the log file.

Notes for step 6: The audit service can generate extensive output.

In the example, the `audit_syslog` plug-in is being activated and the audit flags that are to be activated for the log are indicated. You want to track failed login and login attempts, failed changes in the system state, and successful uses of the `profile` command.

Next, you add the `audit.notice` entry to the `syslog.conf` file and then create the file by using the `touch` command. With the final two commands, you refresh the `syslog` service and the audit service.

Configuring the Audit Service in Zones

- Configuring all zones identically for auditing
  - Single audit service is used.
  - Audit service runs in the global zone.
  - Audit records are collected for both the global zone and all non-global zones.
- Specifying per-zone auditing
  - An audit service is used per zone.
  - An audit service can be disabled on a zone-by-zone basis.
  - Each zone collects its own audit records, which are visible to the non-global zone and the global zone.

ORACLE

The audit service audits the entire system, including activities in zones. A system that has installed non-global zones can run a single audit service to audit all zones identically, or it can run one audit service per zone, including the global zone.

When you audit the non-global zones exactly as the global zone is audited, the audit service runs in the global zone. The service collects audit records from the global zone and all the non-global zones. The non-global zone administrators might not have access to the audit records.

The advantages of per-zone auditing are a customized audit trail for each zone, and the ability to disable auditing on a zone-by-zone basis. Each zone collects its own audit records. The records are visible to the non-global zone and the global zone. These advantages can be offset by the administrative overhead. Each zone administrator must administer auditing. Each zone runs its own audit daemon, and has its own audit queue and audit logs. These audit logs must be managed.

In this section you are shown how to configure the audit service for both situations.

Configuring All Zones Identically for Auditing

1. Configure the global zone for auditing.
2. Copy modified audit configuration files from the global zone to every non-global zone by using one of the following options:
   - Loopback mount the changed `audit_class` and `audit_event` files.
     - From the global zone, halt the non-global zone.
     - Create a read-only loopback mount for every audit configuration file that you modified in the global zone.
     - Boot the non-global zone to make the changes effective.
   - Copy the files.
     - From the global zone, list the `/etc/security` directory in the non-global zone.
     - Copy the changed `audit_class` and `audit_event` files to the zone's `/etc/security` directory.

ORACLE

Notes for step 1: Configuring a zone for auditing is the same as configuring a system with the following exceptions:
- Do not enable `perzone` audit policy.
- Do not enable the audit service. You enable the audit service after you have configured the non-global zones for auditing.
- Set the `zonename` policy. This policy adds the name of the zone to every audit record.

Notes for step 2: If you modified the `audit_class` or `audit_event` file, copy it. Otherwise, skip this step. You have two options. You can loopback mount the files, or you can copy the files. The non-global zone must be running.

The non-global zones are audited when the audit service is enabled in the global zone.

Configuring All Zones Identically
for Auditing: Example

```
# auditconfig -getpolicy
configured audit policies = ahlt,arge,argv
active audit policies = ahlt,arge,argv
# auditconfig -setpolicy +zonename
# auditconfig -getpolicy
configured audit policies = ahlt,arge,argv,zonename
active audit policies = ahlt,arge,argv,zonename
# cp /etc/security/audit_class \
    /zones/zone1/root/etc/security/audit_class
# cp /etc/security/audit_event \
    /zones/zone1/root/etc/security/audit_event
# ls -l /zones/zone1/root/etc/security/audit_*
-rw-r--r-- 1 root sys  2878 2011-12-16 07:04
    /zones/zone1/root/etc/security/audit_class
-rw-r--r-- 1 root sys 29472 2011-12-16 07:05
    /zones/zone1/root/etc/security/audit_event
-rwxr----- 1 root sys  7823 2011-12-03 15:24
    /zones/zone1/root/etc/security/audit_warn
# audit -s
```

ORACLE

In the example in the slide, you configure all the zones for auditing. You begin by checking the current audit policy to verify that auditing for the global zone has not been configured. Next, you configure all zones for auditing by setting the `zonename` policy. You then verify that zones are now part of the audit policy. By adding the `zonename` policy, the audit records will be tagged with the zone name. Next you copy the modified `audit_event` and `audit_class` configuration files from the global zone to the non-global zone called `zone1`. You then verify that the audit configuration files are in the `/etc/security` file for `zone1`, which they are. Your final step is to start the audit service.

Specifying Per-Zone Auditing

1. In the global zone, configure auditing.
2. In each non-global zone, configure the audit files.
   a. Complete each of the tasks for configuring the audit service.
   b. Do not configure system-wide audit settings.
3. If auditing is not enabled in the global zone, enable it.
4. Enable auditing in your zone by using `audit -s`.

Notes for step 2b: Specifically, do not add the `perzone` or `ahlt` policy to the non-global zone.

Notes for step 3: The global zone administrator must enable the audit service for the system.

Specifying Per-Zone Auditing: Example

```
# zlogin zone1
[Connected to zone 'zone1' pts/2]
Last login: Fri Dec 16 10:42:38 on pts/2
Oracle Corporation SunOS 5.11 11.0    November 2011
# auditconfig -getcond
audit condition = noaudit
# auditconfig -getflags
active user default audit flags = no(0x0,0x0)
configured user default audit flags = lo(0x1000,0x1000)
# auditconfig -getnaflags
active non-attributable audit flags = no(0x0,0x0)
configured non-attributable audit flags = lo(0x1000,0x1000)
# auditconfig -getpolicy
configured audit policies = cnt
active audit policies = cnt,perzone
# audit -s
# ls /var/audit
20111216141435.not_terminated.zone1
# exit
logout
```

ORACLE

In the example shown in the slide, an auditing is being set up in the non-global zone called zone1. The assumption is that the global zone is already configured. The first step is to log in to the zone. Then the audit files are configured by using the auditconfig command, to include the audit condition, the user default audit flags, the active non-attributable audit flags, and the audit policies. Next, the audit service is enabled. Then it is verified that auditing is occurring in the zone by checking /var/audit, which in this example has been set up as the primary audit directory. You then exit the non-global zone.

Lesson Agenda

- Planning for Oracle Solaris Auditing
- Configuring Oracle Solaris Auditing
- **Administering the Audit Service**
- Managing Audit Records on Local Systems

Administering the Audit Service

This section covers the following topics:

- Enabling the audit service
- Disabling the audit service
- Refreshing the audit service

Enabling the Audit Service

1. Use the `audit -s` command to enable the audit service.
2. Verify that auditing is enabled by using `auditconfig -getcond`.

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

Auditing is an SMF service. You configure the service by using the `auditconfig` command and enable it with the `audit -s` command. The steps for enabling the audit service for all zones are listed in the slide. You must be assigned the Audit Control rights profile to perform these tasks.

Note: If the `perzone` audit policy is set in the global zone, zone administrators can enable, refresh, and disable the service in their non-global zones.

Notes for step 2: The output should reflect that the `audit condition` is set to `auditing`, as shown in the example.

Note: Before a zone administrator can enable the audit service in a non-global zone by using the `audit -s` command, the following actions must be completed:

- The global zone administrator sets the `perzone` policy in the global zone and enables auditing.
- The zone administrator of the non-global zone configures the audit service and per-user exceptions.

Disabling the Audit Service

To disable the audit service, run `audit -t`.

```
# audit -t
```

The step for disabling the audit service for all zones is shown in the slide. This action returns the system to the system state before auditing was enabled.

Note: If the `perzone` audit policy is not set, auditing is disabled for all zones. If the `perzone` audit policy is set in the global zone, the policy remains in effect in the non-global zones that have enabled auditing. The non-global zone continues to collect audit records across global zone reboots and non-global zone reboots until the zone administrator disables the non-global zone by using the `audit -t` command from within the non-global zone.

Refreshing the Audit Service

1. Refresh the audit service by using the `audit -s` command.
2. Update the preselection masks of users who are currently being audited.
    a. Terminate the users' existing sessions.
    b. Use the `auditconfig -setflags` command to dynamically change each logged-in user's preselection mask.
        - Determine the logged-in user's audit ID and audit session ID by using the `who` command.
        - Determine the user's audit ID by using the `getent passwd` *loginname* command.
        - Change the user's preselection mask by using `auditconfig -setumask` and `auditconfig -setsmask`.
        - Verify that the preselection mask has changed by using `auditconfig -getpinfo`.

Any time you make configuration changes to the audit service after it has been enabled, you will need to refresh the service.

Notes for step 1: When you refresh the audit service, all temporary configuration settings are lost. Audit policy and queue controls enable temporary settings.

Notes for step 2: Audit records are generated based on the audit preselection mask that is associated with each process. Refreshing the audit service does not change the masks of existing processes. To explicitly reset the preselection mask for an existing process, you must update each user's preselection mask. To change the systemwide audit preselection mask, the users must be logged in. You have two ways to complete this task. You can terminate the existing sessions or use the `auditconfig` command, as shown in steps 2a and 2b in the slide.

Notes for step 2a: Users can log out and log back in, or you can manually terminate (kill) active sessions. The new sessions will inherit the new preselection mask.

Practice 9-1 Overview: Configuring and Administering Oracle Solaris Auditing

This practice covers the following topics:

- Configuring the audit service
- Configuring audit logs
- Configuring the audit service in zones
- Administering the audit service

The practices for this lesson are designed to reinforce the concepts that have been presented in the lecture portion. These practices cover the following tasks:

- **Practice 9-1:** Configuring and administering Oracle Solaris auditing
- **Practice 9-2:** Managing audit records on local systems

Practice 9-1 should take you about 45 minutes to complete.

Lesson Agenda

- Planning for Oracle Solaris Auditing
- Configuring Oracle Solaris Auditing
- Administering the Audit Service
- **Managing Audit Records on Local Systems**

Managing Audit Records on Local Systems

This section covers the following topics:
- Displaying audit record definitions
- Merging audit files
- Selecting audit events to examine
- Viewing contents of binary audit files

Assume that the audit service has been up and running for a while, and you are now ready to collect and analyze the data from the audit trail.

Displaying Audit Record Definitions

To display audit record definitions, use `auditrecord -a`.

```
# auditrecord -a
terminal login
  program     /usr/sbin/login      See login(1)
              /usr/dt/bin/dtlogin  See dtlogin
  event ID    6152                 AUE_login
  class       lo                   (0x00001000)
      header
      subject
      [text]                       error message
      return

login: logout
  program     various              See login(1)
  event ID    6153                 AUE_logout
  class       lo                   (0x00001000)
---
```

The audit record definitions provide the audit event number, audit class, selection mask, and record format of an audit event. By viewing the audit record definitions, you can determine the set of audit tokens included in a specific type of audit record. The example in the slide contains the partial audit record format for the `login` program. Here you can see that the `lo` class format has three audit tokens: `header`, `subject`, and `return` with `text` being an optional token.

Note: The `-a` option for the `auditrecord` command lists all audit event record definitions. You can use the `-h` option to put the list in an HTML format that can be displayed in a browser. After you have the `*html` file displayed in a browser, you can use the browser's Find tool to find specific audit record definitions.

Merging Audit Files

1. Create a directory for storing merged audit files.
2. Merge the audit records in the audit trail as follows:
   a. Change directories to the `audit-trail-directory`.
   b. Merge the audit records into a file with a named suffix by using the following command:
      - `# auditreduce -Uppercase-option -O suffix`

```
$ cd /var/audit/audit_summary.dir
$ auditreduce -C -O Complete
$ ls *Complete
20111216183214.20111216214217.Complete
```

If you need to analyze the contents of the entire audit trail, you can do so more easily by merging all audit files in all the audit directories by using the `auditreduce` command. The command merges all the records from its input files into a single output file. The input files can then be deleted. If you do not specify a path for your merged file, the `auditreduce` command uses the `/var/audit` directory.

Notes for step 1: To complete this step, follow the instructions for creating a ZFS file system for audit files that were covered earlier.

Notes for step 2b: All directories in the audit trail on the local system are merged. The uppercase options (`-Uppercase-option`), which are used to manipulate files in the audit trail include, but are not limited to, the following:

- `-A`: Selects all of the files in the audit trail
- `-C`: Selects complete files only. This option ignores files with the suffix `not_terminated`.

- -M: Selects files with a particular suffix. The suffix can be a machine name, or it can be a suffix that you have specified for a summary file.
- -O: Creates an audit file with 14-character time stamps for both the start time and the end time, with the suffix `suffix` in the current directory.

Note: For the full list of options, see the `auditreduce(1M)` man page.

In the example in the slide, only complete files are copied from the audit trail into a merged file.

Selecting Audit Events to Examine

To select audit events to examine, use `auditreduce`
`-lowercase-option argument [optional-file].`

```
$ cd /var/audit/audit_summary.dir
$ auditreduce -c na -O nasumm
$ ls *nasumm
20111216183214.20111216215318.nasumm
```

You can select specific kinds of records to examine from the audit trail or from a file by using the `auditreduce` command. Some of the more commonly used options for the `auditreduce` command are as follows:

- `-d`: Selects all of the events on a particular date. The date format for *argument* is *yyymmdd*. Other date options, `-b` and `-a`, select events before and after a particular date.
- `-u`: Selects all of the events attributable to a particular user. The *argument* is a user name. Another user option, `-e`, selects all of the events attributable to an effective user ID.
- `-c`: Selects all of the events in a preselected audit class. The *argument* is an audit class name.
- `-m`: Selects all of the instances of a particular audit event. The *argument* is an audit event.
- *argument*: Specific argument that a lowercase option requires. For example, the `-c` option requires an argument of an audit class, such as `ua`.
- *optional-file*: Is the name of an audit file

Note: For the full list of options, see the `auditreduce(1M)` man page.

In the example in the slide, all the records of audit events in the `na` class are collected into one file.

Viewing Contents of Binary Audit Files

To view the contents of binary audit files, use one of the following `praudit` commands:

- `praudit -s`: Displays audit records in a short format
- `praudit -r`: Displays audit records in their raw format
- `praudit -x`: Displays audit records in XML format

```
$ auditreduce -c lo | praudit -s
header,69,2,AUE_screenlock,,mach1,2011-12-16 08:02:56.348 -07:00
subject,jjones,root,staff,jjones,staff,856,50036632,82 0 mach1
return,success,0
sequence,1298
```

ORACLE

The `praudit` command enables you to view the contents of binary audit files. You can pipe the output from the `auditreduce` command, or you can read a particular audit file. There are three `praudit` command options as listed in the slide.

Note: Each of the `praudit` commands displays the format in one token per line. For the `praudit -s` and `praudit -r` commands, you can use the `-l` option to place each record on one line. For the `praudit -x` command, you can use the `-l` option to place the XML output for one record on one line.

In the example in the slide, the `praudit -s` command is being used to display audit records in a short format.

Practice 9-2 Overview:
Managing Audit Records on Local Systems

This practice covers the following topics:

- Displaying audit record definitions
- Selecting audit events from the audit trail
- Viewing the contents of binary audit files
- Cleaning up an audit file currently in use (named `not_terminated`)

This practice should take you about 30 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Implement a plan for Oracle Solaris auditing
- Configure Oracle Solaris auditing
- Administer the audit service
- Manage audit records

Managing Processes and Priorities

ORACLE

Objectives

After completing this lesson, you should be able to:

- Implement a plan for executing a process in an appropriate scheduling class
- Manage process scheduling priority
- Manage the scheduling class of zones
- Configure the fair share scheduler
- Monitor the fair share scheduler

Workflow Orientation

Before you begin the lesson, take a moment to orient yourself in your job workflow. You have successfully installed the operating system and have updated it. You have configured the data storage environment as well as the physical and virtual networks. You have also ensured that all the system services are up and running that both users and processes have been granted the appropriate level of privilege. You have also set up the Oracle Solaris audit service. In this lesson you are shown how to manage the priority and scheduling of system and user processes that the Oracle Solaris 11 operating system uses to run business functions. As the system administrator, you are responsible for controlling and managing these system processes to ensure the system operates smoothly.

Lesson Agenda

- **Planning Process Execution in an Appropriate Scheduling Class**
- Managing Process Scheduling Priority
- Configuring the Fair Share Scheduler
- Managing the Scheduling Class of Zones

Planning Process Execution
in an Appropriate Scheduling Class

The process execution in an appropriate scheduling class plan ensures that:

- System resources are used appropriately
- Processes are prioritized in accordance with business needs and requirements
- Process workload distribution is controlled
- Processes are assigned to the appropriate scheduling class
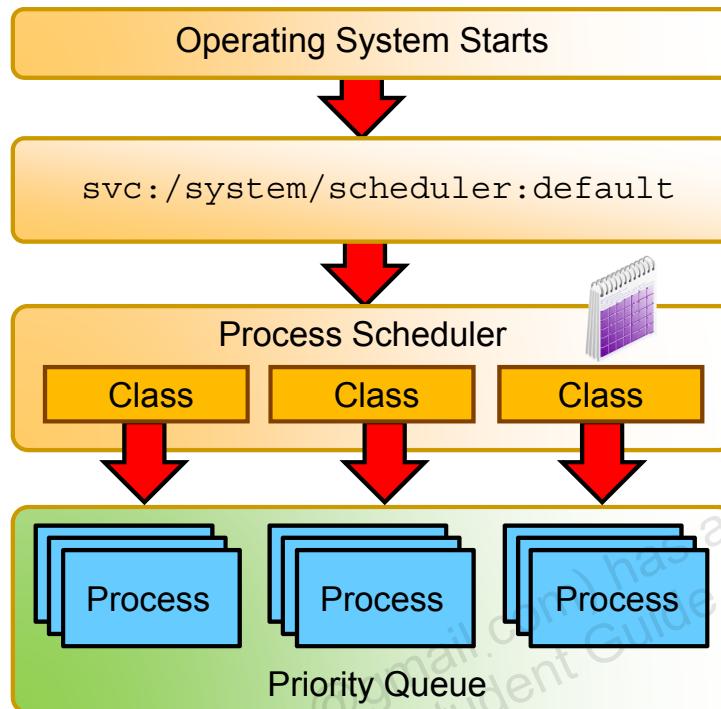
Not all processes are created to be equal, and given that there can be hundreds of processes active on the system at any time, it is important for a system administrator to be able to prioritize the processes and control their load distribution. Through these means, the system administrator ensures that the system resources, such as CPU, memory, and network, are not overused to the point where the system becomes bogged down or comes to a complete halt.

Understandably your company wants to ensure that its business applications run interrupted and that they are available when needed. As part of the predeployment activities, your company wants you to test the Oracle Solaris 11 process priority and scheduling class functionality to determine the best approach for distributing process workload.

In this topic, you are introduced to process priorities and the scheduling classes.

Process Scheduler

Operating System Starts

↓

`svc:/system/scheduler:default`

↓

Process Scheduler

| Class | Class | Class |

↓ ↓ ↓

| Process | Process | Process |

Priority Queue

A fundamental job of the operating system is to arbitrate which processes get access to the system's resources. The process scheduler, which is also called the dispatcher, is the portion of the kernel that controls allocation of the CPU to processes. It is managed by the SMF service `svc:/system/scheduler:default`.

The process scheduler supports the concept of scheduling classes. Each class defines a scheduling policy that is used to schedule processes within the class. The scheduling policy of a process determines its position in the priority queue.

Process Priority

- Global priority:
  - Based on scheduling class
- Designated priority:
  - Affects global priority assignment and position in a priority queue
  - Both scheduling class and user priority can be designated.
  - User priority is based on the assigned priority range of the scheduling class.

Based on scheduling class, each process has a global priority that identifies its position in the priority queue and its access to system resources, specifically CPU resources. The higher the global priority number the greater the priority.

As a system administrator, you might want to specify that certain processes be given more resources than others. You can do this by designating a priority for a process, thereby impacting its global priority assignment and position in the priority queue. You can designate a scheduling class for the process as well as a user priority. The user priority is based on the process's scheduling class and the priority range assigned to that scheduling class. By designating a priority for a process, you as the system administrator can control how the system should prioritize the running of each process, taking into account those processes that by their nature and their system-assigned scheduling class have a higher priority.

Note: You will take a look at the priority ranges for each scheduling class in just a moment.

Based on changing business needs and requirements, you can always modify the priority of a process. You learn how to designate and modify a process's priority later in this lesson.

Process Scheduling Classes

Scheduling Class	Description
Timesharing (TS)	Default class for processes and their associated kernel threads. Priorities in this class are dynamically adjusted in an attempt to allocate processor resources evenly.
Interactive (IA)	Enhanced version of the TS class that applies to the in-focus window in the GUI. Its intent is to give extra resources to processes associated with that specific window.
Fair Share Scheduler (FSS)	This class is share based rather than priority based. Threads managed by FSS are scheduled based on their associated shares and the processor's utilization.
Fixed-Priority (FX)	Priorities for threads associated with this class are fixed. In other words, they do not vary dynamically over the lifetime of the thread.
System (SYS)	Used to schedule kernel threads. Threads in this class are "bound" threads, which means that they run until they block or complete.
Real-Time (RT)	Threads in the RT class are fixed-priority, with a fixed-time duration called quantum.

ORACLE

The table shown in the slide identifies the process scheduling classes that can be configured on your system. The RT class offers the highest scheduling priorities and can preempt other scheduling class priorities.

Note: By default, any new processes that are created are assigned the TS class. However, as discussed, you can change the scheduling class designation based on business requirements and the importance of the application. You can also change the default scheduling class for the entire system so that all the processes including the non-global zones will run in the same scheduling class. You learn how to do this later in the lesson.

Although the TS scheduling class is the system's default scheduling class, using the fair share scheduler (FSS) as the default scheduling class is highly desirable. The FSS gives you the control to specify that certain processes should be given more resources than others. This is exceptionally beneficial when you are trying to balance workloads for multiple projects or non-global zones. Because the FSS is recommended as the default scheduling class, a good deal of time is spent in this lesson, teaching you how to use it.

Priority Ranges for Scheduling Classes

Scheduling Class	Priority Range
Real-time (RT)	100 through 159
System (SYS)	60 through 99
Fair share scheduler (FSS)	0 through 59
Fixed priority (FX)	0 through 59
Interactive (IA)	0 through 59
Timesharing (TS)	0 through 59

The scheduling classes have an assigned range of priorities. The table in the slide presents the ranges.

The higher the number, the greater the priority. This means that a real-time process will always run before either a system process or a process that is assigned to any of the other scheduling classes (FSS, FX, IA, TS).

Note: The priority of a process is inherited from the parent process.

Combining FSS with Other Scheduling Classes

- Avoid having the FSS, TS, IA, and FX scheduling classes share the same processor set (pset).
- All processes that run on a processor set must be in the same scheduling class so that they do not compete for the same CPUs.
- To avoid starving applications in the FSS class, use processor sets for FSS class and FX class applications.
- The following classes can be in the same processor sets:
    - TS and IA classes
    - FSS and RT classes

    Note: FSS has no control over the RT class processes.

As you just saw, by default, the FSS scheduling class uses the same range of priorities (0 through 59) as the time sharing (TS), interactive (IA), and fixed priority (FX) scheduling classes. Therefore, you should avoid having processes from these scheduling classes share the same processor set. A mix of processes in the FSS, TS, IA, and FX classes could result in unexpected scheduling behavior.

With the use of processor sets, you can mix TS, IA, and FX with FSS in one system. However, all the processes that run on each processor set must be in one scheduling class, so they do not compete for the same CPUs. The FX scheduler in particular should not be used in conjunction with the FSS scheduling class unless processor sets are used. This action prevents applications in the FX class from using priorities high enough to starve applications in the FSS class.

You can mix processes in the TS and IA classes in the same processor set, or on the same system without processor sets.

Because RT and FSS are using disjointed, or non-overlapping ranges of priorities, FSS can coexist with the RT scheduling class within the same processor set. However, the FSS scheduling class does not have any control over processes that run in the RT class.

For example, on a four-processor system, a single-threaded RT process can consume one entire processor if the process is CPU bound. If the system also runs FSS, regular user processes compete for the three remaining CPUs that are not being used by the RT process. Note that the RT process might not use the CPU continuously. When the RT process is idle, FSS uses all four processors.

Using CPU Shares with the FSS

- The FSS uses CPU shares to control the allocation of available CPU resources among workloads.
- Assigning a greater number of CPU shares to a project gives that project more CPU resources from the FSS.
- CPU share allocation and CPU resource usage are not the same.
    - CPU shares define the relative importance of workloads in relation to other workloads.
    - Resource utilization is the percentage of CPU capacity being used.
- When allocating CPU shares, you should know:
    - How many shares the project has in comparison with other projects
    - How many of the other projects are competing for CPU resources

The primary benefit of using the FSS is its ability to enable you to control the allocation of available CPU resources among workloads, based on their importance. This control is achieved by using CPU shares. You identify the importance of each workload by the number of shares of CPU resources that you assign to each workload. The term "share" is used to define a portion of the system's CPU resources that is allocated to a project. If you assign a greater number of CPU shares to a project, relative to other projects, the project receives more CPU resources from the FSS.

Note: Processes in projects with zero shares always run at the lowest system priority (0). These processes run only when projects with nonzero shares are not using CPU resources.

CPU share allocation is not the same as CPU resource usage. Shares are used to define the relative importance of workloads in relation to other workloads, whereas CPU resource usage is the percentage of CPU capacity being used. A project that is allocated 50 percent of the CPU resources might average only a 20 percent CPU use. Moreover, shares serve to limit CPU usage only when there is competition from other projects. Regardless of how low a project's allocation is, it always receives 100 percent of the processing power if it is running alone on the system.

When allocating CPU shares, it is important to know how many shares the project has in comparison with other projects and how many of the other projects are competing for CPU resources.

Note: The maximum number of shares that can be assigned to any one project is 65535.

You learn how to allocate CPU shares later in this lesson.

Scheduling Class on a System
with Zones Installed

- Non-global zones use the system's default scheduling class.
- For a new default scheduling class setting, non-global zones obtain the new setting when booted or rebooted.
- To ensure that all zones get a fair share of the system CPU resources, set the FSS as the system default scheduling class.

Non-global zones use the default scheduling class for the system. If the system is updated with a new default scheduling class setting, non-global zones obtain the new setting when booted or rebooted.

As discussed earlier, the recommended scheduler to use with zones is the FSS. The preferred way then is to set the FSS to be the system default scheduling class and then configure CPU shares for the zones. All zones then benefit from getting a fair share of the system CPU resources.

You learn how to configure CPU shares for zones later in this lesson.

Implementing the Process Execution in an Appropriate Scheduling Class Plan

Your assignment is to:

- Determine the scheduling priorities and classes for the process running on the system
- Modify scheduling priorities
- Set the FSS as the default scheduler
- Configure CPU shares for zones

As part of the predeployment test, you have been given the assignment to learn how to determine, designate, modify, and monitor scheduling priorities and classes for the processes running on the system. You have also been tasked with learning how to make the FSS the default scheduling class for zones, and then how to configure CPU shares for the zones.

Quiz

For the operating system to prioritize processes, all processes must have the same scheduling class.

a. True

b. False

ORACLE

Answer: b

Quiz

Which scheduling class has the highest range of user priority designations?

a. Fair share scheduler (FSS)

b. Real-time (RT)

c. System (SYS)

d. Time sharing (TS)

Answer: b

Quiz

What is the purpose of CPU shares?

a. To control the allocation of available CPU resources among workloads

b. To increase CPU capacity

c. To change the global priority of a project in the priority queue

d. To cap the CPU resource usage of a process
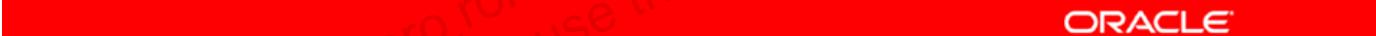
Answer: a

Quiz

Non-global zones use the default system scheduling class for the system.
a. True
b. False

Answer: a

Lesson Agenda

- Planning Process Execution in an Appropriate Scheduling Class
- **Managing Process Scheduling Priority**
- Configuring the Fair Share Scheduler
- Managing the Scheduling Class of Zones

Managing Process Scheduling Priority

This section covers the following topics:

- Displaying processes with the `top` command
- Displaying process class information
- Determining the global priority of a process
- Designating a process priority
- Modifying a process priority

Displaying Processes with the `top` Command

To display the processes that are using the most CPU resources, use `top number_of_processes time_interval`.

```
# top 10 -s 10
last pid:  1121;  load avg:  0.20,  0.14,  0.12;  up 0+01:50:30        14:10:30
87 processes: 83 sleeping, 3 running, 1 on cpu
CPU states: 81.8% idle,  5.1% user, 13.1% kernel,  0.0% iowait,  0.0% swap
Kernel: 609 ctxsw, 9 trap, 327 intr, 1935 syscall, 4 flt
Memory: 1024M phys mem, 84M free mem, 977M total swap, 977M free swap

   PID USERNAME NLWP PRI NICE  SIZE    RES STATE    TIME   CPU COMMAND
   991 oracle      2  59    0   87M    19M sleep    0:11 4.03% gnome-terminal
   733 oracle      3  59    0   65M    53M run      0:23 3.82% Xorg
   929 oracle     20  59    0  160M   140M run      2:01 1.75% java
   934 oracle      1  56    0   12M  5552K run      0:06 1.46% xscreensaver
  1120 root        1  59    0 4296K  2480K cpu      0:00 0.25% top
   917 oracle      1  49    0  107M    36M sleep    0:01 0.22% nautilus
   913 oracle      1  59    0   27M    15M sleep    0:01 0.08% metacity
    11 root       18  59    0   12M    11M sleep    0:41 0.06% svc.configd
   536 root        7  59    0 9420K  1856K sleep    0:03 0.04% VBoxService
```

A quick and convenient way to view the processes running on the system that are using the most CPU resources is by using the `top` command. The output of the command is very similar to the `prstat` command.

Note: The `top` utility iteratively examines all active processes on the system and reports statistics in descending order–based CPU usage.

The command displays the following information:
- `Last pid`: Last process ID assigned to a process
- `Load avg`: These are the CPU load averages. The averages are based on one-, five-, and 15-minute intervals.
- `up`: System uptime and current time
- Number of processes currently active on the system and their respective states
- CPU states by percentage: Shows the percentage of CPU time in the following modes: `idle`, `user`, `kernel`, `iowait`, and `swap`.
- `Kernel`: Statistics on the following kernel-related activity: context switches, traps, interrupts, system calls, and page faults.

- `Memory`: Statistics on memory usage, including physical memory, free memory, total swap, and free swap
- `PID`: Process ID of the process
- `USERNAME`: Login name or UID of the owner of the process
- `NLWP`: Number of lightweight processes (LWPs) in the process

    Note: The kernel and many applications are now multithreaded. A thread is a logical sequence of program instructions written to accomplish a particular task. Each application thread is independently scheduled to run on an LWP, which functions as a virtual CPU. LWPs in turn, are attached to kernel threads, which are scheduled to run on actual CPUs.

- `PRI`: Priority of the process. Processes with higher numbers are given precedence.

    Note: The priority of a process is determined by the policies of its scheduling class and by its `nice` number.

- `NICE`: Value used in priority computation. Only processes in certain scheduling classes have a `nice` value.

    Note: The `nice` numbers range from 0 through +39, with 0 representing the highest priority.

- `SIZE`: Total virtual memory size of the process
- `RES`: Resident memory, which represents the amount of physical memory being used by the process, in megabytes (M)
- `STATE`: State of the process
    - `cpuN`: Process is running on the CPU.
    - `sleep`: Process is waiting for an event to complete.
    - `run`: Process is in the run queue.
    - `zombie`: Process is terminated, and the parent is not waiting.
    - `stop`: Process is stopped.
- `TIME`: Cumulative execution time for the process, given in hours, minutes, and seconds.
- `CPU`: Percentage of recent CPU time used by the process
- `COMMAND`: Command name of the process

Displaying Process Class Information

To display information about process classes, use `priocntl -l`.

```
# priocntl -l
CONFIGURED CLASSES
==================

SYS (System Class)

TS (Time Sharing)
        Configured TS User Priority Range: -60 through 60

SDC (System Duty-Cycle Class)

FSS (Fair Share)
        Configure FSS User Priority Range: -60 through 60

FX (Fixed priority)
        Configured FX User Priority Range: 0 through 60
```

To display process scheduling classes and priority ranges, you use the `priocntl -l` command.

Note: The `priocntl` command is used to display or set scheduling parameters for a specified process. You can also use it to display the current configuration information for the system's process scheduler (as is being done here) or you can use it to execute a command with specified scheduling parameters (which will be looked at in the next few slides).

In the output example, you can see all the classes being used at this time: system class (SYS), time sharing (TS), fixed priority (FX), and interactive (IA). You can also see the priority ranges for the time sharing (-60 through 60), fixed priority (0 through 60), and interactive (-60 through 60). You need to know these ranges when you designate the priority of a process, which will be looked at in the next few slides.

Determining the Global Priority of a Process

To determine the global priority of a process, use `ps -ecl`.

```
$ ps -ecl
 F S UID PID  PPID CLS PRI  ADDR     SZ   WCHAN    TTY      TIME   CMD
19 T 0   0    0    SYS 96   f00d05a8 0             ?        0:03   sched
 8 S 0   1    0    TS  50   ff0f4678 185  ff0f4848 ?        36:51  init
19 S 0   2    0    SYS 98   ff0f4018 0    f00c645c ?        0:01   pageout
19 S 0   3    0    SYS 60   ff0f5998 0    f00d0c68 ?        241:01 fsflush
 8 S 0   269  1    TS  58   ff0f5338 303  ff49837e ?        0:07   sac
 8 S 0   204  1    TS  43   ff2f6008 50   ff2f606e console  0:02   sh
```

The `-e` option displays information about every process that is currently running. The `-c` option displays information about scheduler properties. The `-l` option generates a long listing.

The command displays the following information:

- `F`: Flags associated with the process
- `S`: State of the process. States include:
  - `O`: Process is running on a processor.
  - `S`: Sleeping. Process is waiting for an event to complete.
  - `R`: Runnable. Process is on run queue.
  - `Z`: Zombie state. Process terminated and parent not waiting.
  - `T`: Process is stopped, either by a job control signal or because it is being traced.
- `UID`: Effective user ID number of the process
- `PID`: Process ID of the process
- `CLS`: Scheduling class

- `PRI`: Priority of the process
- `ADDR`: Memory address of the process
- `SZ`: Size (in pages) of the swappable process's image in main memory
- `WCHAN`: Address of an event for which the process is sleeping. If blank, the process is running.
- `TTY`: Controlling terminal for the process. The message `?` is printed when there is no controlling terminal.
- `TIME`: Cumulative execution time for the process
- `CMD`: Command name

In the example in the slide, the values in the priority (`PRI`) column show that the `pageout` process has the highest priority (98), whereas the `sh` process has the lowest priority (43).

Designating a Process Priority

1. Start a process with a designated priority by using `priocntl -e -c` *class* `-m` *user-limit* `-p` *user-priority* *command-name*.

2. Verify the process status by using `ps -ecl | grep` *command-name*.

```
# priocntl -e -c TS -m 60 -p 60 find . -name core -print
# ps -ecl | grep find
 0 S      0 2959  2771   TS  60        ? 1865          ? pts/1
0:01 gfind
ps -ecl | grep find
 0 S      0 2959  2771   TS  60        ? 1961          ? pts/1
0:01 gfind
ps -ecl | grep find
 0 R      0 2959  2771   TS  59        ? 1985            pts/1
0:02 gfind
```

As discussed in the first topic, you can designate the priority of a process. To do this, you use the `priocntl` command. The steps listed in the slide show how to designate the scheduling class as well as the user priority.

Notes for step 1: The options that are used with the `priocntl` command are as follows:

- `-e`: Executes a specified command with the class and scheduling parameters associated with a set of processes

- `-c` *class*: Specifies the class to be set. The valid class arguments are:
  - `RT` for real-time
  - `TS` for time sharing
  - `IA` for interactive
  - `FSS` for fair-share
  - `FX` for fixed priority

- `-m` *user-limit*: When you use the `-p` option in conjunction with this option, it specifies the maximum amount you can raise or lower the priority.

- `-p` *user-priority*: Designates the user priority

In the example, you designate the time sharing (TS) class for the process and the highest possible time-share priority, which is 60, to the `find` command. You then run the `ps -ecl | grep` command to verify that the priority is being used at all times, which as you can see, it is not. Based on workloads and available priorities, the system might not use the designated priority at all times.

Modifying a Process Priority

1. Change the priority of the process using `priocntl -s -p` *user-priority pid*.

2. Verify the process status using `ps -ecl | grep` *command-name*.

```
# priocntl -s -p 30 3084
# ps -ecf | grep myprog
    root   3093   2909   RT 130 09:09:34 pts/3      0:00 /bin/bash /root/myprog
    root   3124   2771   IA  32 09:15:25 pts/1      0:00 grep myprog
```

Because of changing business priorities, you might need to modify the priority of a running process. To do this, you use the `priocntl` command. The steps in the slide show how to complete this task.

Notes for step 1: The options that are used with the `priocntl` command are as follows:

- `-s`: Sets the scheduling parameters associated with a set of processes
- `-p` *user-priority*: Designates the user priority

In the example in the slide, you are changing the current user priority on a process called `myprog` (PID 3093). You now want the `myprog` process to have a priority of 30. You then verify the change. Here you can see that the `myprog` process now has a global priority of 130. The system added 100 to the RT priority of 30 to create the global priority.

Lesson Agenda

- Planning Process Execution in an Appropriate Scheduling Class
- Managing Process Scheduling Priority
- **Configuring the Fair Share Scheduler**
- Managing the Scheduling Class of Zones

Configuring the Fair Share Scheduler (FSS)

This section covers the following topics:

- Making the FSS the default scheduling class
- Manually moving processes from other classes into the FSS class
- Manually moving a project's processes into the FSS class
- Tuning scheduler parameters

ORACLE

Making FSS the Default Scheduling Class

To set the default scheduler for the system to be FSS,
use `dispadmin -d FSS`.

```
# dispadmin -d FSS
# dispadmin -d
FSS(Fair Share)
```

```
# dispadmin -l
CONFIGURED CLASSES
==================

SYS(System Class)
TS(Time Sharing)
SDC(System Duty-Cycle Class)
FSS(Fair Share)
FX(Fixed Priority)
```

The fair share scheduling class enables you to allocate CPU time based on shares instead of the priority scheme of the time sharing (TS) scheduling class. To make FSS the default scheduling class for the system, you use the `dispadmin -d` command, as shown in the slide.

Note: The `dispadmin` command displays or changes process scheduler parameters while the system is running. The `-d` option sets or displays the name of the default scheduling class to be used on reboot when starting `svc:/system/scheduler:default`.

This command does not change the scheduling classes of the currently running process, which you can see if you run the `dispadmin -l` command, as shown in the second example. Here you can see all the classes currently being used. The command does, however, impact any new processes that might be created. The new processes will all be assigned the FSS class.

Manually Moving Processes from Other Classes into the FSS Class

To move all processes into the FSS class, use `priocntl -s -c FSS -i all`.

```
# priocntl -s -c FSS -i all
```

```
# ps -ef -o class,zone,fname | grep -v CLS | sort -k2 | more
 FSS    global automoun
 FSS    global bash
 FSS    global bonobo-a
 FSS    global clock-ap
 FSS    global console-
 FSS    global cron
 FSS    global fmd
 SYS    global fsflush
 TS     global init
```

You can manually move all processes into the FSS scheduling class without changing the default scheduling class and rebooting (assuming you have not made the FSS the default scheduling class). To move all the processes from other classes into the FSS class, use the `priocntl` command as shown in the slide.

Note: This is only a temporary change. After reboot, all processes will again run in the default scheduling class.

The options that are used with the `priocntl` command are as follows:

- `-s`: Sets the upper limit on the user priority range and changes the current priority
- `-c` *class*: Specifies the class to be set
- `-i` *idtype*: Specifies one or more processes to which the `priocntl` command is to apply. The `-i all` option specifies to apply the `priocntl` command to all existing processes.
  Note: For a complete list of valid *idtype* arguments, see the `priocntl` man page.

To verify that all the processes have been moved into the FSS class, you can use the `pf -ef` command, as shown in the second example. The `-o` option is being used to specify the format that is to be displayed. In this case, you want to view the class, zone type, and file name. You use the `grep` command to specify that you want to view the class (`CLS`) output and the `sort` command to indicate that you want to sort by the second column, which in this case, is the zone.

Note: To display all the processes running in a specific class, such as `FSS` or `TS`, replace `CLS` with the class type.

In this partial output, you can see that most but not all of the scheduling classes for the processes have been changed to FSS. Some processes retain their scheduling class based on the nature or scope of the process.

Manually Moving the `init` Process into the FSS Class

To move the `init` process into the FSS class, use `priocntl -s -c FSS -i pid 1`.

```
# ps -ecf | grep init
    root    1    0   TS  59 07:42:52 ?              0:00 /sbin/init
# priocntl -s -c FSS -i pid 1
# ps -ef -o class,zone,fname | grep init
 FSS    global init
```

To move the `init` process into the FSS class, use the `priocntl` command with the `init` process ID number (`PID 1`) as shown in the slide.

Note: Because you are specifying only the `init` process for the global zone (`PID 1`), any `init` processes that are associated with non-global zones are not affected.

In the example in the slide, you begin by displaying the scheduling class for the `init` process. Notice that the scheduling class is `TS`. You then run the command to move the `init` process into the FSS class. Your final step is to verify that the change has been made, and it has.

Note: Again, this is only a temporary change. After reboot, the `init` process will again run in its default scheduling class.

Manually Moving a Project's Processes into the FSS Class

To move the processes that run in a project to the FSS scheduling class, use `priocntl -s -c FSS -i projid projectID_number`.

```
# ps -o user,pid,uid,projid,project,class
   USER   PID   UID PROJID  PROJECT  CLS
   root  2771     0      1 user.root  TS
   root  3000     0      1 user.root  TS
# priocntl -s -c FSS -i projid 1
# ps -o user,pid,uid,projid,project,class
   USER   PID   UID PROJID  PROJECT  CLS
   root  2771     0      1 user.root  FSS
   root  3015     0      1 user.root  FSS
```

ORACLE

You can also manually move a project's processes from their current scheduling class to the FSS scheduling class. The commands for completing this task are identical to moving processes into FSS with one exception. Instead of specifying a process, you specify a project ID number, as shown in the slide. As with the processes, this change is only temporary. After reboot, the project's processes will again run in the default scheduling class.

In the example in the slide, you start by displaying the current scheduling class for the current projects. As you can see, you have one project (PROJID 1) that has a scheduling class of TS. Using the priocntl command, you move the project's processes into the FSS class. Your last step is to verify the change.

Tuning Scheduler Parameters

To tune the scheduler parameters, use `dispadmin -c scheduler -g [-r resolution]`.

```
$ dispadmin -c FSS -g
#
# Fair Share Scheduler Configuration
#
RES=1000
#
# Time Quantum
#
QUANTUM=110
$ dispadmin -c FSS -g -r 100
#
# Fair Share Scheduler Configuration
#
RES=100
#
# Time Quantum
#
QUANTUM=11
```

ORACLE

You can use the `dispadmin` command to display or change process scheduler parameters while the system is running. For example, you can use `dispadmin` to examine and tune the FSS scheduler's time quantum value. Time quantum is the amount of time that a thread is allowed to run before it must relinquish the processor. You can specify the resolution that is used for displaying time quantum values. If no resolution is specified, time quantum values are displayed in milliseconds by default. You might find it easier to work with smaller digits; specifying 10 is much easier than specifying 100000 for quantum values.

In the example in the slide, you are tuning the time quantum parameter for FSS by modifying the resolution. First, you display the current time quantum for the FSS scheduler.
As you can see, currently, the quantum values are specified in 1/1000th of a second. By using the `-r` option, you change the time quantum to 1/100th of a second.

Practice 10-1 Overview:
Modifying Process Scheduling Priority

This practice covers the following topics:

- Managing scheduling class and process priorities
- Configuring the fair share scheduler

The practices for this lesson are designed to reinforce the concepts that have been presented in the lecture portion. These practices cover the following tasks:

- **Practice 10-1:** Modifying the scheduling priority for a process
- **Practice 10-2:** Configuring CPU shares and FSS in an Oracle Solaris zone

Practice 10-1 should take you about 30 minutes to complete.

Lesson Agenda

- Planning Process Execution in an Appropriate Scheduling Class
- Managing Process Scheduling Priority
- Configuring the Fair Share Scheduler
- **Managing the Scheduling Class of Zones**

Managing the Scheduling Class of Zones

This section covers the following topics:

- Configuring CPU shares configuration in a non-global zone
- Measuring CPU performance in the zones
- Assigning CPU shares to the global zone
- Removing the CPU shares configuration from a zone

Note: The assumption is that FSS has been made the default scheduling class for the system.

Configuring CPU Shares Configuration in a Non-Global Zone

1. Add the CPU shares to the zone by using `zonecfg -z` *zone*.

2. Set the number of shares for the global zone by using `set cpu-shares=`*number*.

3. Exit `zonecfg`.

4. Verify the configuration change by using `zonecfg -z` *zone* `info`.

Configuring CPU Shares in a
Non-Global Zone: Example

```
# zonecfg -z hrzone
zonecfg:hrzone> set cpu-shares=80
zonecfg:hrzone> exit
# zonecfg -z hrzone info
zonename: hrzone
zonepath: /zones/hrzone
brand: solaris
autoboot: true
bootargs:
pool:
limitpriv:
scheduling-class:
ip-type: exclusive
hostid:
fs-allowed:
[cpu-shares: 80]
net:
address not specified
allowed-address not specified
physical: vnic1
defrouter not specified
rctl:
name: zone.cpu-shares
value: (priv=privileged,limit=80,action=none)
```

ORACLE

In the example shown in the slide, you configure the CPU shares for hrzone from the global zone by using the zonecfg -z command. You set the CPU shares to 80, exit, and then confirm the configuration change. Here, you can see that hrzone now has 80 CPU shares.

Measuring CPU Performance in the Zones

To measure CPU performance in the zones, use `prstat -Z`.

```
# prstat –Z
…
…
…
ZONEID    NPROC    SWAP    RSS MEMORY       TIME  CPU ZONE
     1       27     34M    43M   4.2%    0:20:09 8.3% hrzone
     2       27     34M    43M   4.2%    0:16:15 2.4% itzone
     0       98    348M   451M    44%    0:00:50 0.3% global
```

In this mode, `prstat` displays separate reports about processes and zones at the same time. The output of the command is as follows:

- `ZONEID`: ID number of the zone
- `NPROC`: Number of processes in the zone
- `SWAP`: Total virtual memory size of the process, including all mapped files and devices, in kilobytes (K), megabytes (M), or gigabytes (G)
- `RSS`: Resident set size of the process in kilobytes (K), megabytes (M), or gigabytes (G)
- `MEMORY`: Percentage of memory used by a specified collection of processes
- `TIME`: Cumulative execution time for the process
- `CPU`: Percentage of recent CPU time used by the process
- `ZONE`: Zone name

As the output is dynamically updated, you will notice the percentage of CPU time shifting closer to the ratio you specified. Assuming that you allocated more CPU shares to `hrzone`, you will see a higher percentage of CPU time being used by that zone.

Assigning CPU Shares to the Global Zone

To assign CPU shares to the global zones, use `prctl -n zone.cpu-shares -v number_of_shares -r -i zone global`.

```
# prctl -n zone.cpu-shares -v 60 -r -i zone global
```

ORACLE

You can also assign CPU shares to the global zone by using the `prctl -n zone.cpu-shares` command, as shown in the slide.

The options for the `prctl -n zone.cpu-shares` command are as follows:

- `-n`: Specifies the name of the resource
- `-v value`: Specifies the value for the resource control for a set operation
- `-r`: Replaces the first resource control value with the new value specified through the `-v` option
- `-i idtype`: Specifies the type of the `id` operands. Valid `idtypes` are process, task, project, and zone

In the example in the slide, you are assigning 60 CPU shares to the global zone. Again, you are making the assumption that FSS is the default scheduling class for the global zone.

Removing the CPU Shares Configuration from a Zone

1. Remove the CPU shares configuring the zone by using `zonecfg -z` *zone* `clear cpu-shares`.

2. Verify the configuration change by using `zonecfg -z` *zone* `info`.

3. Reboot the zone to make the configuration effective.

To remove the CPU shares configuration from either the global zone or a non-global zone, use the `zonecfg -z clear cpu-shares` command. The steps for completing this task are listed in the slide.

Removing the CPU Shares Configuration
from a Zone: Example

```
# zonecfg -z hrzone clear cpu-shares
# zonecfg -z hrzone info
zonename: hrzone
zonepath: /zones/hrzone
brand: solaris
autoboot: true
bootargs:
pool:
limitpriv:
scheduling-class:
ip-type: exclusive
hostid:
fs-allowed:
net:
        address not specified
        allowed-address not specified
        configure-allowed-address: true
        physical: vnic1
        defrouter not specified
…
…
…
# zoneadm -z hrzone shutdown -r
```

In the example shown in the slide, you remove the CPU shares configuration from `hrzone` by using the `zonecfg -z clear cpu-shares` command. You then confirm the configuration change. The CPU shares entry is no longer part of the zones configuration. Your final step is to reboot the zone by using the `shutdown -r` command to make the configuration changes effective.

Practice 10-2 Overview: Configuring FSS in an Oracle Solaris Zone

This practice covers the following topics:

- Configuring CPU shares and monitoring FSS in two zones
- Removing the CPU shares configuration

This practice should take you about 30 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Implement a plan for executing a process in an appropriate scheduling class
- Manage process scheduling priority
- Manage the scheduling class of a zone
- Configure the fair share scheduler
- Monitor the fair share scheduler

11

Evaluating System Resources

Objectives

After completing this lesson, you should be able to:
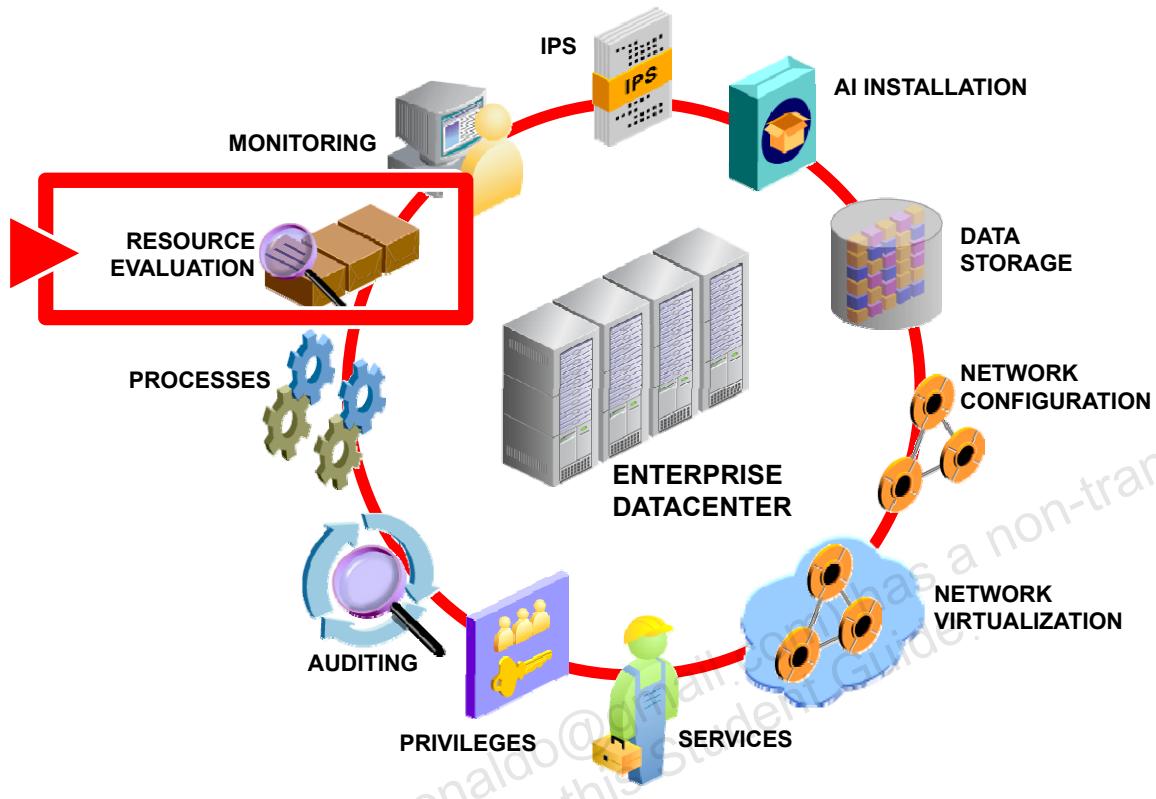
- Implement a plan to evaluate resource allocation and system performance
- Configure system resources
- Monitor system performance

ORACLE

In this lesson, "Evaluating the System Resources," you are introduced to resource controls and shown how to configure system resources to use them. You are also introduced to a number of system utilities that you can use to monitor the usage of these system resources.

Workflow Orientation

Before you begin the lesson, take a moment to orient yourself in your job workflow. Up to this point you have been configuring all the pieces of your system to create a fully functional and secure operating environment. In this lesson you are first shown how to optimize the use of your system resources by configuring the resources and then allocating them. You are then shown how to monitor the usage of these resources to ensure that the system resources have been appropriately allocated to the existing processes.

Lesson Agenda

- **Planning for Resource Allocation and System Performance Evaluation**
- Configuring and Administering System Resources
- Monitoring System Performance

Planning for Resource Allocation and System Performance Evaluation

The resource allocation and system performance evaluation plan ensures that:

- Business applications are being given the appropriate priority in terms of system resource allocation
- Resource allocation is being monitored regularly
- Adjustments are made as necessary to resource controls to ensure continued optimal use of system resources

As part of the predeployment testing activities, your company has put a plan in place that addresses what business application processes should be given priority. The company knows that Oracle Solaris 11 supports resource management, so they are looking to you to create a resource configuration that presents the least compromise to the service goals of the business while working within the limitations of the system's capabilities. The plan also calls for system resources to be monitored on a regular basis and resource controls to be adjusted as necessary to ensure the continued optimal use of the system's resources.

In this topic you are introduced to resource controls as a means of controlling system resource allocation. You are also introduced to a number of tools for monitoring resource usage.

Resource Management

- Resource management enables you to control how applications use available system resources.
- With resource management, you can:
  - Allocate system resources
  - Monitor how the allocations are being used
  - Adjust the allocations as necessary
  - Increase resource usage

You have already learned about resource management in the context of zones, where you controlled your resource allocations through the use of resource pools. In this lesson, you expand your understanding of resource management.

The ability to minimize cross-workload performance compromises, along with the facilities that monitor resource usage, is referred to as resource management. Resource management enables you to control how applications use available system resources. You can allocate system resources, such as processor time and memory, to ensure that your applications have the required response times. You can then monitor how the allocations are being used and adjust the allocations as necessary to address the needs of the business.

You can also use resource management to increase resource usage. By categorizing and prioritizing usage, you can effectively use reserve capacity during off-peak periods, thereby often eliminating the need for additional processing power.

Resource Management Control Mechanisms

- **Constraint:** Limits the consumption of specific resources for a workload
- **Scheduling:** Makes a sequence of allocation decisions at specific intervals
- **Partitioning:** Binds a workload to a subset of the system's available resources

ORACLE

The Oracle Solaris operating system uses three types of resource management control mechanisms: constraints, scheduling, and partitioning.

The constraint mechanism enables you to set bounds on the consumption of specific resources for a workload. You can use bounds to control ill-behaved applications that might negatively compromise system performance or availability through unregulated resource requests. An example of a constraint mechanism is a resource capping.

Scheduling mechanism refers to making a sequence of allocation decisions at specific intervals. An application that has had a scheduling mechanism applied to it leaves the resource available for another application's use if it does not need its current allocation. Scheduling-based resource management enables full usage of an undercommitted configuration, while providing controlled allocations in a critically committed or overcommitted situation. An example of a scheduling mechanism is the fair share scheduler (FSS).

A partitioning mechanism is used to bind a workload to a subset of the system's available resources. This binding guarantees that a known amount of resources is always available to the workload. An example of a partitioning mechanism is a resource pool.

You have already had exposure to using the scheduling and partitioning mechanisms as a means of controlling resources. In the previous lesson on managing processes, you learned how to use scheduling classes to control resource allocation to processes in both the global zone and non-global zones. In the lesson on zones, you used resource pools to control resource allocation. In this lesson, you focus on using constraint mechanisms to set resource controls on the processes associated with projects and tasks.

Projects and Tasks

- **Project:** Provides a network-wide administrative identifier for related work
- **Task:** Collects a group of processes into a manageable entity that represents a workload component
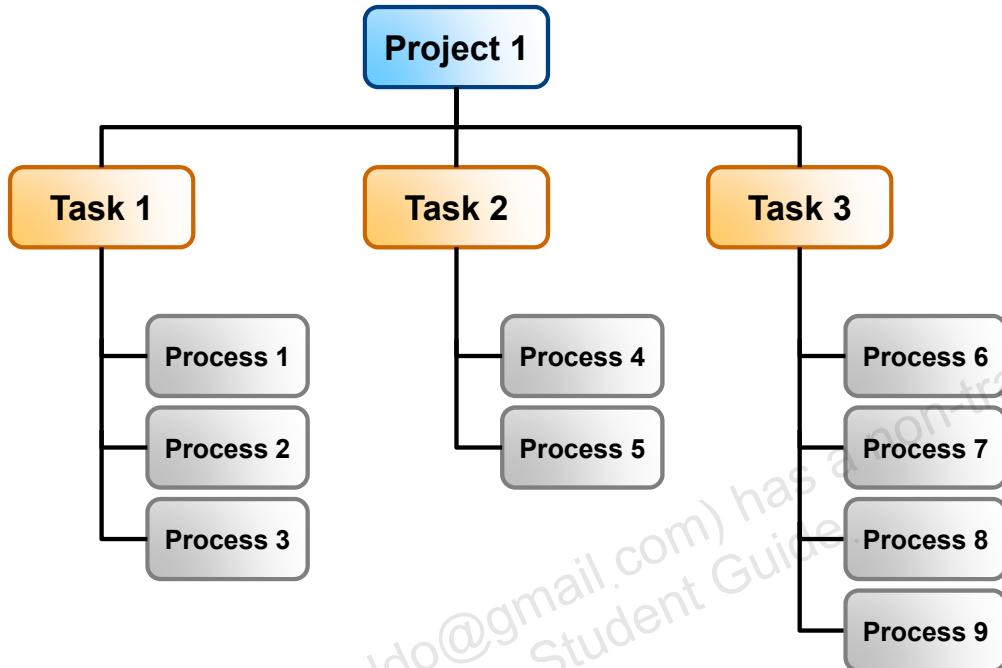
Projects and tasks are used to separate and identify workloads.

To optimize workload response, you must first be able to identify the workloads that are running on the system you are analyzing. This information can be difficult to obtain by using either a purely process-oriented or a user-oriented method alone. In the Oracle Solaris system, you have two additional facilities that can be used to separate and identify workloads: the project and the task. The project provides a network-wide administrative identifier for related work. The task collects a group of processes into a manageable entity that represents a workload component.

Project/Task/Process Relationship

A user or group can belong to one or more projects. These projects can be used to represent the workloads in which the user (or group of users) is allowed to participate. This membership can then be the basis of chargeback that is based on, for example, usage or initial resource allocations. Although a user must be assigned to a default project, the processes that the user launches can be associated with any of the projects of which that user is a member.

Each successful login into a project creates a new task that contains the login process. The task is a process collective that represents a set of work over time. A task can also be viewed as a workload component. Each task is automatically assigned a task ID.

As illustrated by the graphic, each process is a member of one task, and each task is associated with one project.

Resource Controls

- Resource controls can be set at the process, task, project, and zone levels.
- For a list of available resource controls see `resource_controls`.
- Example resource controls include:
  - `process.max-cpu-time`: Maximum CPU time that is available to this process, expressed as a number of seconds
  - `task.max-lwps`: Maximum number of LWPs simultaneously available to this task's processes, expressed as an integer
  - `project.cpu-caps`: Maximum amount of CPU resources that a project can use
  - `zone.max-processes`: Maximum number of processes simultaneously available to a zone, expressed as an integer

You can set resource controls at the process, task, project, and zone levels. You can find a list of the available resource controls for each level on the `resource_controls` man page. Examples of a few resource controls are provided in the slide.

Resource Control Values

Threshold Value:

- A point at which local or global actions can occur
- Associated with the following local actions:
  - `none`: Takes no action on resource requests for an amount that is greater than the threshold
  - `deny`: Denies resource request for an amount that is greater than the threshold
  - `signal=`: Enables a global signal message action when the resource control is exceeded
- Must have an associated privilege level

You define the constraints for a resource control through threshold values and privilege levels. A threshold value on a resource control constitutes a point at which local actions can be triggered or global actions, such as logging, can occur.

Note: Local actions are taken on a process that attempts to exceed the control value. Global actions apply to resource control values for every resource control on the system.

For each threshold value that is placed on a resource control, you can associate one or more actions. There are three types of local actions: `none`, `deny`, and `signal=`. These are defined in the slide.

Note

- The `deny` action is useful for monitoring resource usage without affecting the progress of applications.
- Not all of the actions can be applied to every resource control. For example, a process cannot exceed the number of CPU shares assigned to the project of which it is a member. Therefore, a `deny` action is not allowed on the `project.cpu-shares` resource control.

Each threshold value on a resource control must be associated with a privilege level. You will look at these privilege levels next.

Privilege Levels of Resource Controls

Privilege levels:

- `basic`: Can be modified by the owner of the calling process
- `privileged`: Can be modified by the current process or by the `prctl`(1). Can be abbreviated as `priv`.
- `system`: Is fixed for the duration of the operating system instance

```
task.max-lwps=(priv,1K,deny)
```
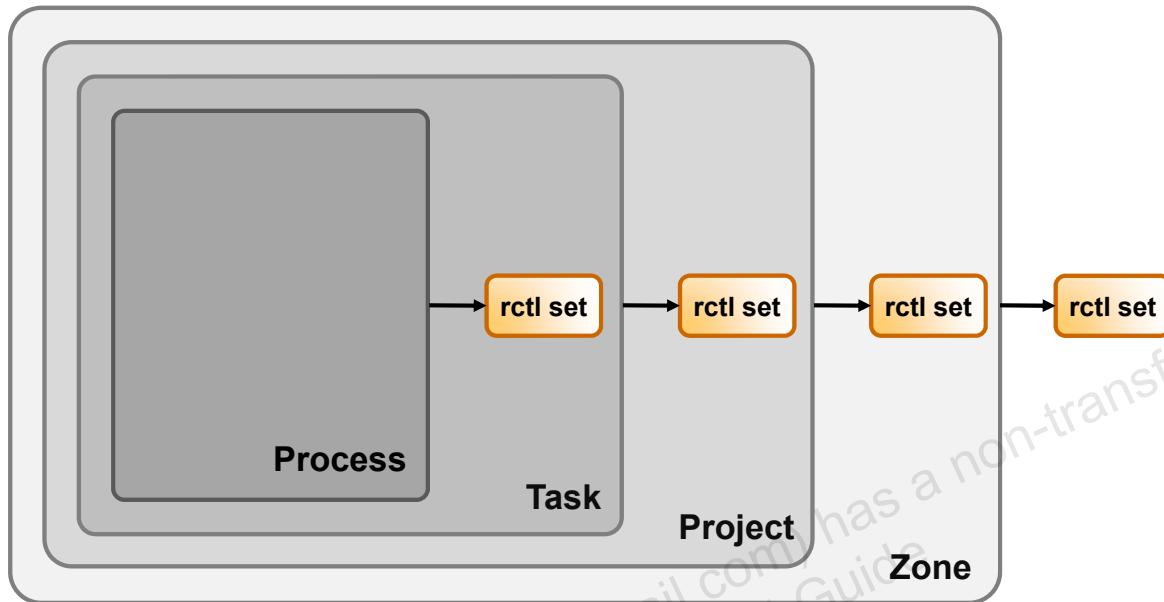
The privilege level for a resource control must be one of these three types: `basic`, `privileged` (`priv`), or `system`. The definitions for all the types are shown in the slide.

A resource control is guaranteed to have one `system` value, which is defined by the system, or resource provider. The `system` value represents how much of the resource the current implementation of the operating system is capable of providing.

You can define any number of `privileged` values, and only one `basic` value is allowed. Operations that are performed without specifying a privilege value are assigned a `basic` privilege by default.

The example shows the `task.max-lwps` resource control. It has been assigned a privilege level of `privileged` (`priv`), which means only the user or current process can modify this limit, a threshold value of `1K`, and the `deny` action.

Enforcing Multiple Resource Controls

More than one resource control can exist on a resource. A resource control can exist at each containment level in the process model. If resource controls are active on the same resource at different container levels, the smallest container's control is enforced first. Thus, action is taken on `process.max-cpu-time` before `task.max-cpu-time` if both controls are encountered simultaneously.

Setting Resource Controls

Use the utilities in the following table to set and modify resource controls:

Utility	Description
`prctl`	Get or set the resource controls of running processes, tasks, and projects.
`projadd`	Administer a new project on the system, to include specifying resource control attributes.
`projmod`	Modify a project's information on the system, to include modifying a project's resource control attributes.
`rctladm`	Display or modify the global state of system resource controls.

You can set and modify the resource controls through the utilities listed in the table shown in the slide. You learn how to configure the resource controls by using each of these utilities later in this lesson.

Default `/etc/project` File

Format of an `/etc/project` file entry:

```
projname:projid:comment:user-list:group-list:attributes
```

Default `/etc/project` file:

```
# cat /etc/project
system:0::::
user.root:1::::
noproject:2::::
default:3::::
group.staff:10::::
```

The resource controls facility is configured through the `project` database.

Note: Updates to entries in the project database, whether to the `/etc/project` file or to a representation of the database in a network naming service, are not applied to currently active projects. The updates are applied to new tasks that join the project when either the `login` or the `newtask` command is used.

Each entry in the `project` database consists of one line of text containing six fields separated by colons (`:`). The format of each entry is shown in the slide. The description for each field is as follows:

- *projname*: Name of the project
- *projid*: Project's unique numerical ID (PROJID) within the system. Project IDs below 100 are reserved for the use of the operating system.
- *comment*: Description of the project

- *user-list*: Comma-separated list of users allowed in the project. An empty field indicates no users are allowed.

  Note: See the project man page for special project exceptions.

- *group-list*: Comma-separated list of groups of users allowed in the project. An empty field indicates no groups are allowed.

  Note: As with the *user-list* there are exceptions. See the project man page for these exceptions.

- *attributes*: Semicolon-separated list of name-value pairs, the most frequent use of which is resource controls. See the project man page for a list of accepted name-value pairs.

An example of the default /etc/project file is shown in the slide.

Setting Zone-Wide Resource Controls

- The total resource usage of all process entities in a zone is limited.
- You can specify limits for both the global and non-global zones using either:
  - `zonecfg` command (limits are persistent)
  - `prctl` command (limits are not persistent)
- Examples of zone-wide resource controls include:
  - `zone.cpu-cap`: Limits the amount of CPU resource for the zone
  - `zone.cpu-shares`: Number of fair share scheduler (FSS) CPU shares for the zone
  - `zone.max-locked-memory`: Total amount of physical locked memory that is available to a zone

Zone-wide resource controls limit the total resource usage of all process entities within a zone. These limits are specified for both the global and non-global zones by using the `zonecfg` command.

You can also specify these limits for running processes by using the `prctl` command. However, the limits you specify through the `prctl` command are not persistent. They are in effect only until the system is rebooted.

Some examples of zone-wide resource controls are shown in the slide. For a complete listing and description of the zone-wide resource controls, see the "Setting Zone-Wide Resource Controls" section of *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management*.

Monitoring Resource Consumption

Performance tools enable you to:
- View current resource consumption
- Evaluate the need to:
    - Restrict access to a given resource
    - Isolate particular workloads from other workloads

In this lesson, you learn how to use these utilities:
- `vmstat`: Reports virtual memory statistics
- `iostat`: Reports I/O statistics
- `df`: Displays status of disk space on file systems
- `sar`: Reports on system activities

Oracle Solaris 11 supports a number of performance tools that enable you to view the current resource consumption of workloads that are running on your system. By using these tools, you can evaluate whether you must restrict access to a given resource or isolate particular workloads from other workloads.

In this lesson you learn how to use the `vmstat`, `iostat`, and `df` utilities to evaluate memory and disk resource usage. You also learn how to use the `sar` utility to monitor system activities.

Implementing the Resource Allocation and System Performance Evaluation Plan

Your assignment is to:

- Configure system resources
- Put resource controls in place
- Monitor the use of these resources

As part of the predeployment testing effort, you have been assigned the task of configuring system resources, putting resource controls in place, and then monitoring the use of these resources.

In the topics that follow, you learn how to complete each of these tasks.

Quiz

Which of the following resource management control mechanisms limits the consumption of specific resources for a workload?

a. Constraint
b. Scheduling
c. Partitioning

ORACLE

Answer: a

Quiz

Resource controls can be set at the process, task, project, and zone levels.

  a. True

  b. False

Answer: a

Quiz

A resource control threshold value must have an associated privilege level.

a. True

b. False

Answer: a

Quiz

In the following resource control, which value defines a local action?

```
task.max-lwps=(priv,1K,deny)
```

 a. `task.max-lwps=`
 b. `priv`
 c. `1K`
 d. `deny`

Answer: d

Oracle Solaris 11 Advanced System Administration   11 - 24

Quiz

You can specify limits for both the global and non-global zones by using `zonecfg`, but the limits are not persistent.

a. True

b. False
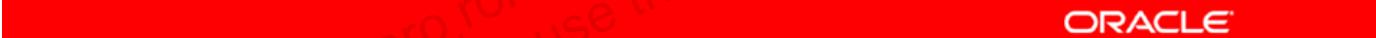
Answer: b

Lesson Agenda

- Planning for Resource Allocation and System Performance Evaluation
- **Configuring and Administering System Resources**
- Monitoring System Performance

Configuring and Administering System Resources

This section covers the following topics:

- Administering projects and tasks
- Administering resource controls and attributes

Administering Projects and Tasks

- Displaying the default projects in a system
- Defining a project
- Obtaining project membership information
- Modifying a project
- Adding attributes and attribute values to a project
- Substituting attributes and attribute values for a project
- Removing attributes or attribute values from a project
- Displaying currently running projects
- Creating a new task
- Moving a running process into a new task
- Deleting a project

Displaying the Default Projects in the System

To display the default projects in a system, use `project -l`.

```
# projects -l
system
        projid : 0
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
user.root
        projid : 1
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
<continued on next slide>
```

ORACLE

Default `/etc/project` File

```
<continued from previous slide>
noproject
        projid : 2
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
default
        projid : 3
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
group.staff
        projid : 10
        comment: ""
        users  : (none)
        groups : (none)
        attribs:
```

The continuation of the default `/etc/project` file is shown in the example in the slide.

Defining a Project

1. View the default projects on your system by using `projects -l`.

2. Add a project by using `projadd -U username -p projid project`.

3. View the `projects` file again to verify that the new project has been added.

```
# projects -l
# projadd -U jjones -p 4115 testproj
# projects -l
<output omitted>
testproj
        projid : 4115
        comment: ""
        users  : jjones
        groups : (none)
        attribs:
```

ORACLE

To define a project, you add it by using the `projadd` command. The steps listed in the slide show how to define a project.

Notes for step 1: Check to see what projects have been defined in the system and determine what project ID number is available for your project.

Notes for step 2: The options that are used with the `projadd` command are as follows:

- `-U user`: Specifies a user for the project. Multiple users can be specified by using a comma-separated list.
- `-p projid`: Sets the project ID for the new project

  Note: The `projid` should be specified as a non-negative decimal integer below `UID_MAX` as defined in `limits.h`. The `projid` defaults to the next available unique number above the highest number currently assigned. For example, if `projids` 100, 105, and 200 are assigned, the next default `projid` is 201. `projids` between 0 and 99 are reserved by the Oracle Solaris operating system.

For a full list of options, see the `projadd` man page.

In the example in the slide, after you have checked the project file, you create a new project called `testproj` with project ID `4115` and assign it to the user `jjones`. You then verify that your new project has been added to the `projects` file.

Obtaining Project Membership Information

To obtain information about project membership, use `id -p`.

```
# /usr/bin/id -p
uid=0(root) gid=0(root) projid=4015(testproj)
```

To obtain information about project membership, you use the `id -p` command. The `id` command is used to return user identity. The `-p` option provides information about the current project membership of the invoking process.

In the example in the slide, you are displaying the identity of the current user, which in this case is `root`. You can see that the project you just created, `testproj` (`4015`), has been assigned to this user.

Modifying a Project

1. Modify the project by using `projmod`.
2. View the `projects` file to verify that the modifications to the project have been added.

```
# projmod -G testers -c "Oracle Solaris test team" testproj
# projects -l
<output omitted>
testproj
        projid : 4115
        comment: "Oracle Solaris test team"
        users  : jjones
        groups : testers
        attribs:
```

To modify the information associated with a project, such as giving the project a new name or project ID or adding a comment, you use the `projmod` command, as shown in the steps listed in the slide.

For a list of the options you can use with the `projmod` command, see the `projmod` man page.

In the example in the slide, you are making several modifications to the `testproj` project. You are adding a group called `testers` by using the `-G` option, and you are adding a short description of the project, "`Oracle Solaris test team`", by using the `-c` option. You then verify that your modifications are reflected in the `projects` file.

Adding Attributes and Attribute Values
to a Project

1. Add an attribute to a project by using `projmod -a -K`
    name=value project.

2. Add another value to the existing list of values by using the
    same options.

3. View the `projects` file to verify that the attribute and
    attribute values have been added.

```
# projmod -a -K "task.max-lwps=(priv,100,deny)" testproj
# projmod -a -K "task.max-lwps=(priv,1000,signal=KILL)" testproj
# projects -l
<output omitted>
testproj
        projid : 4115
        comment: "Oracle Solaris test team"
        users  : jjones
        groups : testers
        attribs: task.max-lwps=(priv,100,deny),(priv,1000,signal=KILL)
```

You can also use the `projmod` command to edit project attributes. The steps for adding an attribute are shown in the slide.

The `-K` option specifies a replacement list of attributes. Attributes are delimited by semicolons (`;`). When the `-K` option is used with the `-a` option, the attribute or attribute value is added.

Notes for step 1: The value consists of a privilege level, a threshold value, and an action associated with reaching the threshold.

Notes for step 2: Multiple values are separated by commas.

In the example in the slide, you are adding a resource control attribute to the project that will restrict the maximum number of lightweight processes (`max-lwps`) to 100. You then add another resource control attribute that generates a `KILL` signal to the project if the number of lightweight processes exceeds `1000`.

Your last step is to verify that the attribute and attribute values have been added to the `projects` file, and they have.

Substituting Attributes and Attribute Values for a Project

1. Replace an attribute to a project by using `projmod -s -K name=value project`.

2. View the `projects` file to verify that the attribute and attribute values have been replaced.

```
# projmod -s -K "task.max-lwps=(priv,120,deny),(priv,800,signal=KILL)" testproj
# projects -l
<output omitted>
testproj
        projid : 4115
        comment: "Oracle Solaris test team"
        users  : jjones
        groups : testers
        attribs: task.max-lwps=(priv,120,deny),(priv,800,signal=KILL)
```

You can substitute attributes and attribute values for a project by using the `projmod` command with the `-s` and `-K` options, as shown in the steps in the slide.

Notes for step 1: If the attribute does not exist, it is created.

Notes for step 2: Multiple values are separated by commas.

In the example in the slide, you are replacing the current `task.max-lwps` values that you defined previously with the new values shown.

To verify that the substitution has been made, you view the `projects` file. You can see here that the substitution for the resource control attribute has been made.

Removing Attributes or Attribute Values
from a Project

1. Remove an attribute or attribute value from a project by using `projmod -r -K name=value project`.

2. View the `projects` file to verify that the attribute or attribute value has been removed.

```
# projmod -r -K "task.max-lwps=(priv,120,deny)" testproj
# projects -l
<output omitted>
testproj
        projid : 4115
        comment: "Oracle Solaris test team"
        users  : jjones
        groups : testers
        attribs: task.max-lwps=(priv,800,signal=KILL)
```

```
# projmod -r -K task.max-lwps testproj
```

If you want to remove an attribute or attribute value from a project, you use the `-r` and `-K` options with the `projmod` command, as shown in the steps listed in the slide.

In the first example, you are removing the attribute value that restricts the maximum number of lightweight processes (`max-lwps`) to `120`. You then verify that the attribute value has been removed from the project's attribute entry in the `projects` file, and it has. The second attribute value that you added previously still remains.

In the second example, you are removing the entire resource control attribute. If you were to view the `projects` file again, you would see nothing listed in the `attribs` field.

Displaying Currently Running Processes and Projects

To display the processes and projects that are currently running on the system, use `prstat -JR`.

```
# prstat -JR
…
…
…
PROJID    NPROC   SWAP    RSS MEMORY      TIME   CPU  PROJECT
  4015        2   312K  7328K   0.7%   2:35:44   50%  testproj
     1        3  2912K    17M   1.6%   0:00:00  0.3%  user.root
     0       99   142M   170M    17%   0:00:47  0.0%  system
```

To display statistical information, such as memory and CPU usage, for the processes and projects that are currently running on the system, you can use the `prstat -JR` command, as shown in this example.

The command displays the following information:

- `PROJID`: ID number of the project
- `NPROC`: Number of processes in the project
- `SWAP`: Total virtual memory size of the process, including all mapped files and devices, in kilobytes (`K`), megabytes (`M`), or gigabytes (`G`)
- `RSS`: Resident set size of the process in kilobytes (`K`), megabytes (`M`), or gigabytes (`G`)
- `MEMORY`: Percentage of memory used by a specified collection of processes
- `TIME`: Cumulative execution time for the process
- `CPU`: Percentage of recent CPU time used by the process
- `PROJECT`: Project name

Creating a New Task

To create a new task, use `newtask -v -p project`.

```
# newtask -v -p testproj
16
```

To create a new task and associate it with a project, you use the `newtask` command, as shown in the slide. You can use the `-v` option with this command to obtain the system task ID. The `-p` option specifies the project. The `newtask` command creates a new task in the specified project and places the user's default shell in this task.

For a full list of options, see the `newtask` man page.

In the example in the slide, you are creating a task for the `testproj` project. The task ID is `16`.

Moving a Running Process into a New Task

1. Determine the process ID by using `pgrep process`.

2. Associate the process ID with a task ID in a project by using `newtask -v -p projid -c pid`.

3. Confirm the task to process ID mapping by using `pgrep -T taskID`.

```
# pgrep test1
    8103
# newtask -v -p testproj -c 8100
    15
# pgrep -T 15
    8103
```

If you are handling a critical process that cannot be restarted in order to place it into a new project, you can take a running process and put it into an existing project by creating a new task. To associate a running process with a new task in an existing project, use the `newtask` command, as shown in the steps in the slide.

Note: To perform this task, you must either be the superuser, have the required rights profile, or be the owner of the process and be a member of the new project.

Notes for step 1: Check to see what projects have been defined in the system and determine what project ID number is available for your project.

Notes for step 2: The options that are used with the `newtask` command are as follows:

- `-p project_name`: Specifies the project name
- `-c pid`: Specifies the process ID of the process that is being mapped to the task

In the example in the slide, you have a running process called `test1` that you want to map to a task associated with the `testproj` project. First, you determine the process ID for `test1`; it is `8103`. You then map the running process's PID to `testproj` by using the `newtask` command, which generates a new task with the task ID `15`. Your last step is to verify that the new task is mapped to the running process, and it is.

Deleting a Project

1. Remove the project by using `projdel` *project*.
2. Display the `projects` file by using `project -l` to verify that the project has been deleted.
3. Log in as a user and enter `projects` to view the projects that are assigned to this user.

```
# projdel testproj
# projects -l
<output omitted>
# su - jjones
# projects
default
```

If you no longer need a project, you can delete it by using the `projdel` command. The steps listed in the slide show how to remove a project from the `/etc/project` file.

Notes for step 3: You should no longer see the deleted project listed.

In the example in the slide, you are deleting the `testproj` project by using the `projdel` command. You then verify that the project no longer appears in the `projects` file. Next, you log in as the user `jjones` to again verify that the project is no longer assigned to this user. As you can see, the `testproj` project is no longer associated with `jjones`. The only project assigned to `jjones` is the `default` project.

Administering Resource Controls and Attributes

- Displaying the default resource controls
- Displaying information about a given resource control
- Displaying current resource control settings
- Monitoring resource control events globally

In this section, you learn how to display the default resource controls for the system; how to display information for a specific resource control; how to display the current resource control settings for a process, task, project, or zone; and how to set up system-wide resource control monitoring.

Displaying the Default Resource Controls

To display the default resource controls, use `prtcl $$`.

```
# prctl $$
process: 3320: bash
NAME        PRIVILEGE       VALUE    FLAG   ACTION                        RECIPIENT
process.max-port-events
            privileged      65.5K      -    deny                                  -
            system          2.15G     max   deny                                  -
…
…
task.max-cpu-time
            usage              0s
            system          18.4Es    inf   none                                  -
…
…
project.max-contracts
            privileged      10.0K      -    deny                                  -
            system          2.15G     max   deny                                  -
…
…
zone.max-lofi
            usage              0
            system          18.4E     max   deny                                  -
…
…
```

To determine what resource controls are available for a process, such as the current shell that is running, you use the `prtcl $$` command.

Note: `$$` refers to the current shell process.

This command can be used only on a system on which you have not set or changed the resource controls. There can be only non-default entries in the `/etc/system` file or in the `project` database.

Note: The `prtcl` command can be used to get or set the resource controls of running processes, tasks, projects, and zones.

In the example in the slide, which contains only a partial sample, the resource controls that are available for the `bash` process are listed. They include resource controls for processes, tasks, projects, and zones. The threshold value, flags, actions, and recipient are listed for each resource control attribute.

Note: For a complete list of local flags, global flags, and their definitions, see `rctlblk_set_value` (3C).

You will have a chance to see the full list of available resources during the practice.

Displaying Current Resource Control Settings

To display the current resource control settings,
use `prctl -i id`.

```
# ps -o taskid -p $$
TASKID
96
# prctl -i task 96
task: 96
NAME      PRIVILEGE         VALUE    FLAG    ACTION                        RECIPIENT
task.max-cpu-time
        usage               26s
        system              18.4Es   inf     none                                  -
…
…
project.cpu-shares
        usage               1
        privileged          1        -       none                                  -
        system              65.5K    max     none                                  -
zone.max-lofi
        usage               0
        system              18.4E    max     deny                                  -
…
…
```

To determine what the resource control settings are for a process, project, task, or zone, you can use the `prctl -i` command with the process, project, task, or zone ID.

In the example in the slide, you want to display the current resource control settings for a particular task. You run the `ps -o` command to determine the task ID for the currently running process. The task ID is `96`. You then run the `prctl` command for task `96` to display the current control settings for that task.

Displaying Information About a Given Resource Control

To display information about a specific resource control, use `prtcl -n control.attribute $$`.

```
# prctl -n task.max-lwps $$
process: 3220: bash
NAME       PRIVILEGE        VALUE     FLAG    ACTION                              RECIPIENT
task.max-lwps
        usage               3
        privileged          3         -       deny                                -
        system              2.15G     max     deny
```

To display information about a specific resource control, use the `prtcl` command with the `-n` option to specify the name of the resource control, followed by the resource control attribute and `$$`.

Enabling Global Resource Control Monitoring

To enable the global resource control monitoring,
use `rctladm -e syslog control.attribute`.

```
# rctladm -e syslog task.max-lwps
```

```
# rctladm
process.max-port-events    syslog=off  [ deny count ]
process.max-msg-messages   syslog=off  [ deny count ]
process.max-msg-qbytes     syslog=off  [ deny bytes ]
process.max-sem-ops        syslog=off  [ deny count ]
…
…
task.max-cpu-time          syslog=off     [ no-deny cpu-time no-obs inf seconds ]
task.max-processes         syslog=off     [ count ]
task.max-lwps              syslog=notice  [ count ]
…
…
zone.max-lofi              syslog=off  [ no-basic deny count ]
zone.max-swap              syslog=off  [ no-basic deny bytes ]
zone.max-locked-memory     syslog=off  [ no-basic deny bytes ]
…
…
```

After you have set a resource control, you can enable system-wide resource controls to monitor resource consumption and log a notification to `syslog` when a resource control threshold value is exceeded.

To enable the global `syslog` attribute of a resource control, use the `rctladm -e syslog` command with the global `syslog` attribute for the resource control, as shown in the slide.

Note: The `rctladm` command is used to display or modify the global state of system resource controls. For a list of options that can be used with this command, see the `rctladm` man page.

In the example in the slide, you are enabling the global `syslog` attribute of `task.max-lwps`.

By using the `rctladm` command without arguments, you can view the global logging state of each resource control on a system-wide basis, as shown in the second example. In the example in the slide, you can see that because you have enabled global resource control monitoring for the `task.max-lwps` resource control, `syslog` messaging for that resource control has been set to `notice`. When the threshold for this resource control value is exceeded, a log entry will be generated in the `/var/adm/messages` file.

Practice 11-1 Overview: Managing Resource Controls in Global and Non-Global Zones

This practice covers the following topics:

- Administering projects and tasks
- Configuring resource controls and attributes

The practices for this lesson are designed to reinforce the concepts that have been presented in the lecture portion. These practices cover the following tasks:

- **Practice 11-1:** Managing resource controls in global and non-global zones
- **Practice 11-2:** Evaluating system performance levels

Practice 11-1 should take you about 30 minutes to complete.

Lesson Agenda

- Planning for Resource Allocation and System Performance Evaluation
- Configuring and Administering System Resources
- **Monitoring System Performance**

Monitoring System Performance

This section covers the following topics:

- Displaying virtual memory statistics and information
- Displaying disk usage information
- Monitoring system activities

Displaying Virtual Memory Statistics and Information

- Displaying virtual memory statistics
- Displaying system event information
- Displaying swapping statistics

To obtain virtual memory statistics and information about system events, such as CPU load, paging, number of context switches, device interrupts, and system calls, you can use the `vmstat` command. You can also use this command to display statistics on swapping, cache flushing, and interrupts. In this section, you focus on using the `vmstat` command to display virtual memory statistics, system event information, and swapping statistics.

Note: To see information about how to use `vmstat` to gather other types of virtual memory-related statistics, see *Oracle Solaris Administration: Common Tasks*.

Displaying Virtual Memory Statistics

To display virtual memory statistics, use `vmstat n`.

```
# vmstat 5
kthr     memory          page          disk      faults      cpu
r b w   swap  free  re mf pi p fr de sr s0 s1 s2 s3   in  sy  cs  us sy id
0 0 0 11456 4120   1 41 19 1  3  0  2  0  4  0  0   48 112 130   4 14 82
0 0 1 10132 4280   0  4 44 0  0  0  0  0 23  0  0  211 230 144   3 35 62
0 0 1 10132 4616   0  0 20 0  0  0  0  0 19  0  0  150 172 146   3 33 64
0 0 1 10132 5292   0  0  9 0  0  0  0  0 21  0  0  165 105 130   1 21 78
1 1 1 10132 5496   0  0  5 0  0  0  0  0 23  0  0  183  92 134   1 20 79
1 0 1 10132 5564   0  0 25 0  0  0  0  0 18  0  0  131 231 116   4 34 62
1 0 1 10124 5412   0  0 37 0  0  0  0  0 22  0  0  166 179 118   1 33 67
1 0 1 10124 5236   0  0 24 0  0  0  0  0 14  0  0  109 243 113   4 56 39
```

The `vmstat` command reports virtual memory statistics regarding kernel thread (`kthr`), virtual memory (`memory`), disk (`disk`), trap (`faults`), and CPU (`cpu`) activity. A five-second interval is a good choice for live monitoring.

Note: For a description of each field, see the `vmstat` man page.

By using this command, you can determine virtual memory performance and identify memory bottlenecks. Turn your attention to the `page` and `cpu` fields. In the page field you want to look for `po` (page outs) and `sr` (scan rate). When both are consistently high (more than 100 per second) at the same time, the page daemon is being forced to steal free memory from running processes.

Note: The `free` column (located in the memory section to the right of the `swap` column) may not be a good indication of the available memory in the system. This is because, after memory pages are used by the file system buffer cache, they are not returned to the free list. When the page daemon detects a memory shortfall, it scans for pages that can be freed. Pages are then freed from the file system buffer cache for the use of applications.

In the `cpu` field, when the system is consuming less CPU, more memory is available to the system. This usage is reflected in the `sy` column and by the amount of CPU idle time reflected in the `id` column.

Note: The `us` column displays user time.

In the example in the slide, on the last line of the output, you can see that the system is staying on the CPU longer, which means the CPU idle time is lower, which equates to less available memory. However, on the first line of the output, the system is consuming very little CPU time and the idle time is very high, which means more memory is available.

Displaying System Event Information

To display system event information, use `vmstat -s`.

```
# vmstat -s
        0 swap ins
        0 swap outs
        0 pages swapped in
        0 pages swapped out
   522586 total address trans. faults taken
    17006 page ins
       25 page outs
    23361 pages paged in
       28 pages paged out
    45594 total reclaims
    45592 reclaims from free list
        0 micro (hat) faults
   522586 minor (as) faults
    16189 major faults
    98241 copy-on-write faults
   137280 zero fill page faults
    45052 pages examined by the clock daemon
        0 revolutions of the clock hand
       26 pages freed by the clock daemon
     2857 forks
       78 vforks
<output omitted>
```

ORACLE

To display system event information (specifically the system events that have occurred since the last reboot), you use the `vmstat -s` command, as shown in the slide. This command can give you an indication of what is occurring in the system that might be causing a load on the system memory. The number of reclaims from free list is an indication of how quickly the system was running out of memory. Because programs require memory to run, it might explain why there is a load on the system.

Other system events that can impact memory are the number of forks that have occurred.

Note: Forks refer to the number of processes launching subprocesses.

Each subprocess that is launched creates a workload that requires memory and CPU resources to run.

Displaying Swapping Statistics

To display swapping statistics, use `vmstat -S`.

```
# vmstat -S
kthr      memory            page            disk         faults      cpu
 r b w   swap   free  si  so pi po fr de sr dd f0 s1 --   in   sy   cs us sy id
 0 0 0 862608 364792  0   0  1  0  0  0  0  0  0  0  0  406  394  213  1  0 99
```

To display swapping statistics, use the `vmstat` command with the `-S` option. With this command, you can evaluate the workload created by one job running in the background.

Displaying Disk Usage Information

- Displaying general disk usage information
- Displaying disk space information

If you are specifically interested in monitoring disk usage, you can use the `iostat` command, in both a normal and an extended format. If you want to find out about disk space, you can use the `df` command.

Displaying General Disk Usage Information

To display general disk usage information, use `iostat n`.

```
# iostat 5
   tty          sd0              sd1              sd2              sd3             cpu
 tin tout kps tps serv   kps tps serv   kps tps serv   kps tps serv   us sy wt id
   0    3 138   4   51     1   0    7     0   0    0     0   0    0    4 10  0 86
   0   47   0   0    0     0   0    0     0   0    0     0   0    0    8 18  0 74
   0   16  50  18    3     0   0    0     0   0    0     0   0    0    8 18  0 74
   0   16   0   0    0     0   0    0     0   0    0     0   0    0    8 18  0 74
```

The `iostat` utility provides statistics on terminal, disk, tape I/O, and CPU usage activity. The first line of output shows the statistics from the last time the system was booted. Each subsequent line shows the interval statistics. The default is to show statistics for the terminal (`tty`), disks (`fd` and `sd`), and CPU (`cpu`).

Note: For a description of each field, see the `iostat` man page.

With this command, you can determine which disks are taking more time to service transactions by comparing the service times (`serv` column under each disk) for each disk. In the example in the slide, you can see that the service time for transactions for the `sd1` disk is 7 milliseconds as compared to the 51 milliseconds it is taking the `sd0` disk to service transactions. Based on this information, you could determine that the `sd0` disk is taking longer to service transactions; however, you need to keep in mind the nature of the transactions, which can impact the length of time it takes a disk to service a transaction.

Displaying Disk Space Information

To display disk space information, use `dh -h`.

```
# df -h | more
Filesystem          Size  Used Avail Use% Mounted on
rpool/ROOT/solaris   14G  3.5G   11g  25% /
swap                1.2G  388K  1.2G   1% /etc/svc/volatile
/usr/lib/libc/libc_hwcap3.so.1
                     14G  3.5G   11g  25% /lib/libc.so.1
swap                1.2G   56K  1.2G   1% /tmp
swap                1.2G   60K  1.2G   1% /var/run
ora                 202G   60G  142G  30% /opt/ora
rpool/export         11g   35K   11g   1% /export
rpool/export/home    11g   34K   11g   1% /export/home
rpool/export/home/jholt
                     11g   31K   11g   1% /export/home/jholt
rpool/export/home/oracle
                     11g  5.0M   11g   1% /export/home/oracle
rpool/export/home/tshane
                     11g   31K   11g   1% /export/home/tshane
```

To show the amount of disk space occupied by the mounted file systems, the amount of used and available space, and how much of the file system's total capacity has been used, you can use the `df -h` command, as shown in the slide.

Note: The usable disk space that the `df` command reports reflects only 90 percent of full capacity. This is because the reporting statistics allow for 10 percent above the total available space. The percentage of disk space actually reported by the `df` command is used space divided by usable space.

In the example in the slide, you can see that the ZFS file system has used up 3.5 GB out of 14 GB, which equates to 25% of the file system's total capacity.

Monitoring System Activities

- Checking file access operation statistics
- Checking buffer activity
- Checking system call statistics
- Checking disk activity
- Checking unused memory
- Setting up automatic data collection

ORACLE

There are a number of system activities that you can monitor by using the sar utility. In this section, you focus on five: file access operation statistics, buffer activity, system call statistics, disk activity, and unused memory. You conclude this section by learning about how to collect data automatically.

For a full list of activities you can monitor with the sar utility, see *Oracle Solaris Administration: Common Tasks*.

Checking File Access Operation Statistics

To display file access operation statistics, use `sar -a`.

```
# sar -a

SunOS s11-desktop 5.11 11.1 i86pc     12/20/2012

00:00:00  iget/s namei/s dirbk/s
01:00:00       0       3       0
02:00:00       0       3       0
03:00:00       0       3       0
04:00:00       0       3       0
05:00:00       0       3       0
06:00:00       0       3       0
07:00:00       0       3       0
08:00:00       0       3       0
08:20:01       0       3       0
08:40:00       0       3       0
09:00:00       0       3       0
09:20:01       0      10       0
09:40:01       0       1       0
10:00:02       0       5       0

Average        0       4       0
```

To display file access operation statistics, use the `sar -a` command. The `-a` option is helpful for viewing how disk-dependent an application is. The output of the command is as follows:

- `iget/s`: Number of requests made for inodes that were not in the directory name look-up cache (DNLC)
- `namei/s`: Number of file system path searches per second
- `dirbk/s`: Number of directory block reads issued per second

Note: You can set the number of displays you want displayed by time intervals in seconds. For example, if you want four displays provided every 10 seconds, you use the command:

```
# sar -a 10 4
```

The amount of time reflects how heavily programs and applications are using the file systems. The larger the reported values for these operating system routines, the more time the kernel is spending to access user files. At the system level, if this number is high, then you need to be concerned.

Checking Buffer Activity

To display buffer activity, use `sar -b`.

```
# sar -b

SunOS s11-desktop 5.11 11.1 i86pc    12/20/2012

00:00:04 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00       0       0     100       0       0      94       0       0
02:00:01       0       0     100       0       0      94       0       0
03:00:00       0       0     100       0       0      92       0       0
04:00:00       0       1     100       0       1      94       0       0
05:00:00       0       0     100       0       0      93       0       0
06:00:00       0       0     100       0       0      93       0       0
07:00:00       0       0     100       0       0      93       0       0
08:00:00       0       0     100       0       0      93       0       0
08:20:00       0       1     100       0       1      94       0       0
08:40:01       0       1     100       0       1      93       0       0
09:00:00       0       1     100       0       1      93       0       0
09:20:00       0       1     100       0       1      93       0       0
09:40:00       0       2     100       0       1      89       0       0
10:00:00       0       9     100       0       5      92       0       0
10:20:00       0       0     100       0       0      68       0       0
10:40:00       0       1      98       0       1      70       0       0
11:00:00       0       1     100       0       1      75       0       0

Average        0       1     100       0       1      91       0       0
```

To display buffer activity, use the `sar -b` command.

Note: The buffer is used to cache metadata. Metadata includes inodes, cylinder group blocks, and indirect blocks.

The most important entries are the cache hit ratios `%rcache` and `%wcache`. These entries measure the effectiveness of system buffering. If `%rcache` falls below 90 percent or if `%wcache` falls below 65 percent, you might be able to improve performance by increasing the buffer space. In the example in the slide, the `%rcache` and `%wcache` buffers are not causing any slowdowns. All the data is within acceptable limits.

Checking System Call Statistics

To display system call statistics, use `sar -c`.

```
# sar -c

SunOS s11-desktop 5.11 11.1 i86pc    12/20/2012

00:00:04 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
01:00:00      89      14       9    0.01    0.00    2906    2394
02:00:01      89      14       9    0.01    0.00    2905    2393
03:00:00      89      14       9    0.01    0.00    2908    2393
04:00:00      90      14       9    0.01    0.00    2912    2393
05:00:00      89      14       9    0.01    0.00    2905    2393
06:00:00      89      14       9    0.01    0.00    2905    2393
07:00:00      89      14       9    0.01    0.00    2905    2393
08:00:00      89      14       9    0.01    0.00    2906    2393
08:20:00      90      14       9    0.01    0.01    2914    2395
08:40:01      90      14       9    0.01    0.00    2914    2396
09:00:00      90      14       9    0.01    0.01    2915    2396
09:20:00      90      14       9    0.01    0.01    2915    2396
09:40:00     880     207     156    0.08    0.08   26671    9290
10:00:00    2020     530     322    0.14    0.13   57675   36393
10:20:00     853     129      75    0.02    0.01   10500    8594
10:40:00    2061     524     450    0.08    0.08  579217  567072
11:00:00    1658     404     350    0.07    0.06 1152916 1144203

Average      302      66      49    0.02    0.01   57842   55544
```

To display system call statistics, such as number of system calls, reads, writes, and forks, use the `sar -c` command. Typically, reads and writes account for about half of the total system calls. However, the percentage varies greatly with the activities that are being performed by the system.

Note: For a description of each field, see the `sar` man page.

This information is useful when you are developing metrics or want to use dtrace to track down a very high number of system calls.

Checking Disk Activity

To display disk activity, use `sar -d`.

```
# sar -d

SunOS s11-desktop 5.11 11.1 i86pc   12/2/2012
12:36:32   device        %busy  avque  r+w/s  blks/s  avwait  avserv

12:40:01   dad1            15    0.7     26    399    18.1    10.0
           dad1,a          15    0.7     26    398    18.1    10.0
           dad1,b           0    0.0      0      1     1.0     3.0
           dad1,c           0    0.0      0      0     0.0     0.0
           dad1,h           0    0.0      0      0     0.0     6.0
           fd0              0    0.0      0      0     0.0     0.0
           nfs1             0    0.0      0      0     0.0     0.0
           nfs2             1    0.0      1     12     0.0    13.2
           nfs3             0    0.0      0      2     0.0     1.9
           nfs4             0    0.0      0      0     0.0     7.0
           nfs5             0    0.0      0      0     0.0    57.1
           nfs6             1    0.0      6    125     4.3     3.2
           nfs7             0    0.0      0      0     0.0     6.0
           sd1              0    0.0      0      0     0.0     5.4
...
...
...
```

To display disk activity, use the `sar -d` command. The output will provide you with information about the name of the device that is being monitored (`device`), the percentage of time the device was busy servicing a transfer request (`%busy`), the average number of requests (`avque`), the number of read/write transfers in seconds (`r+w/s`), the number of block transfers (`blks/s`), average wait time (`avwait`), and average time it took for a request to be completed by the device (`avserv`).

Note: For a description of each field, see the `sar` man page.

Queue lengths and wait times are measured when something is in the queue. If `%busy` is small, large queues and service times probably represent the periodic efforts by the system to ensure that altered blocks are promptly written to the disk. If any of these numbers are too high for your application, there could be a disk issue.

Checking Unused Memory

To display unused memory, use `sar -r`.

```
# sar -r

SunOS s11-desktop 5.11 11.1 i86pc    12/20/2012

00:00:04 freemem freeswap
01:00:00   44717  1715062
02:00:01   44733  1715496
03:00:00   44715  1714746
04:00:00   44751  1715403
05:00:00   44784  1714743
06:00:00   44794  1715186
07:00:00   44793  1715159
08:00:00   44786  1714914
08:20:00   44805  1715576
08:40:01   44797  1715347
09:00:00   44761  1713948
09:20:00   44802  1715478
09:40:00   41770  1682239
10:00:00   35401  1610833
10:20:00   34295  1599141
10:40:00   33943  1598425
11:00:00   30500  1561959

Average    43312  1699242
```

To display unused memory, use the `sar -r` command. The output will provide you with the number of currently unused memory pages and swap-file disk blocks. The `freemem` column displays the average number of pages available to user processes. The `freeswap` column displays the average number of disk blocks available for page swapping.

By monitoring these numbers over time to establish a trend, you can determine if you are in danger of running out of memory and then take appropriate action to correct the situation.

Setting Up Automatic Data Collection

1. Run the `svcadm enable system/sar:default` command (as necessary).
2. Edit the `/var/spool/cron/crontabs/sys` crontab file by using `crontab -e sys`.
3. Uncomment the last entry to run the system script `sa2`.

```
# svcadm enable system/sar:default
# crontab -e sys
…
…
…
#0 * * * 0-6 /usr/lib/sa/sa1
#20,40 8-17 * * 1-5 /usr/lib/sa/sa1
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

Instead of having to manually gather system performance information, you can set up automatic data collection by following the steps listed in the slide.

Notes for step 1: This command writes a special record that marks the time when the counters are reset to zero (boot time).

Notes for step 2: You do not edit a `crontab` file directly. Instead, you use the `crontab -e` command to make changes to an existing `crontab` file.

Notes for step 3: By uncommenting this entry, the `sa2` script will run every day Monday through Friday at 6:05 PM. The monitoring start time is 8 AM and ends at 6:01 PM. The performance data interval is every 1200 seconds (every 20 minutes). The `-A` option at the end of the entry means that the script will report overall system performance. The data files are placed in the `/var/adm/sa` directory. Each file is named `sadd`, where `dd` is the current date.

Note: For other ways to set up automatic data collection, see "Collecting System Activity Data Automatically (`sar`)" in *Oracle Solaris Administration: Common Tasks*.

System Monitoring Commands: Summary

Commands	Description
`vmstat n`	Displays virtual memory statistics
`vmstat -s`	Displays system event information
`vmstat -S`	Displays swapping statistics
`iostat n`	Displays general disk usage information
`iostat -xtc`	Displays extended disk statistics
`df -h`	Displays disk space information
`sar -a`	Checks file access operation statistics
`sar -b`	Checks buffer activity
`sar -c`	Checks system call statistics
`sar -d`	Checks disk activity
`sar -r`	Checks unused memory

ORACLE

The table displayed in the slide contains a list of the system monitoring commands you covered in this topic.

Practice 11-2 Overview:
Evaluating System Performance Levels

This practice covers the following topics:

- Displaying virtual memory statistics (`vmstat`)
- Displaying disk usage information
- Monitoring system activities
- Collecting system activity data automatically (`sar`)
- Setting up automatic data collection (`sar`)

This practice should take you about 30 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Implement a plan to evaluate resource allocation and system performance
- Configure system resources
- Monitor system performance

ORACLE

12

Monitoring and Troubleshooting Software Failures

ORACLE
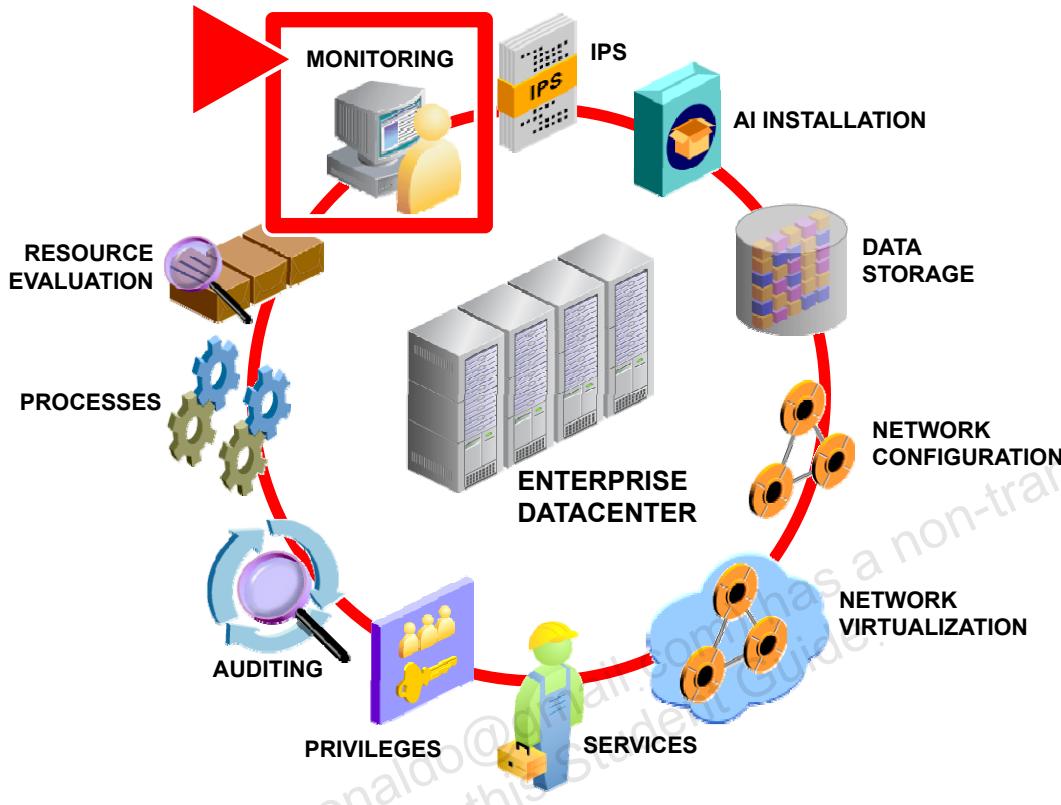
Objectives

After completing this lesson, you should be able to:

- Implement a plan for system messaging and diagnostic facilities implementation
- Configure the following:
    - System messaging
    - System crash facilities
    - Dump facilities for business application failure

Workflow Orientation

Before you start the lesson, orient yourself in the job workflow. You have reached the end of the workflow. You have successfully performed all major administrative tasks: installation, software updates, data storage management, network, zones, and services configuration. You have also put system security controls in place with role-based access control (RBAC) and Oracle Solaris auditing. You have ensured that the system resources are being used appropriately with the resource controls that you have set up for the processes running on the system. In this last lesson, you configure the facilities that you will need to monitor and capture issues with the software.

Lesson Agenda

- **Planning System Messaging and Diagnostic Facilities Implementation**
- Configuring System Messaging
- Configuring System Crash Facilities
- Configuring Dump Facilities for Business Application Failure

Planning System Messaging and Diagnostic Facilities Implementation

The system messaging and diagnostic facilities implementation plan ensures that:

- Controls are in place to monitor system activity so that issues can be addressed quickly and efficiently

- System crashes and core dumps are captured and reported so that major problems can be analyzed and corrected

Knowing what issues the operating system is encountering and what actions to take to correct those issues is an important part of your role as a system administrator. Recognizing this, your company has developed a plan that identifies the system monitoring and diagnostic tools that they want in place to quickly and efficiently identify and resolve issues that might occur within the Oracle Solaris operating system. The plan includes time for you to be trained on how to configure and use these tools. In addition to system logging, your company wants you to set up crash and core dump files so that any major issues with the operating system or with any processes or applications can be captured and sent to a support engineer for analyses and resolutions.

In this section, you are introduced to system messaging and crash and core dump file configuration.

Configuring the `/etc/syslog.conf` File

You configure this file to:

- Define target locations for the `syslog` message files
- Use a selector level of `err` to indicate that all events of priority error (and higher) are logged to the target defined in the action field

```
*.err;kern.notice;auth.notice                /dev/sysmsg
*.err;kern.debug;daemon.notice;mail.crit     /var/adm/messages
*.alert;kern.err;daemon.err                   operator
*.alert                                       root
usr.emerg                                     *
```

Note: Whenever you make changes to this file, you must restart the `syslogd` daemon.

The first thing that you need to do to set up system messaging is to identify target locations for the `syslog` message files. The target locations are defined in the `/etc/syslog.conf` file.

Note: A configuration entry in the `/etc/syslog.conf` file consists of two tab-separated fields: selector and action. The selector field has two components: a *facility* and a level written as *facility.level*. Facilities represent categories of system processes that can generate messages. Levels represent the severity or importance of the message. The action field determines where to send the message. This is the target location.

Within the `/etc/syslog.conf` file, you use a selector level of `err` to indicate that all events of priority error (and higher) are logged to the target defined in the action field.

In the example in the slide, partial contents of the `/etc/syslog.conf` file are displayed. In the first line, every error event (`*.err`) and all kernel and authorization facility events of level notice, which are not error conditions but might require special handling, will write a message to the `/dev/sysmsg` file.

In the second line, every error event (`*.err`), all kernel facility events of level debug, all daemon facility events of level notice, and all critical level mail events will record a message in the `/var/adm/messages` file. Therefore, errors are logged to both files.

The third line indicates that all alert level events, including the kernel error level and daemon error level events, are sent to the user operator if this user is logged in.

The fourth line indicates that all alert level events are sent to the `root` user if the `root` user is logged in.

The fifth line, which is taken from the "log messages to be logged locally" section of the `/etc/syslog.conf` file, indicates that any event that the system interprets as an emergency will be logged to the terminal of every logged-in user.

Note: You will have the opportunity to examine the `/etc/syslog.conf` file in full during the practice on setting up system messaging.

To alter the event logging mechanism, edit the `/etc/syslog.conf` file and restart the `syslogd` daemon.

Note: You must restart the `syslogd` daemon whenever you make any changes to the `/etc/syslog.conf` file.

Stopping and Starting the `syslogd` Daemon

- The `syslogd` daemon can be started:
  - Automatically during boot
  - Manually from the command line
- Each time the `syslogd` daemon starts, the `/etc/syslog.conf` configuration file is read.
- After you have modified the configuration file, you can:
  - Manually stop or start the `syslogd` daemon
  - Send the `syslogd` daemon a `refresh` command

```
# svcadm disable svc:/system/system-log:default
# svcadm enable svc:/system/system-log:default
# svcadm refresh svc:/system/system-log:default
```

The `syslogd` daemon can be started automatically during boot or it can be manually started from the command line. During each system boot, the `/lib/svc/method/system-log` file starts the `syslogd` process. The `/etc/syslog.conf` configuration file is read each time the `syslogd` daemon starts.

If you have modified the configuration file, you can manually stop or start the `syslogd` daemon, or send it a `refresh` command, which causes the daemon to reread the `/etc/syslog.conf` file. The example in the slide shows the commands for stopping, starting, and refreshing the `syslogd` daemon.

Note: Oracle Solaris 11.1 includes an enhanced version of the `syslog` daemon called `rsyslog` for message logging. The `rsyslog` daemon provides enhanced features, such as failover log destinations, high precision timestamps, queued operations, and filter any message part. These advanced features of `rsyslog` makes it suitable for enterprise-class, encryption-protected applications, while being easy to set up and use. By default, the `rsyslog` daemon is not enabled. Administrators can switch to this new logging daemon by disabling `svc:/system/system-log:default` and enabling `svc:/system/system-log:rsyslog` using SMF administrative utilities.

TCP Tracing

- You can use the `inetadm` command to enable the trace option on one or more services.
- The `inetadm` command uses the `syslog` command to record and log the following:
  - Client's IP address
  - TCP port number
  - Name of the service
- You can configure `/etc/syslog.conf` so that the `syslogd` daemon selectively distributes messages sent to it from the `inetd` daemon.

```
# grep daemon.notice /etc/syslog.conf
*.err;kern.debug;daemon.notice;mail.crit /var/adm/messages
```

A part of setting up system messaging includes enabling TCP tracing. Use the `inetadm` command to modify the settings of a service to enable the trace option. When you enable TCP tracing, the `inetd` daemon uses the `syslog` command to record incoming network connection requests made by using TCP. The client's IP address, TCP port number, and the name of the service are logged.

You can enable tracing on all services or on each service separately.

Note: The change is immediately recognized. There is no requirement to restart any daemon process.

By default, the `/etc/syslog.conf` file is configured such that the `syslogd` daemon selectively distributes the messages that are sent to it from the `inetd` daemon to the `/var/adm/messages` file. This message distribution is achieved through the `daemon.notice` entry in the `/etc/syslog.conf` file.

In the example in the slide, all daemon messages of level notice or higher are sent to the `/var/adm/messages` file.

Note: The `/var/adm/messages` file must exist. If it does not exist, create it, and then stop and start the `syslogd` daemon; otherwise, messages will not be written to the file.

TCP Tracing: Example

```
# inetadm -m telnet tcp_trace=TRUE
# inetadm -l telnet
SCOPE       NAME=VALUE
            name="telnet"
            endpoint_type="stream"
            proto="tcp6"
            isrpc=FALSE
            wait=FALSE
            exec="/usr/sbin/in.telnetd"
            user="root"
default     bind_addr=""
default     bind_fail_max=-1
default     bind_fail_interval=-1
default     max_con_rate=-1
default     max_copies=-1
default     con_rate_offline=-1
default     failrate_cnt=40
default     failrate_interval=60
default     inherit_env=TRUE
default     tcp_trace=TRUE
default     tcp_wrappers=FALSE
```

ORACLE

In this example, you are enabling TCP tracing on telnet sessions. You then verify that the tracing option is enabled, and it is.

Note: The -m option changes the values of the specified properties of the identified service instances.

If you want to enable TCP tracing on all services, use the following command:

```
# inetadm -M tcp_trace=TRUE
```

Note: The -M option changes the value of the specified inetd default property or properties.

Logger Command

- This command enables you to send messages to the `syslogd` daemon.
- You can write administrative shell scripts that report the status of backups or other functions.

```
logger [ -i ] [ -f file ] [ -p priority ] [ -t tag ] [ message ]
```

```
# logger System rebooted
```

```
# logger -p user.err System rebooted
```

ORACLE

The `logger` command enables you to send messages to the `syslogd` daemon. By using the `logger` command, you can write administrative shell scripts that report the status of backups or other functions. The slide gives you the syntax for the command. The description for each option is as follows:

- `i`: Logs the process ID of the `logger` command with each line
- `-f file`: Uses the contents of the file as the message to log (*file* must exist)
- `-p priority`: Enters the message with the specified priority
- `-t tag`: Marks each line that is added to the log file with the specified tag
- *message*: Concatenates the string arguments of the message in the order specified, separated by single-space characters

You can specify message priority as a *facility.level* pair. For example, `-p local3.info` assigns a message priority of `info` *level* in the `local3` *facility*. The default priority is `user.notice`.

In the second example, the message `System rebooted` is logged to the `syslogd` daemon, by using the default priority *level* `notice` and *facility* `user`.

If the `user.notice` selector field is configured in the `/etc/syslog.conf` file, the message is logged to the file that is designated for the `user.notice` selector field. If the `user.notice` selector field is not configured in the `/etc/syslog.conf` file, you can either add the `user.notice` selector field to the `/etc/syslog.conf` file, or you can prioritize the output as shown in the third example.

Changing the priority of the message to `user.err` routes the message to the `/var/adm/messages` file as indicated in the `/etc/syslog.conf` file.

You can also specify a message priority numerically. For example, `logger -i -p 2 "crit"` creates an entry in the message log that identifies the `user.crit-facility.level` pair as follows:

```
Nov 3 09:49:34 hostname root[2838]: [ID 702911 user.crit] crit
```

`/etc/dumpadm.conf` File

```
# cat /etc/dumpadm.conf
#
# dumpadm.conf
#
# Configuration parameters for system crash dump.
# Do NOT edit this file by hand -- use dumpadm(1m) instead.
#
DUMPADM_DEVICE=/dev/zvol/dsk/rpool/dump
DUMPADM_SAVDIR=/var/crash/client1
DUMPADM_CONTENT=kernel
DUMPADM_ENABLE=no
DUMPADM_CSAVE=on
```

Now you focus on configuring the crash and core dump files. You begin with the crash dump. The `/etc/dumpadm.conf` file contains the crash dump configuration of the current system.

As you can see in the example in the slide, the default values are set as follows:

- `DUMPADM_DEVICE=/dev/zvol/dsk/rpool/dump`: The default dump device is dedicated to a ZFS volume.

  Note: You can choose an unused disk partition to use as a dedicated dump device. The traditional method is to use a swap disk partition. Whichever device you choose, be sure that the dump device is large enough to handle the dump content. A good rule of thumb is 50% of the physical memory.

- `DUMPADM_SAVDIR=/var/crash/client1`: The directory for the `savecore` files is set to `/var/crash/client1`.

- `DUMPADM_CONTENT=kernel`: The dump content is set to kernel memory pages only.

- `DUMPADM_ENABLE=no`: The `savecore` command is *not* set to run automatically on reboot. This is the default.

  Note: When it is enabled, the `savecore` utility saves a crash dump of the kernel (assuming that one was made) and writes a reboot message in the shutdown log. It is invoked by the `dumpadm` service each time the system boots.

- `DUMPADM_CSAVE=on`: The compression of the `savecore` files is enabled. Because crash dump files can be extremely large and require less file system space if they are saved in a compressed format, the default is `on`.

  Note: When you configure `savecore` to save the crash dump data in a compressed format, `savecore` saves the crash dump data in the file directory `/vmcore.N.z`, where `N` in the pathname is replaced by a number, which increments by one each time `savecore` is run in the directory. The compressed file can be uncompressed in a separate step by using the `-f dumpfile` option. You will learn how to do this later in this lesson. For the uncompressed format, `savecore` saves the crash dump data in the file directory `/vmcore.N` and the kernel's `namelist` in the `/unix.N` directory.

You should not edit the `/etc/dumpadm.conf` file directly. Any changes that you want to make to the dump configuration should be made by using the `dumpadm` command. You learn how to modify the dump configuration later in this lesson.

`/etc/coreadm.conf` File

```
# cat /etc/coreadm.conf
#
# coreadm.conf
#
# Parameters for system core file configuration.
# Do NOT edit this file by hand -- use coreadm(1) instead.
#
COREADM_GLOB_PATTERN=
COREADM_GLOB_CONTENT=default
COREADM_INIT_PATTERN=core
COREADM_INIT_CONTENT=default
COREADM_GLOB_ENABLED=no
COREADM_PROC_ENABLED=yes
COREADM_GLOB_SETID_ENABLED=no
COREADM_PROC_SETID_ENABLED=no
COREADM_GLOB_LOG_ENABLED=no
```

The `/etc/coreadm.conf` file contains the current core dump configuration.

As you can see in the example in the slide, the default values for the `/etc/coreadm.conf` file are set as follows:

- `COREADM_GLOB_PATTERN=`: Identifies the name to use for the core files placed in a global directory
- `COREADM_GLOB_CONTENT=default`: Identifies that the content of the core files has the default setting. The resultant core file contains all the process information that is pertinent to debugging.
- `COREADM_INIT_PATTERN=core`: Identifies the default name that the per-process core files must use. This name is set for the `init` process, meaning that it is inherited by all other processes on the system.
- `COREADM_INIT_CONTENT=default`: Indicates that the `init` core file content has the default content structure.
- `COREADM_GLOB_ENABLED=no`: Indicates that the global core files are disabled

- `COREADM_PROC_ENABLED=yes`: Indicates that core file generation is enabled in the current working directory of a process
- `COREADM_GLOB_SETID_ENABLED=no`: Indicates that the generation of global core files with `setuid` or `setgid` permissions is disabled
- `COREADM_PROC_SETID_ENABLED=no`: Indicates that the generation of per-process core files with `setuid` or `setgid` permissions is disabled
- `COREADM_GLOB_LOG_ENABLED=no`: Indicates whether global core dump logging is enabled

Caution: A process that has a `setuid` mode presents security issues with respect to dumping core files. The files may contain sensitive information in their address space to which the current non-privileged owner of the process should not have access. Therefore, by default, `setuid` core files are not generated because of this security issue.

You should not edit the `/etc/coreadm.conf` file directly. Any changes that you want to make to the dump configuration should be made by using the `coreadm` command. You learn how to modify the dump configuration later in this lesson.

Core File Paths

- Per-process core file path:
  - Defaults to `core`
  - Is enabled by default
  - If enabled, produces a core file when a process terminates abnormally
  - Is inherited by a new process from its parent process

    Per-process files are owned and can be viewed only by the process owner.
- Global core file path:
  - Defaults to `core`
  - Is disabled by default
  - If enabled, produces an additional core file with the same content as the per-process core file

    Global core files are owned by the superuser. Non-privileged users cannot read these files.

As you just saw in the `/etc/coreadm.conf` file, there are two configurable core file paths: per-process and global. A per-process core file path defaults to `core` and is enabled by default. If enabled, the per-process core file path causes a core file to be produced when the process terminates abnormally. The per-process path is inherited by a new process from its parent process. When generated, a per-process core file is owned by the owner of the process with read/write permissions for the owner. Only the owning user can view this file.

A global core file path also defaults to `core` but is disabled by default. If it is enabled, an additional core file with the same content as the per-process core file is produced by using the global core file path. When generated, a global core file is owned by the superuser with read/write permissions only for the superuser. Non-privileged users cannot view this file.

Implementing the System Messaging and Diagnostic Facilities Implementation Plan

Your assignment is to configure the following:

- System messaging
- Crash and core dump files

The predeployment testing effort is nearly at an end. Your final testing activities will be to configure system messaging and the crash and core dump files.

In this assignment, you learn how to complete each of these tasks.

Quiz

What is the default target destination for the following `syslog` message type?

`*.err;kern.debug;daemon.notice;mail.crit`

a. `/dev/sysmsg`

b. `/var/adm/messages`

c. `operator`

d. `root`
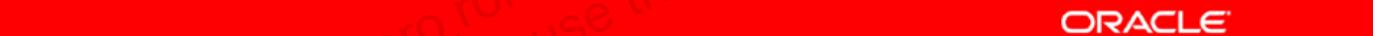
Answer: b

Quiz

You must always restart the `syslogd` daemon after you modify the `etc/syslog.conf` file.

   a. True

   b. False

ORACLE
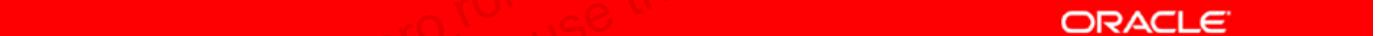
Answer: a

Quiz

Saving of the crash dump file is enabled by default.
a. True
b. False
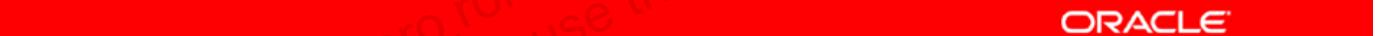
Answer: b

Quiz

You can separately enable or disable two configurable `core` file paths: per-process and global.

a. True

b. False

Answer: a

Lesson Agenda

- Planning System Messaging and Diagnostic Facilities Implementation
- **Configuring System Messaging**
- Configuring System Crash Facilities
- Configuring Dump Facilities for Business Application Failure

ORACLE

Oracle Solaris 11 Advanced System Administration   12 - 23

Configuring System Messaging

This section covers the following topics:

- Setting up message routing
- Restarting the message logging daemon
- Using TCP trace to log a message
- Monitoring message logging in real time
- Adding one-line entries to a system log file

ORACLE

Setting Up Message Routing

1. By using a text editor, edit the contents of the `/etc/syslog.conf` file to append the following entry to the end of the file:

   `local0.notice                    @hostname`

2. Restart the `syslogd` daemon to activate the new configuration.

3. On the local host, create the `/var/log/loca10.log` file.

4. Modify the `/etc/syslog.conf` file and add the entry as follows:

   `local0.notice                    /var/log/local0.log`

5. Restart the `syslogd` daemon to activate the new configuration.

To set up message routing between two hosts (for example, `host1` and `host2`), perform the steps listed in the slide. Steps 1 and 2 are performed on the first host, `host1`. The remaining steps are performed on the second host, `host2`.

Note for step 1: Following our example, the `@hostname` would be `host2`.

Setting Up Message Routing: Example

```
root@host1:~# vi /etc/syslog.conf
<content omitted>
local0.notice                @host2
root@host1:~# svcadm refresh system/system-log
root@host2:~# touch /var/log/local0.log
root@host2:~# vi /etc/syslog.conf
root@host2:~# grep local0 /etc/syslog.conf
local0.notice                /var/log/local0.log
root@host2:~# svcadm refresh system-log
```

In the example in the slide, the local0.notice entry is added to the /etc/syslog.conf file on the host system to enable users to record messages. The destination of the message is host2. After you have modified the configuration, restart the syslog daemon by using the refresh command. Next, you create a log file on host2 for the local0 log messages. You then edit the /etc/syslog.conf file configuration on host2 to include the local0.notice entry. Notice that the destination of the message is the log that you created in an earlier step. Finally, you restart the syslog daemon to activate the configuration change. Now if any message is written to this log, it will be displayed.

Logging a Message by Using TCP Trace

To enable TCP tracing, use `inetadm -m tcp_trace=TRUE`.

```
# inetadm -m tcp_trace=TRUE
# inetadm -l telnet
SCOPE       NAME=VALUE
            name="telnet"
            endpoint_type="stream"
<output omitted>
default     bind_addr=""
default     bind_fail_max=-1
default     bind_fail_interval=-1
default     max_con_rate=-1
default     max_copies=-1
default     con_rate_offline=-1
default     failrate_cnt=40
default     failrate_interval=60
default     inherit_env=TRUE
default     tcp_trace=TRUE
default     tcp_wrappers=FALSE
default     connection_backlog=10
default     tcp_keepalive=FALSE
```

ORACLE

You enable TCP tracing with the `inetadm` command, as shown in the slide. The `-M` option is used to change the values of the specified `inetd` default property. To verify that TCP tracing is enabled, use the `inetadm -p` command.

In the example in the slide, you enable TCP tracing, and then verify that it is enabled. In the example in the slide, you can see that `tcp_trace` is now set to `TRUE`.

Note: To disable TCP tracing, set `tcp_trace` to `FALSE`.

Monitoring a `syslog` File in Real Time

To view the messages sent to the `/var/adm/messages` file, use `tail -f /var/adm/messages`.

```
# tail -f /var/adm/messages
…
…
Dec 20 06:10:05 client1 inetd[655]: [ID 317013 daemon.notice]
ftp[3044] from 192.168.0.100 61017
```

ORACLE

You can monitor the designated `syslog` file in the `/var/adm` directory, in real time, by using the `tail -f /var/adm/messages` command. The `tail -f` command holds the file open so that you can view the messages that are being written to the file by the `syslogd` daemon. To exit the `/var/adm/messages` file, press Ctrl + C.

In the example in the slide, you can see that a TCP tracing-related notice message has been generated by the `syslog` daemon. The message contains the following general information:

- The date and time stamp when the message was generated (`Aug 18 06:10:05`)
- The local host name (`client1`)
- The process name and PID number for the process that was involved in the action (`inetd[655]`)
- The message ID number (`ID 317013`)
- The facility that generated the message; for example, the kernel, a system daemon, or the `syslogd` daemon (`daemon`)
- Level of severity for the message; for example, emergency, error, warning, notice, or information (`notice`)
- The problem or event (`ftp[3044] from 192.168.0.112 61017`)

Practice 12-1 Overview:
Setting Up System Messaging

This practice covers the following topics:

- Setting up message routing
- Using TCP trace to log a message

ORACLE

The practices for this lesson are designed to reinforce the concepts that have been presented in the lecture portion. These practices cover the following tasks:

- **Practice 12-1:** Setting up system messaging
- **Practice 12-2:** Configuring system and application crash facilities

Practice 12-1 should take about 30 minutes to complete.

Lesson Agenda

- Planning System Messaging and Diagnostic Facilities Implementation
- Configuring System Messaging
- **Configuring System Crash Facilities**
- Configuring Dump Facilities for Business Application Failure

Now that you know how to configure system messaging, you will next look at how to configure the system crash facilities.

Configuring System Crash Facilities

This section covers the following topics:

- Displaying the current crash dump configuration
- Modifying the crash dump configuration
- Saving the crash dump file
- Uncompressing the crash dump file
- Displaying the crash dump file contents

ORACLE

Displaying the Current Crash Dump Configuration

To display the current crash dump configuration, use `dumpadm`.

```
# dumpadm
     Dump content: kernel pages
       Dump device: /dev/zvol/dsk/rpool/dump
Savecore directory: /var/crash/client1
  Savecore enabled: no
   Save compressed: on
```

To view the current dump configuration, use the `dumpadm` command without arguments, as shown in the slide.

Note: The configuration in the slide example matches the configuration that you saw earlier in the `/etc/dumpadm.conf` file.

Modifying the Crash Dump Configuration

To modify the crash dump configuration, use
```
/usr/sbin/dumpadm [-nuy] [-c content] [-d
dump-device][-m mink | minm | min%]
[-s savecore-dir] [-r root-dir] [-z on | off].
```

```
# dumpadm -y -d /dev/dsk/c0t1d0s1
      Dump content: kernel
       Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash/client1
  Savecore enabled: yes
    Save compressed: on
```

ORACLE

As discussed in the topic on planning system messaging and diagnostic facilities implementation, if you want to modify the configuration of the crash dump file, you use the `dumpadm` command.

You can use several options with this command, as shown in the slide. The description for each option is as follows:

- `-n`: Specifies that `savecore` should not be run when the system reboots. Although this is the default setting, this dump configuration is not recommended. If system crash information is written to the swap device and `savecore` is not enabled, the crash dump information is overwritten when the system begins to swap.
- `-u`: Forcibly updates the kernel dump configuration based on the contents of the `/etc/dumpadm.conf` file. Normally, this option is used only on reboot when starting `svc:/system/dumpadm:default`, when the `dumpadm` settings from the previous boot must be restored. Your dump configuration is saved in the configuration file for this purpose.
- `-y`: Modifies the dump configuration to automatically execute the `savecore` command on reboot

- -c *content*: Specifies the type of data to dump. Use kernel to dump all kernel memory, all to dump all memory, or curproc to dump kernel memory and the memory pages of the process whose thread was executing when the crash occurred. The default dump content is kernel memory.
- -d *dumpdevice*: Specifies the device that stores the dump data temporarily when the system crashes. The primary swap device is the default dump device.
- -m *min*k | *min*m | *min*%: Specifies the minimum free disk space for saving crash dump files by creating a minfree file in the current savecore directory. This parameter can be specified in KB (nnnk), MB (nnnm), or file system size percentage (nnn%).
- -s *savecore-dir*: Specifies an alternative directory for storing crash dump files. The default savecore-dir directory is /var/crash/hostname, where host name is the output of the uname -n command.
- -r *root-dir*: Specifies an alternative root directory relative to which the dumpadm command should create files. If the -r argument is not specified, the default root directory "/" is used.
- -z on | off: Modifies the dump configuration to control the operation of the savecore command on reboot. The on setting enables the saving of a core file in a compressed format. The off setting automatically uncompresses the crash dump file.

In the example in the slide, the kernel pages are dumped to a different dump device, /dev/dsk/c0t1d0s1, which is labeled as a dedicated dump device. In addition, the dump configuration is set to automatically execute the savecore command upon reboot by using the -y option.

Saving the Crash Dump File

To save the crash dump file to the designated dump device, use `savecore -L`.

```
# savecore -L
dumping to /dev/dsk/c0t1d0s1, offset 65536, content:
kernel
 0:04 100% done
100% done: 103879 pages dumped, dump succeeded
savecore: System dump time: Tue Oct 18 10:23:31 2011

savecore: Saving compressed system crash dump in
/var/crash/client1/vmdump.0
savecore: Decompress the crash dump with
'savecore -vf /var/crash/client1/vmdump.0'
```

To save the contents of the crash dump file to the dump device that you have designated, use the `savecore -L` command. The `-L` option saves a crash dump of the live running Oracle Solaris system without actually rebooting or altering the system in any way. This option forces `savecore` to save a live snapshot of the system to the dump device, and then immediately to retrieve the data and to write it to a new set of crash dump files in the specified directory. Live system crash dumps can be performed only if you have configured your system to have a dedicated dump device by using the `dumpadm` command.

The `vmdump.0` file that you see in the example in the slide contains the recently created dump in compressed format.

Uncompressing the Crash Dump File

To uncompress the crash dump file, use `savecore -vf`
`/var/crash/hostname/vmdump.0`.

```
# savecore -vf /var/crash/client1/vmdump.0
savecore: System dump time: Tue Dec 20 10:23:31 2011

savecore: saving system crash dump in
/var/crash/client1/{unix,vmcore}.0
Constructing namelist /var/crash/lient1/unix.0
Constructing corefile /var/crash/client1/vmcore.0
 0:24 100% done: 103879 of 103879 pages saved
2266 (2%) zero pages were not written
0:24 dump decompress is done
```

After you have saved the contents of the crash dump file to the dump device, you can uncompress the `vmdump.0` file by using the `savecore -vf` command, as shown in the slide.

In the example in the slide, notice that this command (specifically the `-f` option) uncompresses the file to `vmcore.0`.

Displaying the Crash Dump File Contents

To display the contents of the crash dump file, perform the following steps:

1. Change directories to the `/var/crash` directory.

2. List the files in the crash directory.

3. Use the file command to access the crash dump file, either `vmcore.0` or `vmdump.0`.

4. View the contents of the file by using the string command.

To view the contents of the crash dump files, you first need to go to the `/var/crash` directory. Next, you list the files that are in the directory. You should see these files listed: `bounds`, `unix.0`, `vmcore.0`, and `vmdump.0`. To view the contents of the `vmcore.0` and `vmdump.0` files, use the `file` command, and then the `string` command.

From this point, you send the crash dump files to an Oracle Solaris support engineer for analysis to determine what caused the system to crash.

Displaying the Crash Dump
File Contents: Example

```
# cd /var/crash/client1
root@client1:/var/crash/client1# ls
bounds unix.0   vmcore.0   vmdump.0
root@client1:/var/crash/client1# file vmcore.0
vmcore.0:  SunOS 5.11 11.0 64-bit Intel live dump from 'client1'
root@client1:/var/crash/client# strings vmcore.0 | more
SunOS
s11-desktop
5.11
11.0
i86pc
i86pc
aefffed4-f452-6dbc-f11e-cdb35c1bc0a2
.symtab
.strtab
.shstrtab
_END_
_START_
__return_from_main
…
…
```

In this example, the contents of the vmcore.0 file is displayed. The contents represent the processes that are running in memory at the time the system crash occurred.

To display the vmdump.0 file, you use the same set of commands.

When you view the contents of the vmdump.0 file and compare it to the vmcore.0 file, you find that the contents of the two files are the same.

Lesson Agenda

- Planning System Messaging and Diagnostic Facilities Implementation
- Configuring System Messaging
- Configuring System Crash Facilities
- **Configuring Dump Facilities for Business Application Failure**

Configuring Dump Facilities
for Business Application Failure

This section covers the following topics:

- Displaying the current core dump configuration
- Modifying the core dump configuration
- Setting a core file name pattern
- Enabling a core file path
- Displaying the contents of the core dump file

Displaying the Current Core Dump Configuration

To display the current core dump configuration, use `coreadm`.

```
# coreadm
     global core file pattern: /var/core/core.%f.%p
     global core file content: default
       init core file pattern: core
       init core file content: default
            global core dumps: disabled
       per-process core dumps: enabled
      global setid core dumps: disabled
 per-process setid core dumps: disabled
      global core dump logging: disabled
```

To view the current core dump configuration, use the `coreadm` command without arguments, as shown in the slide.

Note: The configuration in the slide example matches the configuration that you saw earlier in the `/etc/coreadm.conf` file.

Modifying the Core Dump Configuration

To modify the core dump configuration, use `coreadm [-g pattern] [-i pattern] [-d option ... ] [-e option ... ].`

```
# coreadm -e log
# coreadm
     global core file pattern: /var/core/core.%f.%p
     global core file content: default
       init core file pattern: core
       init core file content: default
             global core dumps: enabled
        per-process core dumps: enabled
      global setid core dumps: disabled
 per-process setid core dumps: disabled
      global core dump logging: enabled
```

As discussed in the first topic on planning system messaging and diagnostic facilities implementation, if you want to modify the configuration of the core dump file, you use the `coreadm` command, as shown in the slide.

The `coreadm` command enables you to control the behavior of core file generation. For example, you can use the `coreadm` command to configure a system such that all process core files are placed in a single system directory. The flexibility of this configuration makes it easier to track problems by examining the core files in a specific directory whenever a process or daemon terminates abnormally.

This flexibility also makes it easy to locate and remove the core files on a system.

In the example in the slide, assume that you have already configured and enabled the global core file path and now you want to enable global logging. This will generate a message when the system creates a global core file. To enable global logging, use the `coreadm -e` command followed by the `log` core file option. You then verify the change by displaying the current core dump configuration.

Note: You can view the dump creation messages in `/var/adm/messages`.

You can use several options with the `coreadm` command. Descriptions for some of the options are as follows. For a full list of options, see the `coreadm` man page.

- `-g pattern`: Sets the global core file name pattern to `pattern`. The pattern must start with a slash (/), and can contain any of the special embedded variables.

  Note: For a list of possible embedded variables for the global core file content, see the `coreadm` man pages.

- `-i pattern`: Sets the per-process core file name pattern from `init` to `pattern`

  Note: For a list of pattern options, see the `coreadm` man pages. This option is the same as the `coreadm –p pattern 1` command that is described in the following list, except that the setting is persistent after a reboot.

- `-d option`: Disables the specified core file option. See the `-e` option for descriptions of possible options. You can specify multiple `-e` and `-d` options on the command line.

- `-e option`: Enables the specified core file option, where option can be any one of the following:

  - `global`: Enables core dumps by using the global core pattern
  - `process`: Enables core dumps by using the per-process core pattern
  - `global-setid`: Enables `setid` core dumps by using the global core pattern
  - `proc-setid`: Enables `setid` core dumps by using the per-process core pattern
  - `log`: Generates a `syslog` (3) message when a user attempts to generate a global core file

- `-u`: Updates system-wide core file options from the contents of the configuration file `/etc/coreadm.conf`. If the configuration file is missing or contains invalid values, default values are substituted. Following the update, the configuration file is resynchronized with the system core file configuration.

- `-p pattern`: Sets the per-process core file name pattern to `pattern` for each of the specified process IDs (PIDs). The pattern can contain any of the special embedded variables and does not have to begin with a slash (/). If `pattern` does not begin with "/," it is evaluated relative to the current directory that is in effect when the process generates a core file.

- `-G content`: Sets the global core file content. You can specify content by using pattern options.

A core file name pattern is a file system path name with embedded variables. The embedded variables are specified with a leading percent (%) character. The operating system expands these variables from the values that are in effect when the OS generates a core file.

Note: Only the `root` user can run the following `coreadm` command options to configure system-wide core file options: `coreadm [-g pattern] [-i pattern] [-d option ... ] [-e option ... ]`. Users can run only the `coreadm` command with the `-p` option to specify the file name pattern for the operating system to use when generating a per-process core file.

Setting a Core File Name Pattern

To set a per-process file name pattern, use `coreadm -p $HOME/corefiles/%f.%p $$`.

```
$ coreadm -p $HOME/corefiles/%f.%p $$
```

To set a global file name pattern, use `coreadm -g /var/core/%f.%p`.

```
# coreadm -g /var/core/%f.%p
```

After you determine whether you want to set a per-process or global core file, you can set the core file name pattern. You can set a core file name pattern on a global, zone, or per-process basis. In addition, you can set the per-process defaults that persist across a system reboot.

To set a per-process file name pattern, use the `coreadm -p` command followed by `$HOME/corefiles/%f.%p $$`.

Note: This command sets up a per-process core dump that will save core dumps in the `$HOME/core` directory by the name of the file or program being executed (`%f`) and the process ID (`%p`). The `$$` symbols represent a placeholder for the process ID of the currently running shell. The per-process core file name pattern is inherited by all child processes.

To set a global file name pattern, use the `coreadm -g` command followed by `/var/core/%f.%p`.

After you have set a per-process or global core file name pattern, you must enable it.

Enabling a Core File Path

- To enable the per-process core file path, use
  `coreadm -e process`.
- To enable the global core file path, use
  `coreadm -e global -g /var/core/core.%f.%p`.
- To verify the configuration, use `coreadm`.

```
# coreadm
     global core file pattern: /var/core/core.%f.%p
     global core file content: default
       init core file pattern: core
       init core file content: default
             global core dumps: enabled
        per-process core dumps: enabled
     global setid core dumps: disabled
 per-process setid core dumps: disabled
     global core dump logging: enabled
```

To verify either configuration change, you use the `coreadm` command to display the current core dump configuration. In the example in the slide, you can see that both core dump files are enabled.

Note: When a process terminates abnormally, it produces a core file in the current directory by default. If the global core file path is enabled, each abnormally terminating process might produce two files: one in the current working directory and one in the global core file location.

Displaying the Contents of
the Core Dump File

To display the contents of the core dump file:

1. Change directories to the `/var/core` directory.

2. List the files in the core directory.

3. Use the `file` command to access the core file.

4. View the contents of the file by using the `string` command.

To view the contents of a core dump file, you first need to go to the `/var/core` directory. Next, you list the files that are in the directory. To view the contents of a file, use the `file` command, and then the `string` command, just as you did to view the contents of the crash dump file.

From this point, you send the core dump files to an Oracle Solaris support engineer for analysis to determine what caused the system to crash.

Displaying the Core Dump File
Contents: Example

```
# cd /var/core
root@client1:/var/core# ls /var/core
core.bash.3811
root@client1:/var/core# file core*
core.bash.3811:ELF 32-bit LSB core file 80386 Version 1, from 'bash'
root@client1:/var/core# strings core.bash.3811 | more
CORE
pMND-
bash
-bash
CORE
i86pc
CORE
CORE
CORE
CORE
pMND-
bash
-bash
…
…
```

ORACLE

In the example in the slide, the contents of a core dump file for a damaged bash process are displayed.

Practice 12-2 Overview: Configuring System and Application Crash Facilities

This practice covers the configuration of:

- System crash facilities
- Dump facilities for business application failure

This practice should take about 30 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Implement a plan for system messaging and diagnostic facilities implementation
- Configure the following:
  - System messaging
  - System crash facilities
  - Dump facilities for business application failure

In this lesson, you were introduced to system logs and learned how to monitor system messages. You also learned how to configure the system to generate crash and core dump files.