# Oracle Solaris 11 System Administration

## Student Guide - Volume II

D72896GC21

Edition 2.1

August 2012

D78843

ORACLE

## Authors

Anies Rahman

Tammy Shannon

## Technical Contributors and Reviewers

Mike Carew

Sreedhar Chalamalasetti

Susan Chang

Mary Ding

Alta Elstad

Al Flournoy

Glynn Foster

Mike Gerdts

Dave Giroux

Tetsuya Harada

Kristi Herd

Darren Kenny

David Laudon

Rosemary Martinak

Dave Maxwell

Dermot McCluskey

Kristi McNeill

Ronan O'Connor

John Powell

Brock Pytlik

Eric Siglin

Enzo Silva

Sue Sohn

Karen Tung

Sean Wilcox

Albert White

Oracle Solaris Documentation Team

## Editor

Rashmi Rajagopal

## Publishers

Michael Sebastian

Jayanthy Keshavamurthy

**Disclaimer**

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

**Restricted Rights Notice**

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS
The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

**Trademark Notice**

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

# Contents

6

# Administering Oracle Solaris Zones

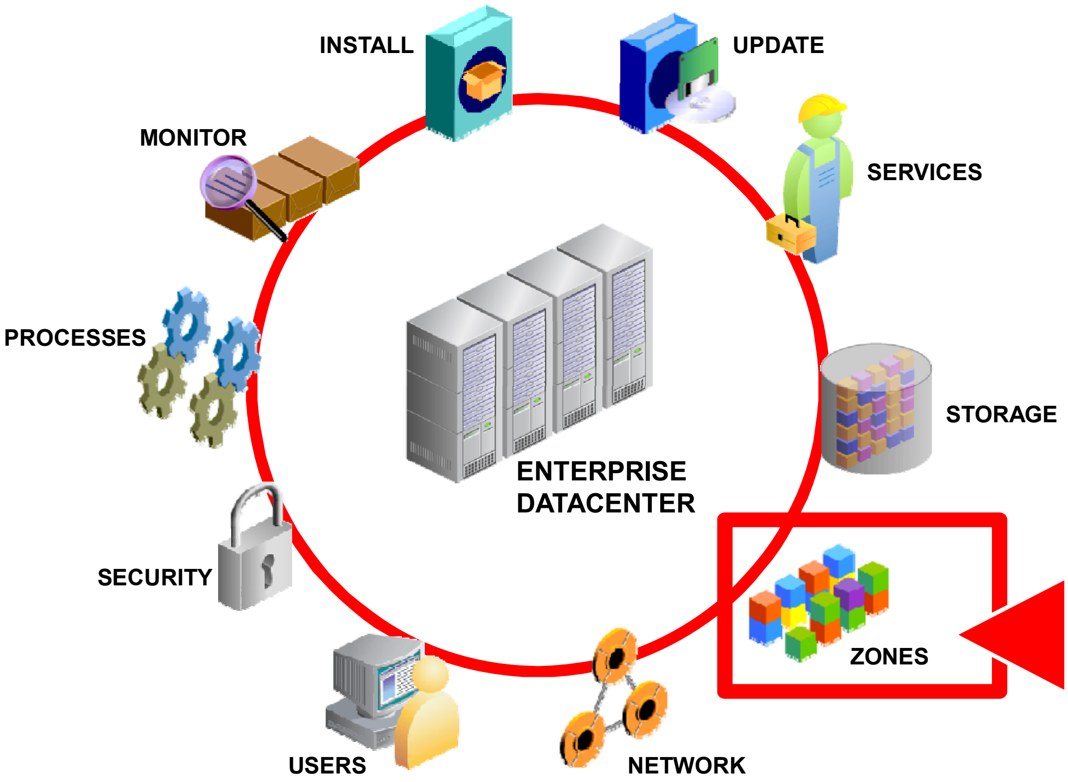
# Objectives

After completing this lesson, you should be able to:

- Implement a plan for Oracle Solaris zone management
- Determine the current zone configuration on the system
- Determine the current zone resource utilization on the system
- Administer an Oracle Solaris zone

Oracle Solaris 11 supports zone technology. In this lesson, you are introduced to the zone technology and you learn how to determine the current zone configuration on your system. You also learn how to determine the resource utilization for each zone. Finally, you learn how to perform basic zone administration on an Oracle Solaris zone.

# Workflow Orientation



Before you begin the lesson, take a moment to orient yourself to where you are in the job workflow. You have successfully installed the operating system, updated it, tested the SMF services as well as the system's boot and shutdown functionality, and set up and administered the data storage environment. Now you are going to be introduced to the world of virtualization and zones. In today's enterprise datacenter, virtualization is a core technology because it offers cost and labor-saving advantages. Because Oracle Solaris 11 supports virtualization, your company is interested in utilizing and benefiting from the technology.

# Lesson Agenda

- Planning for Oracle Solaris Zones
- Determining an Oracle Solaris Zone Configuration
- Administering an Oracle Solaris Zone

# Planning for Oracle Solaris Zones

Oracle Solaris zones planning is required to ensure that:

- Server consolidation is well thought out and executed properly
- Zones configuration supports the needs of the business
- Resources within the zones are allocated properly

For several years now, your company has been concerned about the increased cost and complexity of managing numerous systems, and the company is excited about the opportunity to consolidate many of its applications onto fewer, more scalable servers by using the Oracle Solaris zone technology. The Oracle Solaris 11 implementation team has done its homework and fully understands the benefits of using zones. For example, zones enable more efficient resource utilization on a system. Dynamic resource reallocation permits unused resources to be shifted to other zones as needed. Fault and security isolation mean that poorly behaved applications do not require a dedicated and under-utilized system. With the use of zones, these applications can be consolidated with other applications.

By putting together a very extensive plan for the Oracle Solaris zones implementation, your company hopes to ensure that the server consolidation effort is well thought out and executed properly, and that the cost-saving returns are even greater than anticipated. The plan includes how many zones the company wants to configure based on the business application needs, as well as how to allocate the system resources to each zone. As part of the zones implementation, your company is also very interested in exploring the virtual network capabilities that are supported by Oracle Solaris 11.

In this topic, you are introduced to zone technology and shown how zones are configured.

# Oracle Solaris Zone Technology: Overview

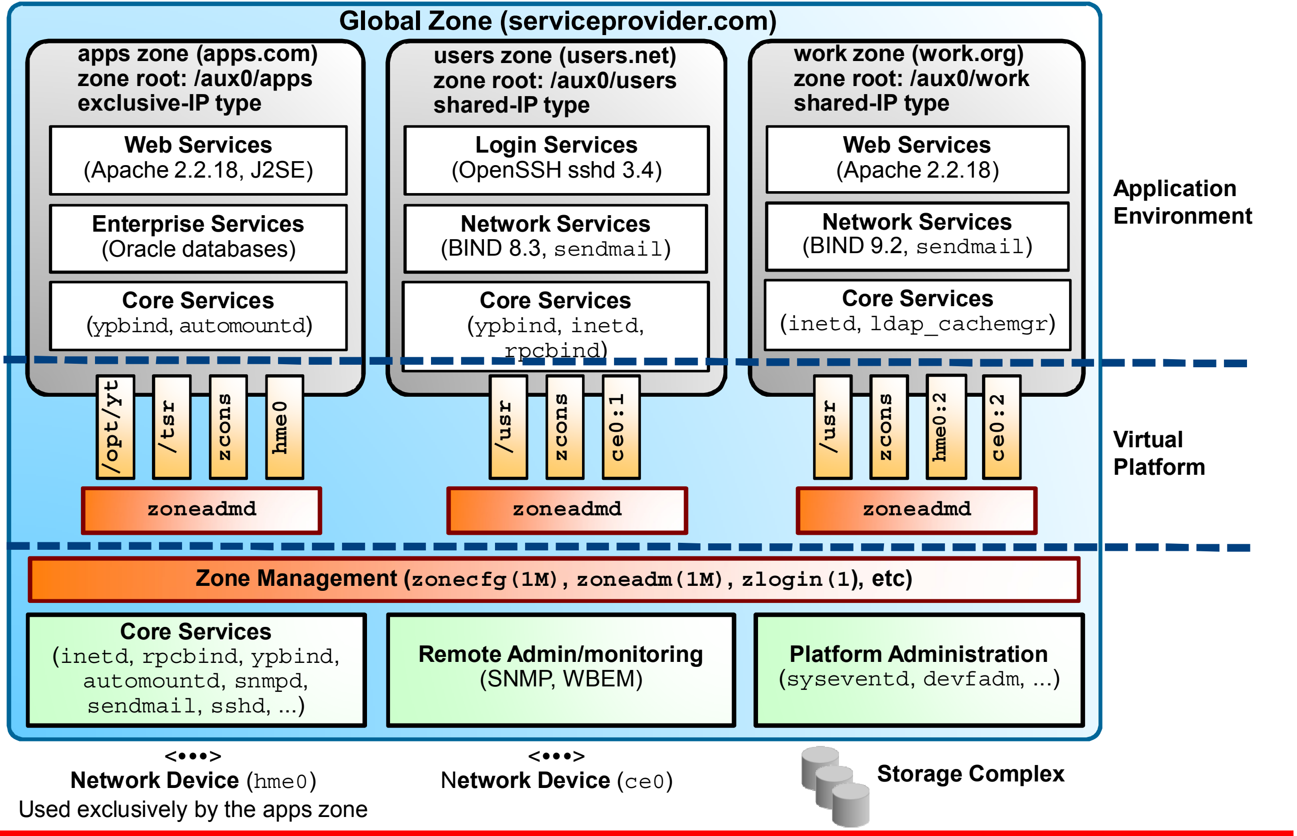
Zones:

- Provide an isolated and secure environment for running applications
- Are virtualized operating system environments, each created within a single instance of the OS
- Are isolated from each other and the rest of the system
- Enable a one-application-per-server deployment model to be maintained while simultaneously sharing hardware resources

Oracle Solaris zones provide an isolated and secure environment for running applications. Zones are virtualized operating system environments, each created within a single instance of the Oracle Solaris operating system.

When you create a zone, you produce an application execution environment in which processes are isolated from the rest of the system. This isolation prevents processes that are running in one zone from monitoring or affecting processes that are running in other zones. Even a process running with superuser credentials cannot view or affect activity in other zones. With Oracle Solaris zones, you can maintain the one-application-per-server deployment model while simultaneously sharing hardware resources.

# Zones Server Consolidation: Example



Global Zone (serviceprovider.com)

apps zone (apps.com)
zone root: /aux0/apps
exclusive-IP type

Web Services
(Apache 2.2.18, J2SE)

Enterprise Services
(Oracle databases)

Core Services
(ypbind, automountd)

users zone (users.net)
zone root: /aux0/users
shared-IP type

Login Services
(OpenSSH sshd 3.4)

Network Services
(BIND 8.3, sendmail)

Core Services
(ypbind, inetd, rpcbind)

work zone (work.org)
zone root: /aux0/work
shared-IP type

Web Services
(Apache 2.2.18)

Network Services
(BIND 9.2, sendmail)

Core Services
(inetd, ldap_cachemgr)

Application Environment

Virtual Platform

**Zone Management** (zonecfg(1M), zoneadm(1M), zlogin(1), etc)

Core Services
(inetd, rpcbind, ypbind, automountd, snmpd, sendmail, sshd, ...)

Remote Admin/monitoring
(SNMP, WBEM)

Platform Administration
(syseventd, devfadm, ...)

<•••>
Network Device (hme0)
Used exclusively by the apps zone

<•••>
Network Device (ce0)

Storage Complex

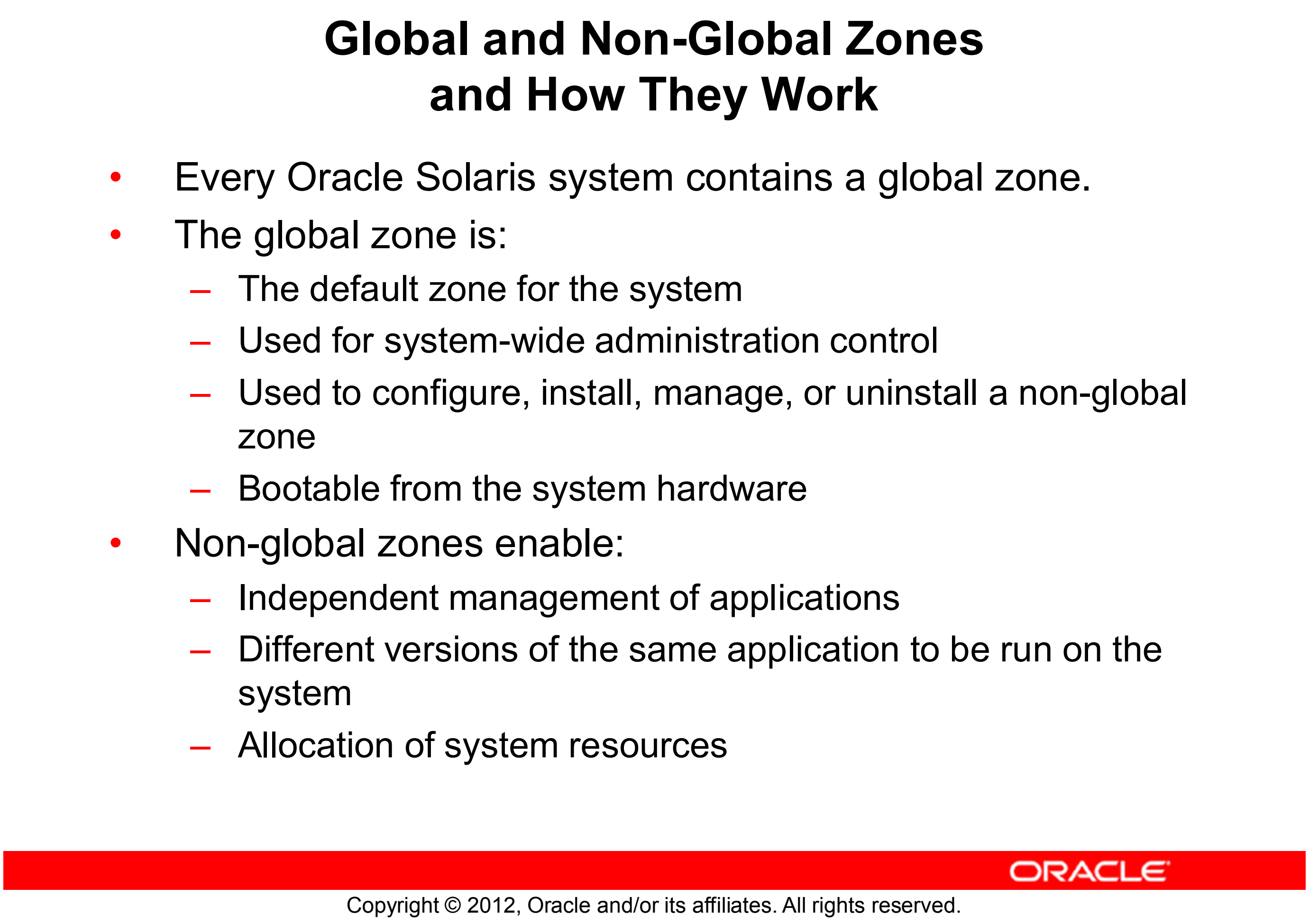
The graphic shows a system with four zones. Each of the zone applications, users, and work is running a workload unrelated to the workloads of the other zones, in a sample consolidated environment. This example illustrates that different versions of the same application can be run without negative consequences in different zones, to match the consolidation requirements. Each zone can provide a customized set of services.

# Oracle Solaris Zones: Requirements and Restrictions

- Zones can be used on any machine that is running Oracle Solaris 10 or later.
- The number of zones is determined by the:
  - Total resource requirements of the application software running in all the zones
  - Size of the system

Zones can be used on any machine that is running Oracle Solaris 10 or later. The number of zones that can be effectively hosted on a single system is determined by the total resource requirements of the application software running in all the zones, and the size of the system.

All the required system software and any additional packages are installed into the private file systems of the zone.



# Global and Non-Global Zones and How They Work

- Every Oracle Solaris system contains a global zone.
- The global zone is:
  - The default zone for the system
  - Used for system-wide administration control
  - Used to configure, install, manage, or uninstall a non-global zone
  - Bootable from the system hardware
- Non-global zones enable:
  - Independent management of applications
  - Different versions of the same application to be run on the system
  - Allocation of system resources

Every Oracle Solaris system contains a global zone. The global zone has a dual function. The global zone is both the default zone for the system and the zone used for systemwide administrative control. All applications run in the global zone if no non-global zones (referred to simply as zones) are created. The global zone is the only zone from which a non-global zone can be configured, installed, managed, or uninstalled. Only the global zone is bootable from the system hardware.

A non-global zone can be thought of as a box. One or more applications can run in this box without interacting with the rest of the system. Applications that are running in the same instance of the Oracle Solaris operating system can then be managed independently of one another. Thus, different versions of the same application can be run in different zones, to match the requirements of your configuration.

A non-global zone provides isolation at almost any level of granularity that you require. A zone does not need a dedicated CPU, a physical device, or a portion of physical memory. These resources can either be multiplexed across a number of zones running within a system, or allocated on a per-zone basis using the resource management features available in the operating system.

# Branded Zones

Branded zones:

- Provide an extension of Oracle Solaris zones
- Contain operating environments different from that of the global zone
- Run applications
- Use a brand (for example, `solaris10` brand) to:
  - Define the operating environment that can be installed in the zone
  - Determine how the system will behave within the zone
  - Identify the correct application type at application launch time
- Use extensions to the standard zone structure to perform branded zone management

By default, a non-global zone on a system runs the same operating system software as the global zone. The branded zone (BrandZ) facility in the Oracle Solaris operating system is a simple extension of Oracle Solaris zones. The BrandZ framework is used to create non-global branded zones that contain operating environments that are different from that of the global zone. Branded zones are used on the Oracle Solaris operating system to run applications. For example using the branded zone facility, you can run Oracle Solaris 10 applications by using Oracle Solaris 10 zones (`solaris10` brand) on a system running Oracle Solaris 11.

The brand defines the operating environment that can be installed in the zone, and determines how the system will behave within the zone so that the software installed in the zone functions correctly. In addition, a zone’s brand is used to identify the correct application type at application launch time. All branded zone management is performed through extensions to the standard zones structure. Most administration procedures are identical for all zones.

For more information about branded zones, see *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management*.

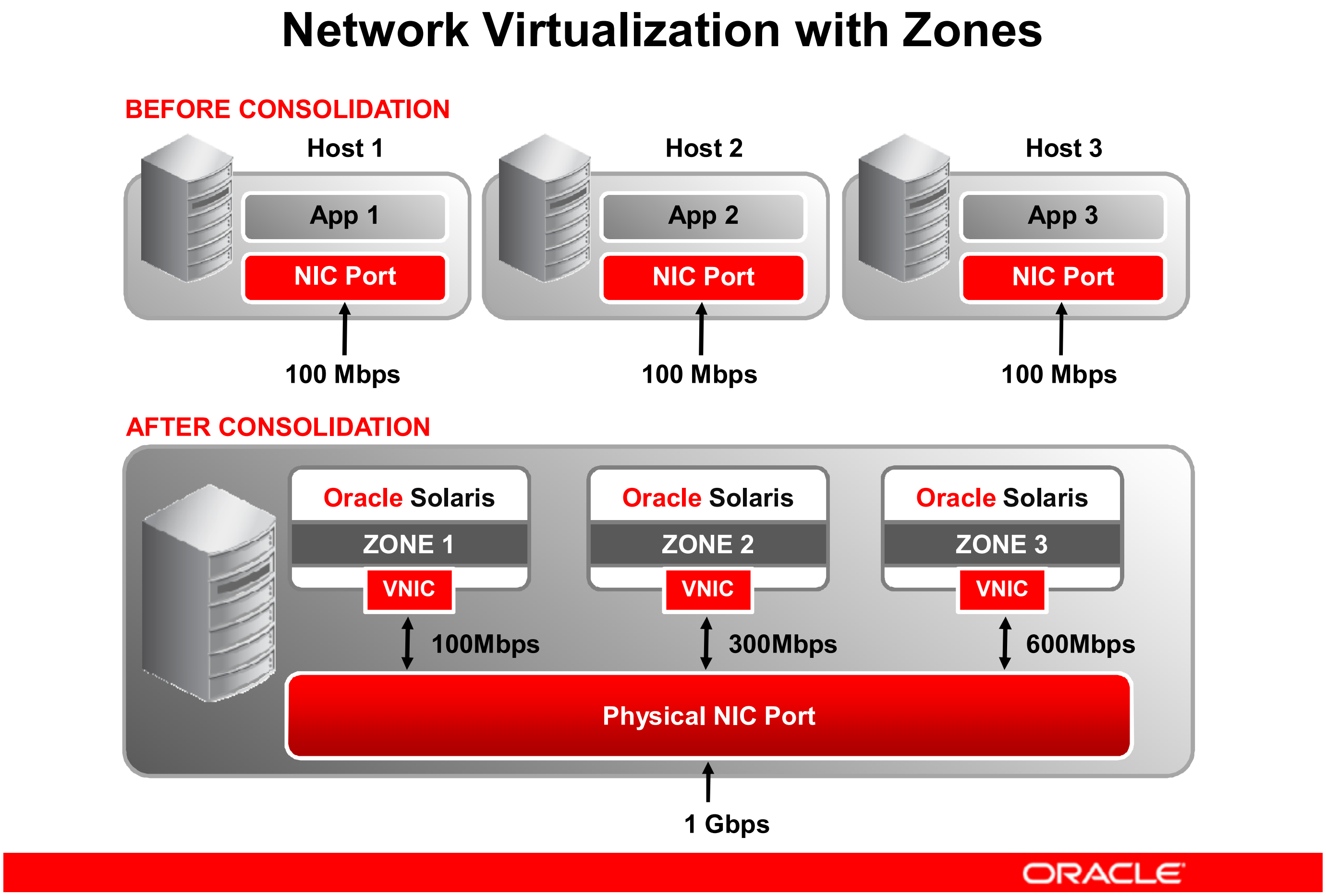
## Zone IP Network Connectivity

- Zones communicate through IP network interfaces.
- The system administrator configures zone network interfaces during zone configuration.
- When a zone is booted, the network interfaces are set up and placed in the zone.
- Two IP types are available for non-global zones:
  - Shared-IP: A network interface is shared with the global zone
  - Exclusive-IP: A network interface is dedicated to the non-global zone

### Zone Network Interfaces

Basic communication between zones is accomplished by giving each zone Internet Protocol (IP) network connectivity. An application running in one zone cannot observe the network traffic of another zone. This isolation is maintained even though the respective streams of packets travel through the same physical interface.

During zone configuration, the system administrator configures the zone network interfaces to provide network connectivity. These network interfaces are set up and placed in the zone when it is booted. Two IP types are available for non-global zones: shared-IP and exclusive-IP, which is the default. The shared-IP zones always share a network interface (or IP layer) with the global zone, and the exclusive-IP zones always have their own dedicated network interface (or instance of the IP layer). Both shared-IP zones and exclusive-IP zones can be used on the same machine.



# Network Virtualization with Zones

BEFORE CONSOLIDATION

Host 1: App 1, NIC Port, 100 Mbps

Host 2: App 2, NIC Port, 100 Mbps

Host 3: App 3, NIC Port, 100 Mbps

AFTER CONSOLIDATION

Oracle Solaris ZONE 1, VNIC, 100Mbps

Oracle Solaris ZONE 2, VNIC, 300Mbps

Oracle Solaris ZONE 3, VNIC, 600Mbps

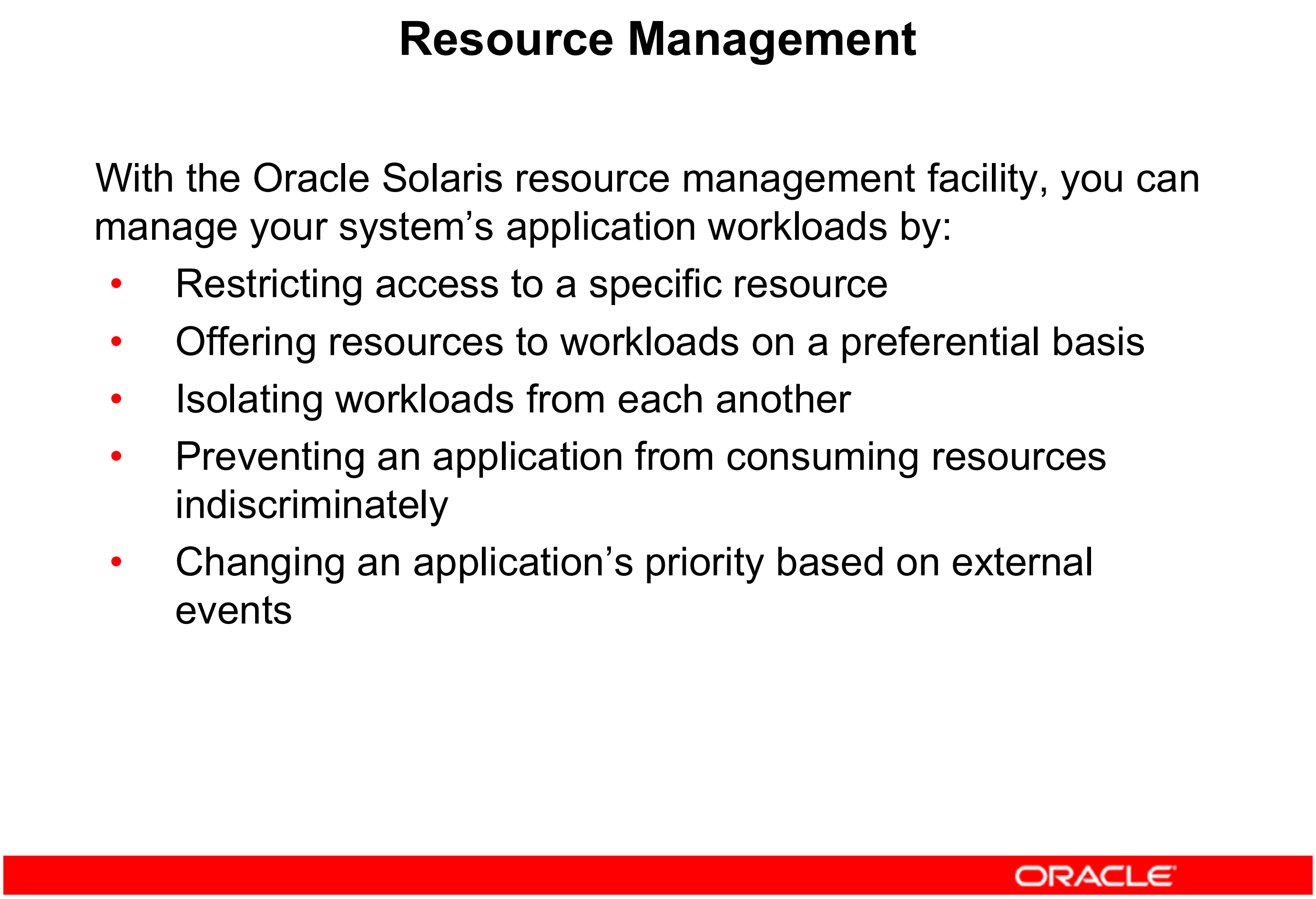
Physical NIC Port

1 Gbps

Oracle Solaris 11 also supports network virtualization, which allows a physical network interface card (NIC) to be partitioned for consolidation purposes. With network virtualization technology, you can share a physical NIC between multiple zones or virtual machines running on the same system, as illustrated by the graphic in this slide.

Virtual network interface cards (VNICs) are the fundamental building blocks of network virtualization. VNICs are created and assigned IP addresses as communication end points. VNICs are created on top of physical interfaces, or on top of etherstubs, and from the application’s point of view, VNICs appear exactly like physical interfaces.

**Note:** An etherstub is a pseudo-interface upon which a virtual network is built.



# Resource Management

With the Oracle Solaris resource management facility, you can manage your system’s application workloads by:

- Restricting access to a specific resource
- Offering resources to workloads on a preferential basis
- Isolating workloads from each another
- Preventing an application from consuming resources indiscriminately
- Changing an application’s priority based on external events

Oracle Solaris 11 provides the option to manage the varying workloads that are generated by different applications on a system. A workload is an aggregation of all processes of an application or group of applications. The ability to minimize cross-workload performance compromises, along with the facilities that monitor resource usage and utilization, is referred to as resource management. If you elect not to use the resource management features, the Oracle Solaris operating system responds to workload demands by adapting to new application requests dynamically. This default response generally means that all activity on the system is given equal access to resources. Oracle Solaris resource management features enable you to treat workloads individually. You can do the following:

- Restrict access to a specific resource
- Offer resources to workloads on a preferential basis
- Isolate workloads from each other
- Prevent an application from consuming resources indiscriminately
- Change an application’s priority based on external events

# Zone Resource Management

- Align resource management control boundaries with those of the zones.
- Zone resources that can be controlled include:
  - Resource pools or assigned CPUs
  - Resource controls
  - Scheduling classes

If you use resource management features, you should align the boundaries of the resource management controls with those of the zones. This alignment creates a more complete model of a virtual machine, where namespace access, security isolation, and resource usage are all controlled.

Resources that can be controlled in a zone include the following:

- Resource pools or assigned CPUs. These are used for partitioning machine resources. A resource pool represents an association between groups of resources that can be partitioned.
- Resource controls, which provide a mechanism for the constraint of system resources. A resource control is a per-process, per-task, per-project limit on the consumption of a resource. A project is a network-wide administrative identifier for related work.

- Scheduling classes, which enable you to control the allocation of available CPU resources among zones, based on their importance.

  The Oracle Solaris operating system uses a process scheduler to control allocation of the CPU to processes. 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 default scheduler in the Oracle Solaris operating system, the timesharing (TS) scheduler, tries to give every process relatively equal access to the available CPUs. However, you might want to specify that certain processes be given more resources than others. To do this, you can use the fair share scheduler (FSS) to control the allocation of available CPU resources among workloads, based on their importance. This importance is expressed by the number of shares of CPU resources that you assign to each zone.

How to manage zone resources is covered in the *Oracle Solaris 11 Advanced System Administration* course. In this course, you are shown how to monitor the zone resources.

# Allocating System Resources to a Zone

To allocate system resources to a zone:

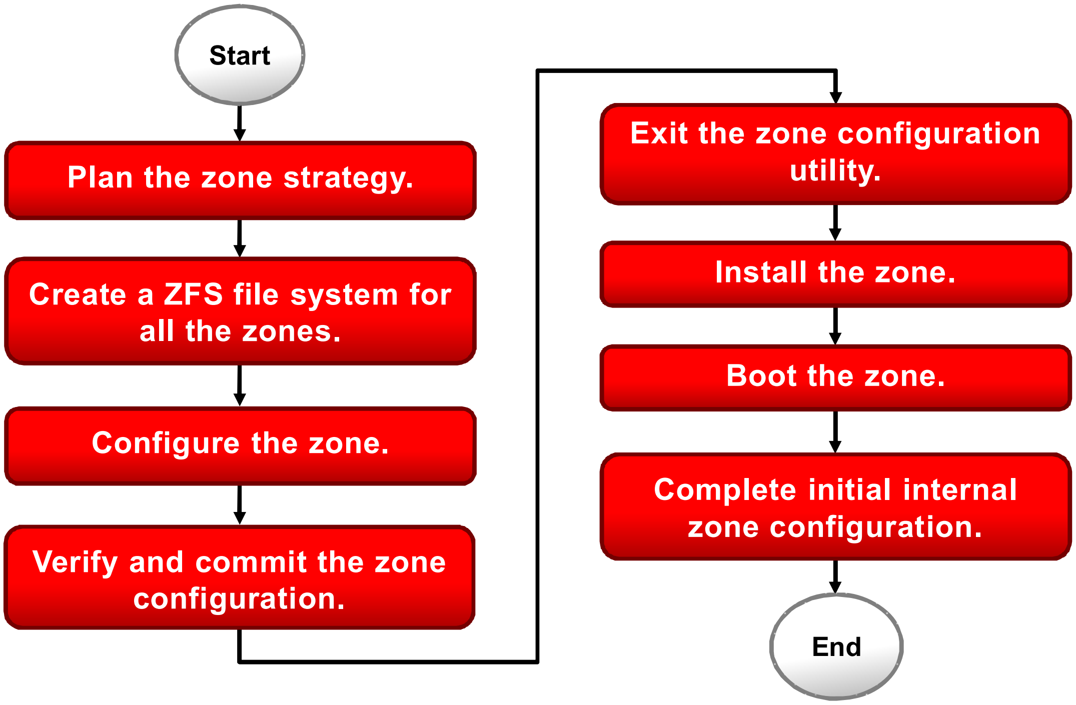
- Limit the amount of CPU resources that can be consumed by a zone
- Control the allocation of available CPU resources among zones, based on their importance
- Limit the amount of physical memory

As part of the zone configuration process, a system administrator can allocate system resources to the zone. Zone resource allocation includes, but is not limited to, the following:

- Limiting the amount of CPU resources that can be consumed by a zone
- Controlling the allocation of available CPU resources among zones, based on their importance. This is where the system administrator can set the fair share scheduler (FSS) as the scheduling class for the zone.
- Limiting the amount of physical memory

As part of planning, the resource allocations for each zone should be identified and then implemented by a designated administrator.

# Non-Global Zone Configuration Process: Overview


Start
Plan the zone strategy.
Create a ZFS file system for all the zones.
Configure the zone.
Verify and commit the zone configuration.
Exit the zone configuration utility.
Install the zone.
Boot the zone.
Complete initial internal zone configuration.
End

Although you are not going to learn how to configure a zone in this course, it is important that you are familiar with the zone configuration process. Knowing at a high level what occurs during zone configuration will help you determine the zone configuration on your system as well as how a specific zone has been configured.

The non-global zone configuration process begins with planning the zone strategy. Planning includes:

- Evaluating the applications running on the system to determine which applications you want to run in a zone
- Assessing the availability of disk space to hold the files that are unique in the zone
- Deciding whether you are going to use the resource management features and, if so, determining how to align the zone with the resource management boundaries
- Deciding whether you are going to use resource pools and, if so, configuring the pools
- Obtaining IP addresses

The next step is to create the ZFS file system for all the zones. All the required system software and any additional packages are installed into the private file systems of the zone.

After the file system for the zone has been created, the zone can be configured. During configuration, the administrator identifies the resources and properties for the zone and identifies whether the zone will be a shared-IP zone or an exclusive-IP zone.

After the configuration has been completed, the zone configuration is verified and then committed. The verify step checks to make sure that the configuration of the specified zone can be safely installed on the system. The commit command takes the configuration from memory and puts it into permanent storage.

After the zone configuration has been committed, the administrator exits the zone configuration utility and can now install the configured zone. The install process automatically creates a ZFS file system (dataset) for the zone path when the zone is installed. If a ZFS dataset cannot be created, the zone is not installed. In addition, the zone installation packages are installed from the Image Packaging System (IPS).

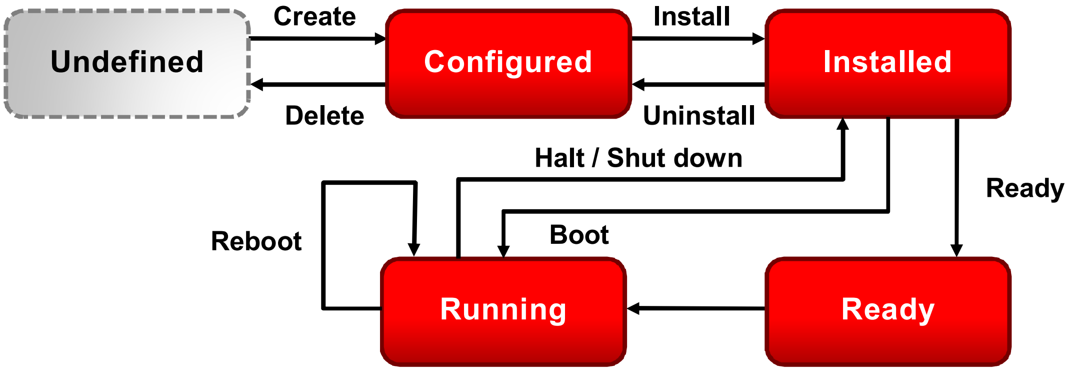
**Note:** Oracle Solaris zones participate fully in the boot environment (BE) framework. When a system is upgraded using IPS, all the zones on that system are cloned using ZFS cloning. Upon reboot of the system, the newly updated clones of the zones will be active. This ensures both a clean upgrade process and the ability to roll back if something goes wrong. In a hosted environment, end users within zones can use ZFS file systems, including creation, destruction, snapshotting, and cloning from within the zone. The global zone system administrator can focus on provisioning top-level datasets to zones and on setting space quotas. They are freed from coordinating with non-global zone users about file system layouts. The end result is that global zone administrators can provide flexible environments that are still safe and easy to update.

After the configured zone has been installed, it can be booted or activated.

After a non-global zone is booted for the first time, the internal configuration of the zone must be created. The internal configuration specifies a naming service to use, the default locale and time zone, the zone’s root password, and other aspects of the operating system environment.

The zone is now ready to be used.

## Identifying Non-Global Zone States



Now that you have a better idea of how the non-global zone configuration process flows, take a few minutes to look at the states that a non-global zone can be in. As a non-global zone is configured, enabled, and used, its status changes. The possible non-global zone states are as follows:

- **Undefined:** In this state, the zone's configuration has not been completed and committed to stable storage. This state also occurs when a zone's configuration has been deleted.
- **Configured:** In this state, the zone's configuration is complete and committed to stable storage. However, those elements of the zone's application environment that must be specified after initial boot are not yet present.
- **Incomplete:** This is a transitional state. During an install or uninstall operation, the state of the target zone is set to incomplete. After successful completion of the operation, the state is set to the correct state. However, a zone that is unable to complete the install process will stop in this state.

- **Installed:** In this state, the zone configuration is instantiated on the system. At this point, the system administrator verifies that the configuration can be successfully used on the designated Oracle Solaris system. Packages are installed under the zone's root path. In this state, the zone has no associated virtual platform.
- **Ready:** In this state, the virtual platform for the zone is established. The kernel creates the zone scheduling process, network interfaces are set up and made available to the zone, file systems are mounted, and devices are configured. A unique zone ID is assigned by the system. At this stage, no processes associated with the zone have been started.
- **Running:** In this state, the user processes associated with the zone application environment are running. The zone enters the running state as soon as the first user process associated with the application environment (`init`) is created.
- **Shutting down, down:** These states are transitional states that are visible while the zone is being halted. However, a zone that is unable to shut down for any reason will stop in one of these states.

# Implementing the Oracle Solaris Zones Plan

Your assignment is to:

- Review the current zone configuration
- Perform basic zone administration tasks

It is now time to help test the zones functionality within Oracle Solaris 11. A senior system administrator has set up several zones on the system. Your task is to review the configuration, to include the network configuration and resource allocations, and then ensure that you are able to perform basic zone administration tasks successfully. In the topics that follow, you will learn the commands that you need to perform these tasks.

# Quiz

Which type of zone is the default zone for a system?

a. Global zone
b. Non-global zone
c. Branded zone

**Answer: a**

# Quiz

Zones are isolated from each other and from the rest of the system.

a. True

b. False

**Answer: a**

# Quiz

A shared-IP zone must share a network interface with at least one other non-global zone.

a. True

b. False

**Answer: b**

# Quiz

Non-global zones can communicate only over a virtual network.

a. True

b. False

**Answer: b**

# Lesson Agenda

- Planning for Oracle Solaris Zones
- Determining an Oracle Solaris Zone Configuration
- Administering an Oracle Solaris Zone

# Determining an Oracle Solaris Zone Configuration

- Displaying the current zone configuration on the system
- Determining the current zone configuration
- Displaying a zone configuration
- Displaying zone network information
- Determining a zone’s resource utilization

# Displaying the Current Zones Configuration on the System

To display all running zones on the system, enter `zoneadm list`.

```
# zoneadm list
```

List options:

- `-c`: Displays all configured zones
- `-i`: Expands the display to all installed zones
- `-v`: Displays verbose information, including zone name, ID, current state, root directory, brand type, IP-type, and options

To display information about one or more zones on a system, you use the `zoneadm list` command.

**Note:** The `zoneadm` command is the primary tool used to administer non-global zones. Operations using the `zoneadm` command must be run from the global zone.

The `list` subcommand has several options. By itself, the list subcommand displays all running zones on the system.

To display all configured zones on the system, you can add the `-c` option. If you want to display all the installed zones on the system, you use the `-i` option. The `-v` option displays the zone name, ID, current state, root directory, brand type, IP-type, and options for the selected zone or zones.

You also have the option of combining options and even using all three options presented here simultaneously. For example, you can use `zoneadm list -civ` and `zoneadm list -cv` to show all the zones in any defined state: configured, incomplete, installed, running, shutting down, or down. The `zoneadm list -iv command` omits those zones that are only configured or incomplete. For more information about the `zoneadm` utility and its subcommands, see the `zoneadm(1M)` man page.

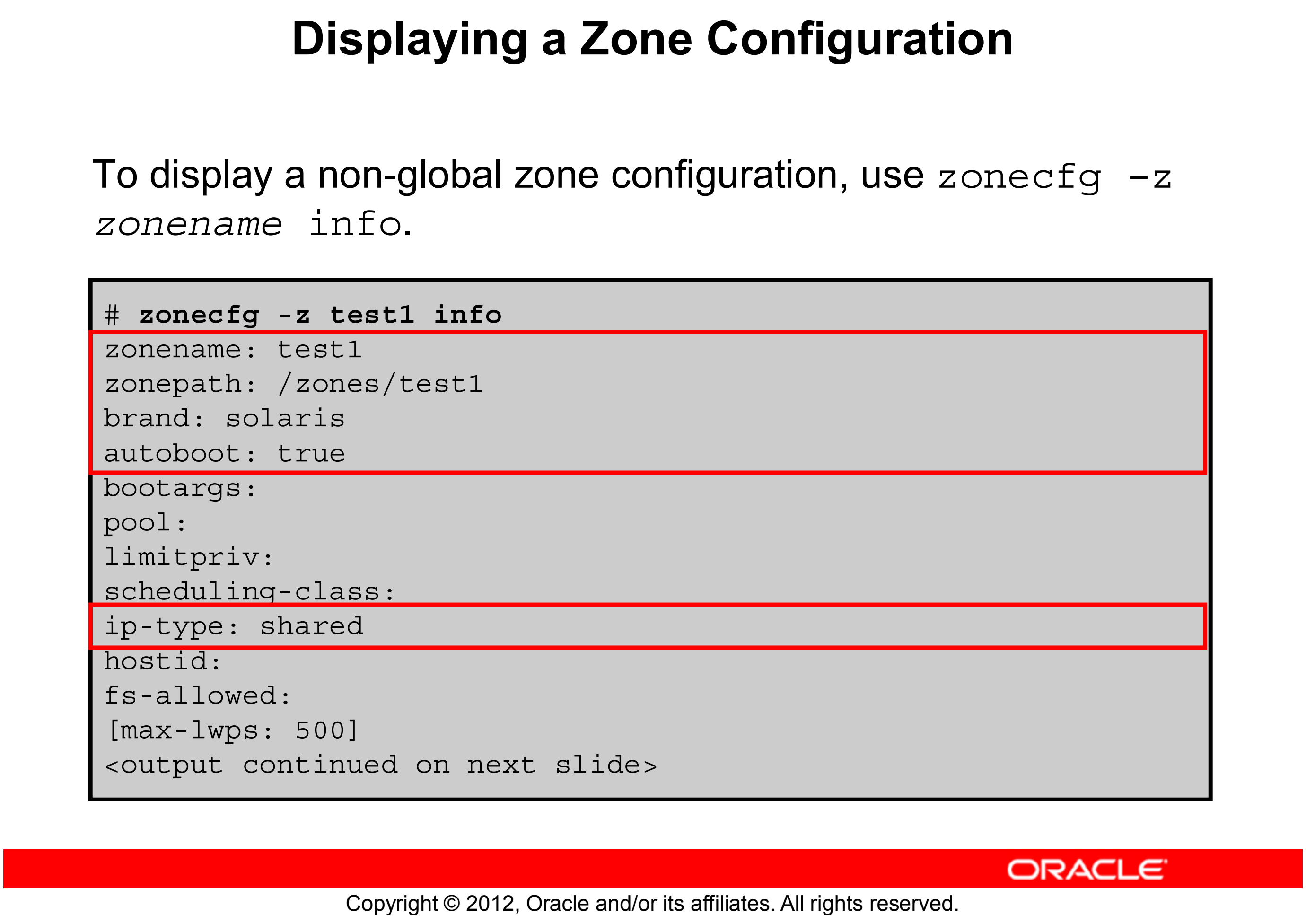
# Determining the Current Zone Configuration

```
# zoneadm list -iv
ID NAME           STATUS     PATH           BRAND     IP
 0 global         running    /              solaris   shared
 3 test1          running    /zones/test1   solaris   shared
```

This slide shows the output of `zoneadm list -iv`. On this system, there are two zones running. One is the global zone and one is `test1`. Each zone, including the global zone, is assigned a zone name. The global zone always has the name `global`. Each zone is also given a unique numeric identifier, which is assigned by the system when the zone is booted. The global zone is always mapped to ID `0`.

Notice the paths for both zones. Each zone has a path to its root directory that is relative to the global zone's root directory. The global zone's path is root (`/`), and the path for the non-global zone is `/zones/test1`. The brand for both zones is `solaris`, which is the default zone brand in the Oracle Solaris 11 release. Both the global and non-global zone are set up as shared-IP.

**Note:** The `solaris` branded zone is on all supported sun4v and x86 architecture machines. The `solaris` branded zone uses the branded zones framework to run zones installed with the same software as is installed in the global zone. The system software must always be in sync with the global zone when using a `solaris` brand.



# Displaying a Zone Configuration

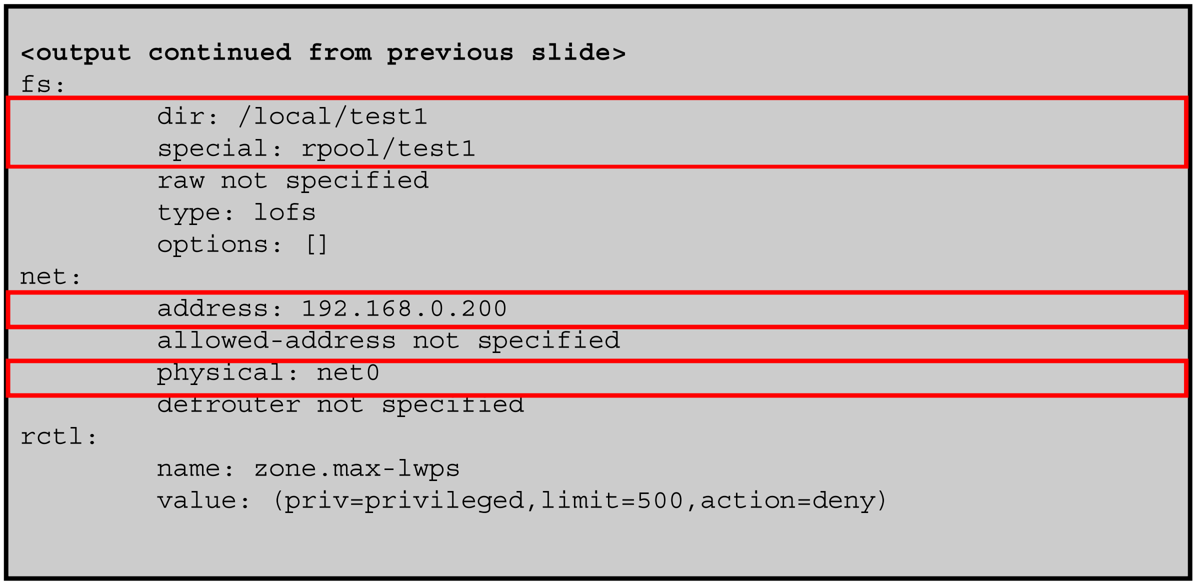
To display a non-global zone configuration, use `zonecfg -z` *`zonename`* `info`.

```
# zonecfg -z test1 info
zonename: test1
zonepath: /zones/test1
brand: solaris
autoboot: true
bootargs:
pool:
limitpriv:
scheduling-class:
ip-type: shared
hostid:
fs-allowed:
[max-lwps: 500]
<output continued on next slide>
```

You must be the global administrator in the global zone, or a user with the correct rights profile, to display a zone's configuration. To display the configuration, you use the `zonecfg -z` command followed by the zone name and the `info` subcommand, as shown in this example.

The first part of the output displays the zone name (`test1`), the zone path (`/zones/test1`), the brand (`solaris`), and the setting of the autoboot option, which, when set to `true`, indicates that the zone should be booted automatically at system boot. Notice also the IP type setting. In this example, the IP type is shared, which, as you recall from the first topic, means that this non-global zone is sharing the IP layer with the global zone.

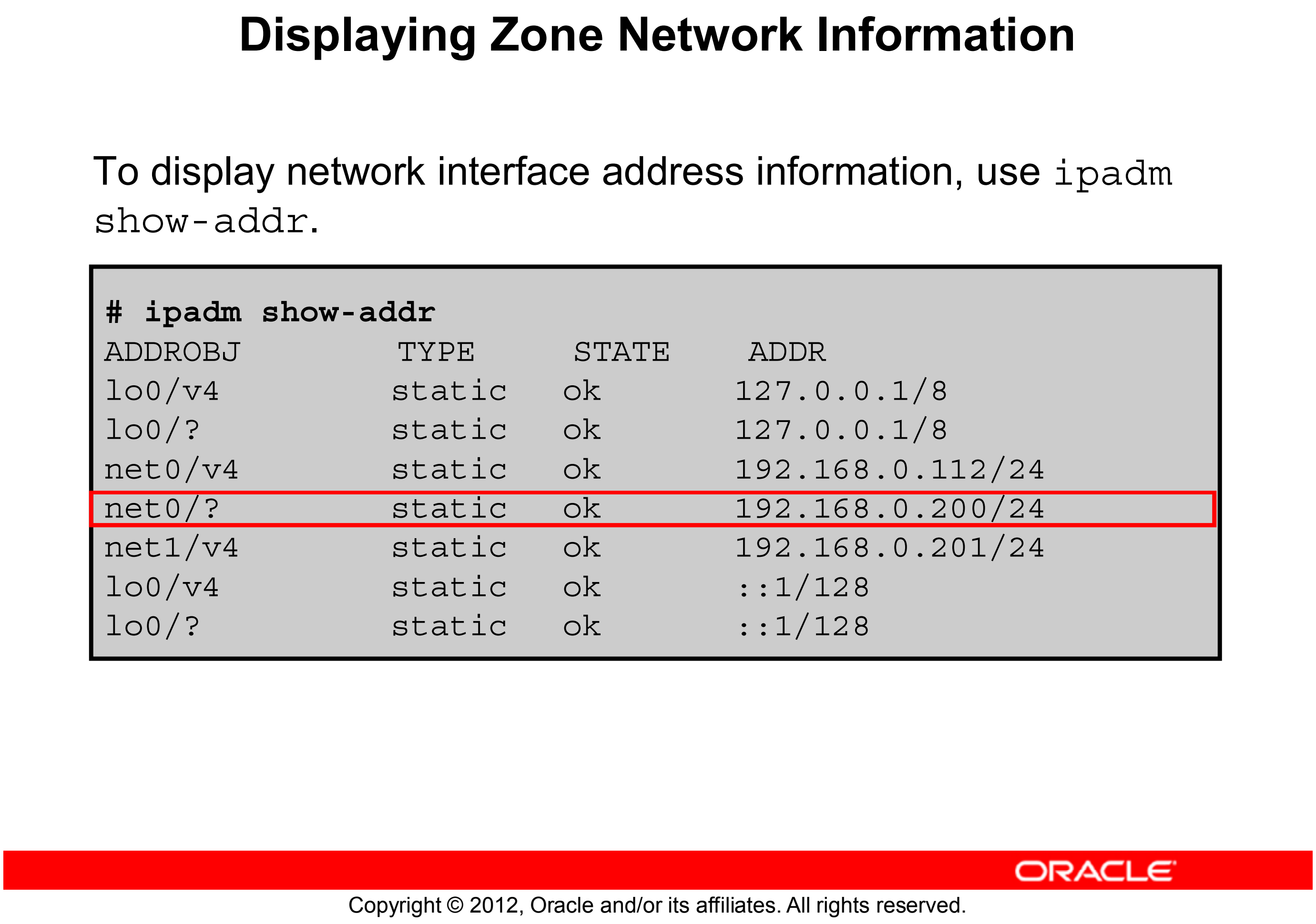
# Displaying a Zone Configuration



```
<output continued from previous slide>
fs:
         dir: /local/test1
         special: rpool/test1
         raw not specified
         type: lofs
         options: []
net:
         address: 192.168.0.200
         allowed-address not specified
         physical: net0
         defrouter not specified
rctl:
         name: zone.max-lwps
         value: (priv=privileged,limit=500,action=deny)
```

The next section of the output (shown here) displays zone's file system information (`fs`), such as the directory location, the zone's network information, such as the IP address (`192.168.0.200`) and NIC (`net0`), the zone attribute information, and the resource control settings (`rctl`).

**Note:** If this zone were using a virtual network interface, after `physical` in the network section, you would see `vnic1` or `vnic2`.



# Displaying Zone Network Information

To display network interface address information, use `ipadm show-addr`.

```
# ipadm show-addr
ADDROBJ         TYPE     STATE    ADDR
lo0/v4          static   ok       127.0.0.1/8
lo0/?           static   ok       127.0.0.1/8
net0/v4         static   ok       192.168.0.112/24
net0/?          static   ok       192.168.0.200/24
net1/v4         static   ok       192.168.0.201/24
lo0/v4          static   ok       ::1/128
lo0/?           static   ok       ::1/128
```

For network interface address information, you can use the `ipadm show-addr` command, as shown in this example. You might recall this command from when you were verifying the network interface information during the operating system installation verification task. As you can see, it is more difficult to directly associate the zone with its IP address with this view. However, you can easily see the state of the interface.

## Determining a Zone's Resource Utilization

To determine a zone's resource utilization, use the `zonestat` utility.

```
# zonestat -r summary 5
Collecting data for first interval...
Interval: 1, Duration: 0:00:05
SUMMARY                Cpus/Online: 1/1    PhysMem: 2047M    VirtMem: 3071M
                      ---CPU----  --PhysMem--  --VirtMem--   --PhysNet--
               ZONE  USED %PART  USED %USED   USED %USED   PBYTE %PUSE
            [total]  0.10 10.7% 1394M 68.0%  1661M 54.0%      0   0.00%
           [system]  0.01 1.23%  968M 47.3%  1260M 41.0%      -      -
             global  0.08 8.76%  256M 12.5%   257M 8.39%      0   0.00%
              test1  0.00 0.22% 54.2M 2.64%  45.6M 1.48%      0   0.00%
              test2  0.00 0.23% 58.2M 2.84%  48.4M 1.57%      0   0.00%
              test3  0.00 0.30% 56.5M 2.76%  48.5M 1.58%      0   0.00%


…
…
…
```

If you are asked to monitor a zone's resource utilization, you can do so with the `zonestat` utility. The `zonestat` utility provides highly accurate reports on the CPU, memory, and resource control utilization of running zones. Each zone's utilization is reported as a percentage of both system resources and the zone's configured limits. You can specify the interval at which you want the utility to print a report. In the example shown here, you have selected one interval with a duration of 5 seconds.

The `zonestat` utility can be used from within a zone, for a localized view of zone system resources, or from the global zone for a system-wide view of zone system resources. With this utility, it is very easy to identify resource bottlenecks or misbehaving applications. The system administrator is able to take prompt remedial action, accurately addressing the identified problem. Capacity planning is also greatly simplified. In addition, `zonestat` also helps the administrator quickly understand how zones and resource management have been configured. This can be particularly useful when taking over administration duties on an unfamiliar system.

One of the uses of the `zonestat` utility is to monitor periodic usage of the system's physical memory, virtual memory, and CPU resources. In the example shown here, we are monitoring the resource utilization of all the zones from the global zone by using the `zonestat` command followed by `-r summary 5`. The output shows how many CPUs are available on the system and how many are being used. If a CPU has been partitioned (that is, if multiple processor sets have been created on the system by dividing up the CPU), usage of the partition is displayed.

To the right of the CPU usage, the physical and virtual memory usage by the system and by each zone are displayed. Physical memory refers to RAM on the machine, and virtual memory refers to RAM plus system swap space. The memory usage shows how much memory is being used and the percentage of memory being used. To the right of the virtual memory is the physical network utilization for each zone. You can use this information to determine the load on a NIC.

**Note:** Oracle Solaris enables you to regulate memory consumption in non-global zones by using a resource capping daemon. For more information on resource capping, see "Physical Memory Control Using the Resource Capping Daemon" in *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management.*

Now, take a closer look at the percentages in each area to see what determinations you can make about the resource utilization for this system. Begin with the CPU usage. As you can see, the system shows the highest usage (1.23% of the CPU). The global zone shows the highest usage ( 8.76% of the CPU) among the zones. The non-global zones are using 0.22%, 0.23%, and 0.30% respectively, which is much lower than the CPU usage of the global zone. The total usage of the available CPU is 10.7%.

Now, look at the physical memory. There is a total of 2047 MB of physical RAM available. Most of the memory (47.3%) is consumed by the system processes. The next highest memory usage is 12.5% by the global zone. This percentage reflects all the processes running for the global and non-global zones. Respectively, the non-global zones are using 2.64%, 2.84%, and 2.76% of the physical memory, which represents the memory being used by the processes running in each of the non-global zones. The total amount of physical memory usage is 68.0%.

Finally, there is the virtual memory usage. The total virtual memory available is 3071 MB. Because you are working with more total virtual memory than total physical memory, the percentages are smaller in the virtual memory column as compared to those displayed in the physical memory column. The system is showing a usage of 41.0%. The global zone is using 8.39%, and the zones are using 1.48%, 1.57% and 1.58% respectively.

Given the low percentages, it does not appear that there are any issues with this system's resource usage. If the numbers were high (for example, 40% of the CPU utilization or 80% of the memory), or if one of the non-global zones was taking up so much memory that very little was left for the global zone or the other zone, you would want to make adjustments to the resource allocations or controls.

For more information about the `zonestat` utility, see the "Using the `zonestat` Utility in a Non-Global Zone" section of *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management* and the `zonestat(1)` man page.

# Quiz

If you want to see additional information about all configured, running, and installed zones on a system, which command would you use?

a. `zoneadm list`

b. `zoneadm list -c`

c. `zoneadm list -civ`

**Answer: c**

# Quiz

Which command would you use to display configuration information about a zone named `myzone`?

a. `zoneadm myzone status`

b. `zoneadm myzone info`

c. `zonecfg -z myzone info`

d. `zonecfg -z myzone verify`

**Answer: c**

# Practice 6-1 Overview: Determining an Oracle Solaris Zone's Configuration

This practice covers the following topics:

- Examining configuration of the current zones
- Determining the current zone resource allocation

In the practices for Lesson 6, you are presented with two tasks designed to reinforce the concepts presented in the lecture portion of this lesson. You will have the chance to perform the following tasks:

- **Practice 6-1:** Determining an Oracle Solaris zone's configuration
- **Practice 6-2:** Administering an Oracle Solaris zone

You will find Practice 6-1 in your *Activity Guide*. It should take you about 50 minutes to complete.

## Lesson Agenda

- Planning for Oracle Solaris Zones
- Determining an Oracle Solaris Zone Configuration
- Administering an Oracle Solaris Zone

ORACLE

# Administering an Oracle Solaris Zone

- Logging in to a zone
- Logging out of a zone
- Shutting down a zone
- Starting up a zone
- Halting a zone

In this topic you are shown how to perform basic zone administration tasks, such as logging in to and out of a zone, shutting down and starting up a zone, and halting a zone. Before you look at each task, take a moment to discuss the delegated administration of zones.

# Delegated Administration for Zones

- Zones are fully integrated with IPS.
- Updating zones with IPS is simple and efficient.
- Zone content can be customized at installation time.
- Benefits for the system administrator include:
  - Reduced time to deploy or update zone environments
  - Reduced length and risk of planned downtime
  - Ability to meet different application requirements without having to install every package

With Oracle Solaris 11, you have the ability to delegate common zone administration tasks for specific zones to different administrators. With delegated administration, for each zone, a user or set of users may be identified with the permissions to log in, manage, or clone from that zone. These specific authorizations are interpreted by the appropriate commands running in the global zone to allow access at the correct authorization level to the correct user.

There are two basic levels of zone administration: global administrator and zone administrator.

A global administrator has superuser privileges or an equivalent rights profile. When logged in to the global zone, the global administrator can monitor and control the system as a whole.

A non-global zone can be administered by a zone administrator. The global administrator assigns the required authorizations to the zone administrator. The privileges of a zone administrator are confined to a non-global zone.

Being able to delegate administration is extremely useful in a shared environment where you want to allow specific users to manage only zones that are relevant to their role.

# Logging In to a Zone

To log in to a zone, use `zlogin` followed by the zone name.

```
# zlogin my-zone
[Connected to zone ‘my-zone’ pts/1]
Oracle Corporation   SunOS 5.11     11.0  November 2011
```

In order to perform administrative tasks in a zone, such as modifying the configuration, taking a backup, or monitoring resource usage, you must be logged in to the zone. The `zlogin` utility is used to enter a non-global zone. Only a user operating in the global system zone can use this utility, and it must be executed with all privileges. To log in to a zone, you use the `zlogin` command followed by the zone name, as shown in the example.

# Exiting a Non-Global Zone

To exit a non-global zone from a pseudo terminal or terminal login, use `exit`.

```
# exit
```

To disconnect from a zone from a virtual console or console login, use `~.`.

```
# ~.
```

When you have completed your administrative tasks in a zone, you must log out of, or exit, the zone. Logging out of the zone can save on system resources, especially if you are running multiple zones on a system.

There are two methods for exiting a non-global zone depending on whether you are exiting the zone from a non-virtual console or disconnecting from a virtual console. To exit a zone from a non-virtual console, use the `exit` command, as shown in the first example. To exit a zone from a virtual console, use the tilde (~) character and a period, as shown in the second example.

# Shutting Down a Non-Global Zone

To shut down a zone, use `zoneadm -z zonename shutdown -i 0`.

```
global# zoneadm -z my-zone shutdown -i 0
```

There are various reasons why you might be directed to shut down a zone. For example, another zone needs to complete a database update before the zone you have been told to shut down can be brought back up.

To shut down a non-global zone you must be the global administrator or a user with appropriate authorizations in the global zone. This procedure is used to cleanly shut down a zone (as opposed to halting a zone).

To shut down a zone, from the global zone you use the `zoneadm -z` command followed by the zone name and specify `shutdown` as the command to run and `init 0` as the state, as shown in the example.

**Note:** At this time, you cannot use the `shutdown` command to place the zone in single-user state.

# Starting Up a Zone

To start a zone, use `zoneadm -z zonename boot`.

```
global# zoneadm -z my-zone boot
```

After a zone has been shut down, you can start it again by using the `zoneadm -z` command followed by the zone name and `boot`, as shown in this example. When a zone is booted, the required services and the facilities for that zone are brought online. These services help the zone to be functional and ready to use.

# Halting a Zone

To halt a zone, run `zoneadm -z zonename halt`.

```
global# zoneadm -z my-zone halt
```

To verify that the zone has been halted, run `zoneadm list -v`.

```
global# zoneadm list -iv
ID  NAME     STATUS       PATH                             BRAND      IP
 0  global   running      /                                solaris    shared
 -  my-zone  installed    /zones/my-zone                   solaris    shared
```

When you need to remove a zone, you use the `zoneadm -z halt` command with the zone name, as shown in the example. When halted, the zone is brought back to the `installed` state.

To verify that the zone has been halted and is no longer running, you can run the `zoneadm list -iv` command, as shown in the second example. As you can see in the output, only the global zone is running. There are no other zones running on the system.

**Note:** Although you can halt a zone, the recommended way to bring a zone down is by using the `zoneadm shutdown` command. This approach brings the zone down more gently.

# Quiz

The privileges of a zone administrator are confined to a non-global zone.

a. True

b. False

**Answer: a**

# Quiz

Which command is used to perform a clean shutdown of a zone?

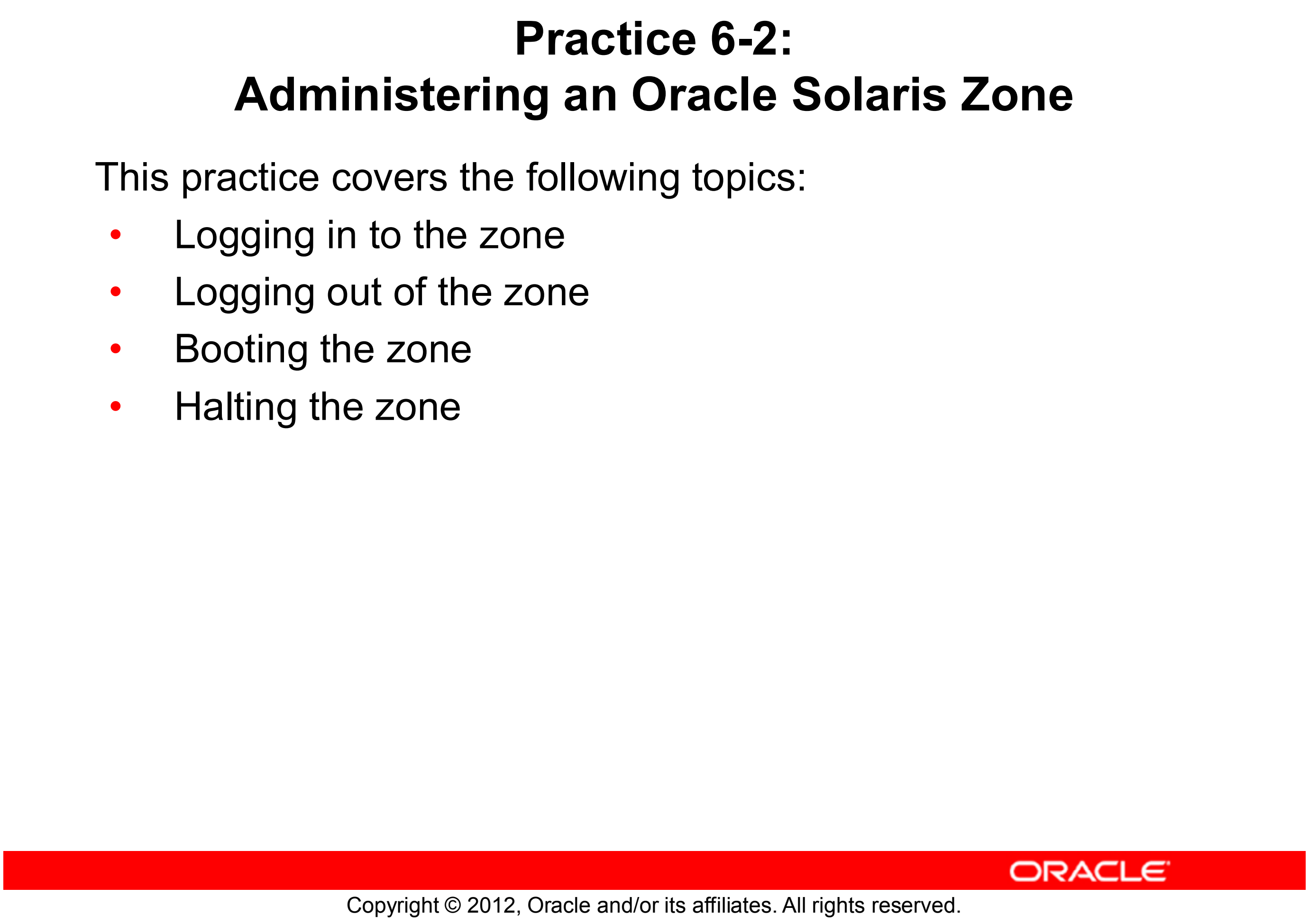
a. exit

b. zoneadm –z *zonename* shutdown

c. zoneadm -z *zonename* halt

d. ~.

**Answer: b**



# Practice 6-2: Administering an Oracle Solaris Zone

This practice covers the following topics:

- Logging in to the zone
- Logging out of the zone
- Booting the zone
- Halting the zone

This practice should take you about 40 minutes to complete.

# Summary

In this lesson, you should have learned how to:

- Implement a plan for Oracle Solaris zone management
- Determine the current zone configuration on the system
- Determine the current zone resource utilization on the system
- Administer an Oracle Solaris zone

In this lesson, you were introduced to zone technology and shown how to determine what zones are currently configured on the system, as well as how to determine what a zone's configuration and resource utilization are. You also learned how to perform basic zone administration tasks in an Oracle Solaris zone, such as logging in and exiting a zone and shutting down and starting a zone.

7

# Administering a Physical Network

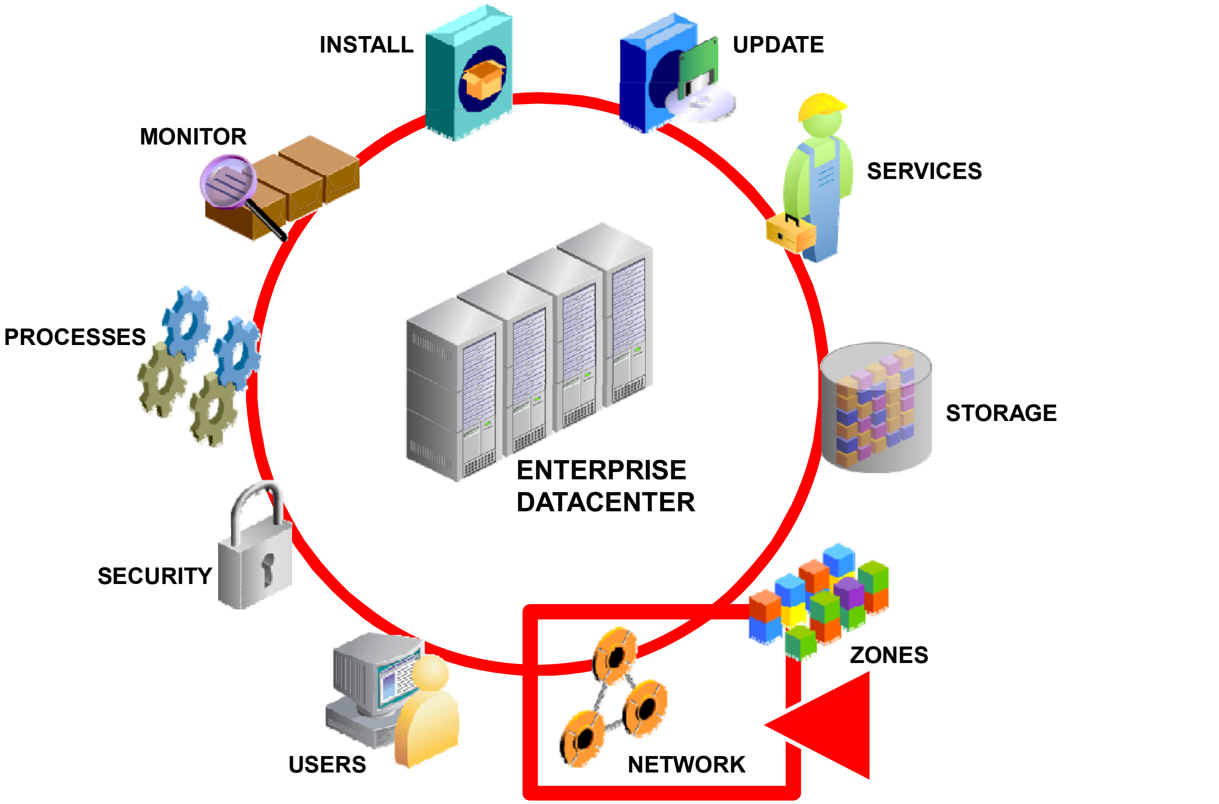
# Objectives

After completing this lesson, you should be able to:

- Implement a plan for network management
- Determine datalink availability
- Configure a network interface
- Administer a network interface
- Verify network operation

In this lesson, you are presented with a plan for the existing client-server network setup. You learn to determine which datalinks are available, and then to configure and administer a network interface. You also learn how to verify that the network is operational.

## Workflow Orientation



Before you start the lesson, orient yourself to where you are in the job workflow. You have successfully installed the operating system, updated it, tested the SMF services, set up and administered the data storage environment, and worked with zones. Now you will be introduced to basic network administration. To be able to support your company's network administration needs, you need to be familiar with how your company's network is set up. In the client-server networking environment, the hosts communicate with each other by sending and receiving business data. One of your responsibilities would be to monitor the network interfaces that support the transfer of data between hosts to ensure that communications continue uninterrupted.

# Lesson Agenda

- Planning for Network Management
- Determining Datalink Availability
- Configuring a Network Interface
- Administering a Network Interface
- Verifying Network Operation

# Planning for Network Management

The network management plan addresses how to configure and manage its TCP/IP network by using Oracle Solaris 11, including the following:

- IP addressing scheme
- Network interfaces
- Datalinks

Your company uses a TCP/IP network that is configured with an IPv4 addressing scheme. The plan that your company has developed for network management specifies how the network will be configured and managed with the implementation of Oracle Solaris 11.

To understand how your company's network is structured and operates, you need to be familiar with the TCP/IP protocol architecture model. In this lesson, you learn about this model as well as how the IP addressing schemes, network interfaces, and datalinks support this model.

# TCP/IP Protocol Architecture Model

| OSI Ref. Layer No. | OSI Layer Equivalent | TCP/IP Layer | TCP/IP Protocol Examples |
|---|---|---|---|
| 5, 6, 7 | Application, (7)<br>Presentation (6)<br>Session (5) | Application | `telnet`, `ftp`, `rlogin`, DNS, LDAP, and NFS |
| 4 | Transport | Transport | TCP |
| 3 | Network | Internet | IPv4, IPv6 |
| 2 | Datalink | Datalink | IEEE 802.2 |
| 1 | Physical | Physical network | Ethernet (IEEE 802.3) |

Most network protocol suites are structured as a series of layers, which is sometimes collectively referred to as a protocol stack. Each layer is designed for a specific purpose. Each layer exists on both the sending and receiving systems. The International Organization for Standardization (ISO) designed the Open Systems Interconnection (OSI) Reference Model that uses structured layers. The OSI model describes a structure with seven layers for network activities. This model describes idealized network communications with a family of protocols.

TCP/IP does not directly correspond to this model. TCP/IP either combines several OSI layers into a single layer, or does not use certain layers at all. The table in the slide shows the layers of the Oracle Solaris implementation of TCP/IP. The table lists the layers from the topmost layer (application) to the bottommost layer (physical network). It shows the TCP/IP protocol layers and the OSI model equivalents. Also shown are examples of the protocols that are available at each level of the TCP/IP protocol stack. Each system that is involved in a communication transaction runs a unique implementation of the protocol stack.

You now look at each of the TCP/IP layers, beginning with the physical layer and working your way up to the application layer.

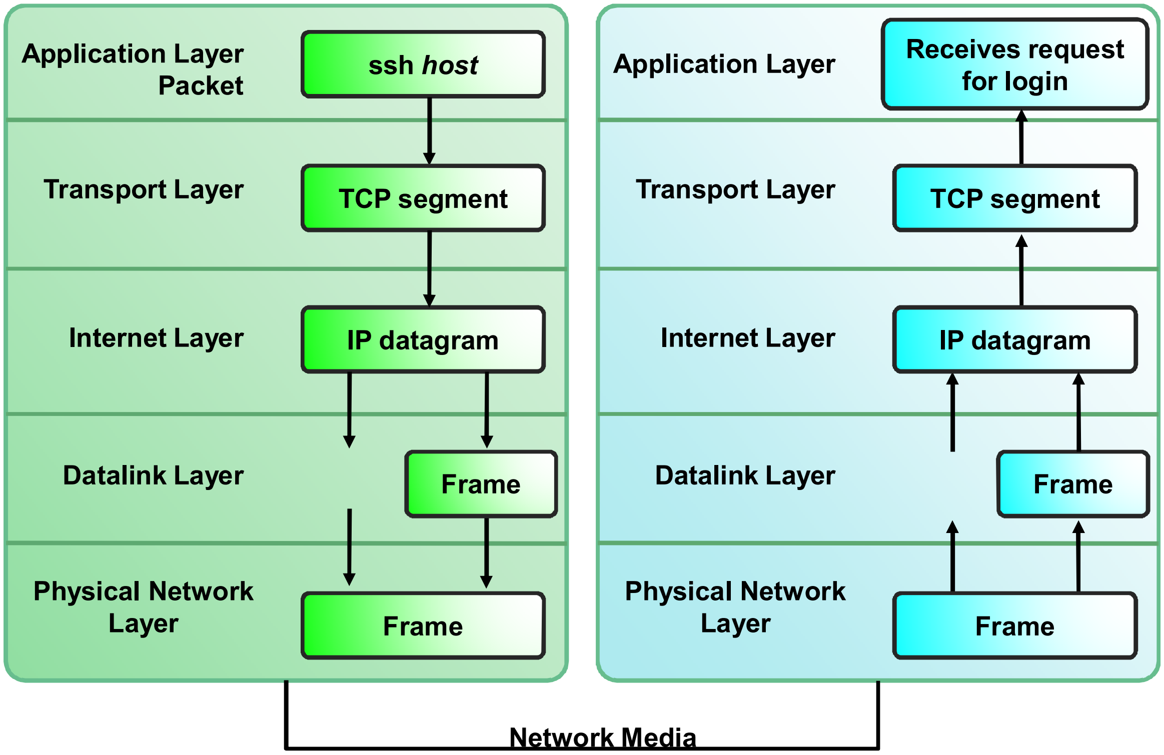
- **Physical network layer:** Specifies the characteristics of the hardware to be used for the network. For example, the physical network layer specifies the physical characteristics of the communications media, such as IEEE 802.3, which is the specification for Ethernet network media.
- **Datalink layer:** Identifies the network protocol type of the packet, in this instance TCP/IP. The datalink layer also provides error control and “framing.” An example of a datalink layer protocol is Ethernet IEEE 802.2 framing.
- **Internet layer:** Accepts and delivers packets for the network. This layer includes the powerful Internet Protocol (IP) and is also known as the network layer or IP layer. The IP protocol and its associated routing protocols are possibly the most significant of the entire TCP/IP suite. IP is responsible for the following:
  - **IP addressing:** The IP addressing conventions (for example, IPv4 and IPv6 addressing) are part of the IP protocol.
  - **Host-to-host communications:** IP determines the path that a packet must take, based on the receiving system’s IP address.
  - **Packet formatting:** IP assembles packets into units that are known as datagrams.
  - **Fragmentation:** If a packet is too large for transmission over the network media, IP on the sending system breaks the packet into smaller fragments. IP on the receiving system then reconstructs the fragments into the original packet.
- **Transport layer:** Ensures that packets arrive in sequence and without error by swapping acknowledgments of data reception, and retransmitting lost packets. This type of communication is known as end-to-end. TCP is an example of a transport layer protocol at this level. TCP enables applications to communicate with each other as though they are connected by a physical circuit. TCP sends data in a form that appears to be transmitted in a character-by-character fashion, rather than as discrete packets. This transmission consists of the following:
  - Starting point, which opens the connection
  - Entire transmission in byte order
  - Ending point, which closes the connection

  TCP attaches a header onto the transmitted data. This header contains many parameters that help the processes on the sending system to connect to the peer processes on the receiving system. TCP confirms that a packet has reached its destination by establishing an end-to-end connection between the sending and receiving hosts. TCP is therefore considered a “reliable, connection-oriented” protocol.

- **Application layer:** Defines the standard Internet services and network applications that anyone can use. These services work with the transport layer to send and receive data. Many application layer protocols exist. Examples of application layer protocols include:
  - Standard TCP/IP services, such as the `ftp` and `telnet` commands
  - UNIX “`r`” commands, such as `rlogin`. The “`r`” is for “remote.”
  - Name services, such as the domain name system (DNS). Name services maintain critical information about the machines on a network, such as the host names, IP addresses, Ethernet addresses, and so forth. DNS is the name service provided by the Internet for TCP/IP networks. DNS provides host names to the IP address service. DNS also serves as a database for mail administration.
  - Directory services, such as Lightweight Directory Access Protocol (LDAP), which Oracle Solaris supports. The distinction between a name service and a directory service is in the differing extent of functionality. A directory service provides the same functionality as a naming service, but provides additional functionalities as well.
  - File services, such as the Network File System (NFS) service. NFS enables users on a network to share file systems.

For more information about the OSI reference model and the TCP/IP model, see *Oracle Solaris Administration: IP Services*.

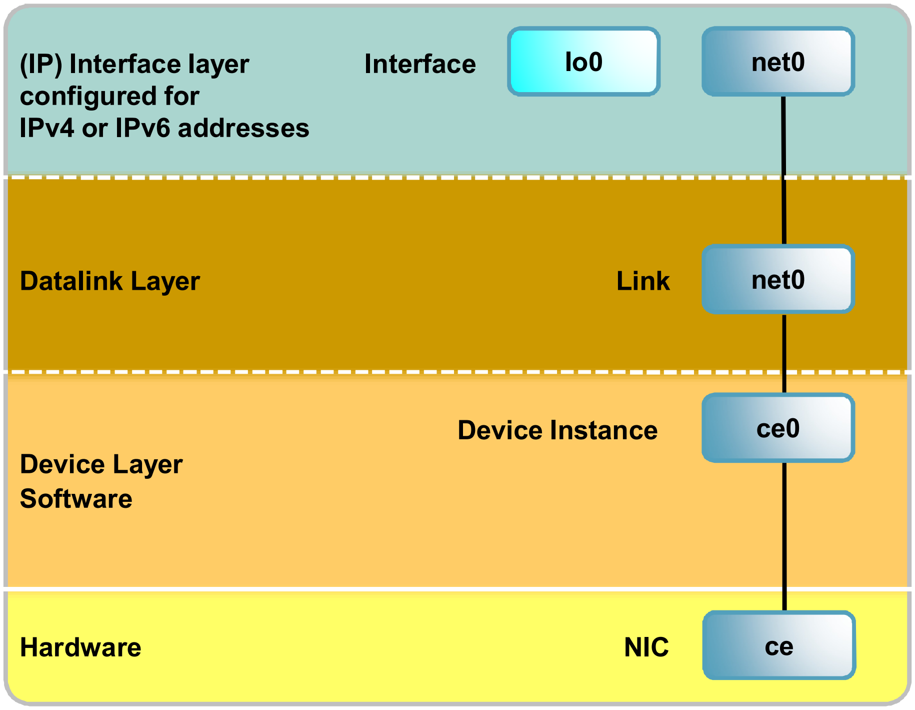
# How TCP/IP Handles Data Communications



When a user issues a command that uses a TCP/IP application layer protocol, a series of events is initiated. The user's command or message passes through the TCP/IP protocol stack on the local system. Then, the command or message passes across the network media to the protocols on the remote system. The protocols at each layer on the sending host add information to the original data. The graphic in the slide illustrates this process.

Now that you have a better understanding of the TCP/IP model and how it handles data communications, you can look at the networking stack in detail, which connects the physical network, datalink, and Internet layers of the TCP/IP protocol stack.

# Networking Stack



To connect to the network, a system must have at least one physical network interface. Each network interface must have its own unique IP address. During Oracle Solaris installation, an IP address is supplied for the first interface that the installation program finds. These interfaces are configured over datalinks, which in turn correspond to instances of hardware devices in the system. Network hardware devices are also called network interface cards (NICs) or network adapters. Certain NICs have only a single interface that resides on the card. Many other brands of NICs have multiple interfaces that you can configure to perform network operations.

From an administrative perspective, a network interface has a link name. The datalink represents a datalink object in the second layer of the TCP/IP model. As was just discussed, the physical link is directly associated with a device and possesses a device name. The device name is essentially the device instance name, and is composed of the driver name and the device instance number.

Driver names can be `nge`, `nxge`, and `bge`, among many other driver names. The variable instance number can have a value from zero through n, depending on how many interfaces of that driver type are installed on the system.

For example, consider a Gigabit Ethernet card, which is often used as the primary NIC on both host systems and server systems. Some typical driver names for this NIC are `nge` and `bge`. When used as the primary NIC, the Gigabit Ethernet interface has a device name such as `nge` or `bge`.

In Oracle Solaris's current model of the network stack, interfaces and links on the software layer build on the devices in the hardware layer. More specifically, a hardware device instance in the hardware layer has a corresponding link on the datalink layer and a configured interface on the interface layer. This one-to-one relationship means that network configuration is dependent on hardware configuration and network topology. Interfaces must be reconfigured if changes are implemented in the hardware layer, such as replacing the NIC or changing the network topology.

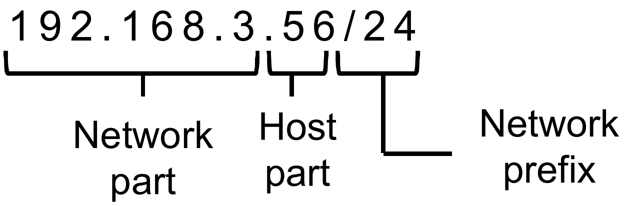
The graphic in the slide illustrates the one-to-one relationship between the network device, its datalink, and the IP interface. As you can see, there is one NIC on the hardware layer: `ce` with a single device instance `ce0`. Device `ce0` has a corresponding link `net0` on the datalink layer.

**Note:** For simplicity and uniformity, Oracle Solaris 11 uses `netx` as a vanity naming scheme for the NIC.

The datalink has a corresponding IP interface (`net0`). This interface can be configured with IPv4 or IPv6 addresses to host both types of network traffic. Note also the presence of the loopback interface `lo0` on the interface layer. This interface is used to test, for example, that the IP stack is functioning properly.

# IPv4 Addressing

- The IPv4 address is:
  - A 32-bit number that uniquely identifies a network interface on a system
  - Written in decimal digits
  - Divided into four 8-bit fields that are separated by periods
- Component parts of an IPv4 address:
  - Network part
  - Host part
  - Network prefix



IPv4 addresses are the original IP addressing format that was designed for TCP/IP. Although you can no longer obtain class-based IPv4 network numbers from an ISP, many existing networks still have them.

The IPv4 address is a 32-bit number that uniquely identifies a network interface on a system. An IPv4 address is written in decimal digits and divided into four 8-bit fields that are separated by periods. Each 8-bit field represents a byte of the IPv4 address. This form of representing the bytes of an IPv4 address is often referred to as the dotted-decimal format.

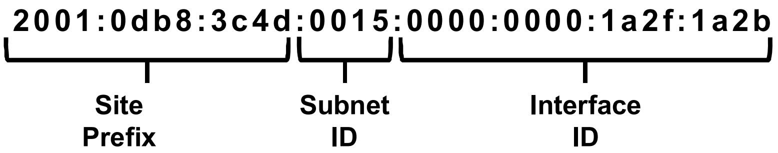
An IPv4 address is composed of the following component parts:

- Network part, which consists of the IPv4 network number that is received from an Internet service provider (ISP) or Internet Registry (IR)
- Host part, which you assign to an interface on a system
- Network prefix, which defines how many bits of the address comprise the network number. The network prefix also provides the subnet mask for the IP address.

Any IPv4 address that you obtain from an ISP is in the Classless Inter-Domain Routing (CIDR) format, as shown in the figure in the slide. These addresses were developed as a short-to-medium term fix for the shortage of IPv4 addresses. The network prefix of the CIDR address indicates how many IPv4 addresses are available for the hosts on your network. Note that these host addresses are assigned to the interfaces on a host. If a host has more than one physical interface, a host address must be assigned for every physical interface that is in use. The network prefix of a CIDR address also defines the length of the subnet mask.

# IPv6 Addressing

- Was developed to address:
  - IPv4 shortage
  - Manual address configuration
- Uses 128-bit addressing
  - Divided into eight, 16-bit fields, with each field bounded by a colon
  - Written in hexadecimal numbers
- Includes component parts such as:
  - Site prefix
  - Subnet ID
  - Interface ID



IPv6 is the most recent version of the IP specification. One of the reasons that IPv6 was developed was to address IPv4 address shortage and to resolve the need for administrators to manually assign IP addresses as is done in IPv4 by editing the `/etc/inet/hosts` file. IPv6 systems configure their IPv6 addresses automatically. Administrators, however, must still administer the name-to-IPv6 address mapping.

An IPv6 address is 128 bits in length and consists of eight, 16-bit fields, with each field bounded by a colon. Each field must contain a hexadecimal number, in contrast to the dotted-decimal notation of IPv4 addresses.

An IPv6 address consists of the following component parts:

- Site prefix (48 bits), which describes the public topology that is usually allocated to your site by an ISP or Regional Internet Registry (RIR)
- Subnet ID (16 bits), which an administrator allocates for your site. The subnet ID describes the private topology, which is also known as the site topology, because it is internal to your site.

- Interface ID (64 bits), which is either automatically configured from the interface's MAC address or manually configured in EUI-64 format (also referred to as a token)

**Note:** Oracle Solaris supports IPv4 and IPv6 addressing on the same host, through the use of dual-stack TCP/IP. As with IPv4 addresses in CIDR format, IPv6 addresses have no notion of network classes or netmasks.

# Unicast, Multicast, and Broadcast Addressing

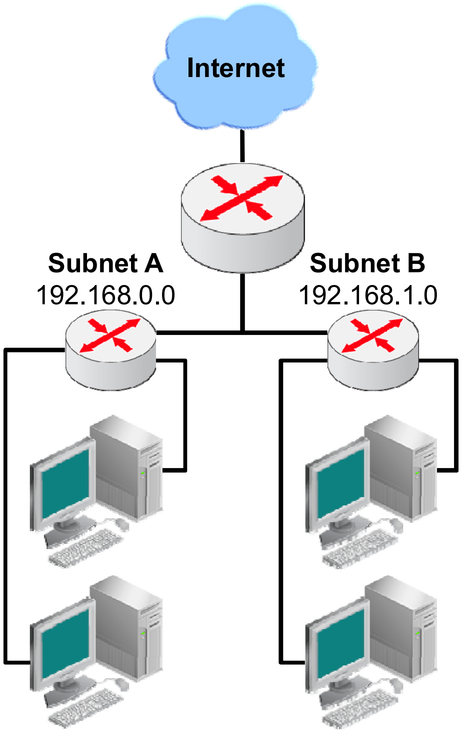
- Unicast: Sends information to a single network interface
- Multicast: Sends information or services to all the network interfaces that are defined as members of the multicast group
- Broadcast: Sends information to all the network interfaces on a specific subnet

NICs are usually configured to listen for three types of messages: messages sent to their specific address, messages sent to a specific group of interfaces, and messages sent to all interfaces on a specific subnet. For each type of message transmission, there is an associated IP addressing type:

- **Unicast:** A unicast address is used to send information to a single network interface.
- **Multicast:** A multicast address is used to send information or services to all the network interfaces that are defined as members of the multicast group. The multicast address identifies a multicast group, which is a group of interfaces, usually on different nodes. An interface can belong to any number of multicast groups. Both IPv4 and IPv6 support the use of multicast addresses. For example, one use of multicast addresses is to communicate with all IPv4 or IPv6 nodes on the local link. If an address begins with ff00n, it is a multicast address.
- **Broadcast:** A broadcast address is used to send information to all the network interfaces on a specific subnet.

# Subnets, Netmasks, and Subnet Masks

- Subnets:
  - Allow allocation of the host address space to network addresses
  - Are created by using a netmask
- Netmasks determine:
  - How many and which bits in the host address space represent the subnet number
  - How many and which bits represent the host number
- Subnet masks determine which bits in the host address bytes are applied to the subnet and host addresses.



Local networks with large numbers of hosts are sometimes divided into subnets by using routers. A subnet is a group of hosts on the same network segment that share the same network address. Subnetting allows you to divide one network address into multiple network addresses (or subnets) by allocating a part of the host address space to network addresses.

Subnets are created by using a netmask. The netmask determines how many and which bits in the host address space represent the subnet number and how many and which bits represent the host number. The bits in the host address bytes that are applied to subnet addresses and those applied to host addresses are determined by a subnet mask. Subnet masks are used to select bits from either byte for use as subnet addresses.

The netmask can be applied to an IPv4 address by using the bitwise logical `AND` operator. This operation selects the network number and subnet number positions of the address. For example, if a netmask `255.255.255.0` is applied to the IPv4 address `192.168.0.100`, the result is the IPv4 address of `192.168.0.0` (`192.168.0.100` and `255.255.255.0` = `192.168.0.0`).

# Implementing the Network Management Plan

Your assignment is to:

- Determine the datalinks that are available
- Configure the network interface
- Administer the network interface
- Verify that the network is operational

You now help to test the physical network functionality, specifically the network interfaces. As part of testing, you determine the availability of the datalinks, configure the network interface, and administer it. You also verify that the network is operational. In the slides that follow, you learn the commands that you need to perform these tasks.

# Quiz

Which layer of the TCP/IP protocol stack is responsible for accepting and delivering packets for the network?

a. Datalink

b. Transport

c. Internet

d. Application

**Answer: c**

# Quiz

The TCP/IP protocol supports only IPv4 addressing.

a. True

b. False

**Answer: b**

# Quiz

This is an example of an IPv4 address: 192.168.3.56/24

a. True

b. False

**Answer: a**

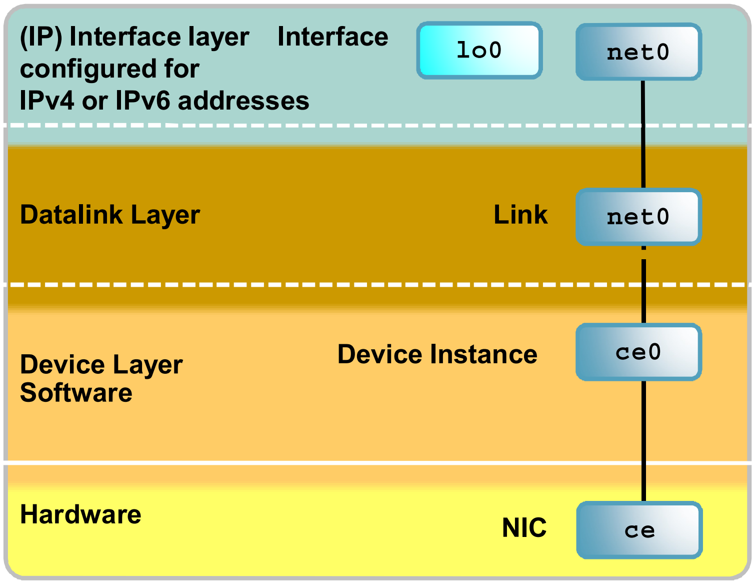
# Lesson Agenda

- Planning for Network Management
- Determining Datalink Availability
- Configuring the Network Interface
- Administering the Network Interface
- Verifying Network Operation

ORACLE

# Determining Datalink Availability

- Determining the physical links that are available
- Determining the datalinks that are available
- Verifying that the network service is running



As was discussed in the first section, there is a one-to-one relationship between the network device, its datalink, and the IP interface, with the datalink providing the connection between the network device and the IP interface. This means that if for some reason the datalink goes down, the connection between the network device and the IP interface is broken. Therefore, knowing how to determine the datalink and the device that it is associated with it is important.

In the slides that follow, you learn to determine the physical links that are available and the datalinks that are available. You also learn to verify that the network service is running.

## Determining Which Physical Links Are Available

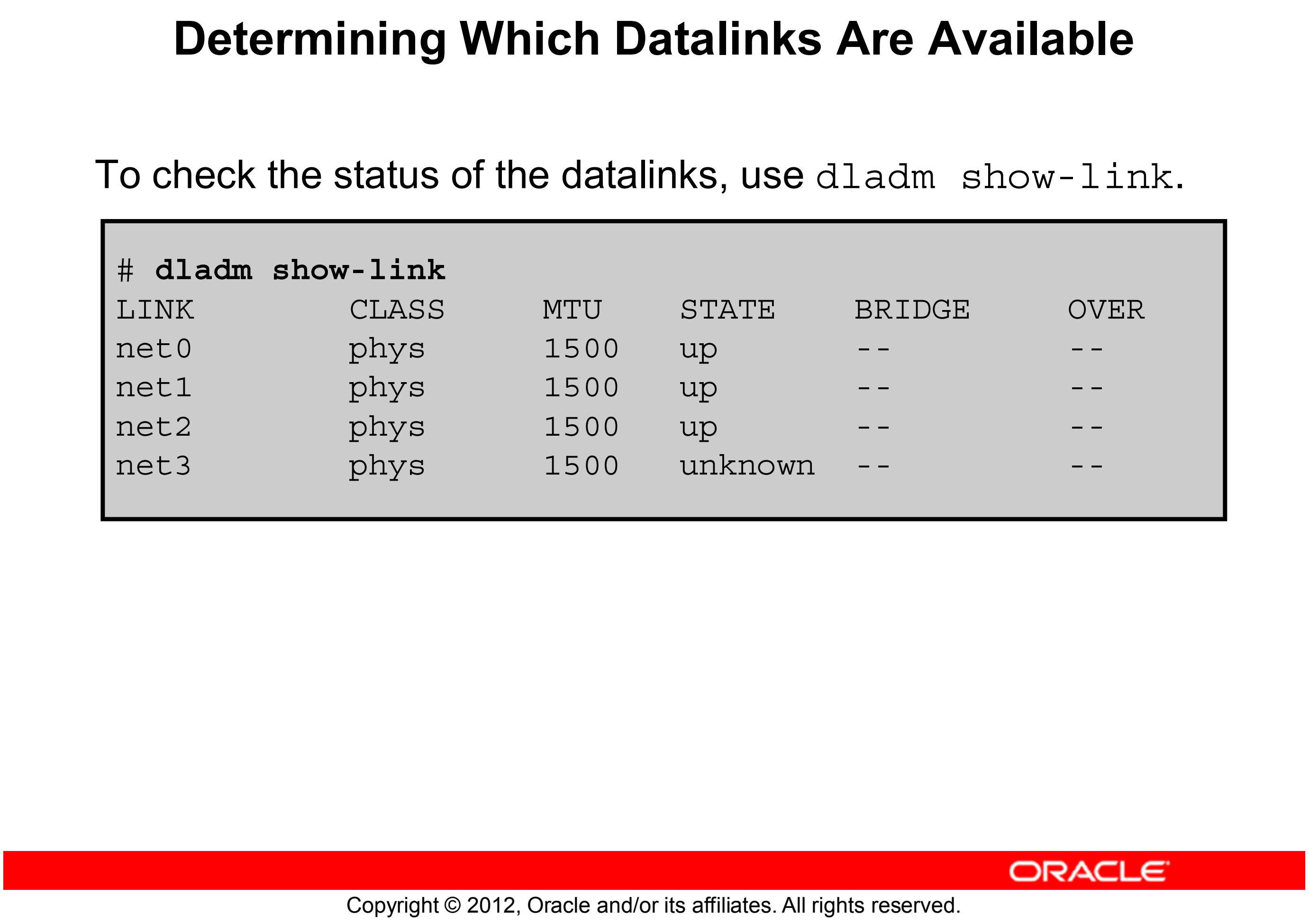
To display information about the physical attributes of datalinks, use `dladm show-phys`.

```
# dladm show-phys
LINK         MEDIA        STATE      SPEED  DUPLEX    DEVICE
net1         Ethernet     up         1000   full      e1000g1
net2         Ethernet     up         1000   full      e1000g2
net0         Ethernet     up         1000   full      e1000g0
net3         Ethernet     unknown    0      unknown   e1000g3
```

To display information about the physical attributes of the datalinks that are currently on the system, use the `dladm show-phys` command, as shown in the example in the slide. This command shows the physical network cards that are installed on your system and some of their properties. In addition to the name of the datalink and the media type, this view displays the state of the link (either `up`, `down`, or `unknown`), the current speed of the link in megabits per second, the full/half duplex status of the link, and the name of the physical device under the link.

In the example, you have four physical links available: `net0` through `net3`.

**Note:** The `dladm` command is used to administer datalinks. Each datalink relies on either a single network device or an aggregation of devices to send packets to or receive packets from a network.



# Determining Which Datalinks Are Available

To check the status of the datalinks, use `dladm show-link`.

```
# dladm show-link
LINK         CLASS     MTU    STATE    BRIDGE     OVER
net0         phys      1500   up       --         --
net1         phys      1500   up       --         --
net2         phys      1500   up       --         --
net3         phys      1500   unknown  --         --
```

ORACLE

To determine the datalinks that are currently available on the system, use the `dladm show-link` command, as shown in the example in the slide.

By default, the system is configured with one datalink for each known network device. The output displays the following:

- Name of the datalink (`LINK`)
- The class of the datalink (`CLASS`). The classes include `aggr` for an IEEE 802.3ad link aggregation, part for an IP-over-IB interface, `phys` for physical datalink, `vlan` for a VLAN datalink, and `vnic` for a virtual network interface.
- Maximum transmission unit size for the datalink being displayed (`MTU`)
- State of the datalink (`STATE`). The state can be `up`, `down`, or `unknown`.

- Name of the bridge to which this link is assigned, if any (`BRIDGE`)
- The physical datalinks over which the datalink operates (`OVER`). This applies to `aggr`, `bridge`, and `vlan` and `part` partition classes of datalinks.

In the example, you have four physical datalinks, all with a maximum transmission unit size of 1500 bytes. Three of the four datalinks are up and one is unknown.

# Verifying That the Network Service Is Running

To verify that the network service is running, use `svcs network/physical`.

```
# svcs network/physical
disabled       17:54:37 svc:/network/physical:nwam
online         17:54:56 svc:/network/physical:upgrade
online         17:54:56 svc:/network/physical:default
```

To verify that the network service is running, use the `svcs network/physical` command, as shown in the example in the slide. Here you can see that the default instance is running. The service assigns an IPv4 or IPv6 address on the local system for each IPv4 or IPv6 interface.

# Lesson Agenda

- Planning for Network Management
- Determining Datalink Availability
- Configuring a Network Interface
- Administering a Network Interface
- Verifying Network Operation

# Configuring a Network Interface

- Displaying network interface configuration information
- Displaying network interface IP address information
- Creating a network interface
- Assigning an IP address to the network interface
- Verifying the IP address assignment

Before you configure a new network interface, you must determine the interfaces that are already configured on the system and the IP addresses that have been assigned to them. When you have this information, you can create the network interface, assign an IP address to the interface, and then verify that the IP address assignment has been made. You can use the `ipadm` command with various subcommands to perform each of these tasks. You begin with displaying the current network interface configuration, covering the other tasks subsequently.

# Displaying Network Interface Configuration Information

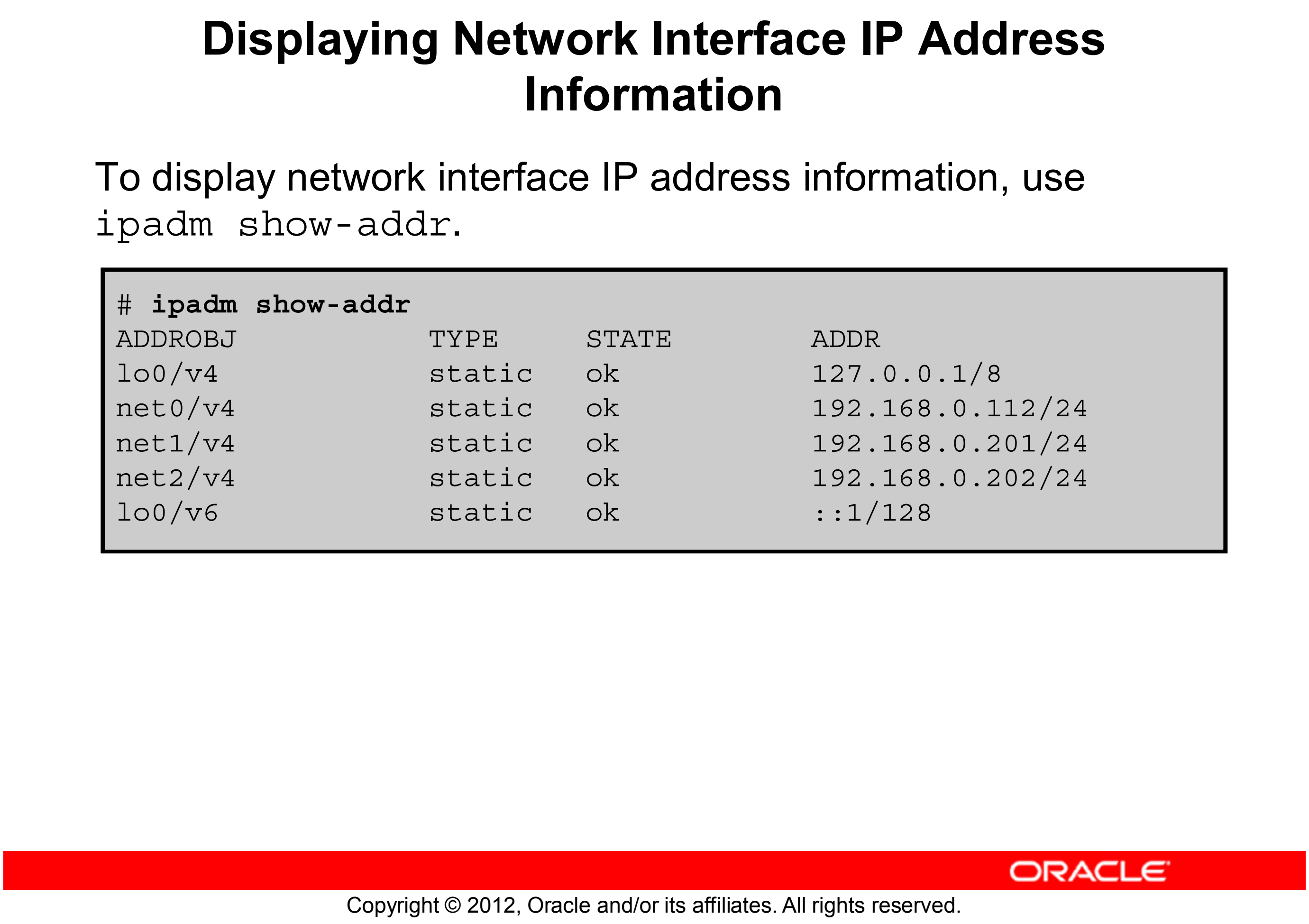
To display information about the current network interface configuration, use `ipadm show-if`.

```
# ipadm show-if
IFNAME     CLASS    STATE    ACTIVE    OVER
lo0        loopback ok       yes       --
net0       ip       ok       yes       --
net1       ip       ok       yes       --
net2       ip       ok       yes       --
```

To display information about the current network interface configuration, use the `ipadm show-if` command, as shown in the example in the slide.

**Note:** The `ipadm` command is used to configure and manage IP network interfaces, addresses, and TCP/IP protocol properties. The `show-if` subcommand displays network interface configuration information, either for all the network interfaces that are configured on the system, including the ones that are only in the persistent configuration, or for the specified network interface.

In this example, you can see that you currently have three network interfaces up and running: `net0`, `net1`, and `net2`.

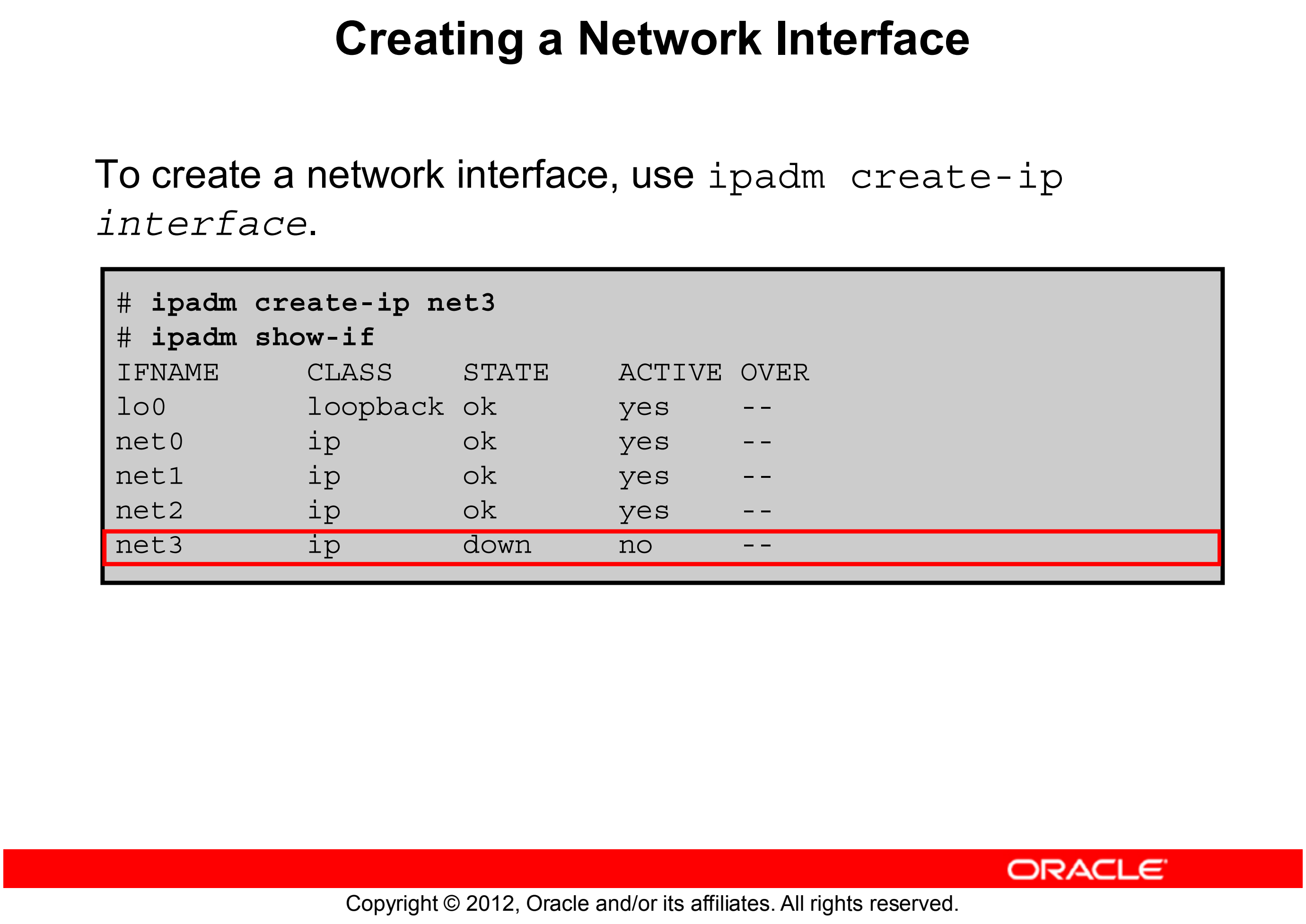


# Displaying Network Interface IP Address Information

To display network interface IP address information, use `ipadm show-addr`.

```
# ipadm show-addr
ADDROBJ              TYPE      STATE         ADDR
lo0/v4               static    ok            127.0.0.1/8
net0/v4              static    ok            192.168.0.112/24
net1/v4              static    ok            192.168.0.201/24
net2/v4              static    ok            192.168.0.202/24
lo0/v6               static    ok            ::1/128
```

To display IP address information for the network interface that is currently configured on the system, use the `ipadm show-addr` command, as shown in the example in the slide.



# Creating a Network Interface

To create a network interface, use `ipadm create-ip` *`interface`*.

```
# ipadm create-ip net3
# ipadm show-if
IFNAME     CLASS    STATE    ACTIVE OVER
lo0        loopback ok       yes    --
net0       ip       ok       yes    --
net1       ip       ok       yes    --
net2       ip       ok       yes    --
net3       ip       down     no     --
```

To create a network interface, use the `ipadm create-ip` command, followed by the network interface name, as shown in the example in the slide. Here you create the network interface `net3`. You then run the `ipadm show-if` command to verify that the network interface has been created. In the example, you can see that the interface exists but it is down. To bring the interface up, you need to assign an IP address to it.

**Note:** The `create-ip` subcommand creates an IP interface that handles both IPv4 and IPv6 packets. The address of the IPv4 interface is set to 0.0.0.0 and the address of the IPv6 interface is set to ::. This subcommand, by default, causes the information to persist, so that on the next reboot, this interface is instantiated.

# Assigning an IP Address to the Network Interface

To assign an IP address to a network interface, use `ipadm create-addr -T address-type -a` *`address/prefixlen addrobj`*.

```
# ipadm create-addr -T static -a 192.168.0.203/24 net3/v4
# ipadm show-if
IFNAME     CLASS    STATE    ACTIVE OVER
lo0        loopback ok       yes    --
net0       ip       ok       yes    --
net1       ip       ok       yes    --
net2       ip       ok       yes    --
net3       ip       ok       yes    --
```

To assign an IP address to the network interface, use the `ipadm create-addr -T` command, followed by the address type, the `-a` option, which specifies the IP address to configure on the interface, the IP address, the prefix length (`prefixlen`), which specifies the length of the network ID (for example, in the address `192.168.0.203/24`, `24` is the prefix length), and the address object (`addrobj`), which specifies an identifier for the unique IP address that is used in the system. The addresses can be either IPv4 or IPv6 types.

**Note:** The `create-addr` subcommand with the `-T static -a` options creates a static IPv4 or IPv6 address on the specified interface. If the interface on which the address is created is not plumbed, this subcommand implicitly plumbs the interface. By default, a configured address is marked `up`, so that it can be used as a source or destination of or for outbound and inbound packets.

In the example, you assign a static IP address with v4 addressing to your new network interface `net3`. You then run the `ipadm show-if` command again to verify that the new network interface is now up—and it is.

# Verifying the IP Address Assignment

To display the network interface IP address assignment, use `ipadm show-addr`.

```
# ipadm show-addr
ADDROBJ            TYPE     STATE        ADDR
lo0/v4             static   ok           127.0.0.1/8
net0/v4            static   ok           192.168.0.112/24
net1/v4            static   ok           192.168.0.201/24
net2/v4            static   ok           192.168.0.202/24
net3/v4            static   ok           192.168.0.203/24
lo0/v6             static   ok           ::1/128
```

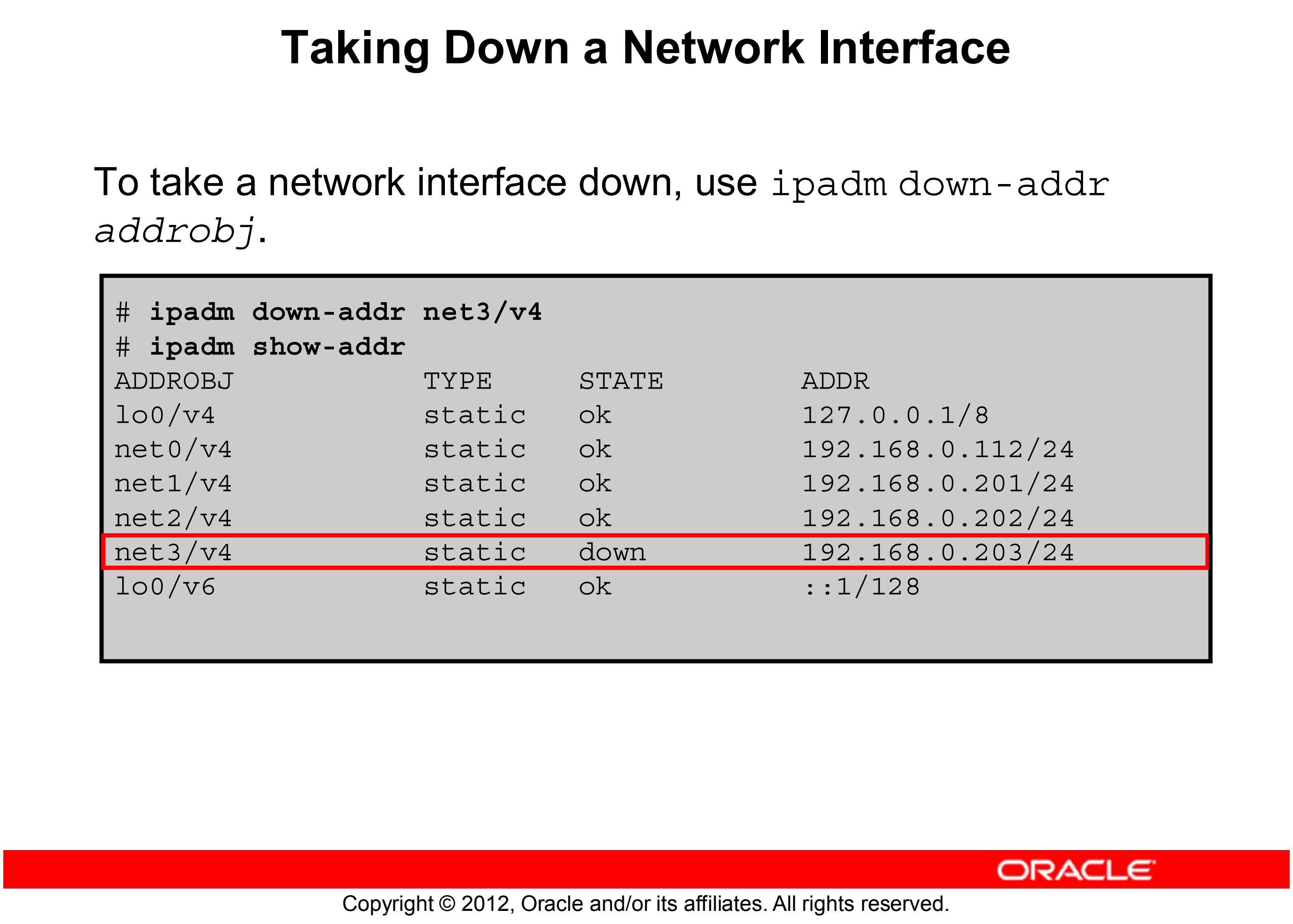
The last step that you need to perform in your network interface configuration is to verify the IP address assignment. You do this by using the `ipadm show-addr` command, as shown in the example. You can see that the IP address assignment that you made to the new network interface is displayed as shown in the example.

# Lesson Agenda

- Planning for Network Management
- Determining Datalink Availability
- Configuring a Network Interface
- Administering a Network Interface
- Verifying Network Operation

# Administering a Network Interface

- Taking down a network interface
- Bringing up a network interface
- Deleting an IP address for a network interface
- Deleting a network interface

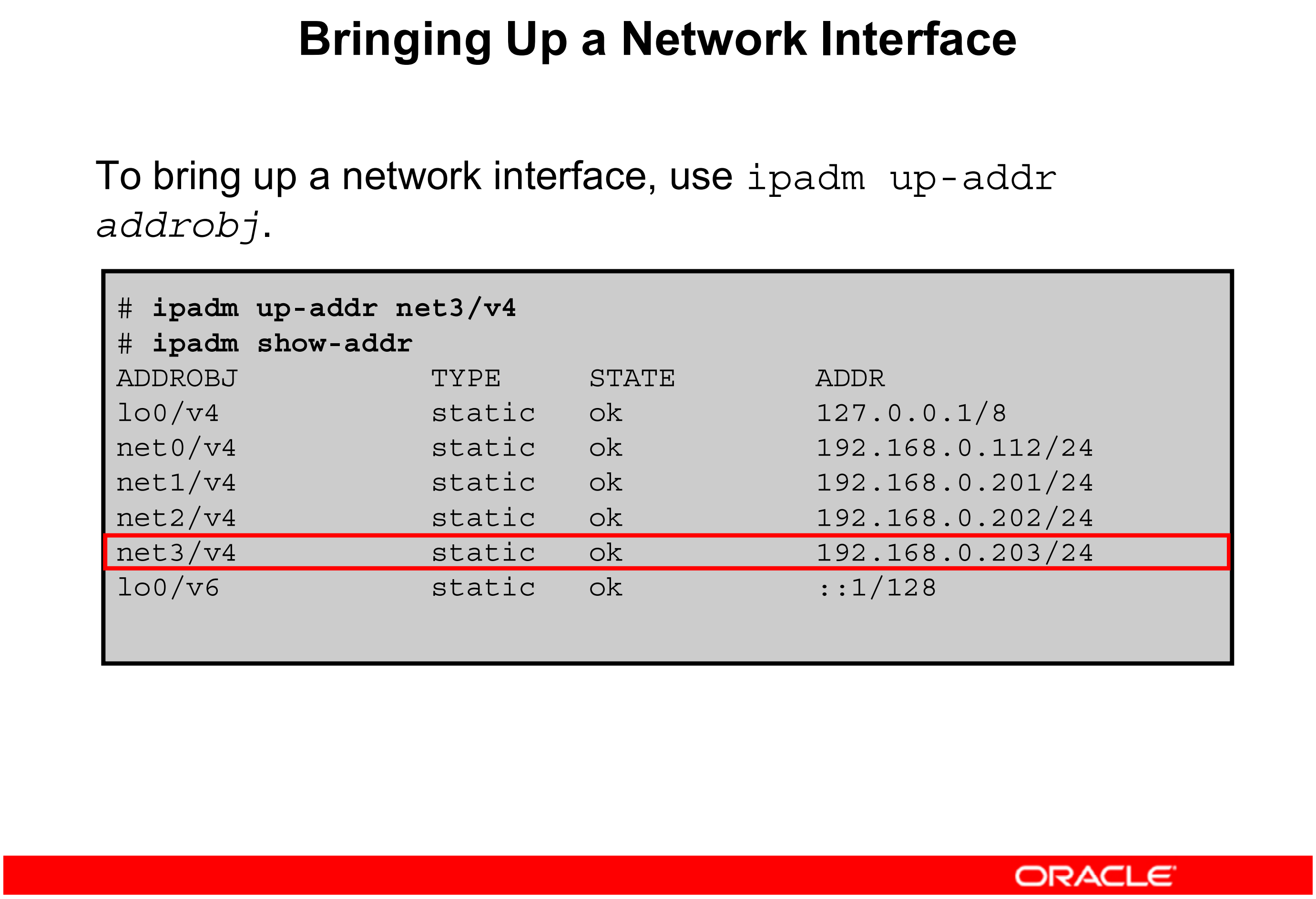


# Taking Down a Network Interface

To take a network interface down, use `ipadm down-addr` *`addrobj`*.

```
# ipadm down-addr net3/v4
# ipadm show-addr
ADDROBJ            TYPE     STATE         ADDR
lo0/v4             static   ok            127.0.0.1/8
net0/v4            static   ok            192.168.0.112/24
net1/v4            static   ok            192.168.0.201/24
net2/v4            static   ok            192.168.0.202/24
net3/v4            static   down          192.168.0.203/24
lo0/v6             static   ok            ::1/128
```

To take a network interface out of service, use the `ipadm down-addr` command, followed by the address object (`addrobj`), as shown in the example in the slide. In the example, you take down the network interface `net3`. You then run the `ipadm show-addr` command to verify that the network interface has been brought down.

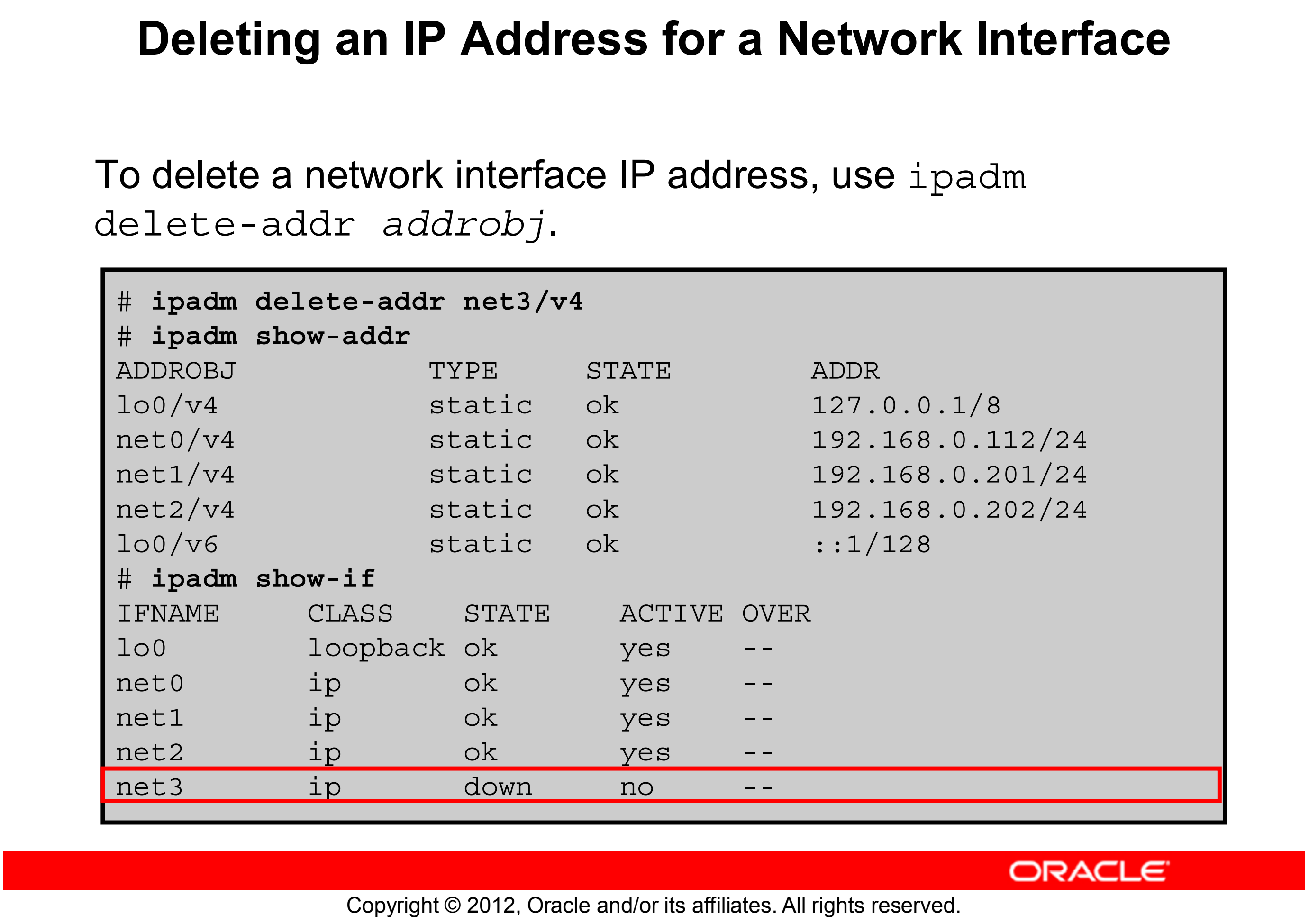


# Bringing Up a Network Interface

To bring up a network interface, use `ipadm up-addr` *`addrobj`*.

```
# ipadm up-addr net3/v4
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4           static   ok           192.168.0.112/24
net1/v4           static   ok           192.168.0.201/24
net2/v4           static   ok           192.168.0.202/24
net3/v4           static   ok           192.168.0.203/24
lo0/v6            static   ok           ::1/128
```

To bring a network interface up, use the `ipadm up-addr` command, followed by the address object (`addrobj`), as shown in the example in the slide. In the example, you bring the network interface `net3` back up. You then run the `ipadm show-addr` command to verify that the network interface has been brought back up.



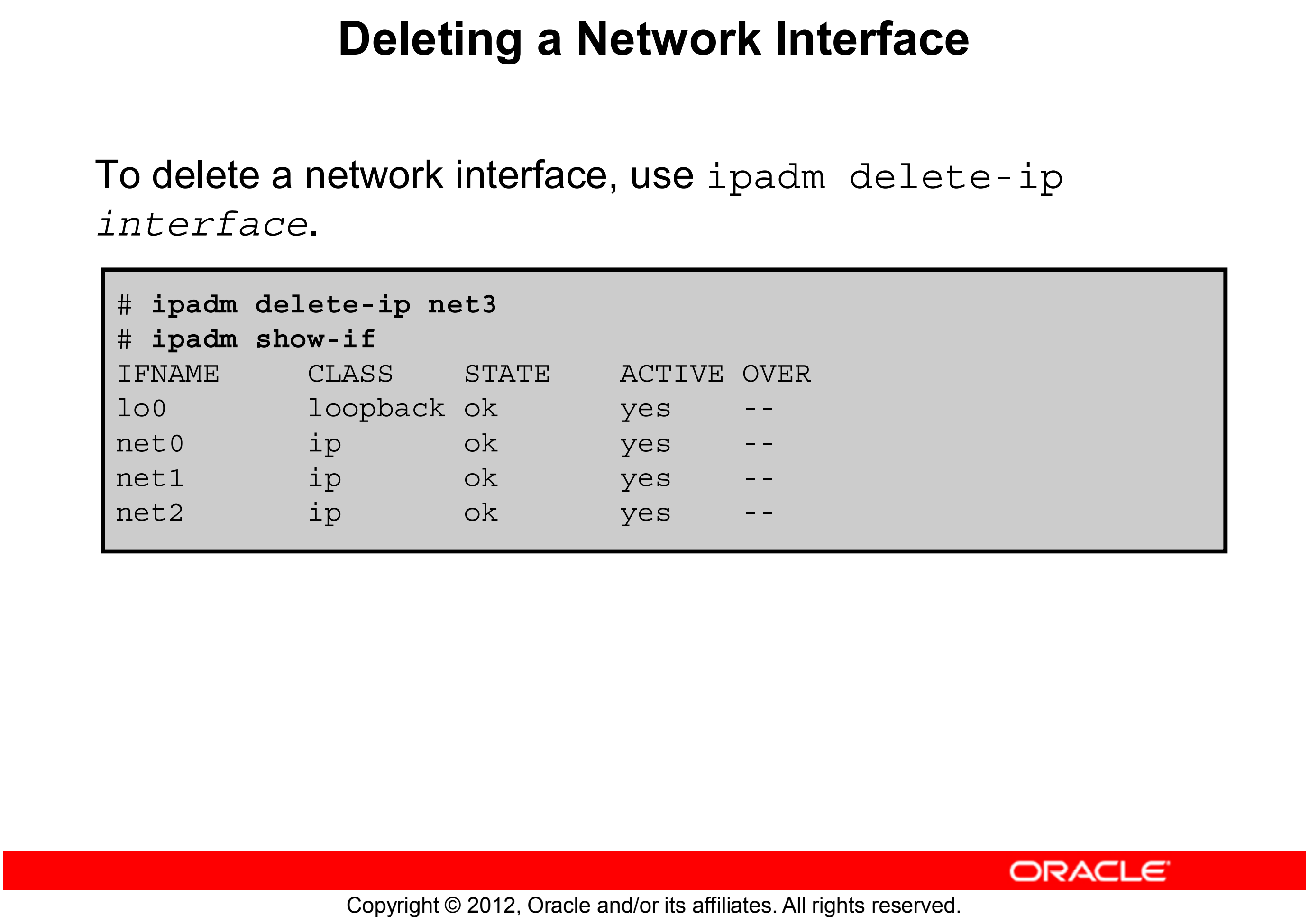
# Deleting an IP Address for a Network Interface

To delete a network interface IP address, use `ipadm delete-addr` *`addrobj`*.

```
# ipadm delete-addr net3/v4
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4           static   ok           192.168.0.112/24
net1/v4           static   ok           192.168.0.201/24
net2/v4           static   ok           192.168.0.202/24
lo0/v6            static   ok           ::1/128
# ipadm show-if
IFNAME     CLASS    STATE    ACTIVE OVER
lo0        loopback ok       yes    --
net0       ip       ok       yes    --
net1       ip       ok       yes    --
net2       ip       ok       yes    --
net3       ip       down     no     --
```

To delete an IP address that is assigned to a network interface, use the `ipadm delete-addr` command, followed by the address object (`addrobj`), as shown in the example in the slide. In the example, you delete the IP address for the network interface `net3`. You then run the `ipadm show-addr` command to verify that the IP address has been deleted—and it is.

If you run the `ipadm show-if` command (as shown in the example), you can see that the network interface exists but that it is now down.



## Deleting a Network Interface

To delete a network interface, use `ipadm delete-ip` *`interface`*.

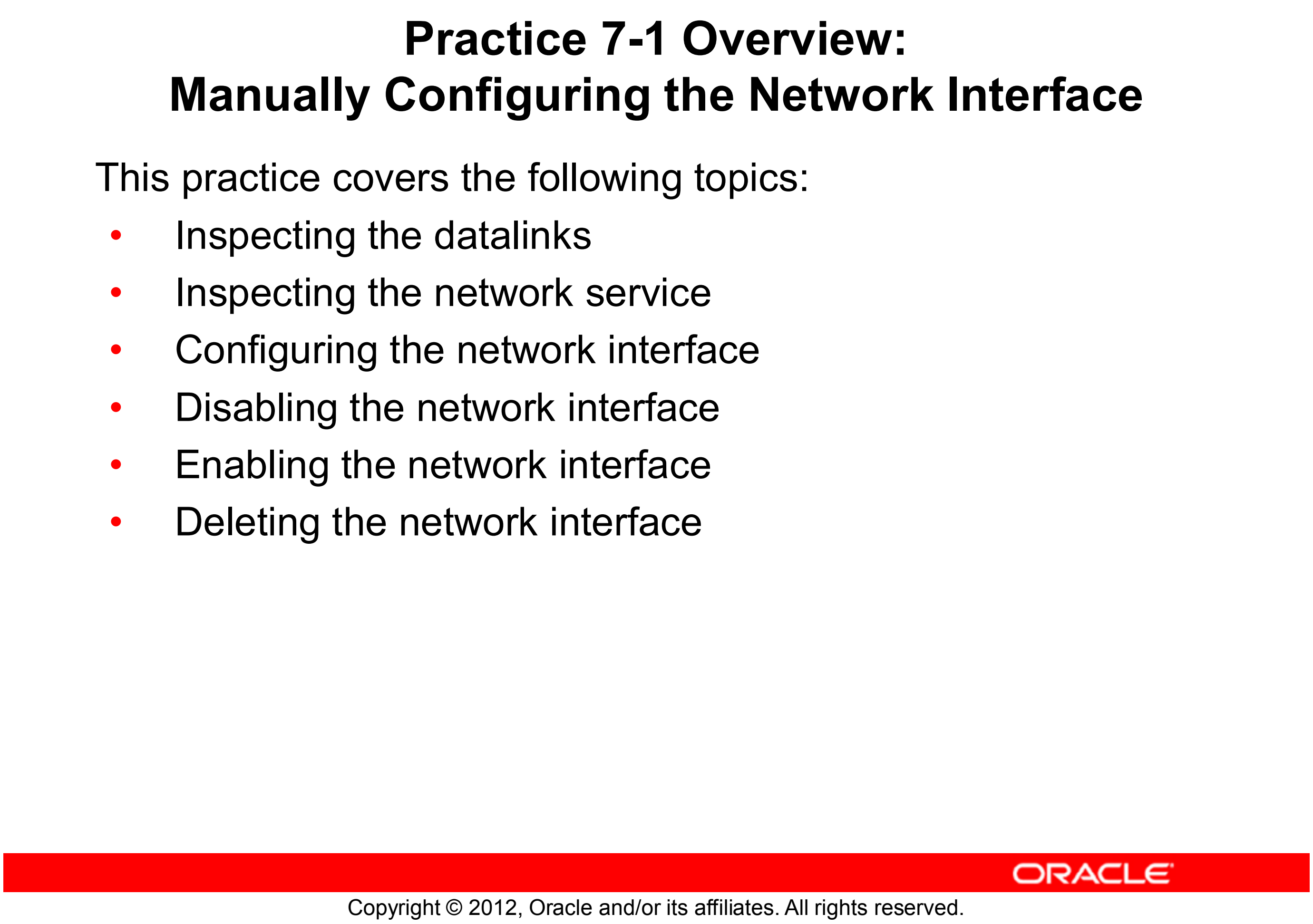
```
# ipadm delete-ip net3
# ipadm show-if
IFNAME      CLASS    STATE    ACTIVE OVER
lo0         loopback ok       yes    --
net0        ip       ok       yes    --
net1        ip       ok       yes    --
net2        ip       ok       yes    --
```

To delete the network interface itself, use the `ipadm delete-ip` command, followed by the network interface name, as shown in the example in the slide. In the example, you delete the network interface `net3`. You then run the `ipadm show-if` command to verify that the network interface has been deleted—and it is.

# Summary of `ipadm` Commands

| Network Interface Task | `ipadm` Command |
|---|---|
| Display network interface information. | `ipadm show-if` |
| Display IP address assignments to network interfaces. | `ipadm show-addr` |
| Create a network interface. | `ipadm create-ip` *`interface`* |
| Assign a static IP address to a network interface. | `ipadm create-addr -T` *`address-type`* `-a` *`address/prefixlen addrobj`* |
| Take down a network interface. | `ipadm down-addr` *`addrobj`* |
| Bring up a network interface. | `ipadm up-addr` *`addrobj`* |
| Delete an IP address assigned to a network interface. | `ipadm delete-addr` *`addrobj`* |
| Delete a network interface. | `ipadm delete-ip` *`interface`* |

The table in the slide contains a summary of the `ipadm` commands that were described in the preceding section.



# Practice 7-1 Overview: Manually Configuring the Network Interface

This practice covers the following topics:

- Inspecting the datalinks
- Inspecting the network service
- Configuring the network interface
- Disabling the network interface
- Enabling the network interface
- Deleting the network interface

In the practices for this lesson, you will have the chance to perform the following tasks:

- **Practice 7-1:** Manually configuring the network interface
- **Practice 7-2:** Verifying network operation

You will find Practice 7-1 in your *Activity Guide*. It should take about 50 minutes to complete the practice.

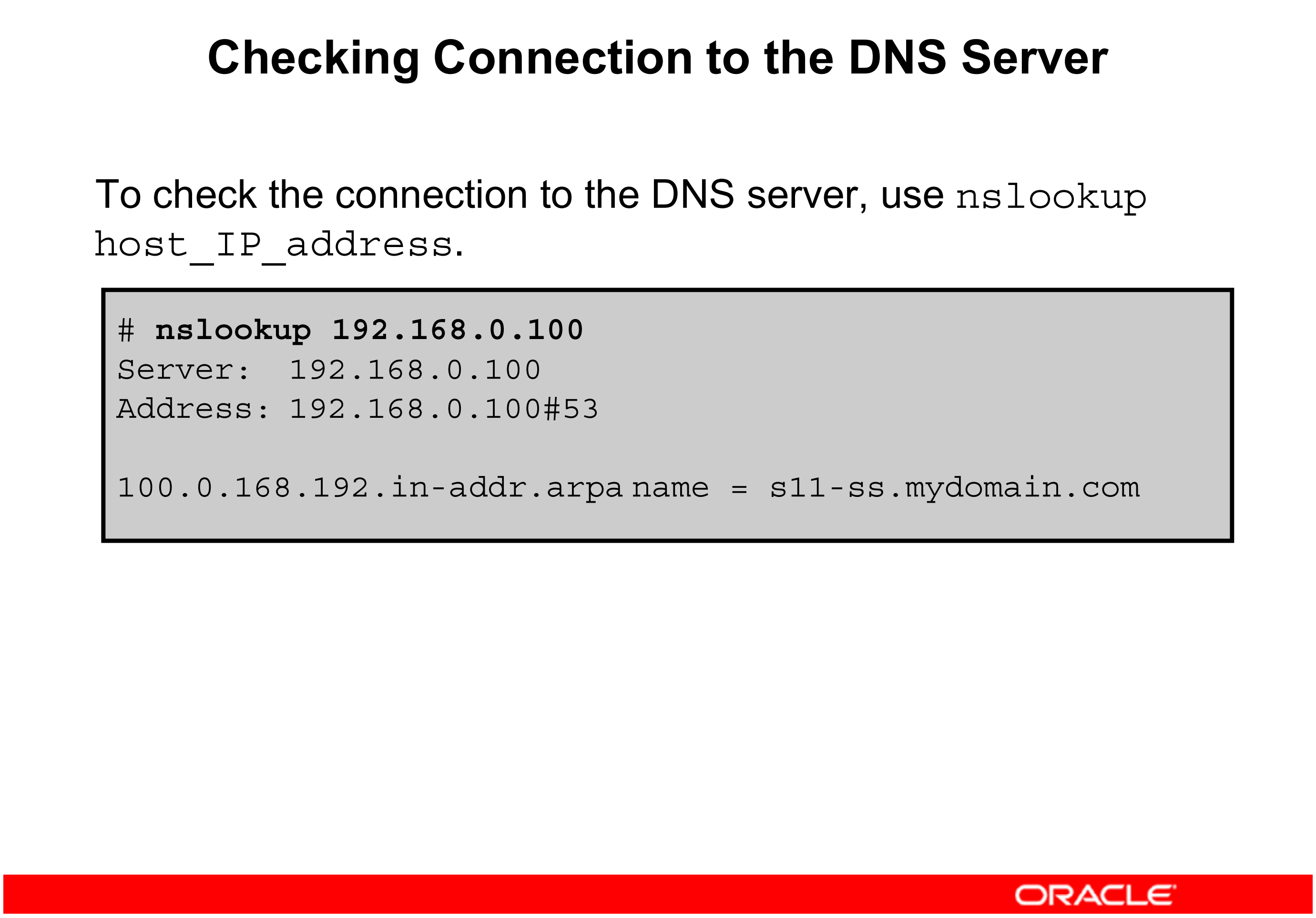
# Lesson Agenda

- Planning for Network Management
- Determining Datalink Availability
- Configuring the Network Interface
- Administering the Network Interface
- Verifying Network Operation

ORACLE

# Verifying Network Operation

- Checking connection to DNS
- Examining the status of all network interfaces
- Checking network connectivity and response times
- Checking network interface traffic status



# Checking Connection to the DNS Server

To check the connection to the DNS server, use `nslookup host_IP_address`.

```
# nslookup 192.168.0.100
Server:  192.168.0.100
Address: 192.168.0.100#53

100.0.168.192.in-addr.arpa name = s11-ss.mydomain.com
```

As was discussed earlier, DNS is the name service provided by the Internet for TCP/IP networks and provides host names to the IP address service.

To check the connection between your system and the DNS server, use the `nslookup` command, followed by the host system's IP address, as shown in the example in the slide. Because a connection exists between the host and the server, DNS returns the server and address information. In this environment, the DNS server is defined on server `s11-ss`.

**Note:** The `nslookup` utility is a program to query Internet domain name servers. It has two modes: interactive and non-interactive. The interactive mode allows users to query name servers for information about various hosts and domains or to print a list of hosts in a domain. The non-interactive mode is used to print only the name and the requested information for a host or domain.

# Examining the Status of All Network Interfaces

To display all the network interfaces, their IP addresses, and status, use `ipadm show-addr`.

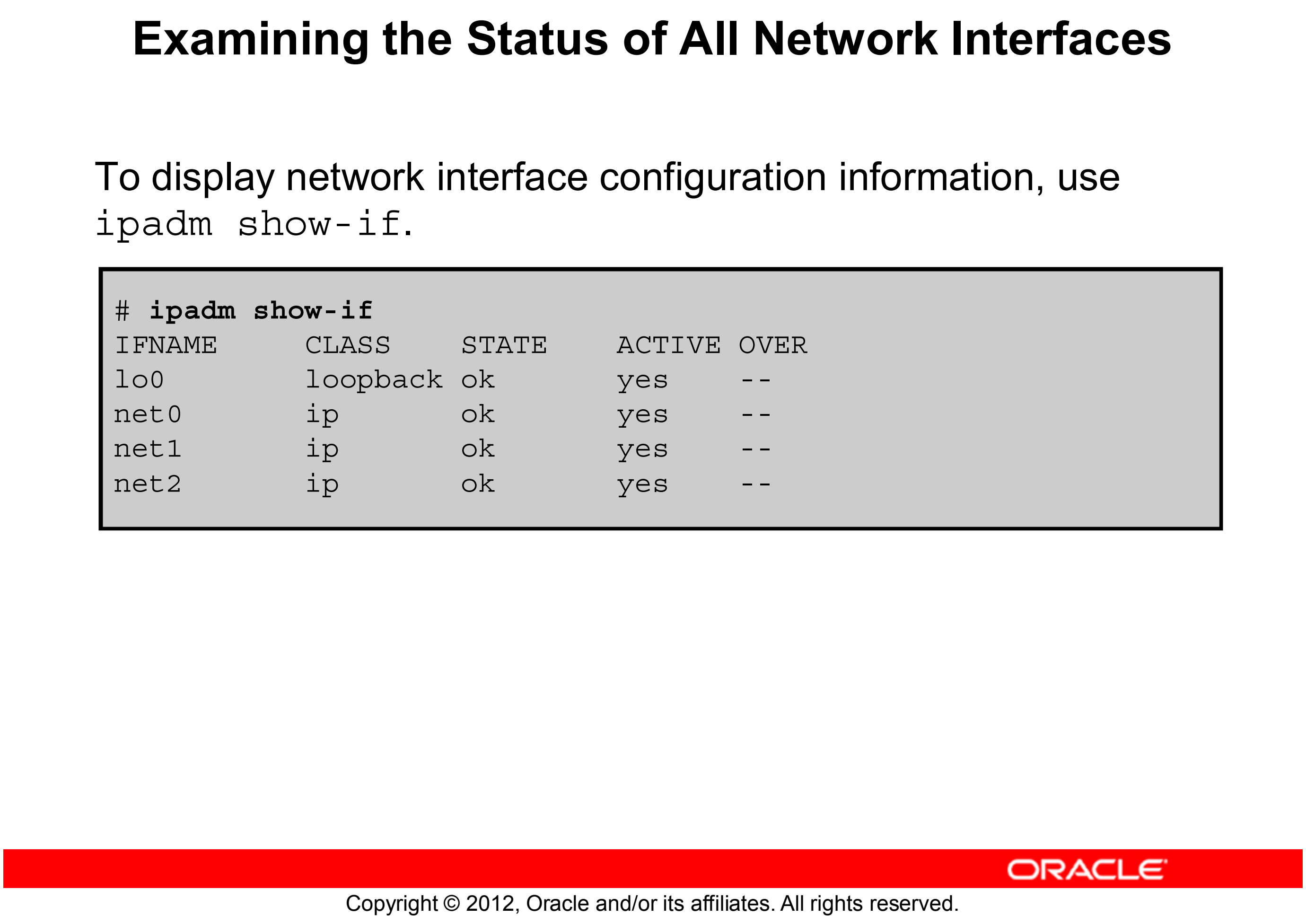
```
# ipadm show-addr
ADDROBJ            TYPE     STATE        ADDR
lo0/v4             static   ok           127.0.0.1/8
net0/v4            static   ok           192.168.0.111/24
net1/v4            static   ok           192.168.0.101/24
net2/v4            static   ok           192.168.0.202/24
lo0/v6             static   ok           ::1/128
```

After you have established that your system is connected to the DNS server, you can examine the status of your network interfaces. To display all the network interfaces, their IP addresses, and status, use the `ipadm show-addr` command, as shown in the example in the slide. The output for this command displays the following information:

- **`ADDROBJ`:** Name of the address object or interface
- **`TYPE`:** Type of the address object. It is `static` (which is a “permanent” address that is associated with an interface specified by the address object), `dhcp` (which is a DHCP-controlled IPv4 address on an interface specified by the address object), `addrconf` (which is an auto-configured IPv6 address on an interface specified by the address object), or `from-gz`. The `from-gz` type is displayed only in non-global zones and indicates that the address was configured based on the allowed-address property configured for the non-global exclusive-IP zone from the global zone. A static IP address is a “permanent” address that is associated with a single interface as opposed to a dynamic IP address, which can change from one time to the next.

- **STATE:** State of the address object. The state can be one of the following:
  - **ok:** Indicates that the address is enabled, up, and functioning properly. The system will accept IP packets destined to this address, and will originate IP packets with this address in accordance with the configured IP source address selection policy.
  - **down:** Indicates that the address is administratively down
  - **duplicate:** Indicates that the address was found to conflict with another system's IP address by duplicate address detection (DAD) and cannot be used until the conflict is resolved
  - **tentative:** Indicates that the address is currently undergoing duplicate address detection
  - **inaccessible:** Indicates that the address cannot be used because the IP interface that it is configured on has failed
  - **Disabled:** Indicates that the address is not part of the active configuration

In the example, you again see the IPv4 and IPv6 loopback interfaces and the three `netX` interfaces. All the interfaces have been configured with static IP addresses and all the addresses for these interfaces are enabled, up, and functioning properly as indicated by the `ok` status. Notice the address format of the three `netX` interfaces. All these are examples of the IPv4 address Classless Inter-Domain Routing (CIDR) format that was discussed earlier.



## Examining the Status of All Network Interfaces

To display network interface configuration information, use `ipadm show-if`.

```
# ipadm show-if
IFNAME     CLASS    STATE    ACTIVE OVER
lo0        loopback ok       yes    --
net0       ip       ok       yes    --
net1       ip       ok       yes    --
net2       ip       ok       yes    --
```

To display network interface configuration information for the entire network or for a specified network interface, use the `ipadm show-if` command, as shown in the example in the slide. The output for this command displays the following information:

- **`IFNAME`:** Name of the IP interface
- **`CLASS`:** Type of network interface. For example, `loopback` for a loopback interface, `ip` for an interface that is plumbed over an underlying datalink, `ipmp` for an IPMP interface that is created over one or more underlying IP interfaces, or `vni` for a virtual IP interface.
- **`STATE`:** State of the interface. Options include:
  - **`ok`:** Indicates that the required resources for an interface are allocated
  - **`offline`:** Indicates that the interface is offline and thus cannot send or receive IP data traffic
  - **`failed`:** Indicates that the datalink is down

  - **down:** Indicates that the interface is administratively down, preventing any IP packets from being sent or received through it
  - **disabled:** Indicates that the interface has been disabled from the active configuration by using the disable-if subcommand
- **ACTIVE:** Indicates the state of the configuration. You see either yes or no, depending on whether the IP interface is used by the system for IP data traffic.
- **OVER:** Indicates the underlying interfaces over which a link aggregation or IPMP interface is created. This field does not apply to other interface classes.

# Checking Network Connectivity and Response Times

To check the connectivity between one host and another, use `ping`.

```
# ping -s 192.168.0.112
PING 192.168.0.112: 56 data bytes
64 bytes from s11-serv1.mydomain.com (192.168.0.112):
   icmp_seq=0. time=1.143 ms
64 bytes from s11-serv1.mydomain.com (192.168.0.112):
   icmp_seq=1. time=0.724 ms
64 bytes from s11-serv1.mydomain.com (192.168.0.112):
   icmp_seq=2. time=1.639 ms
^C
----192.168.0.112 PING Statistics----
3 packets transmitted, 3 packets received, 0% packet
   loss
```

To check the connectivity between one host and another, you can use the `ping -s` command followed by the IP address, as shown in the example in the slide, or a host name. The `-s` option tells the sending host to send one datagram per second and collect statistics.

As you can see in the output, the receiving host received three data packets, thereby confirming that there is a working connection between these two hosts.

**Note:** You can press Ctrl + C to stop the continuous display.

# Checking Network Interface Traffic Status

To check network traffic on the network interface, use `netstat -I interface` *`interval count`*.

```
# netstat -I net0 -i 5
    input   net0      output        input  (Total)    output
packets errs  packets errs  colls  packets errs  packets errs  colls
582     0     69      0     0      2732    0     1364    0     0
0       0     0       0     0      0       0     0       0     0
0       0     0       0     0      1       0     2       0     0
1       0     0       0     0      5       0     1       0     0
0       0     0       0     0      0       0     0       0     0
0       0     0       0     0      0       0     0       0     0
^C
```

To check the status of traffic on a network interface, use the `netstat` command, followed by the `-I` option to specify the interface, the interface (for example, `net0`), the interval at which you want the interface statistics displayed (this is optional), and the number of times per second (`count`) that you want the interface statistics displayed.

If an optional interval is specified, the output continues to display in interval seconds until interrupted by the user.

In the example, you check network traffic for the network interface `net0` at an interval of five seconds.

The output displays the number of input packets, input errors, output packets, output errors, and collisions, respectively, and is divided into two sections. On the left, you have the current traffic statics for the interface. On the right, you have the total number of packets sent and received by this interface.

## Quiz

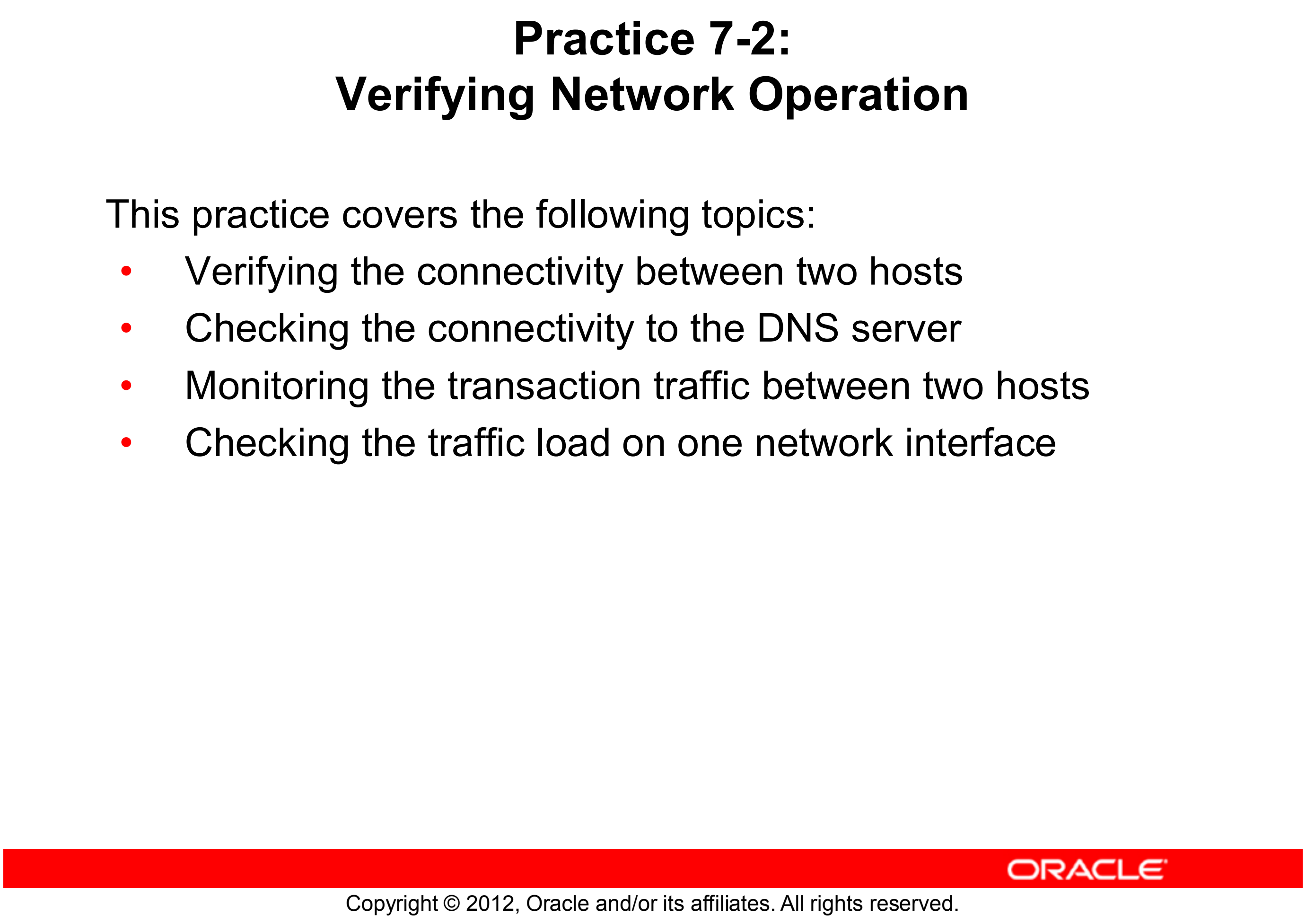
Which command can you use to display your system's current network interface configuration?

a. `ipadm`

b. `ping`

c. `netstat -I`

**Answer: a**



# Practice 7-2: Verifying Network Operation

This practice covers the following topics:

- Verifying the connectivity between two hosts
- Checking the connectivity to the DNS server
- Monitoring the transaction traffic between two hosts
- Checking the traffic load on one network interface

This practice should take about 15 minutes to complete.

# Summary

In this lesson, you should have learned to:

- Implement a plan for network management
- Determine datalink availability
- Configure a network interface
- Administer a network interface
- Verify network operation

In this lesson, you were introduced to the physical network. You learned to determine datalink availability, and configure and administer a network interface. You also learned to verify that the network is operational.

# 8

# Setting Up and Administering User Accounts

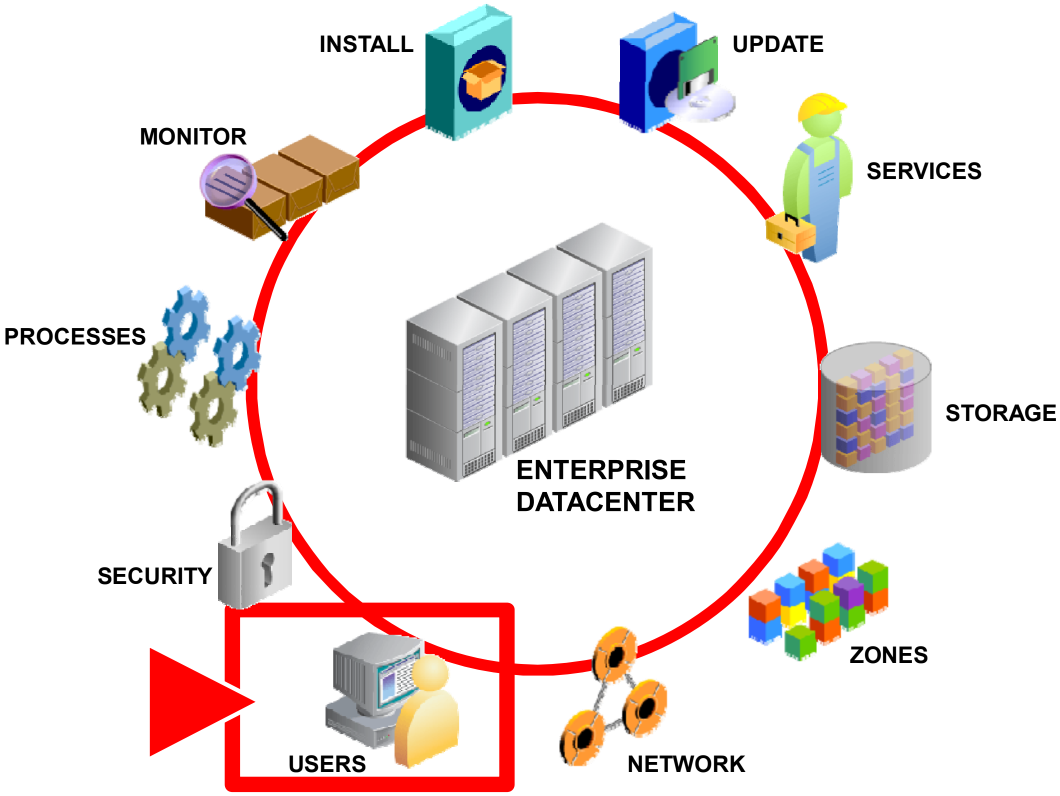
# Objectives

After completing this lesson, you should be able to:

- Implement a plan for user administration
- Set up user accounts
- Manage user accounts
- Manage user initialization files
- Use shell metacharacters
- Configure user disk quotas

In this lesson, you are presented with a plan for administering the user accounts in your company. You learn to set up and manage user accounts and initialization files. You also learn to use shell metacharacters and configure user disk quotas.

# Workflow Orientation



Before you start the lesson, orient yourself to where you are in the job workflow. You have successfully installed and updated the operating system; the SMF services are running as expected; you have set up the data storage environment, and have ensured that the zones have been set up correctly. You just finished administering the physical network's interfaces and datalinks and are now ready to set up and administer the user accounts for your company.

A data center not only includes storage, applications, and a network, it also includes users who can perform business functions. For these users to manage their business applications, they must be recognized by the system and must have appropriate access and privileges that are required to execute certain business functions. Setting up and administering these user accounts is the responsibility of the system administrator.

# Lesson Agenda

- Planning for User Administration
- Setting Up User Accounts
- Maintaining User Accounts
- Managing User Initialization Files
- Using Shell Metacharacters and Configuring User Disk Quotas

# Planning for User Administration

The user administration plan addresses the requirements for supporting the needs of the user community.

- Setting up new accounts
- Maintaining accounts
- Providing access to the system and system resources

Your company has put together a plan for user administration after Oracle Solaris 11 has been implemented. The plan includes a comprehensive approach to managing users and groups that includes setting up new accounts, maintaining accounts, and ensuring that the users have access to the system and system resources that they need to do their work.

In the slides that follow, you are introduced to one of the system administrator’s most important tasks: user administration.

# Types of User Accounts

A user can have the following types of accounts:

- **User:** An individual account that provides a user with a unique account name, a user identification (UID) number, a home directory, and a login shell
- **Group:** A collection of individual users that have a shared set of permissions on files and other system resources
- **Role:** A special account that can be assigned to one or more users and that provides a set of functions and permissions that are specific to the role

A user can be identified in the system in multiple ways: as an individual user, as a member of a group, or by a function or role.

A user account is a login account. It provides an individual with a unique account name, a user identification (UID) number, a home directory, and a login shell. This account cannot administer the system.

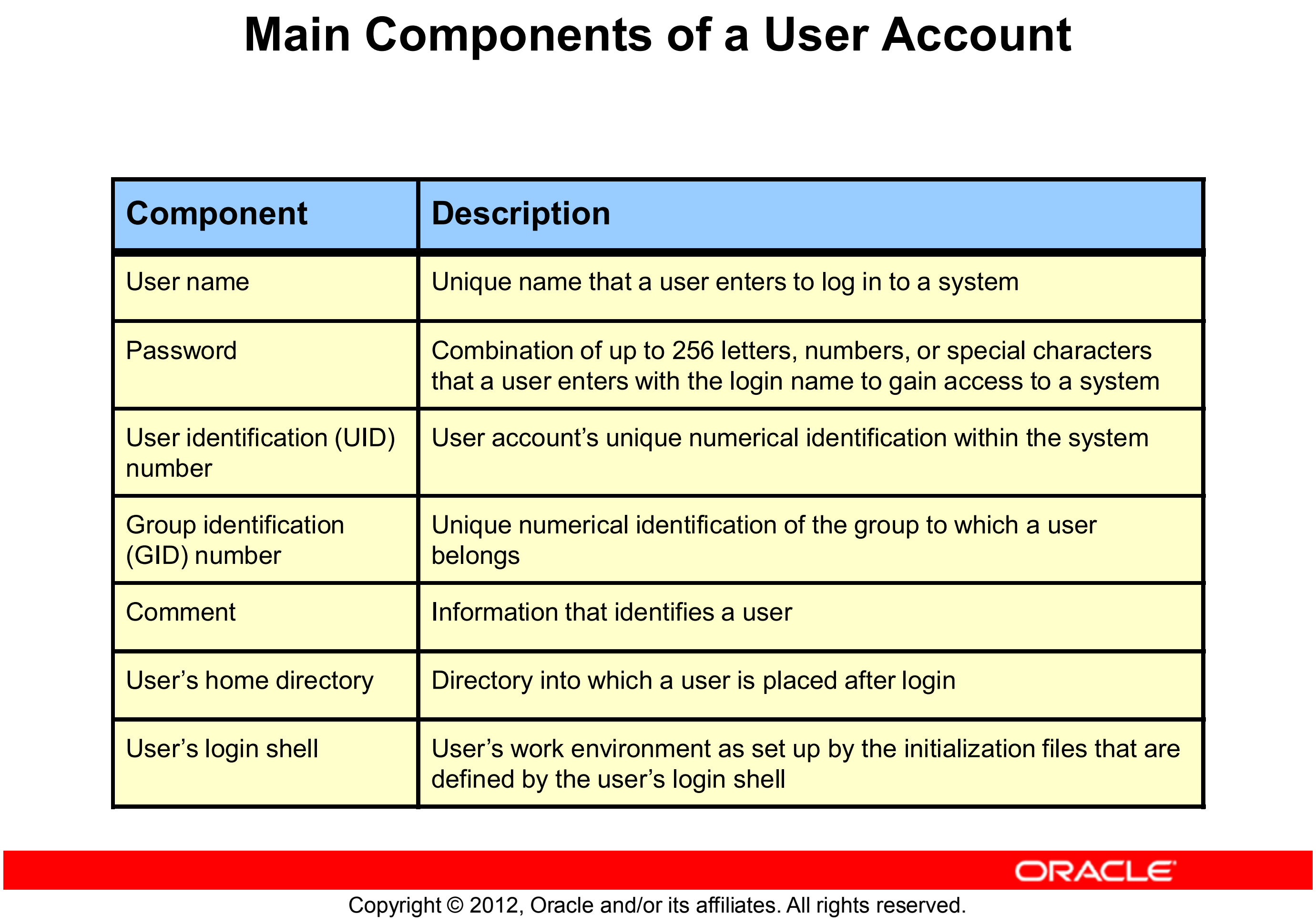
The group account is a collection of individual users. A user must be a member of a primary group and can belong to multiple secondary groups. A typical use of groups is to set up group permissions on files or other system resources, which allows access only to those users who are part of that group.

A role is a special account that can be assigned one or more user accounts. Each role has a defined set of functions and associated permissions that users who have been assigned to the role can perform. A role is not a login account. For example, to assume the `root` role, you would have to first log in by using your user account login name, and then use the `su - root` command to assume the `root` role. A user can assume only those roles that are assigned to the user's login account.

**Note:** The `su` command allows you to become another user without logging off, or to assume a role. The default username is `root` (superuser). For more information about the `su` command, see the `su`(1M) man page. A related command is the `sudo` (`superuser do`) command. This command allows a permitted user to execute a command as the superuser (`sudo su`) or another user for a limited time. The system tracks and logs the actions of the `sudo` command.

**Note:** In the default Oracle Solaris system configuration, the user account that is created during installation is assigned the root role if the text installation method is used. This is referred to as “root as a role.” However, if the text installation method is not used, `root` is set up as an account rather than a role.

In this course, you focus on user accounts and groups. The *Oracle Solaris 11 Advanced System Administration* course covers the use of roles, privileges, and role-based access control (RBAC) in detail.



# Main Components of a User Account

| Component | Description |
|---|---|
| User name | Unique name that a user enters to log in to a system |
| Password | Combination of up to 256 letters, numbers, or special characters that a user enters with the login name to gain access to a system |
| User identification (UID) number | User account’s unique numerical identification within the system |
| Group identification (GID) number | Unique numerical identification of the group to which a user belongs |
| Comment | Information that identifies a user |
| User’s home directory | Directory into which a user is placed after login |
| User’s login shell | User’s work environment as set up by the initialization files that are defined by the user’s login shell |

The user accounts that you will be setting up consists of the following components:

- **Username:** A unique name that a user enters to log in to a system. The username is also called the login name. Usernames allow users to access their own systems and remote systems that have the appropriate access privileges. You must choose a username for each user account that you create.
- **Password:** A combination of up to 256 letters, numbers, or special characters that a user enters with the login name to gain access to a system. You can specify a password for a user when you add the user, or you can force the user to specify a password when the user first logs in.
- **User identification (UID) number:** A user account’s unique numerical identification within the system. The UID number identifies the username to any system on which the user attempts to log in. The UID number is also used by systems to identify the owners of files and directories. If you create user accounts for a single individual on a number of different systems, always use the same username and ID number. That way, the user can easily move files between systems without ownership problems.

- **Group identification (GID) number:** A unique numerical identification of the group to which the user belongs. A group is a collection of users who can share files and other system resources. For example, users who are working on the same project can form a group. Each group must have a name, a group identification (GID) number, and a list of usernames that belong to the group. A GID number identifies the group internally to the system. The two types of groups that a user can belong to are as follows:
  - **Primary group:** Specifies a group that the operating system assigns to files that are created by the user. Each user must belong to a primary group.
  - **Secondary groups:** Specifies one or more groups to which a user also belongs. Users can belong to up to 15 secondary groups.
- **Comment:** Information that identifies the user
- **User's home directory:** A directory into which the user is placed after login. The home directory is the portion of a file system that is allocated to a user for storing private files.
- **User's login shell:** The user's work environment that is set up by the initialization files that are defined by the user's login shell. Besides having a home directory to create and store files, users need an environment that gives them access to the tools and resources that they need to do their work. When a user logs in to a system, the user's work environment is determined by the initialization files. These files are defined by the user's startup shell, which can vary, depending on the release.

For additional guidelines on setting up user accounts, see the section titled "Guidelines for Assigning User Names, User IDs, and Group IDs" in the *Oracle Solaris Administration: Common Tasks* guide.

# System Files That Store User Account Information

| System File for User Account Information | Description |
|---|---|
| `/etc/passwd` | Contains login account entries for authorized system users |
| `/etc/shadow` | Contains encrypted passwords |
| `/etc/default/passwd` | Contains entries for controlling all user passwords on the system |
| `/etc/group` | Defines the default system group entries |

The Oracle Solaris 11 OS stores user account and group entry information in the following system files:

- **`/etc/passwd`:** Contains login account entries for authorized system users. Due to the critical nature of this file, it should not be edited directly. Instead, command-line tools should be used to maintain the file.
- **`/etc/shadow`:** Contains encrypted passwords. Due to the critical nature of this file, it should not be edited directly. Instead, command-line tools should be used to maintain the file. Only the root user can read the `/etc/shadow` file.
- **`/etc/default/passwd`:** Contains entries for controlling properties for all users' passwords on the system
- **`/etc/group`:** Defines the default system group entries for system groups that support some system-wide tasks, such as printing, network administration, or electronic mail. Many of these groups have corresponding entries in the `/etc/passwd` file.

You now take a closer look at the contents of each of these files.

# Interpreting the `/etc/passwd` File

```
root:x:0:0:Super-User:/:/sbin/sh
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
smmsp:x:25:25:SendMail Message Submission Program:/:
listen:x:37:4:Network Admin:/usr/net/nls:
gdm:x:50:50:GDM Reserved UID:/:
webservd:x:80:80:WebServer Reserved UID:/:
postgres:x:90:90:PostgreSQL Reserved UID:/:/usr/bin/pfksh
unknown:x:96:96:Unknown Remote UID:/:
svctag:x:95:12:Service Tag UID:/:
nobody:x:60001:60001:NFS Anonymous Access User:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
```

Even though you will not edit the `/etc/passwd` file directly, you should be familiar with its contents. This slide presents an example of the default system account entries in the `/etc/passwd` file.

The username, associated user ID, and description for the default entries in the `/etc/passwd` file are as follows:

- `root`, `0`: The superuser account
- `daemon`, `1`: The umbrella system daemon that is associated with the routine system tasks
- `bin`, `2`: The administrative daemon that is associated with running system binaries to perform some routine system tasks
- `sys`, `3`: The administrative daemon that is associated with system logging or updating files in temporary directories
- `adm`, `4`: The administrative daemon that is associated with system logging
- `lp`, `71`: The line printer daemon
- `uucp`, `5`: The daemon that is associated with the UNIX-to-UNIX Copy Program (UUCP) functions
- `nuucp`, `6`: Another daemon associated with the UUCP functions

- `dladm`, `15`: The account that is reserved for datalink administration
- `zfssnap`, `51`: The account that is reserved for automatic snapshots
- `upnp`, `52`: The account that is reserved for the UPnP server
- `xvm`, `60`: Reserved for the xVM user
- `openldap`, `75`: Reserved for the OpenLDAP user
- `smmsp`, `25`: The daemon for the Sendmail message submission program
- `webservd`, `80`: The account reserved for WebServer access
- `postgres`, `90`: The account reserved for PostgresSQL access
- `svctag`, `95`: Service Tag Registry access
- `gdm`, `50`: GNOME Display Manager daemon
- `listen`, `37`: Network listener daemon
- `unknown`, `96`: The account that is reserved for unmappable remote groups in NFSv4 ACLs
- `nobody`, `60001`: The account that is reserved for anonymous NFS access
- `noaccess`, `60002`: Assigned to a user or process that needs access to a system through some application but without actually logging in
- `nobody4`, `65534`: SunOS 4.0 or 4.1 version of the nobody user account

# Interpreting an `/etc/passwd` File Entry

```
loginID:x:UID:GID:comment:home_directory:login_shell
```

- *x*: Represents a placeholder for the user’s encrypted password
- *UID*: Contains the UID number that is used by the system to identify the user
- *GID*: Contains the GID number that is used by the system to identify the user’s primary group
- *comment*: Typically contains the user’s full name
- *home_directory*: Contains the autofs-mounted directory name of the user’s *home* directory
- *login_shell*: Defines the user’s login shell

The slide presents an example of an `/etc/passwd` file entry. Each entry in the `/etc/passwd` file contains seven fields. A colon separates each field. The following is the format for an entry:

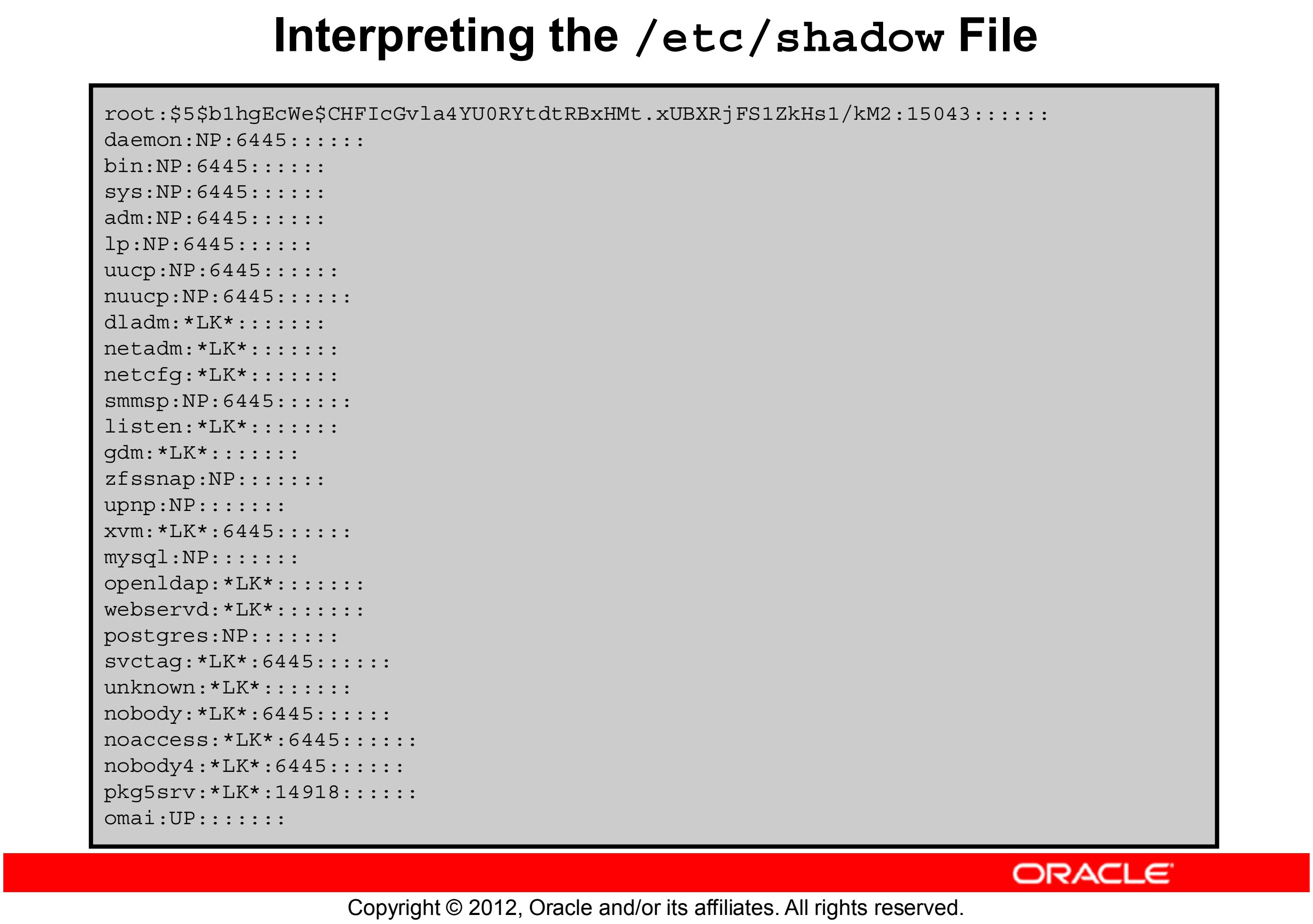
```
    loginID:x:UID:GID:comment:home_directory:login_shell
```

The description and requirement for each field are as follows:

- *loginID*: Represents the user’s login name. It should be unique to each user. The field should contain a string of no more than eight letters (A–Z, a–z) and numbers (`0`–`9`). The first character should be a letter, and at least one character should be lowercase.
- *x*: Represents a placeholder for the user’s encrypted password, which is kept in the `/etc/shadow` file
- *UID*: Contains the UID number that is used by the system to identify the user. UID numbers for users range from 100 to 60000. Values 0 through 99 are reserved for system accounts. UID number 60001 is reserved for the nobody account. UID number 60002 is reserved for the `noaccess` account. Even though duplicate UID numbers are allowed, they should be avoided unless absolutely required by a program.

  **Note:** The maximum value for a UID is 2147483647. However, the UIDs that are greater than 60000 do not have full utility and are incompatible with some Oracle Solaris OS features. Avoid using UIDs that are greater than 60000 so as to be compatible with earlier versions of the operating system.

- *GID*: Contains the GID number that is used by the system to identify the user's primary group. GID numbers for users range from 100 to 60000. (Those between 0 and 99 are reserved for system accounts.)
- *comment*: Typically contains the user's full name
- *home_directory*: Contains the name of the mounted name of the user's home directory (done as a default using AUTOFS) and is created as a ZFS file system automatically
- *login_shell*: Defines the user's login shell. There are six possible login shells in the Solaris OS: the Bourne shell, the Korn shell, the C shell, the Z shell, the Bash shell, and the TC shell. The default shell for Oracle Solaris 11 is Bash.



# Interpreting the /etc/shadow File

```
root:$5$b1hgEcWe$CHFIcGvla4YU0RYtdtRBxHMt.xUBXRjFS1ZkHs1/kM2:15043::::::
daemon:NP:6445::::::
bin:NP:6445::::::
sys:NP:6445::::::
adm:NP:6445::::::
lp:NP:6445::::::
uucp:NP:6445::::::
nuucp:NP:6445::::::
dladm:*LK*:::::::
netadm:*LK*:::::::
netcfg:*LK*:::::::
smmsp:NP:6445::::::
listen:*LK*:::::::
gdm:*LK*:::::::
zfssnap:NP:::::::
upnp:NP:::::::
xvm:*LK*:6445::::::
mysql:NP:::::::
openldap:*LK*:::::::
webservd:*LK*:::::::
postgres:NP:::::::
svctag:*LK*:6445::::::
unknown:*LK*:::::::
nobody:*LK*:6445::::::
noaccess:*LK*:6445::::::
nobody4:*LK*:6445::::::
pkg5srv:*LK*:14918::::::
omai:UP:::::::
```

This slide presents an example of the initial system account entries in the /etc/shadow file, which correspond to the default system account entries found in the /etc/passwd file. Again, you will not be editing this file, but you should be familiar with its contents.

# Interpreting an `/etc/shadow` File Entry

```
loginID:password:lastchg:min:max:warn:inactive:expire:flag
```

Each entry in the `/etc/shadow` file contains nine fields:

- *`loginID`*: The user's login name
- *`password`*: A variable-length encrypted password
- *`lastchg`*: The number of days between January 1, 1970 and the last password modification date
- *`min`*: The minimum number of days required between password changes
- *`max`*: The maximum number of days that the password is valid before the user is prompted to enter a new password at login
- *`warn`*: Number of days that the user is warned before the password expires
- *`inactive`*: Number of inactive days allowed for the user before the user's account is locked
- *`expire`*: Date when the user account expires
- *`flag`*: Used to track failed logins

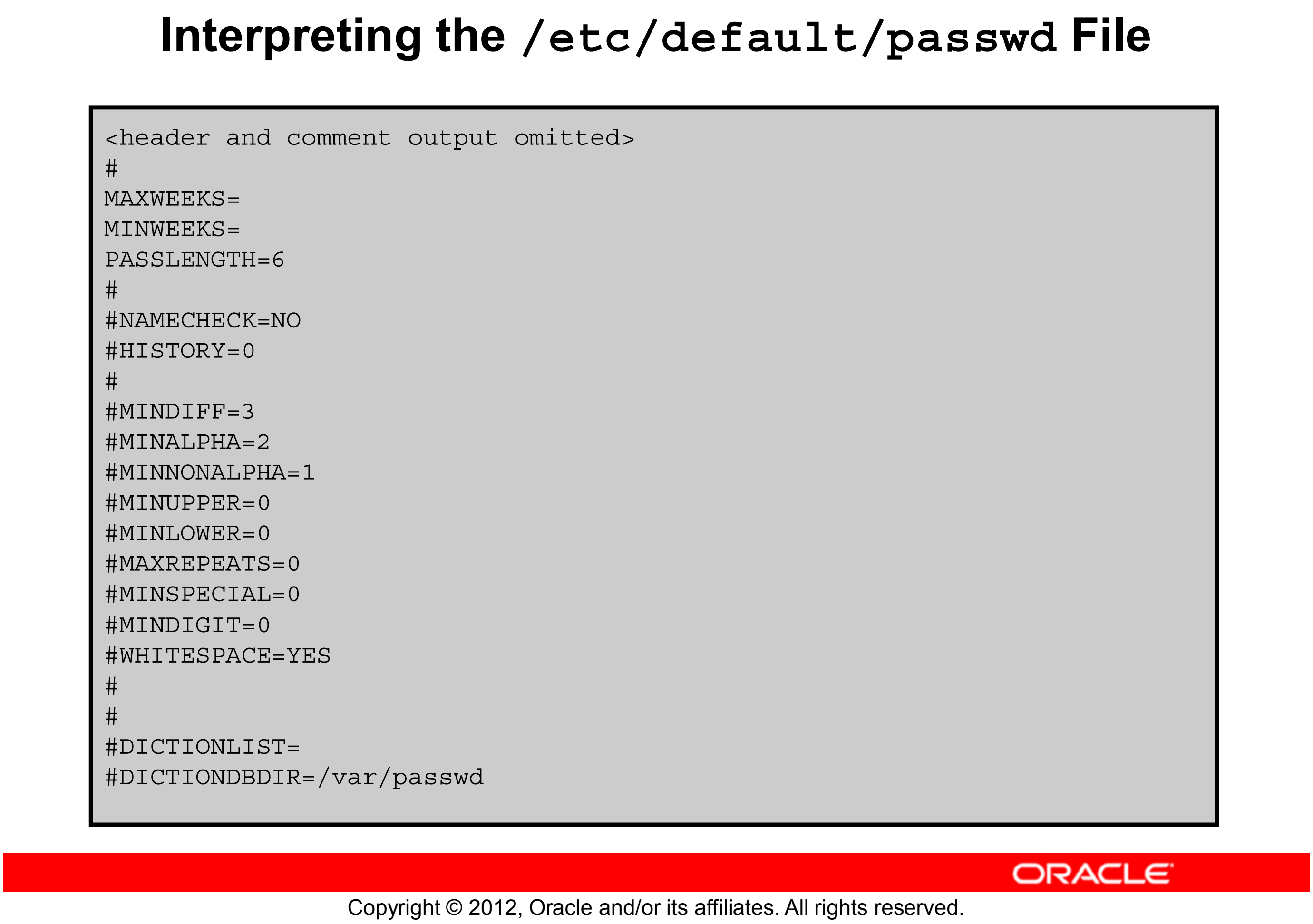
This slide presents an example of a default entry in the `/etc/shadow` file. Each entry in the `/etc/shadow` file contains nine fields. A colon separates each field. The following is the format for an entry:

```
    loginID:password:lastchg:min:max:warn:inactive:expire:flag
```

The description and requirement for each field are as follows:

- *`loginID`*: User's login name
- *`password`*: A variable-length password, depending on the selected hashing algorithm. The string `*LK*` indicates a locked account, and the string `NP` indicates no valid password. Passwords must be constructed to meet the following requirements. Each password must be at least six characters, and contain at least two alphabetic characters and at least one numeric or special character. It cannot be the same as the login ID or the reverse of the login ID.
- *`lastchg`*: Number of days between January 1, 1970 and the last password modification date

- *min*: Minimum number of days required between password changes
- *max*: Maximum number of days that the password is valid before the user is prompted to enter a new password at login
- *warn*: Number of days that the user is warned before the password expires
- *inactive*: Number of inactive days allowed for the user before the user's account is locked
- *expire*: Date (given as number of days since January 1, 1970) when the user account expires. After the date is exceeded, the user can no longer log in.
- *flag*: Used to track failed logins. The count is in low-order four bits. The remainder is reserved for future use, set to zero.



## Interpreting the `/etc/default/passwd` File

```
<header and comment output omitted>
#
MAXWEEKS=
MINWEEKS=
PASSLENGTH=6
#
#NAMECHECK=NO
#HISTORY=0
#
#MINDIFF=3
#MINALPHA=2
#MINNONALPHA=1
#MINUPPER=0
#MINLOWER=0
#MAXREPEATS=0
#MINSPECIAL=0
#MINDIGIT=0
#WHITESPACE=YES
#
#
#DICTIONLIST=
#DICTIONDBDIR=/var/passwd
```

This slide presents the `/etc/default/passwd` file. You can set values for the following parameters in this file to control properties for all users' passwords on the system:

- `MAXWEEKS`: Sets the maximum time period (in weeks) that the password is valid
- `MINWEEKS`: Sets the minimum time period before the password can be changed
- `PASSLENGTH`: Sets the minimum number of characters for a password. Valid entries are `6`, `7`, and `8`.
- `WARNWEEKS` (not shown): Sets the time period before a password's expiration to warn the user that the password will expire

  **Note:** The `WARNWEEKS` value does not exist by default in the `/etc/default/passwd` file, but it can be added.

**Note:** The password aging parameters `MAXWEEKS`, `MINWEEKS`, and `WARNWEEKS` are default values. If set in the `/etc/shadow` file, the parameters in that file override those in the `/etc/default/passwd` file for individual users.

The following password management controls are commented out by default.

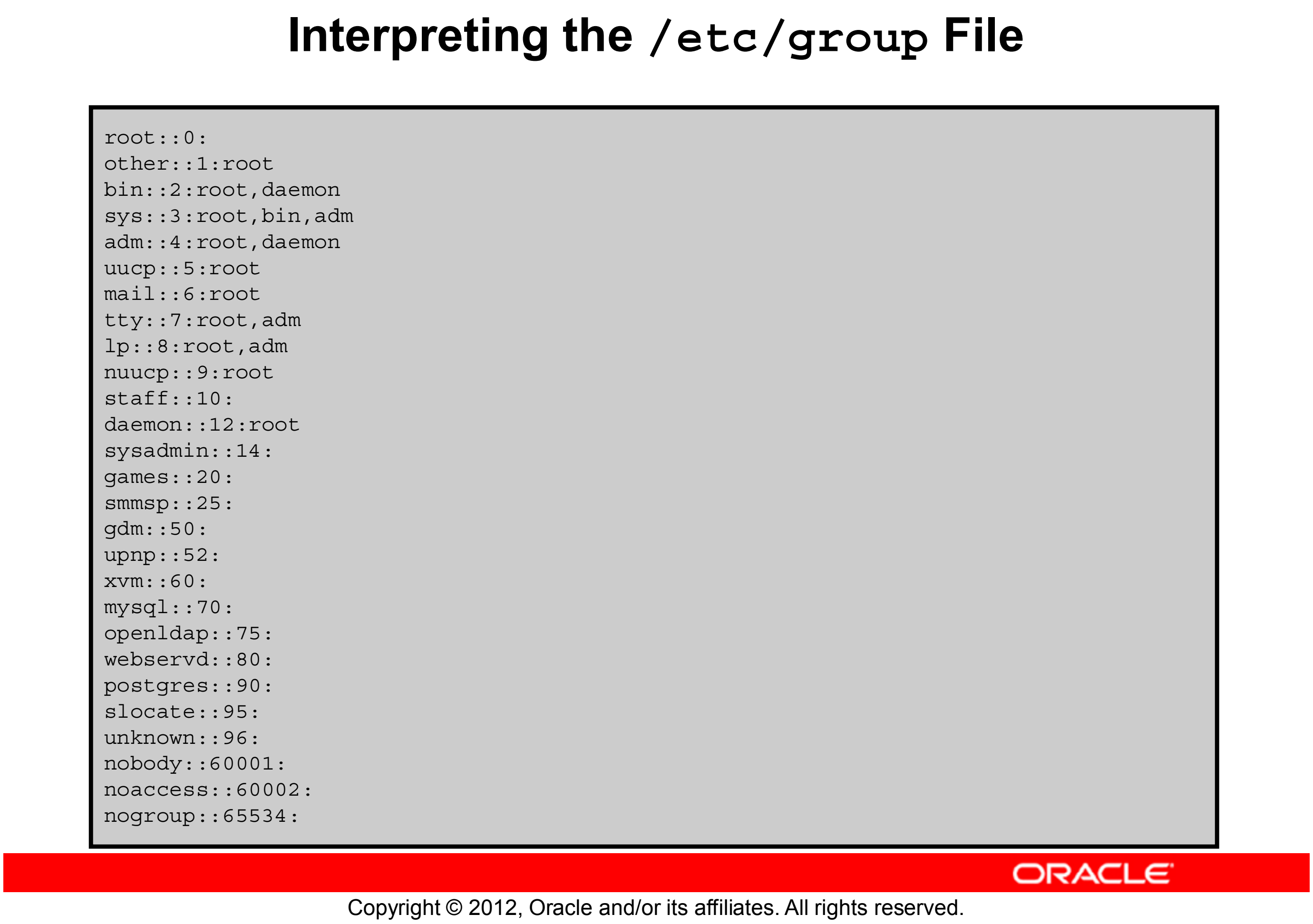
- `NAMECHECK=NO`: Sets the password controls to verify that the user is not using the login name as a component of the password. The default is to do login name checking.
- `HISTORY=0`: Forces the passwd program to log up to 26 changes to the user's password. This prevents the user from reusing the same password for 26 changes. If the `HISTORY` value is set to a number other than zero (`0`), and then set back to zero, it causes the password log for a user to be removed on the next password change.

You can control the complexity of the password by using the following parameters, which by default, are commented out:

- `MINDIFF=3`: Specifies the minimum number of characters in the password that must be different
- `MINALPHA=2`: Specifies the minimum number of alpha characters that must appear in the password
- `MINNONALPHA=1`: Specifies the minimum number of non-alpha characters that must appear in the password
- `MINUPPER=0`: Specifies the minimum number of uppercase characters that must appear in the password
- `MINLOWER=0`: Specifies the minimum number of lowercase characters that must appear in the password
- `MAXREPEATS=0`: Specifies the maximum number of times a password can be repeated
- `MINSPECIAL=0`: Specifies the minimum number of special characters that must appear in the password
- `MINDIGIT=0`: Specifies the minimum number of digits for the password
- `WHITESPACE=YES`: Specifies whether or not whitespace is allowed in the password

**Note:** Be careful with the amount of complexity that you introduce into the password structure. For example, you may inadvertently cause users to write down their passwords because they may be too difficult for them to remember. When setting a password change policy, you must not underestimate the problems that too much complexity may cause.

- `DICTIONLIST=`: Causes the `passwd` program to perform dictionary word lookups from comma-separated dictionary files
- `DICTIONDBDIR=/var/passwd`: Is the location of the dictionary where the generated dictionary databases reside. This directory must be created manually.



# Interpreting the `/etc/group` File

```
root::0:
other::1:root
bin::2:root,daemon
sys::3:root,bin,adm
adm::4:root,daemon
uucp::5:root
mail::6:root
tty::7:root,adm
lp::8:root,adm
nuucp::9:root
staff::10:
daemon::12:root
sysadmin::14:
games::20:
smmsp::25:
gdm::50:
upnp::52:
xvm::60:
mysql::70:
openldap::75:
webservd::80:
postgres::90:
slocate::95:
unknown::96:
nobody::60001:
noaccess::60002:
nogroup::65534:
```

This slide presents an example of the default entries in the `/etc/default/group` file. Many of these groups have corresponding entries in the `/etc/passwd` file.

The group name, associated group ID, and description for the default entries in the `/etc/group` file are as follows:

- `root`, `0`: The superuser group
- `other`, `1`: The optional group
- `bin`, `2`: The administrative group that is associated with the running system binaries
- `sys`, `3`: The administrative group that is associated with system logging or temporary directories
- `adm`, `4`: The administrative group that is associated with system logging
- `uucp`, `5`: The group that associated with the `uucp` functions
- `mail`, `6`: The electronic mail group
- `tty`, `7`: The group that is associated with the `tty` devices
- `lp`, `8`: The line printer group
- `nuucp`, `9`: The group that is associated with the `uucp` functions

- `staff`, `10`: The general administrative group
- `daemon`, `12`: The group that is associated with routine system tasks
- `sysadmin,14`: The administrative group that is useful for system administrators
- `games`, `20`: The group that is reserved for games
- `smmsp`, `25`: The daemon for the Sendmail message submission program
- `gdm`, `50`: The group that is reserved for the GNOME Display Manager daemon
- `upnp`, `52`: The group that is associated with the UPnP server functions
- `xvm`, `60`: The group that is reserved for xVM access
- `mysql`, `70`: The group that is reserved for MySQL access
- `openldap`, `75`: The group that is reserved for OpenLDAP access
- `webservd`, `80`: The group that is reserved for WebServer access
- `postgres`, `90`: The group that is reserved for PostgresSQL access
- `slocate`, `95`: The group that is reserved for `slocate` indexing and query daemon. `slocate` is a secure version of the `locate` command.
- `unknown`, `96`: The group that is reserved for unmappable remote groups in NFSv4 ACLs
- `nobody`, `60001`: The group that is assigned for anonymous NFS access
- `noaccess`, `60002`: The group that is assigned to a user or process that needs access to a system through some application but without actually logging in
- `nogroup`, `65534`: The group that is assigned to a user who is not a member of a known group

# Interpreting an `/etc/group` File Entry

```
groupname:group-password:GID:username-list
```

Each entry in the `/etc/group` file contains four fields:

- *`groupname`*: Contains the name assigned to the group
- *`group-password`*: Usually contains an empty field or an asterisk
- *`GID`*: Contains the group's GID number
- *`username-list`*: Contains a comma-separated list of usernames that represent the user's secondary group memberships

This slide provides an example of a default `/etc/group` file entry. Each entry in the `/etc/group` file contains four fields. A colon separates each field. The following is the format for an entry:

```
groupname:group-password:GID:username-list
```

The description and requirement for each field are as follows:

- *`groupname`*: Contains the name assigned to the group. Group names contain up to a maximum of eight characters.
- *`group-password`*: Usually contains an empty field or an asterisk. This is a relic of the earlier versions of UNIX.
- *`GID`*: Contains the group's GID number. It is unique on the local system and should be unique across the organization. Numbers 0 through 99, 60001, 60002 and 65534 are reserved for system group entries. User-defined groups range from 100 through 60000.
- *`username-list`*: Contains a comma-separated list of usernames that represent the user's secondary group memberships. By default, each user can belong to a maximum of 15 secondary groups.

## Implementing the User Administration Plan

Your assignment is to:

- Set up a few user accounts
- Maintain these user accounts
- Manage user initialization files
- Use shell metacharacters
- Configure user disk quotas

Now you will help to test the user administration functionality. Your assignment is to set up some new user accounts, maintain the accounts, and set up the user initialization files that are used to define the user's work environment. You will also spend some time testing the use of the shell metacharacters. Your final task will be to configure user disk quotas.

# Quiz

A user must belong to at least one group.

a. True

b. False

**Answer: a**

# Quiz

Which file contains encrypted user passwords?

a. `/etc/shadow`

b. `/etc/default/passwd`

c. `/etc/skel`

**Answer: a**

# Lesson Agenda

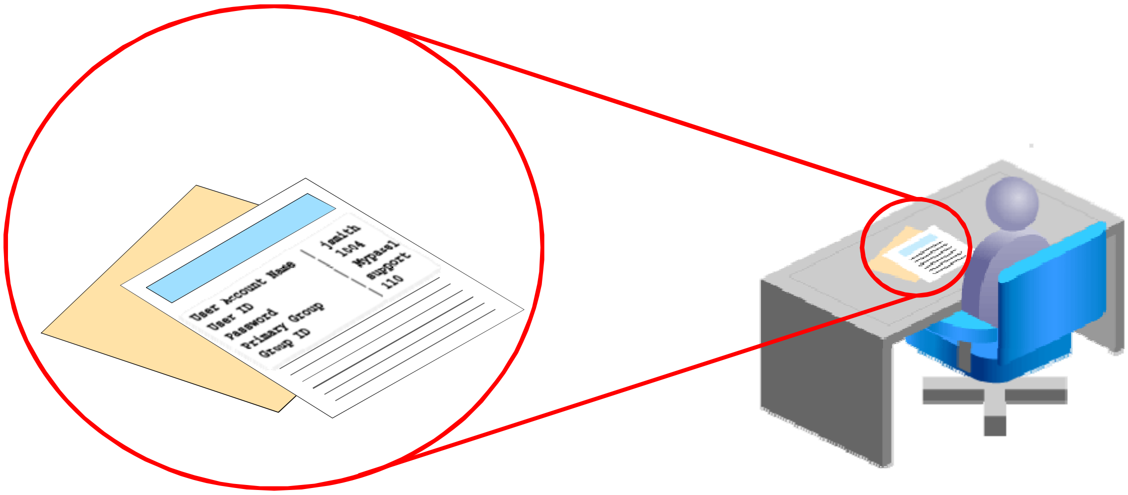
- Planning for User Administration
- Setting Up User Accounts
- Maintaining User Accounts
- Managing User Initialization Files
- Using Shell Metacharacters and Configuring User Disk Quotas

# Setting Up User Accounts

- Gathering User Information
- Creating and Modifying the User Accounts Default File
- Adding a Group
- Adding a User Account
- Verifying the User Account Setup
- Setting a Password to Expire Immediately

In this section, you learn to set up a user account. The tasks that you perform to set up the user account are presented in the slide. You begin with gathering user information and cover each of the other tasks subsequently.

# Gathering User Information



|  |  |
|---|---|
| User Account Name | jsmith |
| User ID | 1004 |
| Password | Mypass1 |
| Primary Group | support |
| Group ID | 110 |

Before you start setting up accounts in the system, it is always a good idea to gather user information first. If your company does not have a form that it uses for this purpose, you can create one. The form should contain the information that you need to complete the user account setup, such as the username and user ID, user password, and group name and group ID.

**Note:** For an example form that you could use, as well as the type of information you might want to collect, see the section titled “Gathering User Information” in the *Oracle Solaris Administration: Common Tasks* guide.

# Creating the User Accounts Default File

To check to see whether the user accounts default file exists, use `ls /usr/sadm/defadduser`.

```
# ls /usr/sadm/defadduser
ls: cannot access /usr/sadm/defadduser: No such file or
directory
```

To create the user accounts default file, use `useradd -D`.

```
# useradd -D
group=staff,10  project=default,3  basedir=/export/home
skel=/etc/skel  shell=/usr/bin/bash  inactive=0
expire=  auths=  profiles=  roles=  limitpriv=
defaultpriv=  lock_after_retries=
```

The next task is to create the user accounts default file if it does not exist. The user accounts default file contains a preset range of default values for a new user's account. When you use the `useradd` command for the first time, it generates a file called `/var/sadm/defadduser` that contains the default values for a new user account. If you modify the contents of this file, the new contents become the default values for the next time that you use the `useradd` command.

To create the user accounts default file, use the `useradd -D` command.

**Note:** The `useradd` command is used to administer a new user login on the system and adds a new user to the `/etc/passwd` and `/etc/shadow` files. This command also automatically copies all the initialization files from the `/etc/skel` directory to the user's new home directory.

The `-D` option displays and sets the default values for the default values in the user accounts default file. The default values are:

- `group=staff` (GID of `10`): An existing group's integer ID or character-string name. Without the `-D` option, it defines the new user's primary group membership and defaults to the default group.
- `project=default,3`: Specifies the default project for user. A project is similar to a group in that it contains a collection of users.
- `basedir=/export/home`: Specifies the user's base or home directory
- `skel=/etc/skel`: Is the directory that contains the default user initialization file templates that are automatically copied to a new user's home directory when the user account is created
- `shell=/user/bin/bash`: Defaults to an empty field, causing the system to use `/user/bin/bash` as the default
- `inactive=0`: Specifies the maximum number of days allowed for the use of a login ID before that ID is declared invalid. Normal values are positive integers. A value of `0` defeats the status.
- `expire=null`: Specifies the expiration date for a login. After this date, no user will be able to access this login.
- `auths=null`: Specifies a set of authorizations for a user. The default is none.
- `profiles= null`: Specifies one or more profiles for a user. The default is none.
- `roles= null`: Specifies one or more profiles for a role. The default is none.
- `limitpriv= null`: Limits the privileges that a user has. The default is none.
- `defaultpriv= null`: Specifies the default privileges that a user has. The default is none.
- `lock_after_retries= null`: Specifies the number of failed login retry attempts before the user account is locked. User accounts are locked by default when added with the `useradd` command.

# Modifying the User Accounts Default File

To modify the user accounts default file, use `useradd -D` *`value`*.

```
# useradd -D -s /bin/ksh
group=staff,10  project=default,3  basedir=/export/home
skel=/etc/skel  shell=/bin/ksh  inactive=0
expire=  auths=  profiles=  roles=  limitpriv=
defaultpriv=  lock_after_retries=

# useradd -D
group=staff,10  project=default,3  basedir=/export/home
skel=/etc/skel  shell=/bin/ksh  inactive=0
expire=  auths=  profiles=  roles=  limitpriv=
defaultpriv=  lock_after_retries=
```

You can modify any of the default values in the user accounts default file. To modify the file, use the `useradd -D` command with the appropriate option and the value that you want to change. In the example, you modify the user's shell on login from the default `/usr/bin/bash` to `/bin/ksh`. To modify the default shell setting, you must use the `-s` option.

To display the current user accounts default file that will be applied to any new user account, you can run the `useradd -D` command again. Notice that the default shell value is now `shell=/bin/ksh`.

For more information about the `useradd` command options, see the `useradd`(1M) man page.

## Adding a Group

To add a group, use `groupadd -g` *`GID groupname`*.

```
# groupadd -g 110 support
```

To verify that the group has been created, use `grep` *`groupname`* `/etc/group`.

```
# grep support /etc/group
support::110:
```

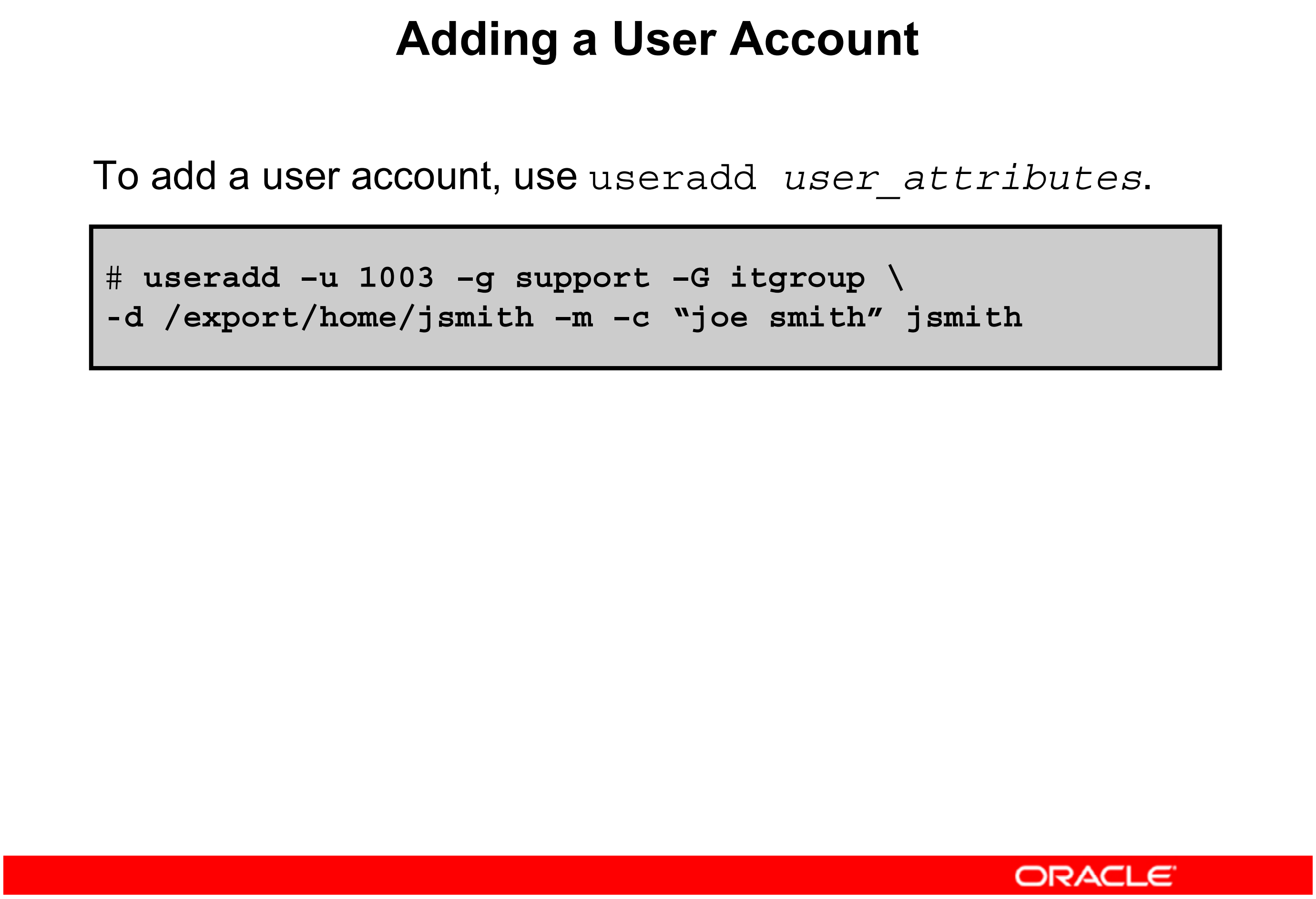
ORACLE

As you learned in the section on planning for user administration, every user must belong to at least one group. Organizing multiple users into groups makes user administration easier because you can set privileges for a group instead of doing it user by user. Each user belongs to a group that is referred to as the user's primary group. The GID number, which is located in the user's account entry within the `/etc/passwd` file, specifies the user's primary group. Each user can also belong to up to 15 additional groups, known as secondary groups. In the `/etc/group` file, you can add users to group entries, thus establishing the user's secondary group affiliations.

For the purposes of training, assume that your new user `jsmith` is part of the support organization. Before you create a user account for `jsmith`, you need to create the group of which he is a part.

To create a new group definition and assign a new group ID (GID) number for the new group, use the `groupadd -g` command, followed by the group ID number and the group name. In the example, you create the group named `support` and assign it the group ID `110`.

The `groupadd` command adds the group definition to the `/etc/group` file. To verify that the group is created, you can `grep` the group name in the `/etc/group` file, as shown in the example. The output of the command displays the group name and the GID.



# Adding a User Account

To add a user account, use `useradd` *`user_attributes`*.

```
# useradd -u 1003 -g support -G itgroup \
-d /export/home/jsmith -m -c “joe smith” jsmith
```

After you have created the user accounts default file, modified it, and created a group or groups for your user, you are ready to add the user by creating the user account. A user account consists of a number of attributes that you assign as part of user account creation (and which you collected during your user information gathering). These attributes and their associated `useradd` command options are as follows:

- `-u` *`uid`*: Sets the UID number for the new user. UID numbers must be a whole number that is less than or equal to 2147483647. UID numbers are required for both regular user accounts and special system accounts. Do not assign UIDs `0` through `99`. These UIDs are reserved for allocation by Oracle Solaris. By definition, root always has UID `0`, daemon has UID `1`, and pseudo-user bin has UID `2`. In addition, you should give uucp logins and pseudo user logins, such as `who`, `tty`, and `ttytype`, low UIDs so that they fall at the beginning of the `/etc/passwd` file.
- `-g` *`gid`*: Defines the new user’s primary group
- `-G` *`gid`*: Defines the new user’s secondary group memberships
- `-d` *`dir`*: Defines the full path name for the user’s home directory

- `-m`: Creates the user's home directory if it does not already exist
- `-s` *`shell`*: Defines the full path name for the shell program of the user's login shell
- `-c` *`comment`*: Specifies any comment, such as the user's full name and location
- *`loginname`*: Defines the user's login name for the user account

When adding a user account, you must assign a primary group for a user or accept the default group, `staff` (group `10`). The primary group should already exist. If the primary group does not exist, specify the group by a GID number.

To add a user, use the `useradd` command with the appropriate options. In the example, you create a user account for the user `jsmith`. You have assigned the user ID `1003` to `jsmith` and have made him a member of the `support` group that you created earlier. This is `jsmith`'s primary group. In addition, you have also made him a member of a secondary group called `itgroup`. Next, you have defined the full path name for the user's home directory, which you created earlier. You have used the comment option to specify the user's full name (Joe Smith), and you have defined the user's login name for the user account (`jsmith`).

Because you have already created the user's home directory, you do not need to specify it here with the `-m` option. Notice also that you did not specify the shell program of the user's login shell. Because you have not assigned a shell program to this user, he will have the default shell, `bash`, that you set up in the user accounts default file.

# Verifying the User Account Setup

As you create a user account, the information is sent to these files:

- `/etc/passwd`
- `/etc/shadow`
- `/etc/group`

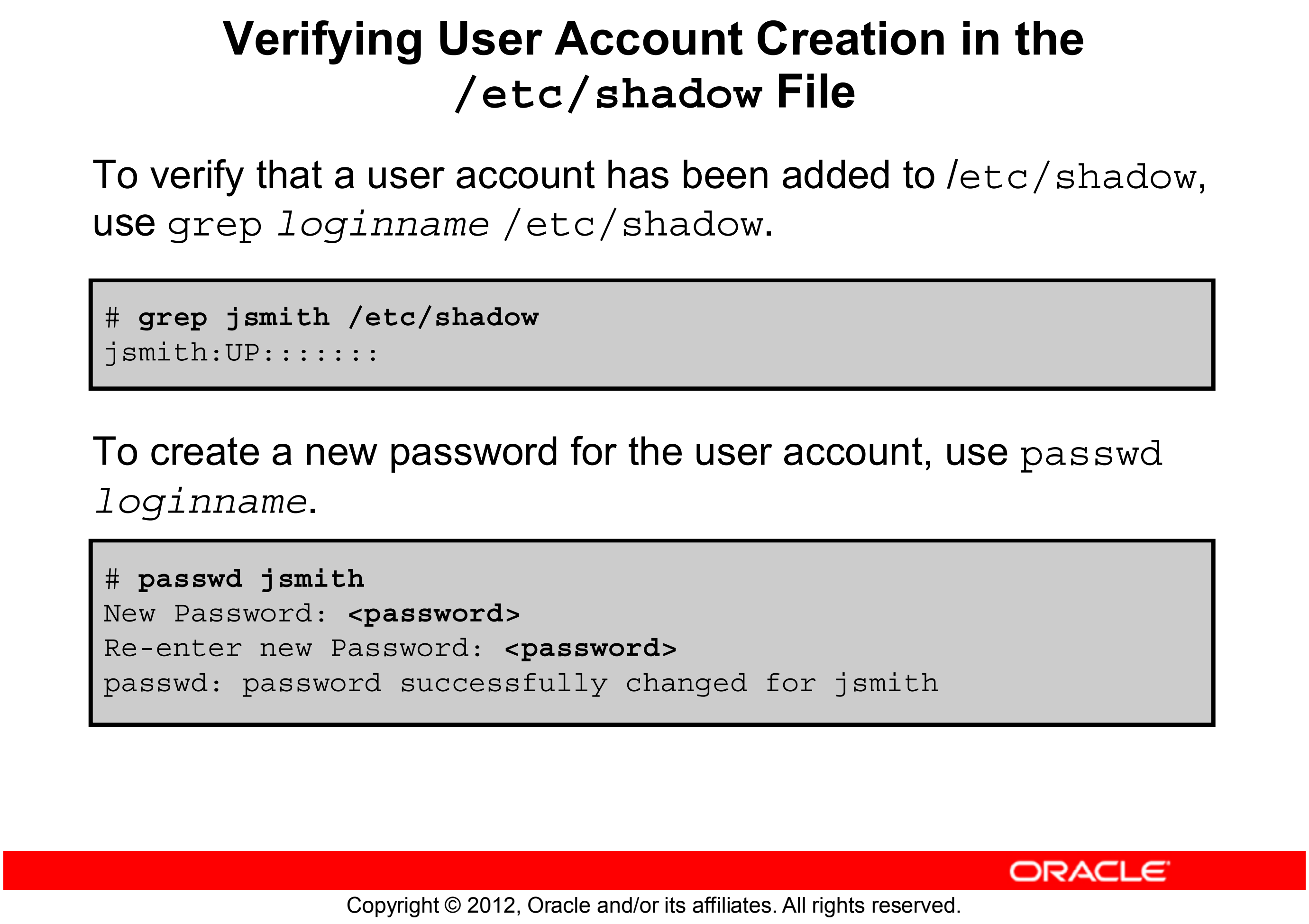
As you create a user account, the account information is automatically sent to the `/etc/passwd`, `/etc/shadow`, and `/etc/group` files. To verify that the user account has been created, you should check each of these files. You can do that beginning with the `/etc/password` file.

# Verifying User Account Creation in the `/etc/passwd` File

To verify that a user account has been added to `/etc/passwd`, use `grep` *`loginname`* `/etc/passwd`.

```
# grep jsmith /etc/passwd
jsmith:x:1003:110:joe smith:/home/jsmith:/bin/bash
```

To verify that a new user account has been added to the `/etc/passwd` file, use the `grep` command, followed by the user's login name and `/etc/passwd`. In the example, you check to see whether an entry for Joe Smith has been created. Because an entry is returned, you can conclude that the user account was created successfully. As you learned in the section on planning for user administration, the entry displays the user's login name (or login ID), a placeholder for the user's encrypted password (`x`), the user's user ID (UID), the group ID (GID) for the user's primary group, a comment (usually the user's full name), the full path name to the user's home directory, and the user's login shell.



## Verifying User Account Creation in the /etc/shadow File

To verify that a user account has been added to /etc/shadow, use grep *loginname* /etc/shadow.

```
# grep jsmith /etc/shadow
jsmith:UP::::::
```

To create a new password for the user account, use passwd *loginname*.

```
# passwd jsmith
New Password: <password>
Re-enter new Password: <password>
passwd: password successfully changed for jsmith
```

To verify that a new user account has been added to the /etc/shadow file, use the grep command, followed by the user's login name and /etc/shadow. In the first example, you check to see whether an entry for Joe Smith has been created. Because an entry is returned, you can conclude that the user account was created successfully. However, because this is a new user account, the account has been tagged, by default, with UP for "undefined password." To remove this tag, create a password for the user.

**Note:** Remember that each password must be at least six characters and contain at least two alphabetic characters and at least one numeric or special character. It cannot be the same as the login ID or the reverse of the login ID.

To create the new password for the user account, use the passwd command, followed by the user's login name. As you can see in the second example, you are prompted to provide the new password, and then re-enter it. If you have entered the password successfully, you will receive a confirmation that the password has been changed.

# Verifying User Account Creation in the `/etc/shadow` File

To view the user account in `/etc/shadow` after the password has been changed, use `grep` *`loginname`* `/etc/shadow`.

```
# grep jsmith /etc/shadow
jsmith:$5$x0aftZOd$d8hbuX/rb9vS485/9OlH63EkPbLzL8eDtFL/LVtbAp3:15168::::::
```

After you have changed the password, you can go back to the `/etc/shadow` file to view the new user account by using `grep` *`loginname`* `/etc/shadow`. Now you can see the entry. In this example, the user's login name is displayed, followed by the encrypted password. The number that appears after the password field is the number of days between January 1, 1970 and the last time the password was modified. You might recall that this is the `lastchg` field. As you can see, the remaining six fields have not been populated.

# Verifying User Account Creation in the /etc/group File

To verify that a user has been added to /etc/group, first confirm that the group exists by using grep *groupname* /etc/group, and then use id loginname.

```
# grep support /etc/group
support::110:
# id jsmith
1003 110
```

The last verification step is to ensure that the new user appears as a member of a group in the /etc/group file. To do this, you must first check that there is an entry for the group in the /etc/group file and make a note of the group ID number. You do this by using the grep command, followed by the group name and /etc/group. In the example, you check to see whether the support group is in the /etc/group file. As you can see, it is and the group ID number is 110. You might recall from an earlier discussion about the /etc/group file that the entry contains four fields, the first of which is the group name, followed by the group password (which is usually empty), and the group ID number (GID). The last field contains a list of usernames that represent the user's secondary group memberships. For example, if users ckent and jdoe had the support group identified as a secondary group in each of their user accounts, the /etc/group entry for the support group would look like this:

```
    support::110: ckent,jdoe
```

You can also use the id command with a user's login name to see the groups that a user is a member of. In the example, you can see that jsmith's user ID is 1003 and that he is a member of the support group, which has a GID of 110.

# Setting a Password to Expire Immediately

To set a password to expire immediately, use `passwd -f` *`loginname`*.

```
# passwd -f jsmith
passwd: password information changed for jsmith
```

To see the effect of `passwd` command changes, use `grep` *`loginname`* `/etc/shadow`.

```
# grep jsmith /etc/shadow
jsmith:$5$iJM6uDL8$1C28YFeERBKOFkA.eE3JCJEjLKkp4r.HBdGqiA7Ql96:0:::::
```

After you have successfully created a user's account, you can set up password aging on the user's password. Password aging enables you to force users to change their passwords periodically or to prevent a user from changing a password before a specified interval. If you want to prevent an intruder from gaining undetected access to the system by using an old and inactive account, you can set a password expiration date when the account becomes disabled. You can set password aging attributes with the `passwd` command.

These attributes and their associated `passwd` command options are as follows:

- `-f`: Forces the user to change the password at the next login by expiring the password for *`loginname`*
- `-l`: Locks the password entry for *`loginname`*
- `-n` *`min`*: Sets the `min` field for *`loginname`*. The `min` field contains the minimum number of days between password changes for name. If `min` is greater than `max`, the user may not change the password. Always use this option with the `-x` option, unless `max` is set to `-1` (aging turned off). In that case, `min` need not be set.
- `-w` *`warn`*: Sets the `warn` field for *`loginname`*. The `warn` field contains the number of days before the password expires and the user is warned.

- `-x` *`max`*: Sets the `max` field for *`loginname`*. The `max` field contains the number of days that the password is valid for name. The aging for name is turned off immediately if max is set to `-1`. If it is set to `0`, the user is forced to change the password at the next login session and aging is turned off.
- `-d`: Deletes the password for name. The login name will not be prompted for password. It is applicable only to the files repository.

**Note:** Only a privileged user can use these options.

To set a password to expire immediately, thereby forcing the user to change passwords at the next login, use the `passwd -f` command, followed by the user's login name, as shown in the example.

The changes that you make to the password attributes are reflected in the `/etc/shadow` file. In the second example, notice the third field that contained `15168` is now set to zero, indicating that the password has expired.

# Lesson Agenda

- Planning for User Administration
- Setting Up User Accounts
- Maintaining User Accounts
- Managing User Initialization Files
- Using Shell Metacharacters and Configuring User Disk Quotas

# Maintaining User Accounts

- Modifying a User Account
- Deleting a User Account
- Modifying a Group Entry
- Deleting a Group Entry

Changes have occurred recently in your company that require you to modify the user and group accounts that you have set up. One employee, `jsmith`, has discovered that he was adopted and that his real last name is Jones. He has requested that his login name be changed. Another employee, `ckent`, has decided to leave the company. In addition, there have been some major organizational changes that have impacted the group structure. The `support` group has been renamed to `itsupport`, and one group, `quality`, has been cut altogether.

In this section, you learn to maintain user accounts. First, you learn how to modify and delete a user account, and then to modify a group and delete a group.

# Modifying a User Account

To modify a user account, use `usermod` *`user_attributes`*.

```
# usermod –u 1003 –m –d /export/home/jjones –c “joe jones” \
–l jjones jsmith
Found user in files repository.

# grep jjones /etc/passwd
jjones x:1003:110:joe jones:/home/jjones:/bin/bash
```

The `usermod` command is used to modify user accounts. The `usermod` command uses many of the same options as the `useradd` command:

- `-u` *`uid`*: Specifies the UID for the current user or a new UID number for the user
- `-g` *`gid`*: Specifies an existing group's integer ID or character-string name. It redefines the user's primary group membership.
- `-G` *`gid`*: Defines the new user's supplementary group membership
- `-d` *`dir`*: Specifies the new home directory of the user. It defaults to *`base_dir/login`*, where *`base_dir`* is the base directory for new login home directories, and *`login`* is the new login.
- `-m`: Moves the user's home directory to the new location that is specified with the `-d` option
- `-s` *`shell`*: Specifies the full path name for the shell program of the user's login shell
- `-c` *`comment`*: Specifies any comment, such as the user's full name and location
- *`loginname`*: Identifies the user's login name for the user account

The `usermod` command also uses the following options and their associated attributes:

- `-o`: Allows a UID number to be duplicated
- `-l` *`new_logname`*: Changes a user's login name for the specified user account
- `-f` *`inactive`*: Sets the number of inactive days that are allowed on a user account. If the account is not logged in to for the specified number of days, it is locked.
- `-e` *`expire`*: Sets an expiration date on the user account; specifies the date (mm/dd/yy) on which a user can no longer log in and access the account. After that date, the account is locked.

**Note:** For a full listing of options for this command, see the `usermod`(1M) man page.

To modify a user's account, use the `usermod` command, followed by the appropriate user attribute. In the example, you change `jsmith`'s login name and home directory to `jjones`. Next, you verify that the change has been made in the `/etc/passwd` file. In the `/etc/passwd` file, you can see that the login name, the user's name, and home directory changes have been made. The user ID, group ID, and shell have remained the same.

# Deleting a User Account

To delete a user account, use `userdel -r` *`loginname`*.

```
# userdel -r ckent
Found user in files repository.
```

The employee `ckent` has left the company, so now you should delete his account from the system. To delete the user's home directory along with the account, use the `userdel` command, followed by the user's login name, as shown in the example in the slide.

**Note:** The `userdel` utility deletes a user account from the system and makes the appropriate account-related changes to the `/etc/passwd`, `/etc/shadow`, and `/etc/group` files.

**Note:** If for some reason, you want to delete only the account and not the home directory, use the `userdel` *`loginname`* command without the `-r` option.

# Modifying a Group Entry

To modify a group entry, use `groupmod` *`group_attribute`*.

```
# groupmod -n itsupport support
Found group in files repository.
# grep support /etc/group
itsupport::110::
# grep itsupport /etc/group
itsupport::110::
# id jjones
1003 110
```

The `groupmod` command is used to modify group entries. The `groupmod` utility modifies the group definitions in the `/etc/group` file. The `groupmod` command uses the following options:

- `-o`: Allows a GID number to be duplicated
- `-g` *`gid`*: Specifies the new GID number for the group
- `-n` *`name`*: Specifies the new name for the group

To modify a group entry, use the `groupmod` command, followed by the group attribute. In the example, you change the name of the `support` group to `itsupport` by using the `-n` option to specify the new name for the group. You then verify the group name change in the `/etc/group` file by running the `grep /etc/group` command against the old group name, and then again with the new group name. For the old group name, only the `itsupport` entry is returned. The `support` group entry is not there. When the new group name is run, you see the entry for that group. You do one last verification check by running the `id` command for a user that you know is part of the old `support` group: `jjones` (formerly `jsmith`). The output for this command also confirms that the group name change has been made successfully.

# Deleting a Group Entry

To reassign a user account to a valid group, use `usermod -u` *`UID`* `-g` *`GID`* `loginname`.

```
# usermod -u 1004 -g 120 jdoe
Found user in files repository.
# grep jdoe /etc/passwd
jdoe:x:1004:120:jane doe:/home/jdoe:/bin/bash
```

To delete a group entry, use `groupdel` *`groupname`*.

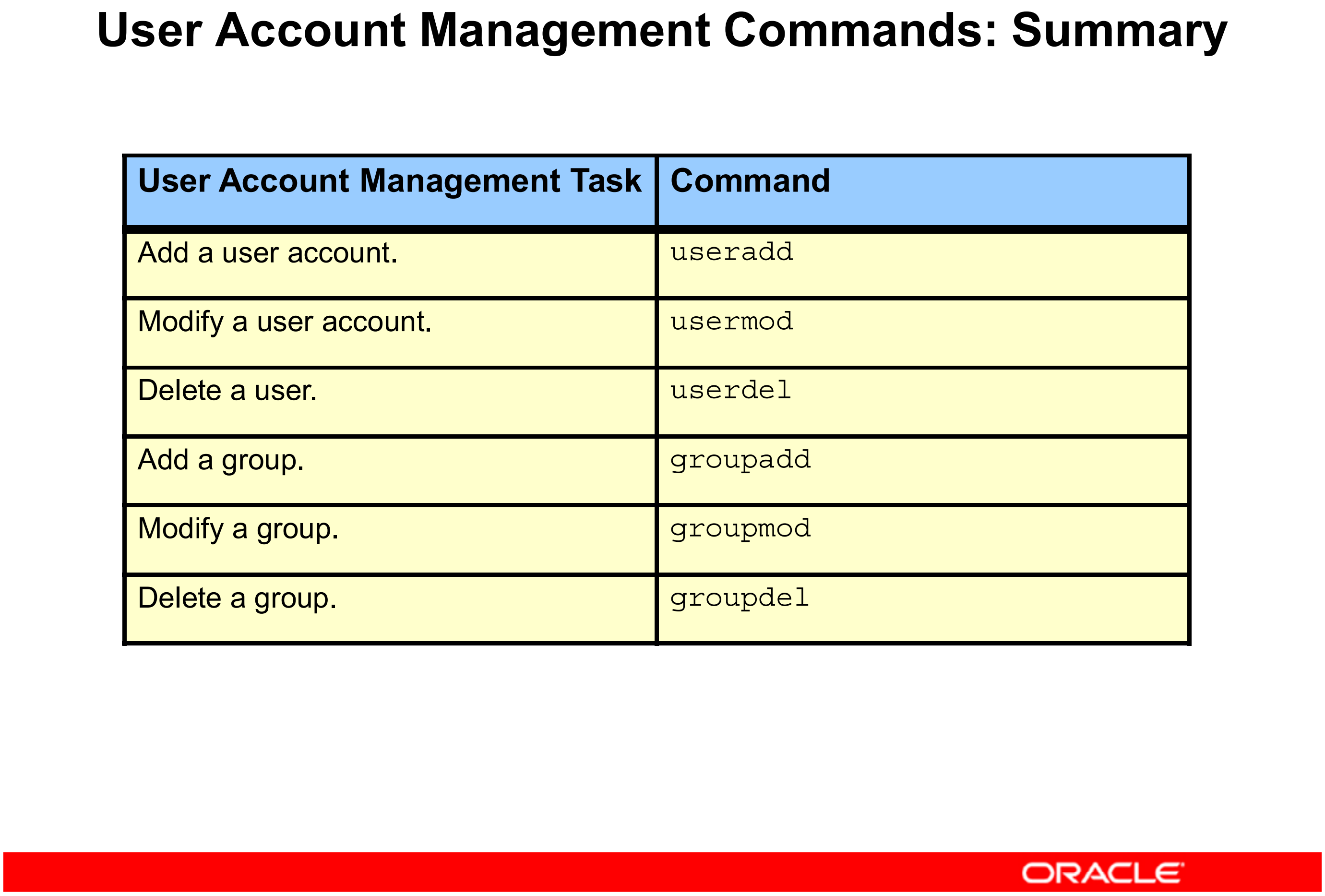
```
# grep quality /etc/group
quality::130:
# groupdel quality
Found group in files repository.
# grep quality /etc/group
# grep 130 /etc/group
```

The `quality` group (GID 130) has been removed as part of a recent organizational restructuring, so you have been directed to delete this group from the system. To do this, you must reassign the user accounts for users who were members of this group to a valid group, and then delete the group entry.

To reassign a user to a valid group, use the `usermod -u` *`UID`* `-g` *`GID loginname`* command. In the example, you reassign the user `jdoe` (UID `1004`) to a group called `hitech` that has a group ID number of `120`. To check that `jdoe` has been reassigned to a valid primary group, you run `grep /etc/passwd` with the user's login name. The output for this command confirms that `jdoe` is now part of the `hitech` group.

After you have reassigned the user accounts, you can delete the group entry. To do this, use the `groupdel` command, followed by the group name. In the example, you are deleting the `quality` group. First, you verify that the group exists, and it does. You then verify that the group entry has been deleted in the `/etc/group` file by running the `grep /etc/group` command with the group name that you just deleted. No entry is returned. You do one last check by running the `grep /etc/group` command with the deleted group's GID. Again, no entry is returned. You have successfully deleted the group entry.

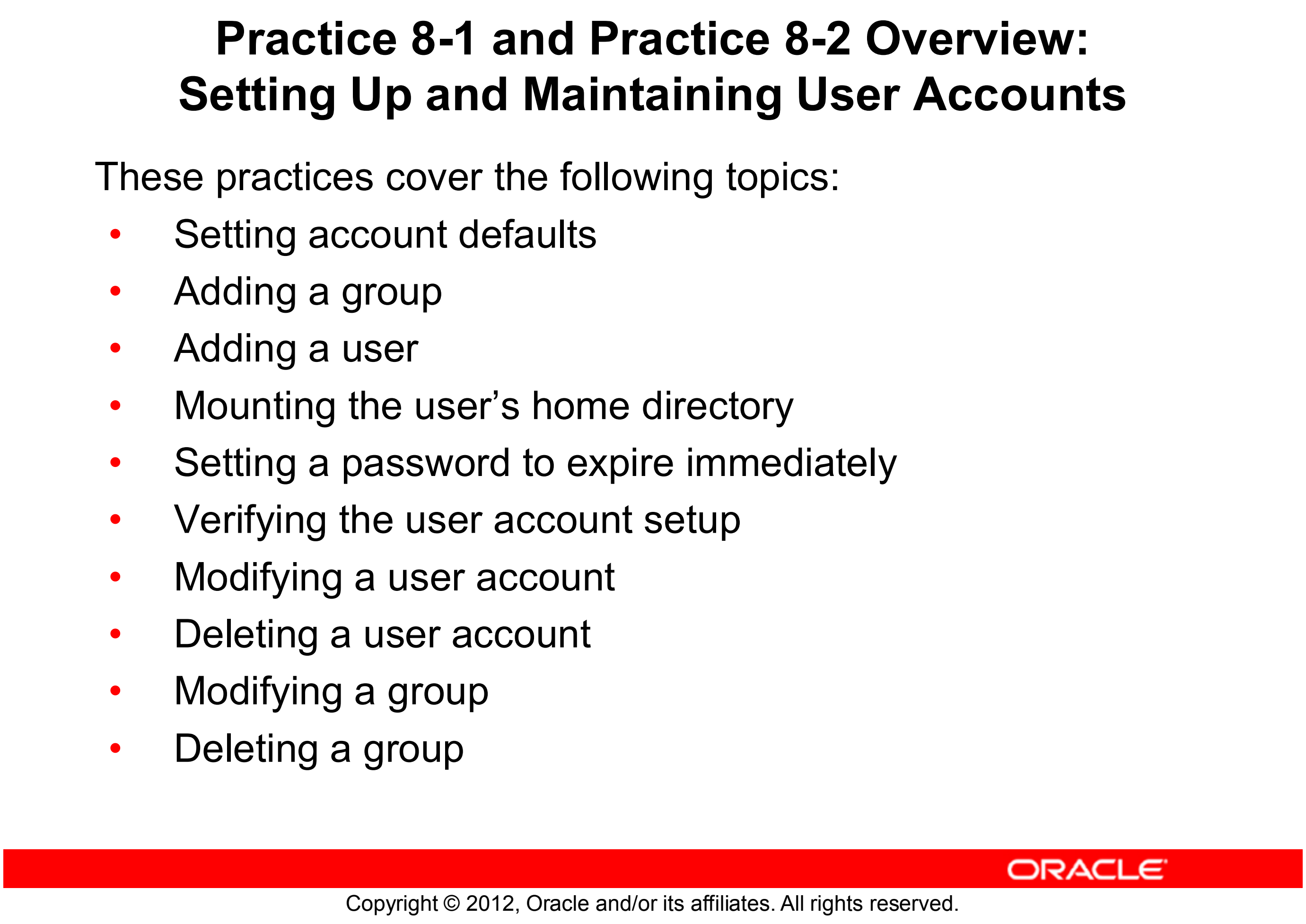
**Note:** The `groupdel` utility deletes a group entry from the system and makes the appropriate changes to the `/etc/group` file.



# User Account Management Commands: Summary

| User Account Management Task | Command |
|---|---|
| Add a user account. | `useradd` |
| Modify a user account. | `usermod` |
| Delete a user. | `userdel` |
| Add a group. | `groupadd` |
| Modify a group. | `groupmod` |
| Delete a group. | `groupdel` |

The table in the slide summarizes the user account management commands by task.



# Practice 8-1 and Practice 8-2 Overview: Setting Up and Maintaining User Accounts

These practices cover the following topics:

- Setting account defaults
- Adding a group
- Adding a user
- Mounting the user’s home directory
- Setting a password to expire immediately
- Verifying the user account setup
- Modifying a user account
- Deleting a user account
- Modifying a group
- Deleting a group

In the practices this lesson, you will perform the following tasks:

- **Practice 8-1:** Setting up user accounts
- **Practice 8-2:** Maintaining user accounts
- **Practice 8-3:** Managing user initialization files
- **Practice 8-4:** Exploring shell metacharacters and user quotas

You will find Practices 8-1 and 8-2 in your *Activity Guide*. It should take about 45 minutes to complete Practice 8-1 and 35 minutes to complete Practice 8-2.

# Lesson Agenda

- Planning for User Administration
- Setting Up User Accounts
- Maintaining User Accounts
- Managing User Initialization Files
- Using Shell Metacharacters and Configuring User Disk Quotas

ORACLE

# Managing User Initialization Files

- Setting up system-wide initialization files
- Setting up the user initialization files
- Customizing the user’s work environment

# Oracle Solaris 11 Shell Features

| Shell | Path | Comments |
|---|---|---|
| Bourne-Again Shell (`bash`) | `/usr/bin/bash` | Default shell for users that are created by an installer, as well as the `root` role |
| Korn Shell | `/usr/bin/ksh` | `ksh93` is the default shell in this Oracle Solaris release |
| C Shell and enhanced C Shell | `/usr/bin/csh` and `/usr/bin/tcsh` | C Shell and enhanced C Shell |
| POSIX-compliant Shell | `/usr/xpg4/bin/sh` | POSIX-compliant shell |
| Z Shell | `/usr/bin/zsh` | Z Shell |

The user account that is created when you install the Oracle Solaris release is assigned the GNU Bourne-Again Shell (`bash`) by default. The standard system shell, `bin/sh`, is the Korn Shell 93 (`ksh93`).

The table in the slide describes the shell options that are supported in this release.

**Note:** The Z Shell (`zsh`) and the enhanced C Shell (`tsch`) are not installed on your system by default. To use either of these shells, you must first install the required software packages. In this course, the focus is on the default shells, `bash` and `ksh`.

# Working with the bash and ksh93 Shells

Both shells feature:

- Command-line editing
- Command history on a per-user basis
- Environment variables
  - To view a list of bash variables, use the declare command.
  - To view a list of ksh93 variables, use the set command.

Both the bash and ksh93 shells feature command-line editing, which means that you can edit commands before executing them. To change to a different shell, you type the path of the shell that you want to use. To exit a shell, you type exit.

Both shells record a history of all the commands that you run. This history is kept on a per-user basis, which means that history is persistent between login sessions and is representative of all your login sessions.

The bash and ksh93 shells store special variable information that is known to the shell as an environment variable. To view a complete list of the current environment variables for the bash shell, use the declare command. To see the current environment variables for the ksh93 shell, use the set command.

The shells support two types of variables:

- **Environment variables:** Variables that provide information about the user's environment to every shell program that is started
- **Shell (local) variables:** Variables that affect only the current shell

The following is a list of environment and shell variables:

- CDPATH: Sets a variable used by the cd command
- HOME: Sets the path to the user's home directory
- LANG: Sets the locale
- LOGNAME: Defines the name of the user that is currently logged in. The default value of LOGNAME is set automatically by the login program to the username specified in the passwd file. You should only need to refer to and not reset this variable.
- MAIL: Sets the path to the user's mailbox
- MANPATH: Sets the hierarchies of the man pages that are available
- PATH: Specifies, in order, the directories that the shell searches to find the program to run when the user types a command. If the directory is not in the search path, users must type the complete path name of a command. As part of the login process, the default PATH is automatically defined and set as specified in .profile.

  **Note:** The order of the search path is important. When identical commands exist in different locations, the first command that is found with that name is used. For example, suppose that PATH is defined in the shell syntax as PATH=/bin:/usr/bin:/usr/sbin:$HOME/bin and a file named sample resides in both /usr/bin and /home/jean/bin. If the user types the command sample without specifying its full path name, the version found in /usr/bin is used.
- PS1: Defines the shell prompt for the bash or ksh93 shell
- SHELL: Sets the default shell that is used by make, vi, and other tools
- TERMINFO: Names a directory where an alternate terminfo database is stored. Use the TERMINFO variable in either the /etc/profile or /etc/.login file. For more information, see the terminfo(4)man page. When the TERMINFO environment variable is set, the system first checks the TERMINFO path defined by the user. If the system does not find a definition for a terminal in the TERMINFO directory defined by the user, it searches the default directory, /usr/share/lib/terminfo, for a definition. If the system does not find a definition in either location, the terminal is identified as "dumb."
- TERM: Defines the terminal. This variable should be reset in either the /etc/profile or /etc/.login file. When the user invokes an editor, the system looks for a file with the same name that is defined in this environment variable. The system searches the directory referenced by TERMINFO to determine the terminal characteristics.
- TZ: Sets the time zone. The time zone is used to display dates, for example, in the ls -l command. If TZ is not set in the user's environment, the system setting is used. Otherwise, Greenwich Mean Time is used.

**Note:** Environment variables do not persist between sessions. To set up environment variables that remain consistent between logins, you must make the changes in the .bashrc file. For more information about the environment variables, see the bash(1) man page.

# Initialization Files

Oracle Solaris 11 provides two types of initialization files:

- System-wide initialization files: Enable you to introduce new functionality to the user’s work environment
- User initialization files: Enable both you and the user to customize the user’s work environment

When users log in to the system, their login shells look for and execute two different types of initialization files: system-wide initialization files and user initialization files. System-wide initialization files enable you to introduce new functionality to the user’s work environment, while enabling the user to customize the user’s initialization file. The user initialization files can be customized by both the administrator and the user.

# System-Wide Initialization Files

- You are responsible for maintaining the system-wide initialization files.
- System-wide initialization files:
  - Provide an environment for all users who log in to the system
  - Reside in the `/etc` directory: `/etc/profile` and `/etc/.login`

The system administrator is responsible for maintaining the system-wide initialization files in accordance with the needs and requirement of the users on the system. These files provide an environment for the entire community of users who log in to the system. The system-wide initialization files reside in the `/etc` directory. The `/etc/profile` file and the `/etc/.login` file are the two main system-wide initialization files. The `bash` and `ksh93` login shells look for and execute the site initialization file `/etc/profile` during login. The `/etc/.login` file is used by `cshell`.

Default versions of the system-wide initialization files are used when the operating system is first installed. You should modify the default files only if directed to do so by a senior system administrator.

**Note:** The default files `/etc/profile` and `/etc/.login` check disk usage quotas, print the message of the day from the `/etc/motd` file, and check for mail. None of the messages are printed to the screen if the `.hushlogin` file exists in the user's home directory.

You can customize a site initialization file the same way that you customize a user initialization file. These files typically reside on a server, or a set of servers, and appear as the first statement in a user initialization file. Also, each site initialization file must be the same type of shell script as the user initialization file that references it.

# Viewing the Default `/etc/profile` Site Initialization File

To view the `/etc/profile` file, use `more /etc/profile`.

```
$ more /etc/profile
<output is presented in the Notes>
```

To view the `/etc/profile` file, use the `more /etc/profile` command, as shown in the example. Take a moment to familiarize yourself with the content of this file.

```
<header and copyright content omitted for training purposes>
# The profile that all logins get before using their own .profile.
trap ""  2 3
export LOGNAME PATH

if [ "$TERM" = "" ]
then
     if /bin/i386
     then
            TERM=sun-color
     else
            TERM=sun
     fi
     export TERM
fi
#    Login and -su shells get /etc/profile services.
#    -rsh is given its environment in its .profile.
case "$0" in
-sh | -ksh | -ksh93 | -jsh | -bash | -zsh)

     if [ ! -f .hushlogin ]
     then
            /usr/sbin/quota
            #      Allow the user to break the Message-Of-The-Day only.
            trap "trap '' 2"  2
            /bin/cat -s /etc/motd
            trap "" 2

            /bin/mail -E
            case $? in
            0)
                   echo "You have new mail."
                   ;;
            2)
                   echo "You have mail."
                   ;;
            esac
     fi
esac
umask 022
trap  2 3
```

# Viewing the Default `/etc/.login` Site Initialization File

To view the `/etc/.login` file, use `more /etc/.login`.

```
# more /etc/.login
<output is presented in the Notes>
```

To view the `/etc/.login` file, use the `more /etc/.login` command, as shown in the example. Take a moment to familiarize yourself with the content of this file.

**Note:** `/etc/.login` is the system-wide profile for C shell.

```
<head content omitted for training purposes>
# Copyright 1998 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident     "%Z%%M%     %I%   %E% SMI"

# The initial machine wide defaults for csh.

if ( $?TERM == 0 ) then
    if { /bin/i386 } then
          setenv TERM sun-color
    else
          setenv TERM sun
    endif
else
    if ( $TERM == "" ) then
          if { /bin/i386 } then
                setenv TERM sun-color
          else
                setenv TERM sun
          endif
    endif
endif

if (! -e .hushlogin ) then
    /usr/sbin/quota
    /bin/cat -s /etc/motd
    /bin/mail -E
    switch ( $status )
    case 0:
          echo "You have new mail."
          breaksw;
    case 2:
          echo "You have mail."
          breaksw;
    endsw
endif
```

# Modifying the System-Wide Initialization Files

To edit a system-wide initialization file, use `vi` or any other UNIX editor.

```
# vi /etc/.login
```

To make the modified file and configuration available to the users on the system, use the `source` command.

```
# source /etc/.login

<or>

# . .login
```

You can modify the system-wide initialization files with `vi` or any other UNIX editor, as shown in the first example in the slide.

After you have finished modifying the file, you can make the system read the modified file and make the configuration available to the users on the system by using the `source` command or the `.  .login` command, as shown in second example in the slide.

# User Initialization Files

When you create a user account by using `useradd -D`, you can modify the contents of the default file or accept the system default files.

```
# useradd -D
group=staff,10  project=default,3  basedir=/home
skel=/etc/skel  shell=/usr/bin/bash  inactive=0
expire=  auths=  profiles=  roles=  limitpriv=
defaultpriv=  lock_after_retries=
```

The user initialization files:

- Define a user’s work environment
- Can be changed or customized by the owner or root user

The Oracle Solaris 11 OS provides default user initialization files for each shell in the `/etc/skel` directory on each system. When you create a new user account for a user by using the `useradd -D` command, all the initialization files from the `/etc/skel` directory are automatically copied to the user’s new home directory. As you saw when you were using the `useradd -D` command and looking at the user accounts default file, you can modify the contents of these files for the user or you can choose to use the system default files.

The primary purpose of the user initialization files is to define the characteristics of a user’s work environment, such as the command-line prompt, the environment variables, and the windowing environment. Only the owners of the files or the root user can change or customize the content of these files.

# User Initialization Files

The initialization files presented in the following table are necessary for each primary shell.

| Shell | User Initialization File | Purpose |
|---|---|---|
| `bash` | `/etc/profile`<br>`$HOME/.bash_profile`<br>`$HOME/.bash_login`<br>`$HOME/.profile` | Defines the user’s environment at login |
| `ksh93` | `/etc/profile`<br>`$HOME/.profile` | Defines the user’s environment at login |
| | `$ENV` | Defines the user’s environment at login in the file, and is specified by the Korn shell’s `ENV` environment variable |

The table in the slide shows the initialization files that are necessary for each primary shell that is available in the Oracle Solaris OS.

When a user logs in to the system, the system invokes the user’s login shell program. The shell program looks for its initialization files in a specific order, executes the commands contained in each file, and displays the shell prompt on the user’s screen.

# Customizing the User's Work Environment

Initialization file templates:

- Are located in `/etc/skel`
- Can be modified by the system administrators to create:
  - A standard working environment that is common to all users
  - Working environments for different types of users
- Can be used by the user to further customize environments

| Shell | Initialization File Templates | User Initialization File |
|---|---|---|
| `bash` | `/etc/skel/local.profile` | `$HOME/.profile` |
| `ksh93` | `/etc/skel/local.profile` | `$HOME/.profile` |

The Oracle Solaris OS provides a set of initialization file templates. The `/etc/skel` directory contains the initialization file templates. The table in the slide shows the default initialization file templates and the user initialization files for the `bash` and `ksh93` shells.

You can use these files as a starting point, and then modify them to create a standard set of files that provide the work environment that is common to all users. You can also modify these files to provide the working environment for different types of users.

Users can then edit their initialization files to further customize their environments for each shell.

## Accessing the Initialization File Templates

- To see the initialization file templates in /etc/skel, change to the /etc/skel directory, and then run ls.

```
# cd /etc/skel
# ls
local.cshrc  local.login  local.profile
```

- To see the contents of a template, use more template_name.

```
# more local.profile
<header output omitted>
stty istrip
PATH=/usr/bin:/usr/ucb
export PATH
#
```

To access the initialization file templates, you need to first see the templates that are available in the /etc/skel directory. Change directories to /etc/skel, and then run the ls command, as shown in the first example in the slide. Here you can see that three initialization file templates are available: local.cshrc, local.login, and local.profile.

To see the contents of a template, you can run the more command, followed by the template name, as shown in the second example in the slide. The contents of the template are purposely sparse to enable users to customize their work environments as they desire.

# Setting Environment Variables in the User Initialization Files

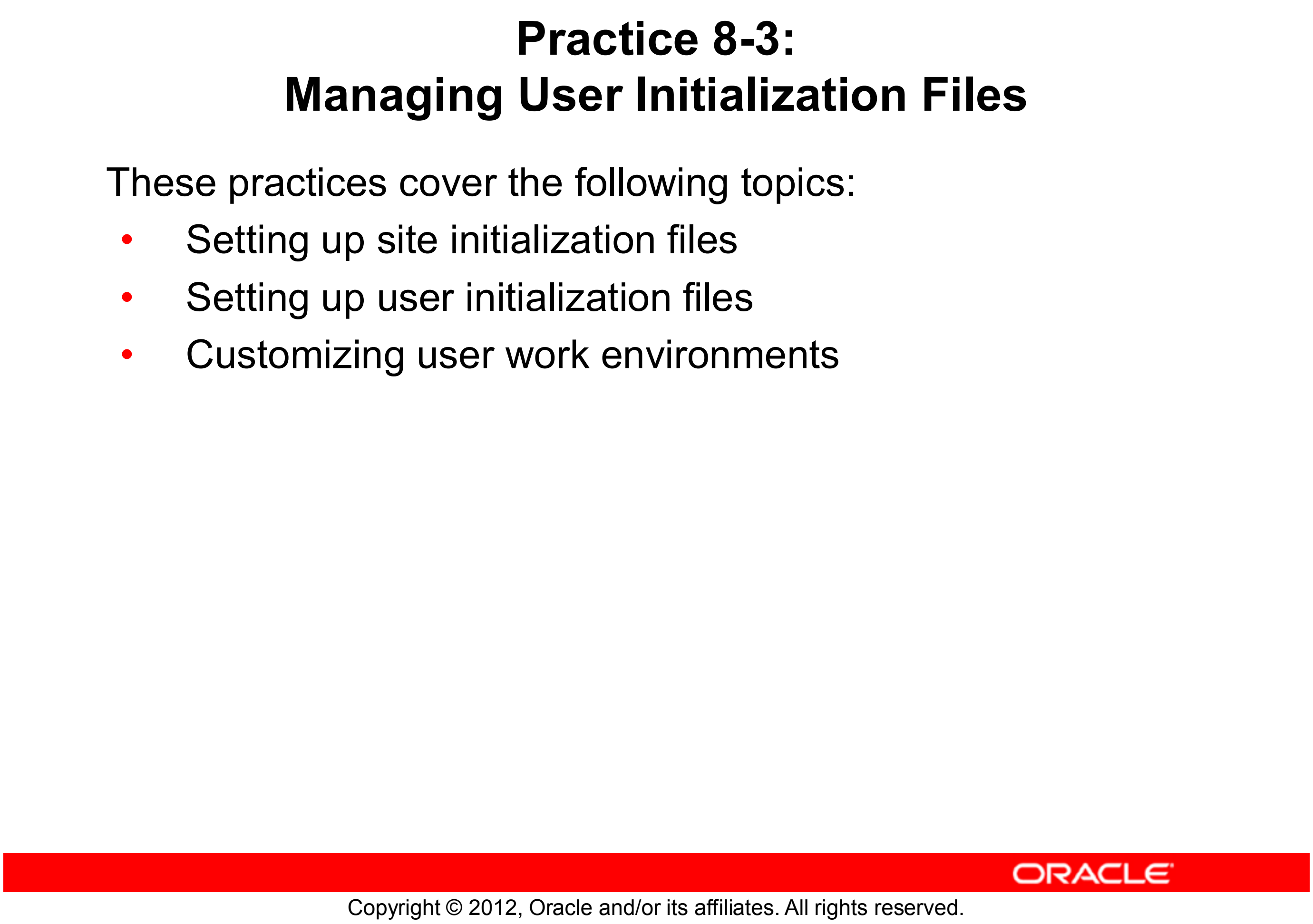
To set environment variables in the user initialization files, use `VARIABLE=value ; export VARIABLE`.

```
PS1="$HOSTNAME "; export PS1
```

To modify the template, use the `vi` editor (or any UNIX editor) as you did with the site initialization file.

To set the environment variables within the files, use the *`VARIABLE`*`=value ; export` *`VARIABLE`* commands. In the example, you set the command prompt variable `PS1`.

**Note:** For a list of environment variables for the `bash` and `ksh93` shells, see the note for the slide titled "Working with the `bash` and `ksh93` Shells" earlier in this lesson. For complete information about all the variables used by the default shells, see the following man pages: `sh`(1), `ksh`(1), `csh`(1), `zsh`(1), `bash`(1), and `tcsh`(1).



# Practice 8-3: Managing User Initialization Files

These practices cover the following topics:

- Setting up site initialization files
- Setting up user initialization files
- Customizing user work environments

This practice should take about 35 minutes to complete.

# Lesson Agenda

- Planning for User Administration
- Setting Up User Accounts
- Maintaining User Accounts
- Managing User Initialization Files
- Using Shell Metacharacters and Configuring User Disk Quotas

Now that you know how to manage user initialization files, you now learn to use shell metacharacters and configure user disk quotas.

# Using Shell Metacharacters and Configuring User Disk Quotas

- Using shell metacharacters
- Configuring user disk quotas

In this section, you see how to use shell metacharacters and configure user disk quotas. As you saw in the last section, the shell is the user’s window to the system.

Knowing how to use shell metacharacters will enable you to move within the system more easily and to locate files and directories more quickly.

One of your primary responsibilities as a system administrator is to administer user accounts. You just spent some time learning how to set up the user’s environment, including  how to create the user’s home directory and file system. Now it is time to learn how to control or limit the amount of disk space each user can consume. If you support a lot of users and are responsible for managing data storage, you can imagine the benefit of being able to control the amount of file system space each user uses.

# Using Shell Metacharacters

- Path name metacharacters include:
  - The tilde (~) character
  - The dash (-) character
- File name substitution metacharacters include:
  - The asterisk (*) character
  - The question mark (?) character
  - The bracket ( [] ) characters

Shell metacharacters are specific characters, generally symbols, that have special meaning for the shell. Two types of metacharacters are path name metacharacters, which include the tilde (~) and dash (-) characters, and file name substitution metacharacters, which include the asterisk (*), question mark (?), and bracket ( [ ] ) characters.

You will now look at each of these characters in turn, beginning with the tilde (~) character.

**Caution:** Do not use these metacharacters when creating file and directory names. These characters hold special meaning to the shell.

# Using the Tilde (~) Character

- The tilde character represents the home directory of the current user.
- To change directories, use `cd ~/directory_name`.

```
$ cd ~/dir1
$ pwd
 /home/student/dir1/
$
```

The tilde (~) character represents the home directory of the current user. It is a substitution that equates to the absolute path name of the user's home directory.

To change directories, use the `cd` command, followed by the tilde (~) character and */directory_name*.

# Using the Dash (–) Character

- Represents the previous working directory
- Is used to switch between two specific directories

```
$ cd
$ pwd
/home/student
$ cd /tmp
$ pwd
/tmp
$ cd -
/home/student
$ cd -
/tmp
$
```

The dash (`-`) character in the shell represents the previous working directory. You can use the dash character to switch between two specific directories. The shell automatically displays the current directory path.

The example in the slide shows how to switch between the `/export/home/student` and `/tmp` directories by using the dash (`-`) character.

# Using the Asterisk (*) Character

- The asterisk character represents zero or more characters, except the leading period (`.`) of a hidden file.
- To list all the files and directories that start with a specific letter, followed by zero or more other characters, use `ls letter*`.

```
$ cd
$ ls f*
feathers file.1 file.2 file.3 file4 fruit2
feathers_6 file1 file2 file3 fruit
$
```

The asterisk `(*)` character is also called the wildcard character and represents zero or more characters, except the leading period (`.`) of a hidden file.

To list all the files and directories that start with a specific letter followed by zero or more other characters, use the `ls` command, followed by the letter and an asterisk (`*`). In the example in the slide, you look for files that begin with the letter `f`.

Another example would be if you wanted to list all the files and directories that end with the number `3`, preceded by zero or more characters. To do this, you would use the following command:

```
$ ls *3
file.3 file3
dir3:
cosmos moon planets space sun vegetables
$
```

# Using the Question Mark (?) Character

- The question mark character represents any single character, except the leading period (.) of a hidden file.
- To list all the files and directories that start with the string dir and are followed by one other character, use ls dir?.

```
$ ls dir?
dir1:
coffees fruit trees
dir2:
beans notes recipes
dir3:
cosmos moon planets space sun vegetables
dir5:
$
```

The question mark (?) character represents any single character, except the leading period (.) of a hidden file. The question mark (?) character is also called a wildcard character.

For example, to list all the files and directories that start with the string dir and are followed by one other character, use the command ls dir?.

# Using the Bracket ( [] ) Characters

- Represents a set or range of characters for a single character position
  - A set of characters is any number of specific characters.
  - A range of characters is a series of ordered characters.

```
$ ls [a-f]*
brands dante_1 file.1 file2 file4
celery feathers file1 file.3 fruit
dante feathers_6 file.2 file3 fruit2
dir1:
coffees fruit trees
dir10:
planets
dir2:
beans notes recipes
dir3:
cosmos moon planets space sun vegetables
$
```

The bracket ( [] ) characters represent a set or range of characters for a single character position.

A set of characters is any number of specific characters (for example, [acb]). The characters in a set do not generally need to be in any order. For example, [abc] is the same as [cab].

A range of characters is a series of ordered characters. A range lists the first character, a hyphen (-), and the last character (for example, [a-z] or [0-9]). When you specify a range, arrange the characters in the order that you want them to appear in the output. Use [A-Z] or [a-z] to search for any uppercase or lowercase alphabetical character, respectively. For example, to list all the files and directories that start with the letters a through f, you would use the command ls [a-f]*, as shown in the example in the slide.

Another example would be to list all the files and directories that start with the letters f or p:

```
$ ls [fp]*
feathers file.1 file.2 file.3 file4 fruit2
practice1:
appointments file.1 file.2 play
$
```

# Configuring User Disk Quotas

The ZFS quota property:

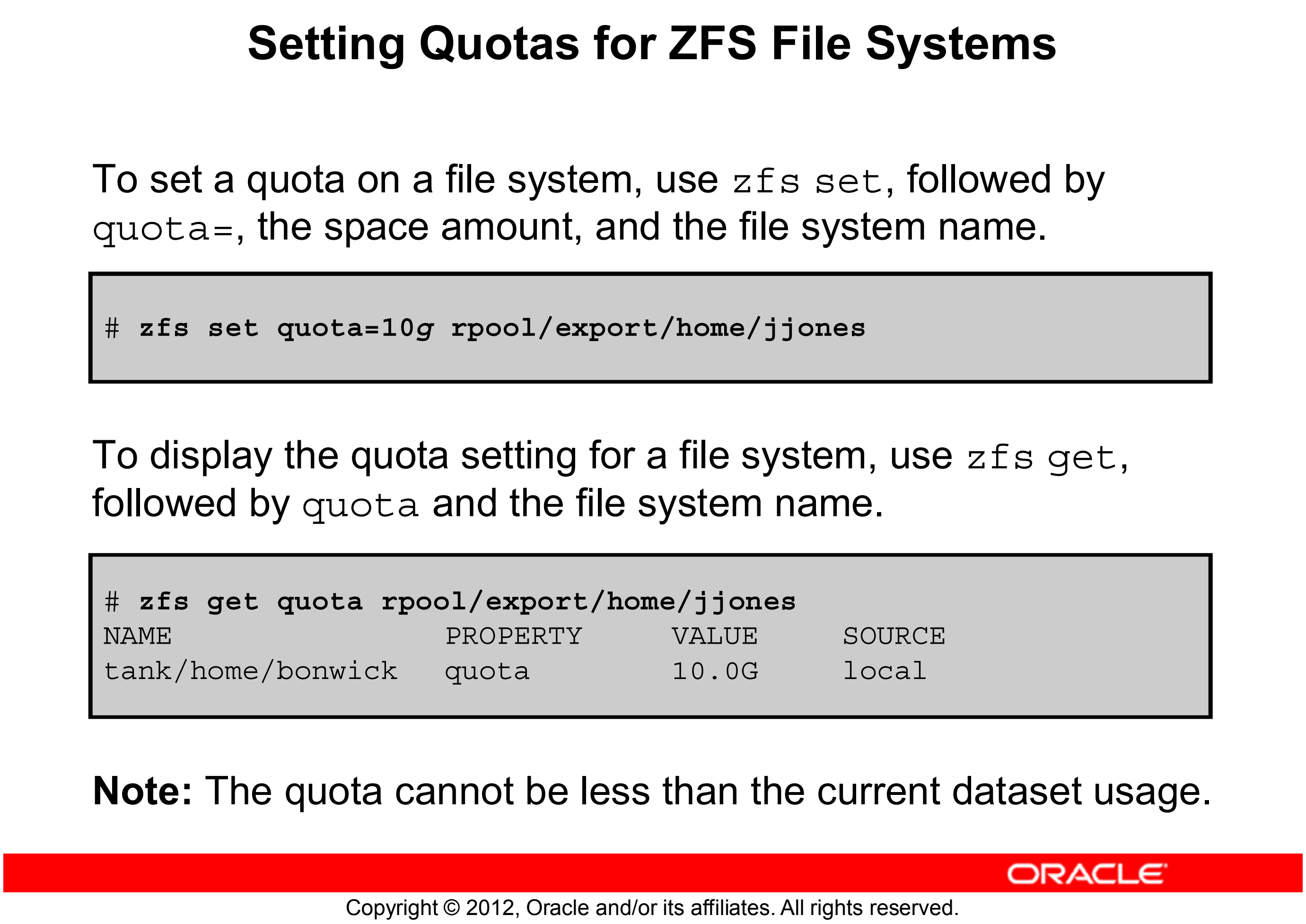
- Sets a space limit on the amount of space used by a file system and user
- Applies to:
  - The dataset that it is set on
  - All descendents of that dataset

Now you will learn about configuring user disk quotas.

ZFS has a `quota` property that you can use to set a limit on the amount of space that a file system can use and the amount of space that a user can use on a file system. User quotas provide a way to more easily manage disk space with many user accounts.

**Note:** ZFS also supports group quotas. To learn more about configuring group quotas, see the section titled “Managing Oracle Solaris ZFS File Systems” in *Oracle Solaris Administration: ZFS File Systems*.

The property applies to the dataset that it is set on and all the descendents of that dataset. For example, if a quota is set on the `tank/home` dataset, the total amount of space used by `tank/home` and all of its descendents cannot exceed the quota.



# Setting Quotas for ZFS File Systems

To set a quota on a file system, use `zfs set`, followed by `quota=`, the space amount, and the file system name.

```
# zfs set quota=10g rpool/export/home/jjones
```

To display the quota setting for a file system, use `zfs get`, followed by `quota` and the file system name.

```
# zfs get quota rpool/export/home/jjones
NAME                PROPERTY     VALUE     SOURCE
tank/home/bonwick   quota        10.0G     local
```

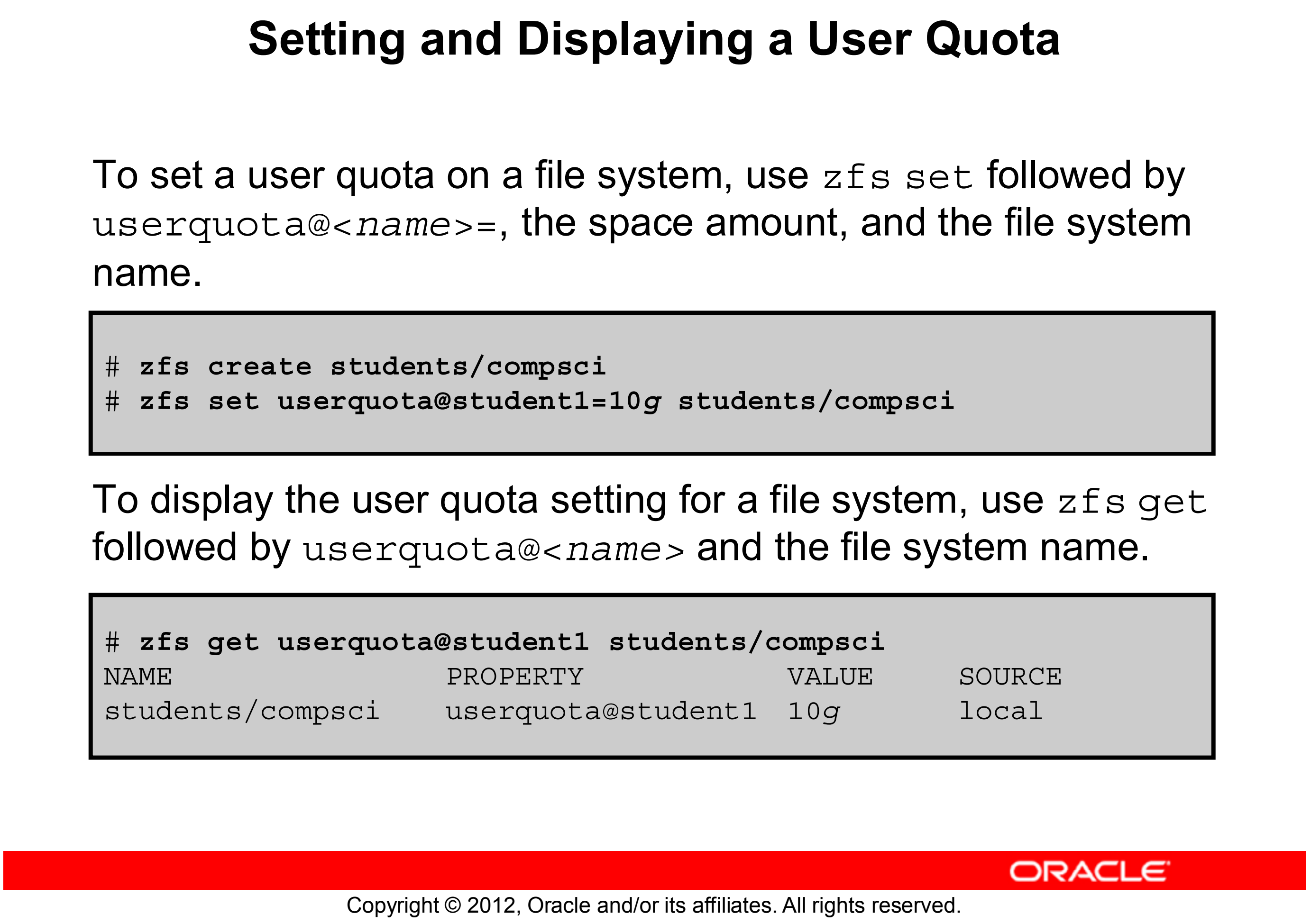
**Note:** The quota cannot be less than the current dataset usage.

ZFS quotas can be set and displayed by using the `zfs set` and `zfs get` commands.

In the first example, you set a quota of 10 GB on `rpool/export/home/jjones`. To do this you use the `zfs set` command, followed by `quota=10g` and the file system name.

In the second example, you display the results of the space allocation. To do this, you use the `zfs get` command, followed by the property name `quota` and the file system name.

**Note:** You cannot set a quota amount that is less than what is currently being used by a dataset.



# Setting and Displaying a User Quota

To set a user quota on a file system, use `zfs set` followed by `userquota@<name>=`, the space amount, and the file system name.

```
# zfs create students/compsci
# zfs set userquota@student1=10g students/compsci
```

To display the user quota setting for a file system, use `zfs get` followed by `userquota@<name>` and the file system name.

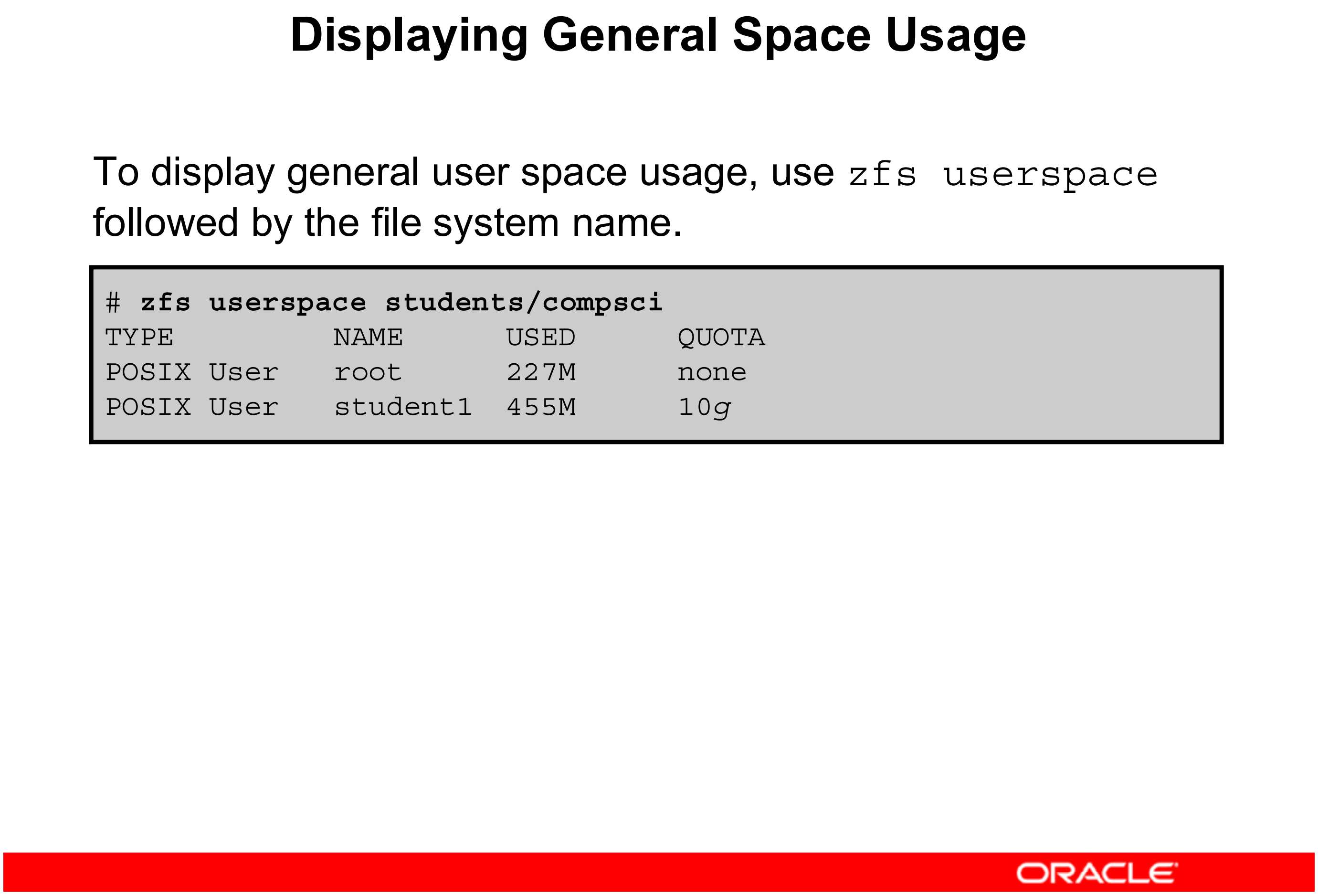
```
# zfs get userquota@student1 students/compsci
NAME                 PROPERTY             VALUE     SOURCE
students/compsci     userquota@student1   10g       local
```

You can set a *user* or *group* quota on the amount of space consumed by the files that are owned by a particular user or group.

You can set a user quota by using the `zfs set userquota` command followed by the amount of space that you want to allocate to the file system and the file system name. In the first example shown in the slide, you first create the file system `students/compsci`. Next, you set the user quota to 10 GB.

**Note:** The amount of space that you allocate for a home directory depends on the kinds of files the user creates, their size, and the number of files that are created.

To display the current user quota, use the `zfs get` command followed by the `userquota` (`userquota@<name>`) command and the file system name.



# Displaying General Space Usage

To display general user space usage, use `zfs userspace` followed by the file system name.

```
# zfs userspace students/compsci
TYPE          NAME      USED      QUOTA
POSIX User    root      227M      none
POSIX User    student1  455M      10g
```

You can display general user space usage by using the `zfs userspace` subcommand as shown in the example in the slide.

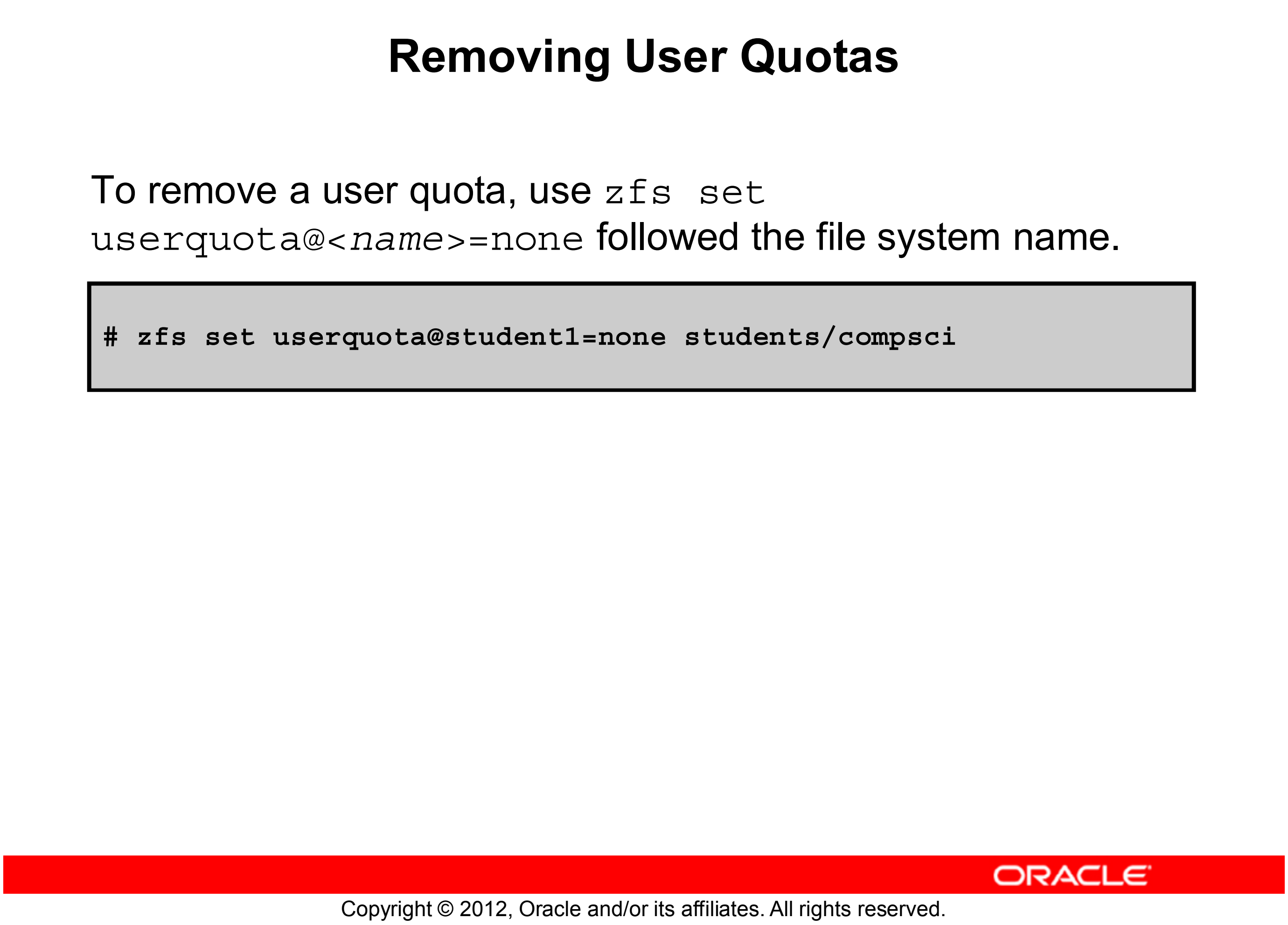
## Identifying Individual User Space Usage

To identify individual user space usage, use `zfs userused@<name>` followed by the file system name.

```
# zfs get userused@student1 students/compsci
NAME                PROPERTY            VALUE     SOURCE
students/compsci    userused@student1   455M      local
```

You can identify individual user space usage by using the `zfs get` command followed by `userused@<name>` and the file system name as shown in the example in the slide. Here you want to identify the individual user space usage for the `students/compsci` file system. You can see that 455 MB of space is being used.

**Note:** The user quota properties are not displayed by using the `zfs get all` *dataset* command that displays a listing of all file system properties.

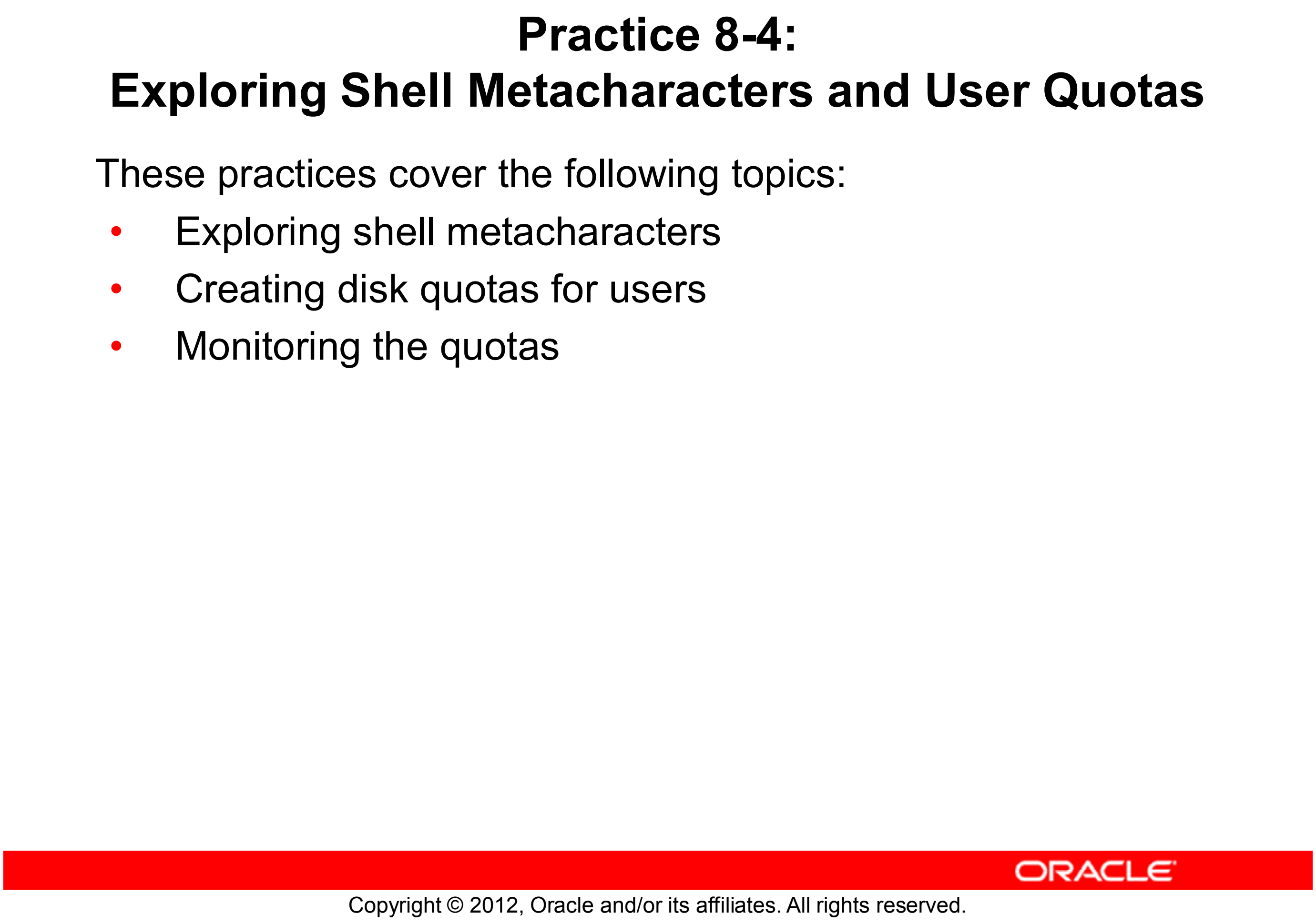


## Removing User Quotas

To remove a user quota, use `zfs set userquota@<name>=none` followed the file system name.

```
# zfs set userquota@student1=none students/compsci
```

You can remove a user quota by using the `zfs set` command to set the user quota property to `none` as shown in the example in the slide.



# Practice 8-4: Exploring Shell Metacharacters and User Quotas

These practices cover the following topics:

- Exploring shell metacharacters
- Creating disk quotas for users
- Monitoring the quotas

This practice should take about 35 minutes to complete.

# Summary

In this lesson, you should have learned to:

- Implement a plan for user administration
- Set up user accounts
- Manage user accounts
- Manage user initialization files
- Use shell metacharacters
- Configure user disk quotas

In this lesson, you were introduced to user administration. You were shown how to set up and manage user accounts. You also learned how to manage user initialization files, use shell metacharacters, and configure user disk quotas.

9

# Controlling Access to Systems and Files

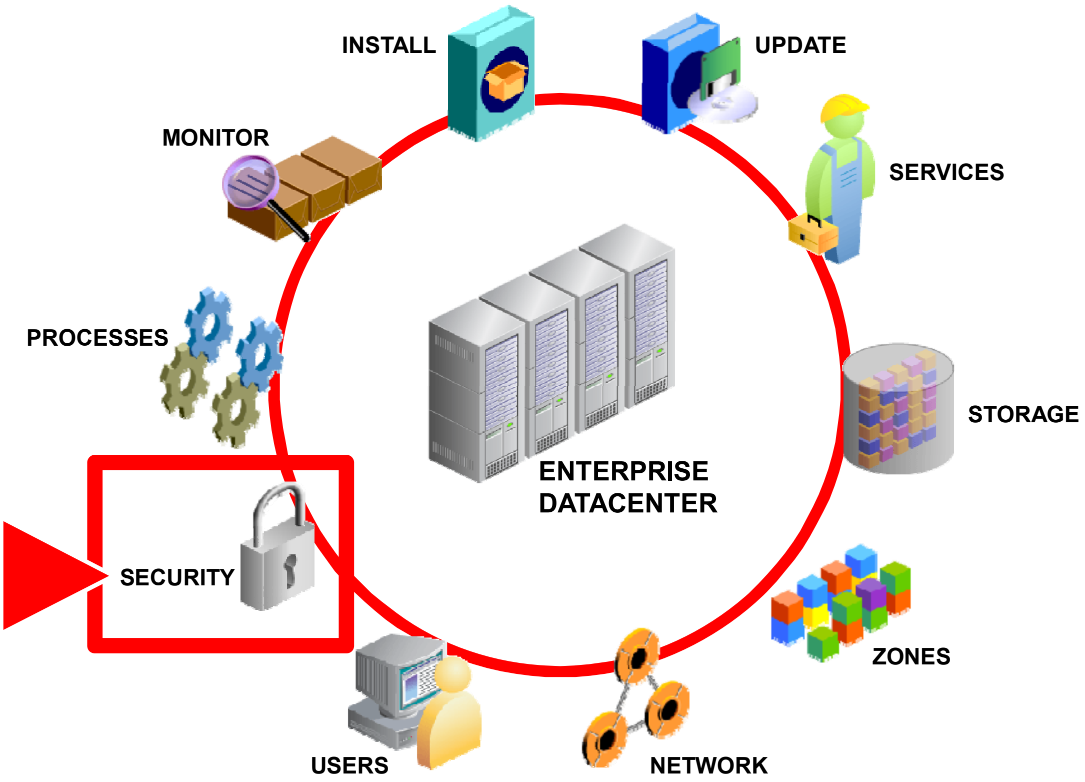
# Objectives

After completing this lesson, you should be able to:

- Implement a plan for system and file access control
- Control access to systems
- Control access to files
- Configure Secure Shell
- Use Secure Shell

In this lesson, you are presented with a plan for system and file access control. You learn how to control user, group, and superuser access to the system and to files. You also learn how to configure and use Secure Shell to control remote access to systems and files.

## Workflow Orientation



Before you start the lesson, orient yourself to where you are in the job workflow. In the lesson titled “Setting Up and Administering User Accounts,” you learned how to set up user accounts to enable users to have access to a system and to their own file system. In this lesson, you learn how to secure a user’s access to a system and to files, both locally and remotely.

# Lesson Agenda

- Planning for System and File Access Control
- Controlling Access to Systems
- Controlling Access to Files
- Configuring and Using Secure Shell

# Planning for System and File Access Control

The system and file access control plan addresses the requirements for controlling:

- User access to systems
- User access to files
- Remote access to systems and files

Your company has a vested interest in ensuring that its business data remains confidential and private. Oracle Solaris 11 provides a number of system security features that help your company keep its data secure. Based on its data protection needs and requirements and an existing security policy that outlines the organization’s security guidelines, your company has developed a multilayered plan that addresses both system and network security issues. Your involvement with the plan will be on the system side, specifically with securing both local and remote access to systems and files.

# Controlling Access to Systems

You can control a user’s access to the system by:

- Securing logins and passwords
- Changing the password algorithm
- Limiting and monitoring the superuser

There are multiple ways in which you can control access to a system. In this lesson, the focus is on three methods:

- Securing logins and passwords
- Changing the password algorithm
- Limiting and monitoring the superuser

You will now take a closer look at each, beginning with securing logins and passwords.

# Login and Password Security

- Use login control and password assignment to prevent unauthorized logins to a system or the network.
- The login command:
  - Verifies the username and password
  - Denies access to the system if the username and/or password are incorrect
- Ensure that all the accounts on a system have a password.
- Passwords are kept secure by being:
  - Encrypted
  - Kept in a separate file from the username and information

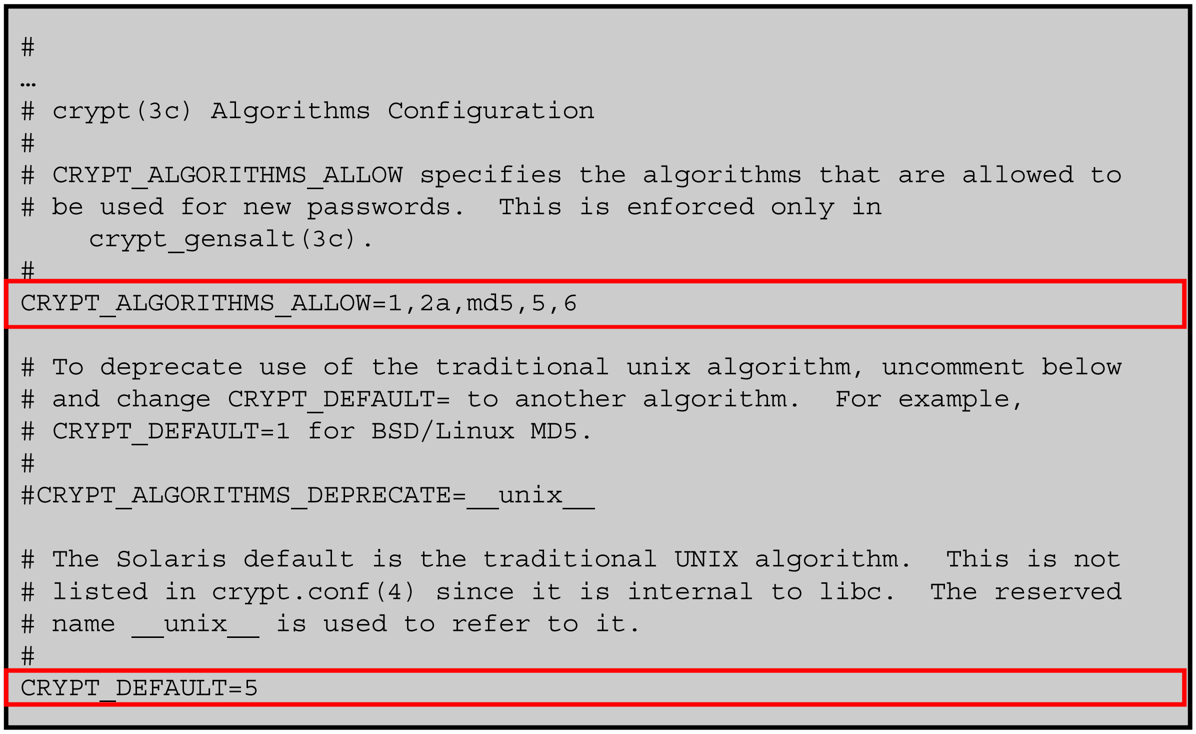
To prevent unauthorized logins to a system or the network, you can use password assignment and login control.

When a user logs in to a system, the login command verifies the username and password that were supplied by the user. If the username is not in the password file, the login command denies access to the system. If the password is not correct for the username that was specified, the login command denies access to the system. When the user supplies a valid username and its corresponding password, the system grants the user access to the system.

All accounts on a system must have a password. A password is a simple authentication mechanism. An account without a password makes your entire network accessible to an intruder who guesses a username.

As you recall from the lesson titled “Setting Up and Administering User Accounts,” passwords are initially created and encrypted when you set up a user account. If your network uses local files to authenticate users, the password information is kept in the system’s `/etc/passwd` and `/etc/shadow` files. The username and other information are kept in the `/etc/passwd` file. The encrypted password itself is kept in a separate shadow file, `/etc/shadow`. This security measure prevents a user from gaining access to the encrypted passwords. Whereas the `/etc/passwd` file is available to anyone who can log in to a system, only the superuser or an equivalent role can read the `/etc/shadow` file.

# Password Algorithms and the `/etc/security/policy.conf` File



```
#
…
# 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

# To deprecate use of the traditional unix algorithm, uncomment below
# and change CRYPT_DEFAULT= to another algorithm.  For example,
# CRYPT_DEFAULT=1 for BSD/Linux MD5.
#
#CRYPT_ALGORITHMS_DEPRECATE=__unix__

# The Solaris default is the traditional UNIX algorithm.  This is not
# listed in crypt.conf(4) since it is internal to libc.  The reserved
# name __unix__ is used to refer to it.
#
CRYPT_DEFAULT=5
```

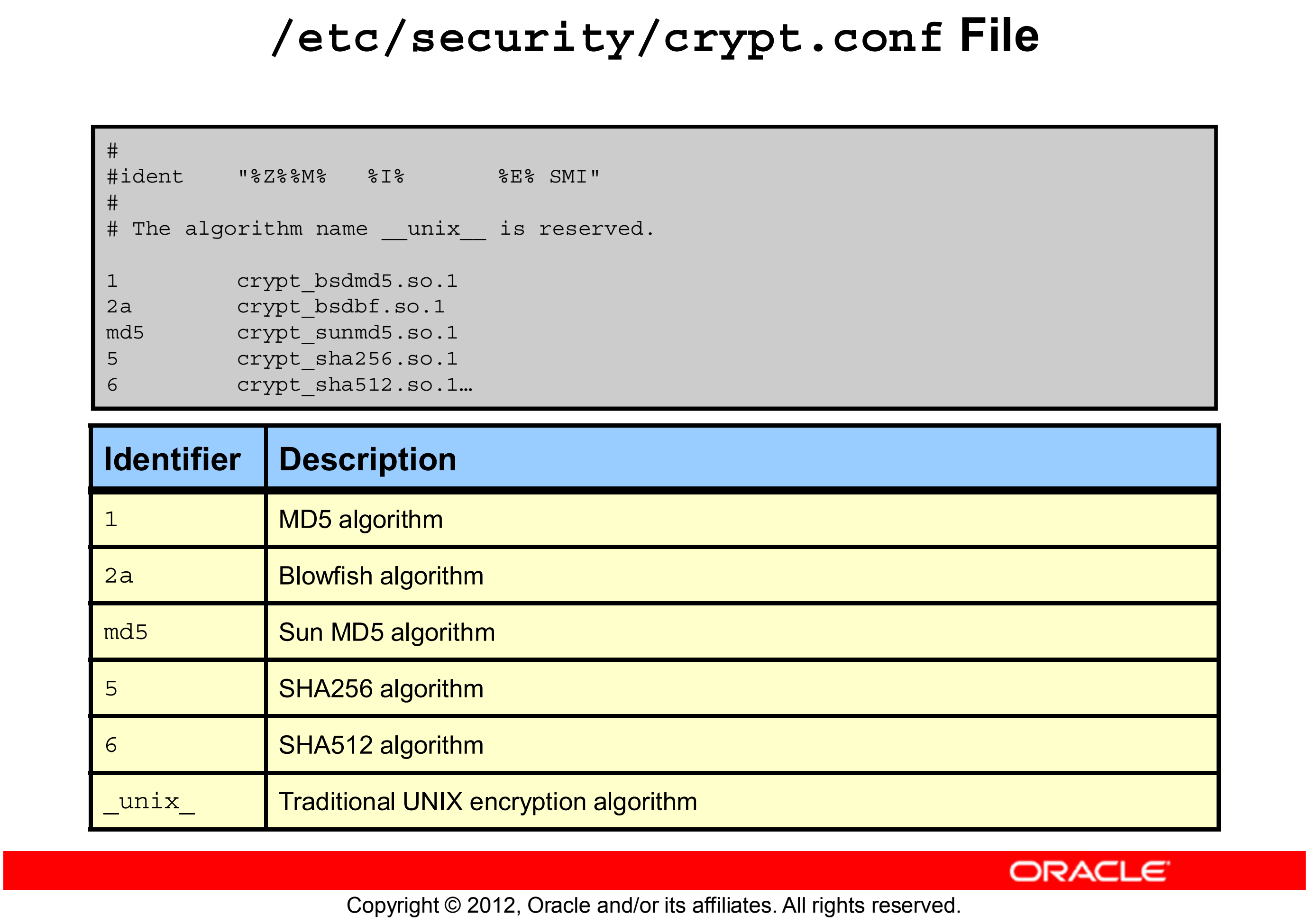
Oracle Solaris uses algorithms to encrypt user passwords. A strong password algorithm protects against brute force attacks.

As a system administrator, you can specify which algorithms configuration to use for your site by modifying the `/etc/security/policy.conf` file. The slide shows the default algorithms configuration in the `policy.conf` file.

In the `policy.conf` file, the algorithms are named by an identifier (for example, `1`, `2a`, `md5`, `5`, `6`, or `_unix_`).

**Note:** The identifiers and their descriptions are presented in the next slide.

When you change the value for `CRYPT_DEFAULT`, the passwords of new users are encrypted with the algorithm that is associated with the new value. You are shown how to change the password algorithm in subsequent slides.



# /etc/security/crypt.conf File

```
#
#ident    "%Z%%M%   %I%       %E% SMI"
#
# The algorithm name __unix__ is reserved.

1         crypt_bsdmd5.so.1
2a        crypt_bsdbf.so.1
md5       crypt_sunmd5.so.1
5         crypt_sha256.so.1
6         crypt_sha512.so.1…
```

| Identifier | Description |
|---|---|
| `1` | MD5 algorithm |
| `2a` | Blowfish algorithm |
| `md5` | Sun MD5 algorithm |
| `5` | SHA256 algorithm |
| `6` | SHA512 algorithm |
| `_unix_` | Traditional UNIX encryption algorithm |

The `/etc/security/crypt.conf` file contains the identifier-algorithm mapping. The identifiers and their descriptions, as presented in the table in the slide, are as follows:

- **`1:`** The MD5 algorithm that is compatible with the MD5 algorithms on BSD and Linux systems. For more information, see the `crypt_bsdmd5`(5) man page.
- **`2a:`** The Blowfish algorithm that is compatible with the Blowfish algorithm on BSD systems. For more information, see the `crypt_bsdbf`(5) man page.
- **`Md5:`** The Sun MD5 algorithm, which is considered stronger than the BSD and Linux version of MD5. For more information, see the `crypt_sunmd5`(5) man page.
- **`5:`** The SHA256 algorithm. SHA stands for Secure Hash Algorithm. This algorithm is a member of the SHA-2 family. SHA256 supports 255-character passwords. For more information, see the `crypt_sha256`(5) man page.
- **`6:`** The SHA512 algorithm. For more information, see the `crypt_sha512`(5) man page.
- **`__unix__:`** The traditional UNIX encryption algorithm. For more information, see the `crypt_unix`(5) man page.

# Superuser Limiting and Monitoring

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

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

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

As a system administrator, it is your responsibility to control and monitor system activity. You can control system activity by setting limits on who can use what resources. You can also log resource use, and you can monitor who is using the resources.

As you know from the lesson titled "Setting Up and Administering User Accounts," there are two common ways to access a system: by using a conventional user login or by using the root login. You also saw how the `su` command can be used to enable a user to run administrative commands without using the root account.

**Note:** The system identifies and tracks logins as either being from the original or real user ID (UID) or from an effective (or switched to) user ID (EUID). The same is true for an original or real group ID (GID) and the effective group ID (EGID). Generally speaking, the effective EUID or EGID is the same as the original UID or GID. One way to determine whether you are logged in as the real or effective user is to run the `who am i` command, which tells you the original user. Otherwise, use the `whoami` command to see the effective UID.

In the default system configuration, a user cannot remotely log in to a system as root. When logging in remotely, a user must log in with the user's username, and then use the `su` command to become root. In situations such as this, you can limit the superuser access and monitor who has been using the `su` command, especially those users who are trying to gain superuser access. In the next section, you are shown how to do this.

# Controlling Access to Files

To secure files and directories in Oracle Solaris, you can use:

- UNIX file permissions
- Access control lists (ACLs)

| Command | Description |
|---|---|
| `ls` | Lists the files in a directory and information about the files |
| `chown` | Changes the ownership of a file |
| `chgrp` | Changes the group ownership of a file |
| `chmod` | Changes permissions on a file. You can use either symbolic mode, which uses letters and symbols, or absolute mode, which uses octal numbers, to change the permissions on a file. |

In Oracle Solaris, files and directories can be secured through UNIX file permissions and through access controls lists (ACLs). Permissions restrict the users and groups that are permitted to read, write, or execute a file or search a directory.

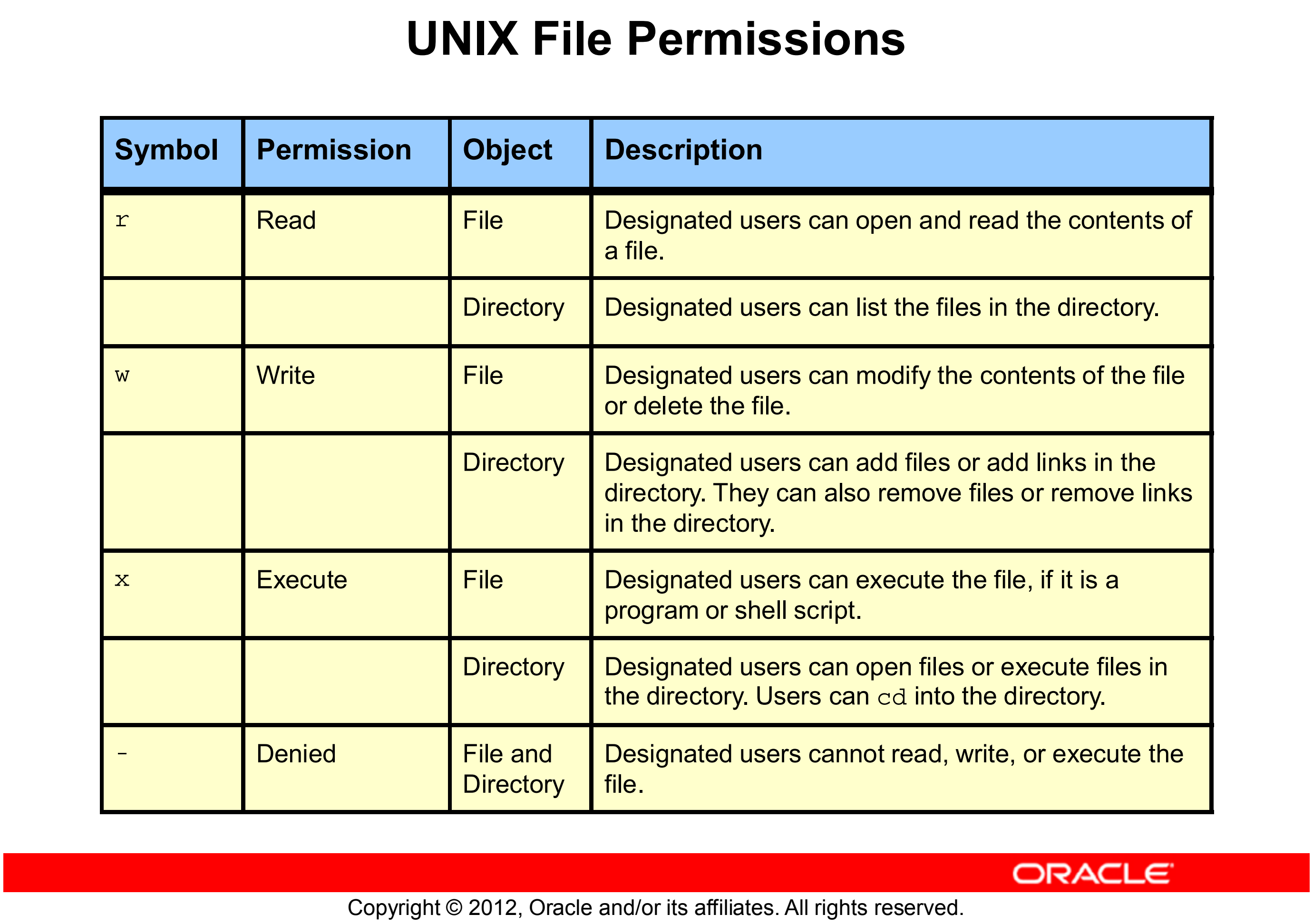
**Note:** In this course, the focus is on using UNIX file permissions. For more information about using ACLs, see the “Using Access Control Lists to Protect UFS Files” section in *Oracle Solaris Administration: Security Services.*

The four basic commands that you use to monitor and secure files and directories are presented in the table shown in the slide.

# File Types

| Symbol | Description |
|---|---|
| `b` | Block special file |
| `c` | Character special file |
| `d` | Directory |
| `l` | Symbolic link |
| `s` | Socket |
| `D` | Door |
| `P` | Named pipe |
| `-` (minus sign) | Regular text file or a program |

A file can be one of eight types, with each type displayed by a symbol. The table lists each file type by its symbol and provides a description of the file type.



# UNIX File Permissions

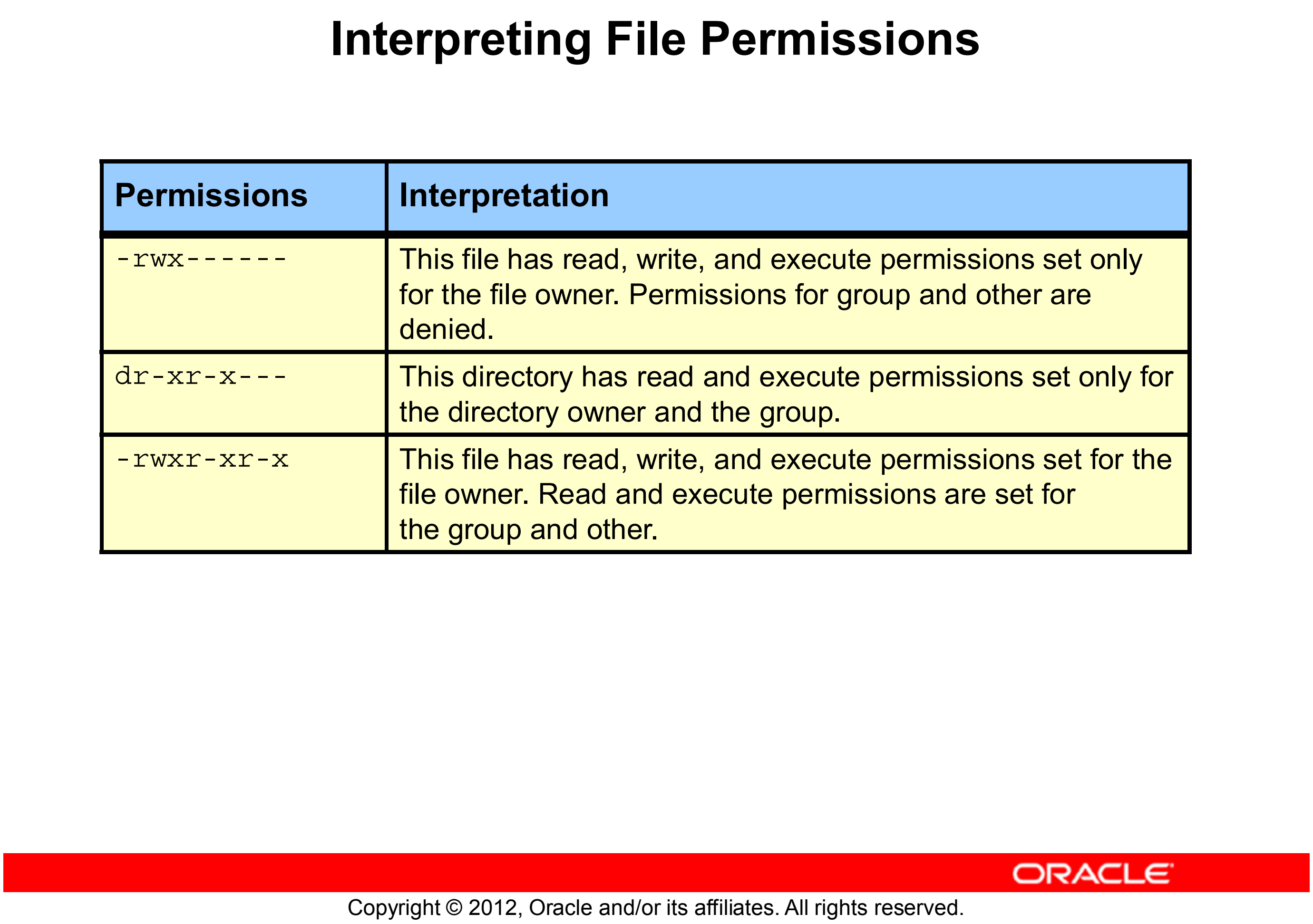
| Symbol | Permission | Object | Description |
|---|---|---|---|
| `r` | Read | File | Designated users can open and read the contents of a file. |
| | | Directory | Designated users can list the files in the directory. |
| `w` | Write | File | Designated users can modify the contents of the file or delete the file. |
| | | Directory | Designated users can add files or add links in the directory. They can also remove files or remove links in the directory. |
| `x` | Execute | File | Designated users can execute the file, if it is a program or shell script. |
| | | Directory | Designated users can open files or execute files in the directory. Users can `cd` into the directory. |
| `-` | Denied | File and Directory | Designated users cannot read, write, or execute the file. |

The table lists and describes the permissions that you can give to each class of users for a file or directory. The three classes of users are:

- **user:** The file or directory owner, which is usually the user who created the file. The owner of a file can decide who has the right to read the file, to write to the file (make changes to it), or, if the file is a command, to execute the file.
- **group:** Members of a group of users
- **others:** All other users who are not the file owner and are not members of the group

The owner of the file can usually assign or modify file permissions. Additionally, the `root` account can change a file's ownership.

You can protect the files in a directory and its subdirectories by setting restrictive file permissions on that directory. Note, however, that the superuser has access to all files and directories on the system.



# Interpreting File Permissions

| Permissions | Interpretation |
|---|---|
| `-rwx------` | This file has read, write, and execute permissions set only for the file owner. Permissions for group and other are denied. |
| `dr-xr-x---` | This directory has read and execute permissions set only for the directory owner and the group. |
| `-rwxr-xr-x` | This file has read, write, and execute permissions set for the file owner. Read and execute permissions are set for the group and other. |

The table in the slide contains three examples of file permissions and their associated interpretations.

## Special File Permissions

- The special permission types for executable files and public directories are:
  - `setuid`: Grants access to the files and directories that are normally available only to the owner
  - `setgid`: Grants access based on the permissions that are granted to a particular group
  - sticky bit: Protects the files within a directory
- When special permissions are used, a user who runs an executable file assumes the ID of the owner (or group) of the executable file.
- Special permissions present a security risk.
- The system should be monitored for any unauthorized use of the `setuid` and `setgid` permissions.

Three special types of permissions are available for executable files and public directories:

- `setuid`: Allows a user to access the files and directories that are normally available only to the owner. When the `setuid` permission is set on an executable file, a process that runs this file is granted access on the basis of the owner of the file. The access is not based on the user who is running the executable file.
- `setgid`: Grants a user access based on the permissions that are granted to a particular group. When the `setgid` permission is applied to a directory, files that were created in this directory belong to the group to which the directory belongs. The files do not belong to the group to which the creating process belongs. Any user who has write and execute permissions in the directory can create a file there. However, the file belongs to the group that owns the directory, not to the group that the user belongs to.
- sticky bit: Protects the files within a directory. If the directory has the sticky bit set, a file can be deleted only by the file owner, the directory owner, or by a privileged user (for example, the `root` user). The sticky bit prevents a user from deleting other users' files from public directories.

When these permissions are set, any user who runs that executable file assumes the ID of the owner (or group) of the executable file.

You must be extremely careful when you set special permissions, because they constitute a security risk. For example, a user can gain superuser capabilities by executing a program that sets the user ID (UID) to `0`, which is the UID of root. Also, all users can set special permissions for files that they own, which constitutes another security concern.

You should monitor your system for any unauthorized use of the `setuid` permission and the `setgid` permission to gain superuser capabilities.

You will be shown how to search for and list all the files that use this special permission later in this lesson. In addition, you will be shown how to protect against other programs that present a security risk, such as an executable stack.

# File Permission Modes

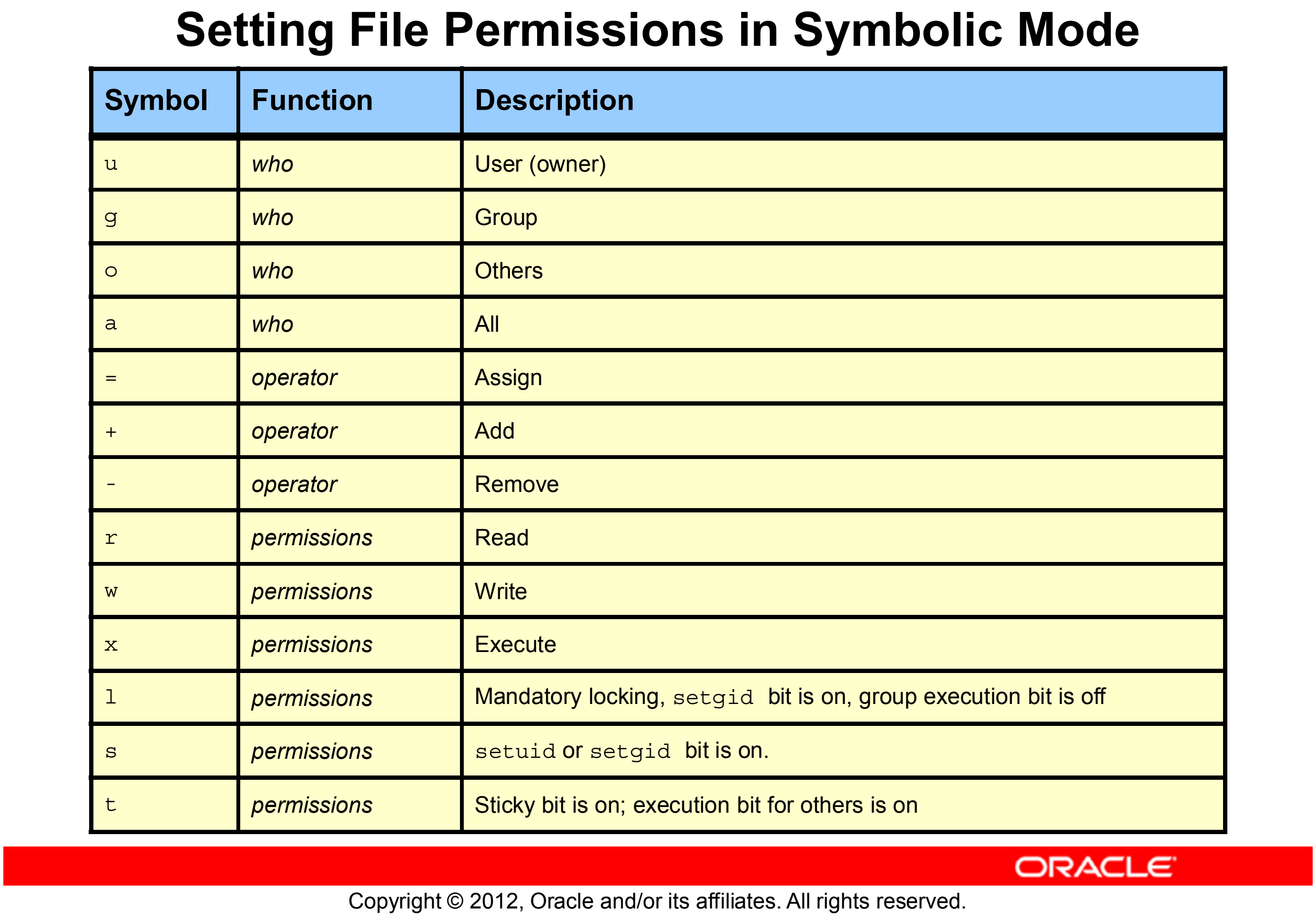
You use chmod to set permissions in either of two modes:

- Symbolic Mode: Combinations of letters and symbols are used to add permissions or remove permissions.
- Absolute Mode: Numbers are used to represent file permissions. This is the most commonly used method to set permissions.

As you saw earlier, the chmod command enables you to change the permissions on a file. You must be the superuser or the owner of a file or directory to change its permissions.

You can use the chmod command to set permissions in either of two modes:

- **Symbolic Mode:** Combinations of letters and symbols are used to add permissions or remove permissions.
- **Absolute Mode:** Numbers are used to represent file permissions. When you change permissions by using the absolute mode, you represent permissions for each triplet by an octal mode number. Absolute mode is the method that is most commonly used to set permissions.



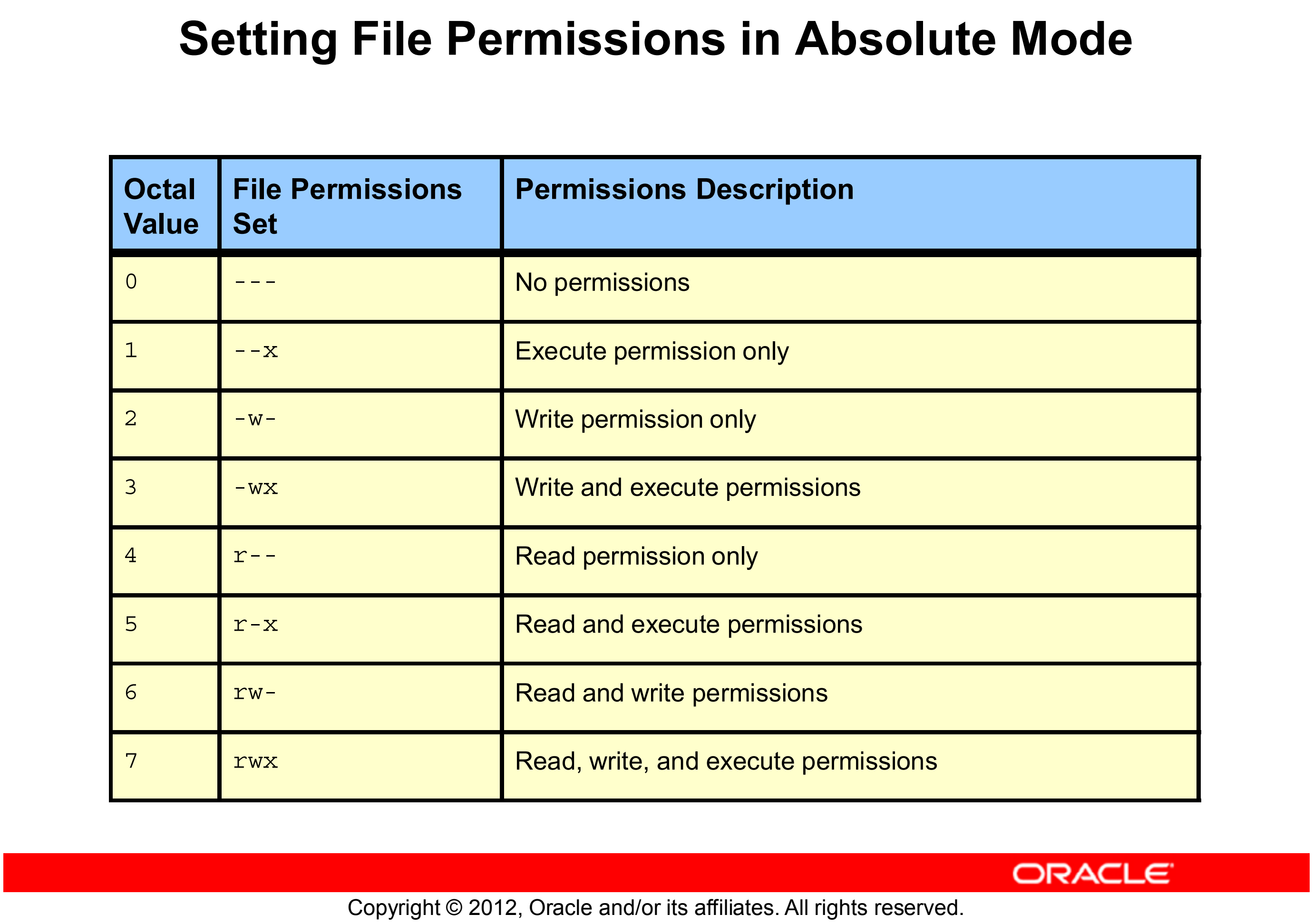
## Setting File Permissions in Symbolic Mode

| Symbol | Function | Description |
|---|---|---|
| `u` | *who* | User (owner) |
| `g` | *who* | Group |
| `o` | *who* | Others |
| `a` | *who* | All |
| `=` | *operator* | Assign |
| `+` | *operator* | Add |
| `-` | *operator* | Remove |
| `r` | *permissions* | Read |
| `w` | *permissions* | Write |
| `x` | *permissions* | Execute |
| `l` | *permissions* | Mandatory locking, `setgid` bit is on, group execution bit is off |
| `s` | *permissions* | `setuid` or `setgid` bit is on. |
| `t` | *permissions* | Sticky bit is on; execution bit for others is on |

The table in the slide lists the symbols for setting file permissions in symbolic mode. Symbols can specify whose permissions are to be set or changed, the operation to be performed, and the permissions that are being assigned or changed.

The who, operator, and permissions designations in the function column specify the symbols that change the permissions on the file or directory.

- ***who***: Specifies whose permissions are to be changed
- ***operator:*** Specifies the operation to be performed
- ***permissions***: Specifies what permissions are to be changed



# Setting File Permissions in Absolute Mode

| Octal Value | File Permissions Set | Permissions Description |
|---|---|---|
| `0` | `---` | No permissions |
| `1` | `--x` | Execute permission only |
| `2` | `-w-` | Write permission only |
| `3` | `-wx` | Write and execute permissions |
| `4` | `r--` | Read permission only |
| `5` | `r-x` | Read and execute permissions |
| `6` | `rw-` | Read and write permissions |
| `7` | `rwx` | Read, write, and execute permissions |

The table in the slide lists the octal values for setting file permissions in absolute mode. You use these numbers in sets of three to set permissions for owner, group, and other, in that order. For example, the value `644` sets read and write permissions for owner and read-only permissions for group and other.

# Setting Special File Permissions in Symbolic or Absolute Mode

- To set special permissions on a file, you can use either the symbolic or absolute mode.
- To set or remove the `setuid` permission on a directory, you must use symbolic mode.
- To set special permissions in absolute mode, you add a new octal value.

| Octal Value | Special File Permissions |
|---|---|
| `1` | Sticky bit |
| `2` | `setgid` |
| `4` | `setuid` |

You can set special permissions on a file in absolute mode or symbolic mode. However, you must use symbolic mode to set or remove `setuid` permissions on a directory.

In absolute mode, you set special permissions by adding a new octal value to the left of the permission triplet. The table in the slide lists the octal values for setting special permissions on a file.

# Oracle Solaris Authentication Services

Oracle Solaris offers the following authentication services:

- Secure RPC: An authentication mechanism that protects NFS mounts and a naming service
- Pluggable Authentication Module (PAM): A framework that enables various authentication technologies to be plugged in to a system entry service without recompiling the service
- Simple Authentication and Security Layer (SASL): A framework that provides authentication and security services to network protocols
- Secure Shell: A secure remote login and transfer protocol that encrypts communications over an unsecure network
- Kerberos service: A client-server architecture that provides encryption with authentication

Oracle Solaris 11 has a number of authentication services that are used to identify a user or service based on predefined criteria. These services range from simple name-password pairs to more elaborate challenge-response systems. Strong authentication mechanisms rely on a user supplying information that only that user knows, and a personal item that can be verified. A username is an example of information that the user knows.

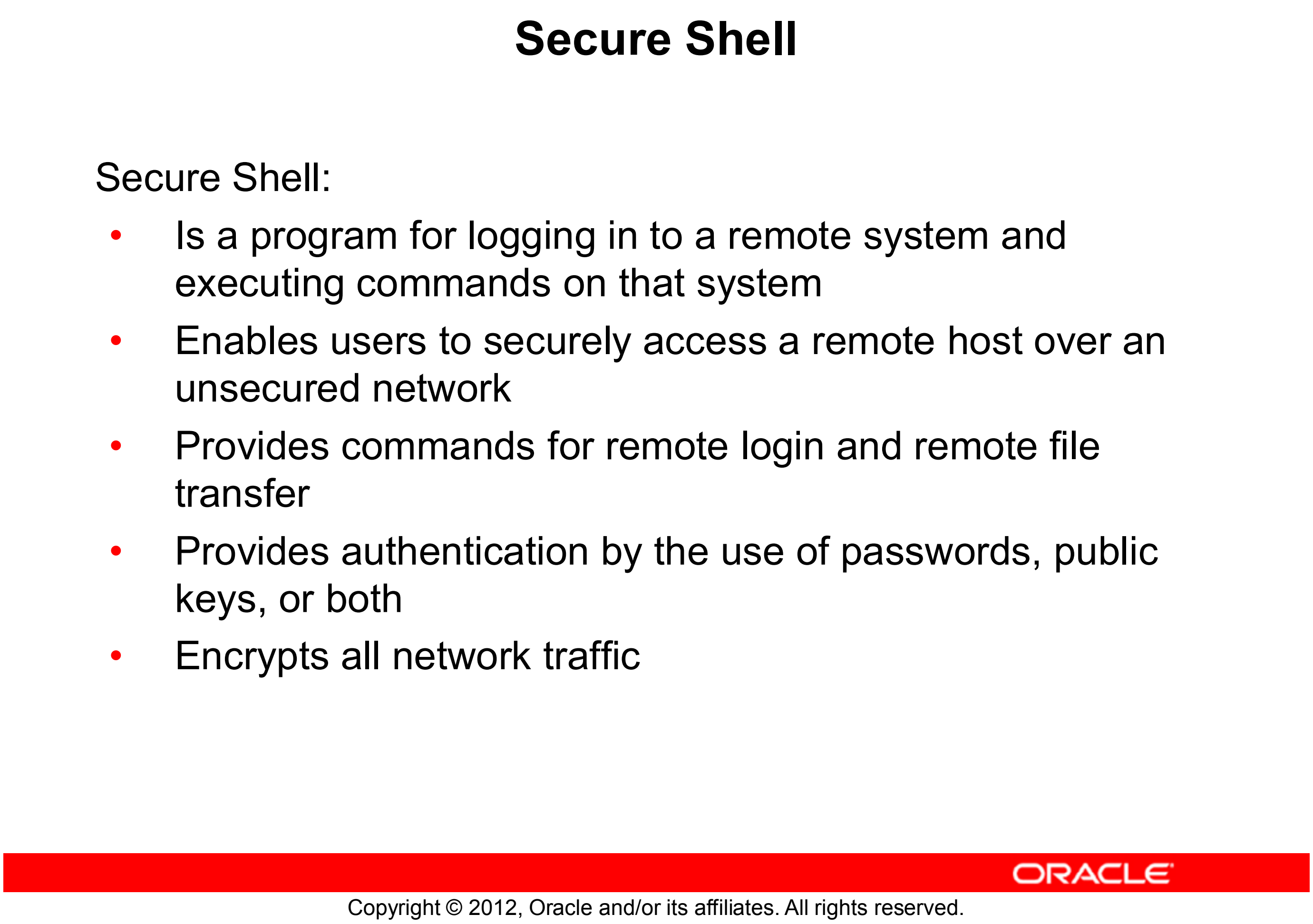
Oracle Solaris offers the following authentication services:

- **Secure RPC:** An authentication mechanism that uses the Diffie-Hellman protocol to protect NFS mounts and a naming service, such as NIS
- **Pluggable Authentication Module (PAM):** A framework that enables various authentication technologies to be plugged in to a system entry service without recompiling the service. Some of the system entry services include login and ftp.
- **Simple Authentication and Security Layer (SASL):** A framework that provides authentication and security services to network protocols

- **Secure Shell:** A secure remote login and transfer protocol that encrypts communications over an unsecure network
- **Kerberos service:** A client-server architecture that provides encryption with authentication

Authentication helps to ensure that the source and the destination are the intended parties. Encryption codes the communication at the source and decodes the communication at the destination. Encryption prevents intruders from reading any transmissions that the intruders might manage to intercept.

In this course, you learn how to configure and use Secure Shell. For information about Oracle Solaris's other authentication services, see *Oracle Solaris Administration: Security Services*.



# Secure Shell

Secure Shell:

- Is a program for logging in to a remote system and executing commands on that system
- Enables users to securely access a remote host over an unsecured network
- Provides commands for remote login and remote file transfer
- Provides authentication by the use of passwords, public keys, or both
- Encrypts all network traffic

Secure Shell is a program that enables users to securely access a remote host over an unsecured network.

The shell provides commands for remote login and remote file transfer.

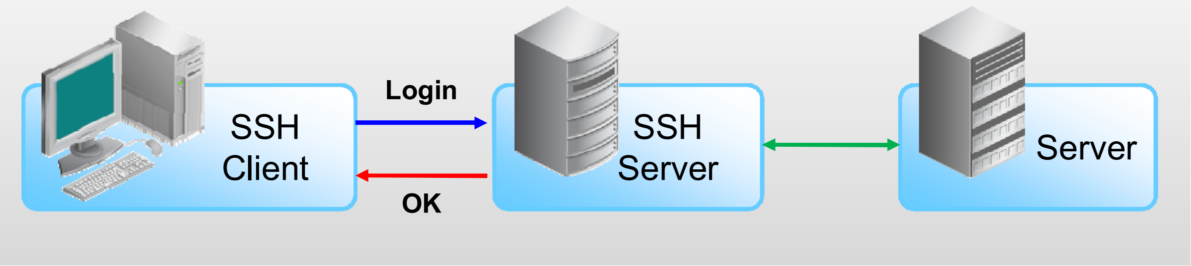
In Secure Shell, authentication is provided by the use of passwords, public keys, or both. All network traffic is encrypted. Thus, Secure Shell prevents a would-be intruder from being able to read an intercepted communication. Secure Shell also prevents an adversary from spoofing the system.

**Note:** Secure Shell can also be used as an on-demand virtual private network (VPN). A VPN can forward X Window system traffic or can connect individual port numbers between the local machines and remote machines over an encrypted network link.

# Secure Shell

With Secure Shell, you can:

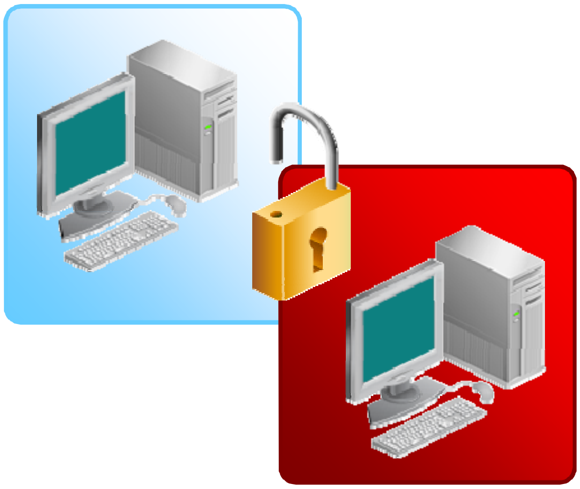
- Log in to another host securely over an unsecured network
- Copy files securely between the two hosts
- Run commands securely on the remote host



With Secure Shell, you can log in to another host securely over an unsecured network, copy files securely between the two hosts, and run commands securely on the remote host.

# Secure Shell and the Secure Shell Protocol

- SSH supports both versions 1 and 2 of the Secure Shell protocol.
- Sites are encouraged to use only version 2.



The current implementation of Secure Shell supports both versions 1 and 2 of the Secure Shell protocol. However, because of inherent security weaknesses in the version 1 protocol, sites are encouraged to use only version 2.

# Secure Shell Protocol Version 2: Parts

- SSH Transfer Protocol: Is used for server authentication, algorithm negotiation, and key exchange. When this part of the SSH protocol completes, an encrypted communication channel is established between the server and the client.
- SSH Authentication Protocol: Is used to verify the identity of the user that runs the ssh client. This protocol uses the established transfer protocol.
- SSH Channel Protocol: Multiplexes the encrypted channel into logical connections. These connections can be used, for example, for user shell sessions, port forwarding, or X11 forwarding. This protocol uses the authentication protocol that the user established.

The Secure Shell protocol version 2 has three major parts.

The Secure Shell Transfer Protocol is used for server authentication, algorithm negotiation, and key exchange. When this part of the Secure Shell protocol completes, an encrypted communication channel is established between the server and the client.

The Secure Shell Authentication Protocol is used for user authentication, that is, to verify the identity of the user that runs the `ssh` client. This protocol uses the established transfer protocol.

The Secure Shell Channel Protocol multiplexes the encrypted channel into logical connections. These connections can be used, for example, for user shell sessions, port forwarding, or X11 forwarding. This protocol uses the authentication protocol that the user established.

**Note:** For a more detailed description of how Secure Shell authentication works, see “Secure Shell Authentication” in *Oracle Solaris Administration: Security Services*.

# Secure Shell Authentication Methods

- GSS-API: Uses credentials for GSS-API mechanisms
- Host-based authentication: Uses host keys and rhosts files
- Public key authentication: Authenticates users with their RSA and DSA public/private keys
- Password authentication: Uses PAM to authenticate users

Secure Shell provides public key and password methods for authenticating the connection to the remote host. Public key authentication is a stronger authentication mechanism than password authentication because the private key never travels over the network. The authentication methods are tried in the following order. When the configuration does not satisfy an authentication method, the next method is tried.

- **GSS-API:** Uses credentials for GSS-API mechanisms such as `mech_krb5` (Kerberos V) and `mech_dh` (AUTH_DH) to authenticate clients and servers. For more information about GSS-API, see "Introduction to GSS-API" in the *Oracle Solaris Security for Developers Guide*.
- **Host-based authentication:** Uses host keys and rhosts files; uses the client's RSA and DSA public/private host keys to authenticate the client; uses the rhosts files to authorize clients to users
- **Public key authentication:** Authenticates users with their RSA and DSA public/private keys
- **Password authentication:** Uses Pluggable Authentication Module (PAM) to authenticate users. The keyboard authentication method in v2 allows for arbitrary prompting by PAM. For more information, see the `SECURITY` section in the `sshd`(1M) man page.

# Host-Based Authentication

| Authentication Method (Protocol Version) | Local Host (Client1) Requirements | Remote Host (Server1) Requirements |
|---|---|---|
| Host-based (v2) | User account<br>Local host private key in `/.ssh/id_rsa` and `/.ssh/id_dsa`<br>`HostbasedAuthentication` yes in `/etc/ssh/ssh_config`<br>Server1 entry in `/etc/ssh/shosts.equiv`, `/etc/hosts.equiv`, `~/.rhosts`, or `~/.shosts` | User account<br>Local host public key in `/.ssh/id_rsa` and `/.ssh/id_dsa`<br>`HostbasedAuthentication yes` in `/etc/ssh/ssh_config`<br>Client1 entry in `/etc/ssh/shosts.equiv`, `/etc/hosts.equiv`, `~/.rhosts`, or `~/.shosts` |

## Editing the Secure Shell Configuration Files for Host-Based Authentication

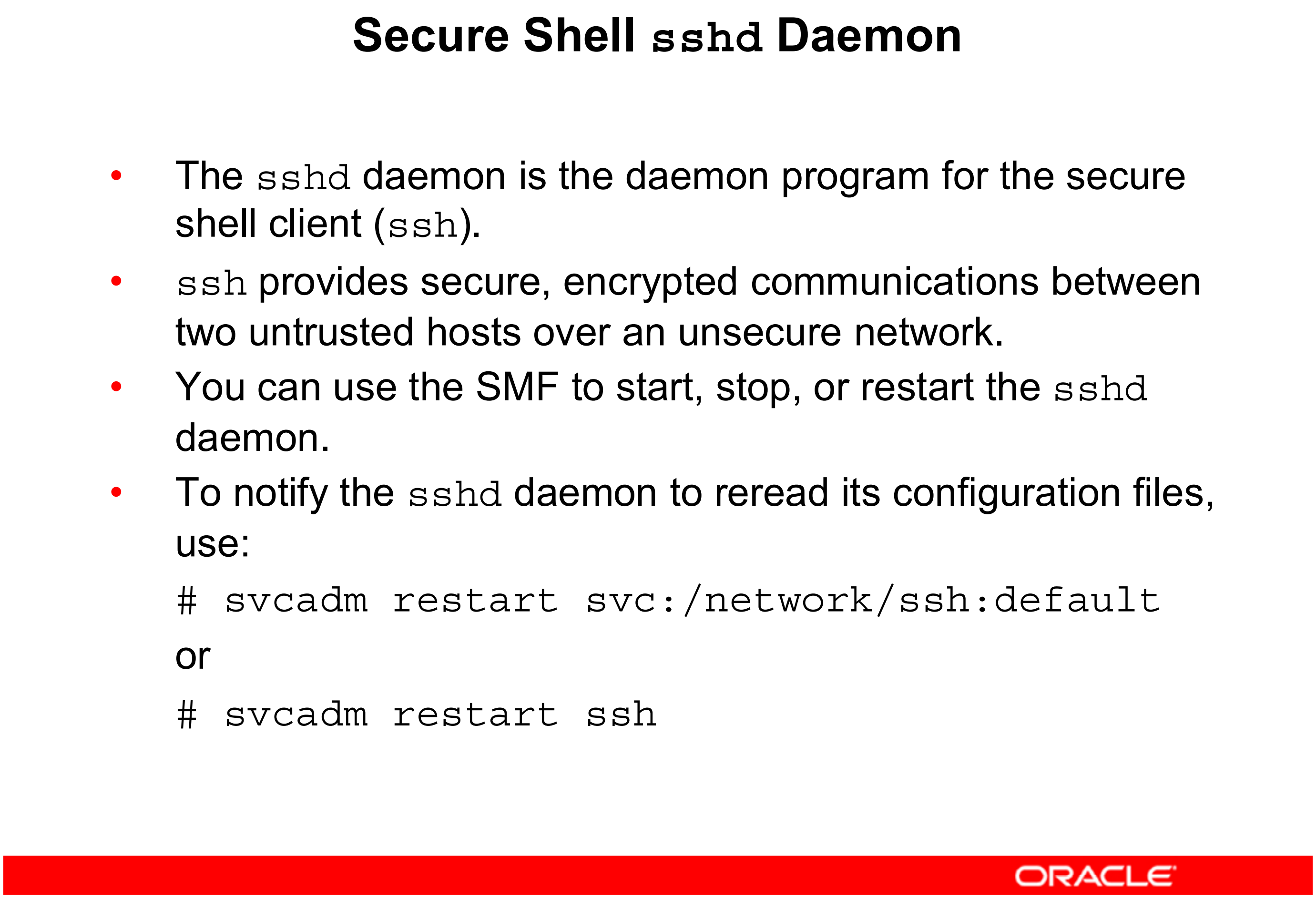
In this course, you are shown how to configure Secure Shell to use the host-based authentication method by using protocol version 2 (v2). To set up host-based authentication, you must edit the Secure Shell configuration files for both the client and the server sides as shown in the table in the slide. Completing the configuration is covered later in this lesson.

# Identifying the Secure Shell Defaults

- Only protocol version 2 is in effect.
- Port forwarding is disabled for server and client sides.
- X11 forwarding is disabled on the server side.
- All authentication methods are enabled, including GSS-API (preferred authentication method).

By default, the following Secure Shell defaults are installed:

- Securing logins and passwords
- Only protocol version 2 is in effect.
- Port forwarding is disabled on both the server and client sides.
- X11 forwarding is disabled on the server side, but is enabled on the client side.
- All authentication methods are enabled, including the generic security service application program interface, or GSS-API for short. GSS-API is the preferred authentication method. Therefore, if Kerberos is configured, Secure Shell uses it out of the box.



## Secure Shell sshd Daemon

- The sshd daemon is the daemon program for the secure shell client (ssh).
- ssh provides secure, encrypted communications between two untrusted hosts over an unsecure network.
- You can use the SMF to start, stop, or restart the sshd daemon.
- To notify the sshd daemon to reread its configuration files, use:

  ```
  # svcadm restart svc:/network/ssh:default
  ```

  or

  ```
  # svcadm restart ssh
  ```

The sshd daemon is the daemon program for the secure shell client (ssh), which is the remote login program. ssh provides secure, encrypted communications between two untrusted hosts over an unsecure network. The daemon listens for connections from a client system. It forks a new daemon for each incoming connection. The forked daemons handle key exchange, encryption, authentication, command execution, and data exchange.

Because Secure Shell is enabled by default during installation, you do not need to do anything to make the program work on your system. You can, however, use the Solaris Management Framework (SMF) to start, stop, or restart the secure shell daemon (sshd). For example, to notify the master Secure Shell daemon to reread its configuration files, you use the following command:

```
svcadm restart svc:/network/ssh:default
```

This simpler command also works:

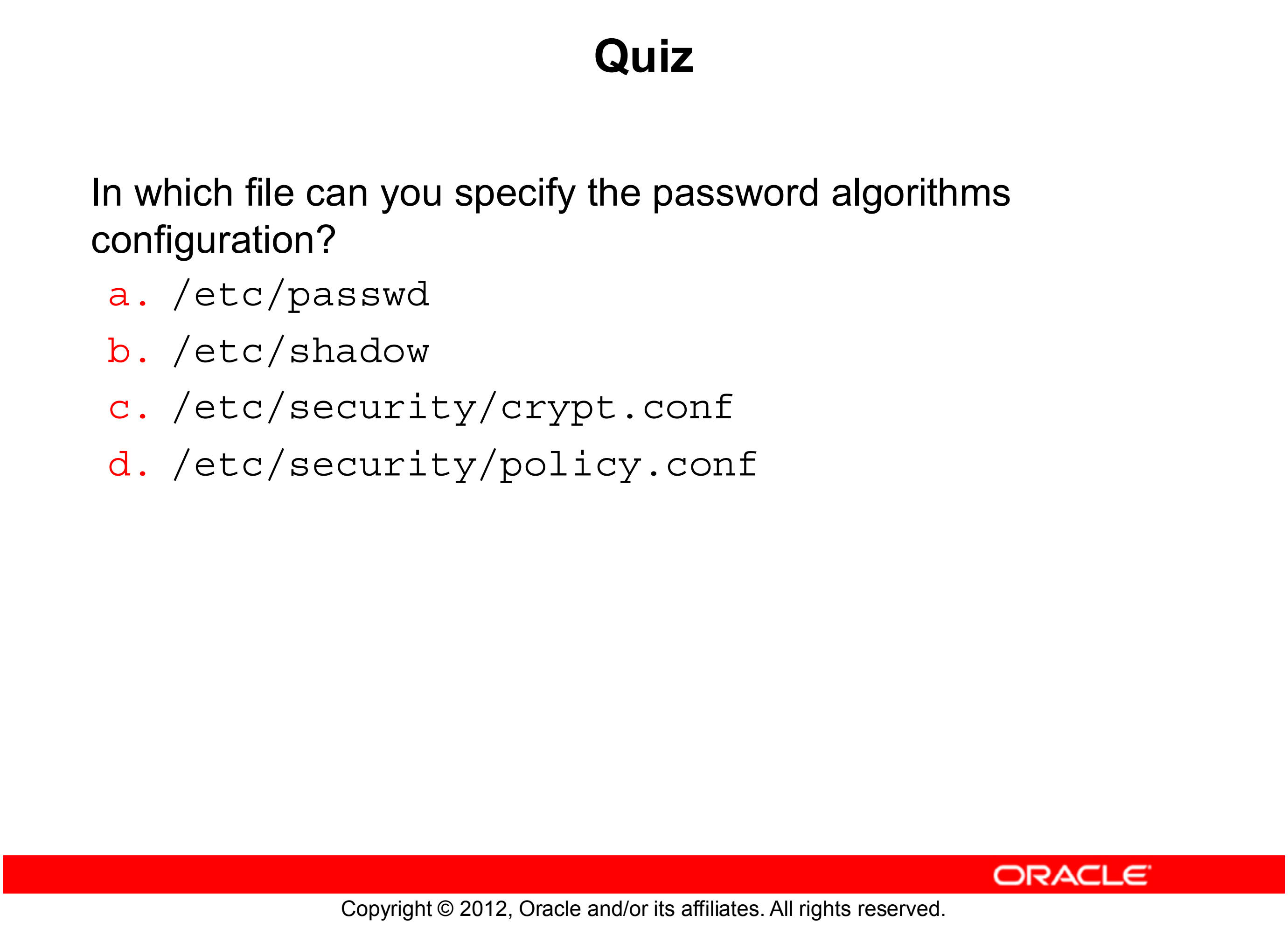
```
svcadm restart ssh
```

# Implementing the System and File Access Control Plan

Your assignment is to:

- Set up and test system and file access controls
- Verify that the controls are working
- Set up and test Secure Shell

Testing a subset of the Oracle Solaris 11 security features is next on the test plan. You have been tasked with setting up system and files access control for users, groups, and the superuser, and then verifying that the controls are working correctly. You have also been tasked with setting up and testing Secure Shell by using the host-based authentication method.



# Quiz

In which file can you specify the password algorithms configuration?

a. `/etc/passwd`

b. `/etc/shadow`

c. `/etc/security/crypt.conf`

d. `/etc/security/policy.conf`

**Answer: d**

## Quiz

Which command enables you to change permissions on a file owned by a group?

a. `chown`

b. `chgrp`

c. `chmod`

**Answer: c**

# Quiz

The `chmod` command can be used only with the absolute mode.

a. True

b. False

**Answer: b**

# Quiz

Which permission gives the following?
This file has read, write, and execute permissions set for the file owner. Read and execute permissions are set for the group and other.

a. `-rwx------`

b. `dr-xr-x---`

c. `-rwxr-xr-x`

**Answer: c**

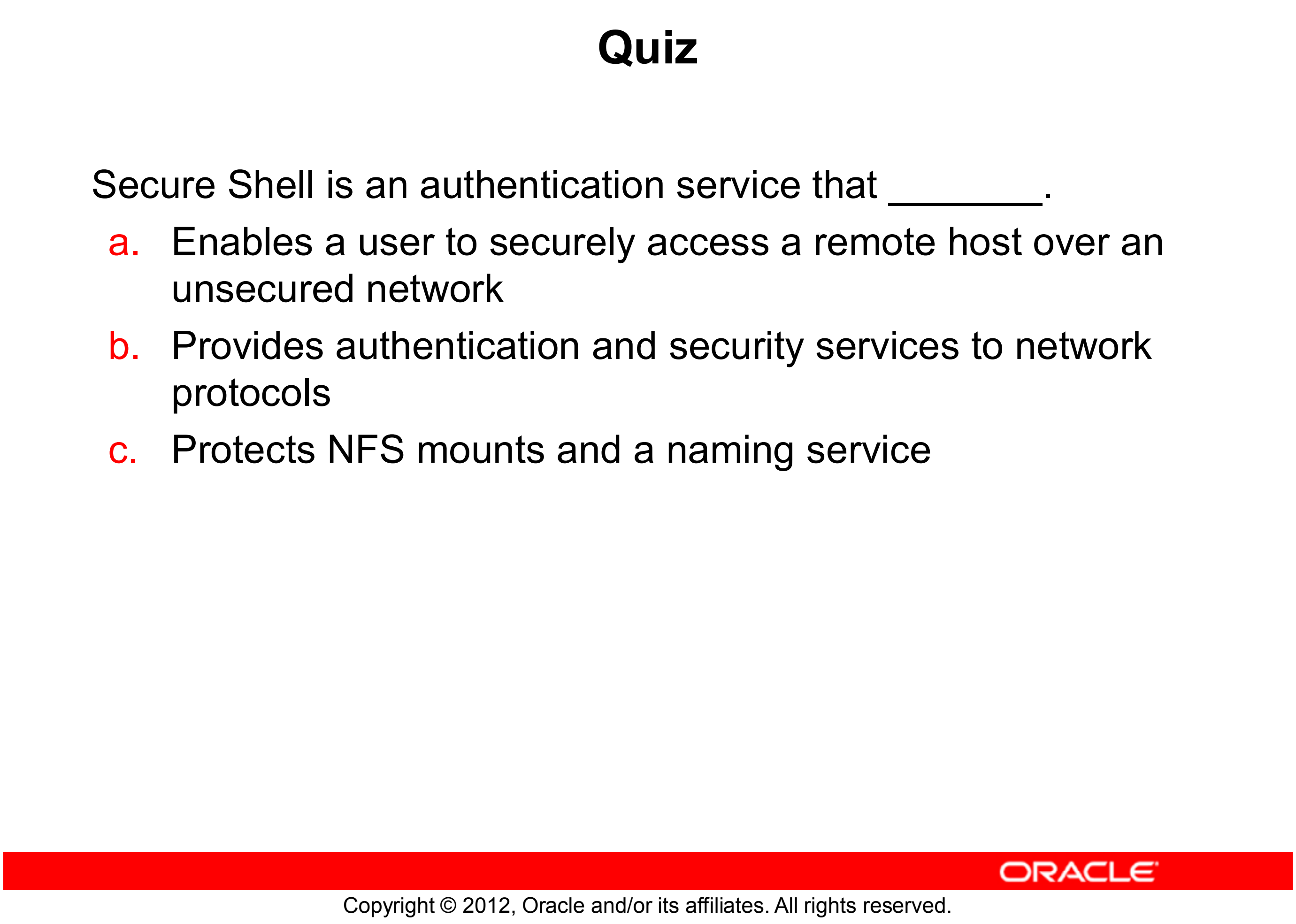
# Quiz

The special permission types `setuid` and `setgid` constitute a risk.

a. True

b. False

**Answer: a**



# Quiz

Secure Shell is an authentication service that _______.

a. Enables a user to securely access a remote host over an unsecured network

b. Provides authentication and security services to network protocols

c. Protects NFS mounts and a naming service

**Answer: a**

# Lesson Agenda

- Planning for System and File Access Control
- Controlling Access to Systems
- Controlling Access to Files
- Configuring and Using Secure Shell

ORACLE

# Controlling Access to Systems

- Securing Logins and Passwords
- Changing the Password Algorithm
- Monitoring and Restricting Superuser Access

In this section, you are shown how to use passwords to control a user's access to the system and how to monitor and restrict the superuser's access to the system. As part of each task, you are shown how to verify that the controls that you have put in place to protect the system are working.

# Securing Logins and Passwords

- Displaying a User’s Login Status
- Displaying Users Without Passwords
- Disabling User Logins Temporarily
- Monitoring Failed Login Attempts
- Monitoring All Failed Login Attempts
- Monitoring Who Is Using the `su` Command

There are several tasks that you can perform to ensure that user logins and passwords are secure. You can check a user’s login status. You can check which users do not have passwords (remember, every user on the system should have a password); you can disable user logins temporarily during a system shutdown or routine maintenance; and you can monitor failed login attempts. You can also monitor who is using the `su` command. When, why, and how often you perform these tasks are dictated in part by your company’s security policies, as well as by indications of suspicious activity or possible security breach attempts.

You will now look at how to perform each of these tasks, beginning with how to display a user’s login status.

# Displaying a User’s Login Status

To display a user’s login status, use `logins -x -l` *`loginname`*.

```
# logins -x -l jjones
jjones          1003    itsupport           110      joe jones
                        /home/jjones
                        /bin/bash
                        PS 120211 56 70 7
```

To display a user’s login status, use the `logins` command followed by the `-x` and `-l` options and the user’s login name.

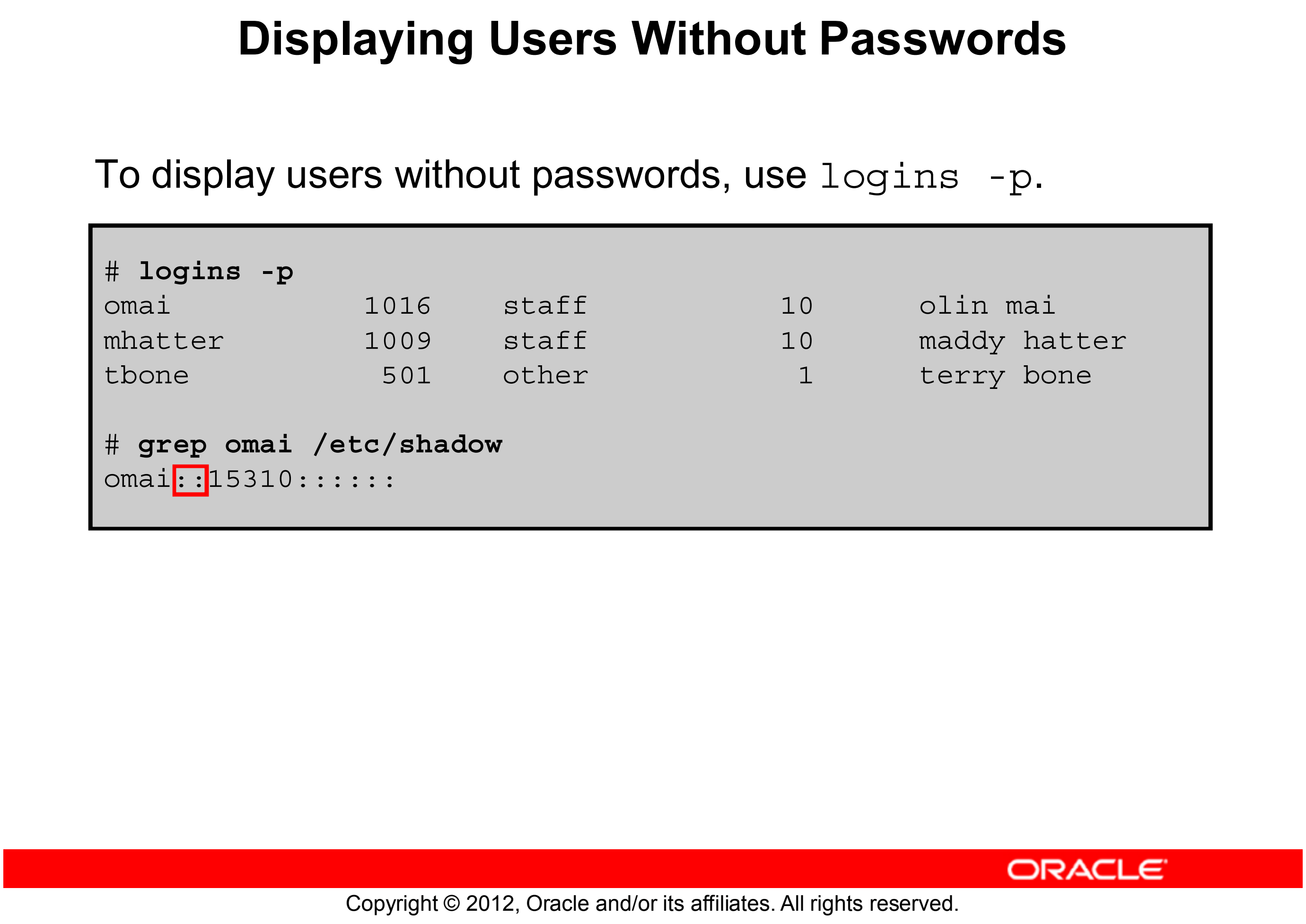
The `logins` command displays information on the user and system logins that are known to the system. The default information is the following: login ID, user ID, primary group name, primary group ID, and the account field value. The `-x` displays an extended set of information about each selected user, including the user’s home directory, login shell, and password aging information, as shown in the example. The `PS` in the example output is a password status associated with the `logins` command. It means that the account probably has a valid password.

If the login is passworded, the password status is followed by the date the password was last changed, the number of days required between changes, and the number of days allowed before a change is required. The password aging information shows the time interval that the user receives a password expiration warning message (when logging on) before the password expires.

The `-l` option displays the login status for the specified user.

**Note:** You can display multiple users by separating their login names with commas.

For more information about the logins command and its options, see the `logins`(1M) man page.



# Displaying Users Without Passwords

To display users without passwords, use `logins -p`.

```
# logins -p
omai              1016    staff             10      olin mai
mhatter           1009    staff             10      maddy hatter
tbone              501    other              1      terry bone

# grep omai /etc/shadow
omai::15310::::::
```

To display all users who have no passwords, use the `logins` command with `-p`. The `-p` option selects logins with no passwords from the appropriate password database.

To verify that a particular user has no password, you can check the `/etc/shadow` file with the user’s login name, as shown in the example. Here you can see that there is no password entry for `omai`.

# Disabling User Logins Temporarily

To temporarily block any non-administrative users from logging in to the system, run `init S`.

```
# init S
```

To enable general user login, run `init 3`.

```
# init 3
```

A common reason for needing to disable user logins temporarily is system maintenance. Before disabling logins, you should use the `init S` command to boot the system to single-user mode to ensure that no users can log in, except users with administrative privilege.

**Note:** In the GUI environment (for example, Desktop), you must use the GRUB menu to bring the system into single-user mode.

When you have completed maintenance and are ready to return the system to the users, you can issue the `init 3` command, which brings the system back up quickly and efficiently.

# Monitoring Failed Login Attempts

1. Create the `loginlog` file in the `/var/adm` directory.
2. Set read and write permissions for the `root` user on the `loginlog` file.
3. Change group membership to `sys` on the `loginlog` file.
4. Verify that the log works.

```
# touch /var/adm/loginlog
# chmod 600 /var/adm/loginlog
# chgrp sys /var/adm/loginlog
# cat /var/adm/loginlog
jjones:/dev/pts/2:Fri Dec 2 10:21:10 2011
jjones:/dev/pts/2:Fri Dec 2 10:21:21 2011
jjones:/dev/pts/2:Fri Dec 2 10:21:30 2011
jjones:/dev/pts/2:Fri Dec 2 10:21:40 2011
jjones:/dev/pts/2:Fri Dec 2 10:21:49 2011
```

By default, the system does not log failed login attempts. To enable logging, you must create the `loginlog` file in the `/var/adm` directory by using the `touch /var/adm/loginlog` command. Then you must set read and write permissions for the `root` user on the `loginlog` file by using the `chmod 600 /var/adm/loginlog` command.

**Note:** You are setting file permissions in absolute mode.

You need to set these permissions and change the group membership in order for the system to be able to write to this file.

After five unsuccessful login attempts, all the attempts are logged in the `/var/adm/loginlog` file. Therefore, a good way to verify that the log works is to log in as a user and attempt to log in five times by using the wrong password. Then check the `loginlog` file by using the `cat /var/adm/loginlog` command.

The `loginlog` file contains one entry for each failed attempt. Each entry contains the user’s login name, address of the terminal window, and the time of the failed attempt. If a person makes fewer than five unsuccessful attempts, no failed attempts are logged.

A growing `loginlog` file can indicate an attempt to break into the computer system. Therefore, it is a good practice to check and clear the contents of this file regularly.

# Monitoring All Failed Login Attempts

1. Edit the `/etc/default/login` file with `SYSLOG=YES` and `SYSLOG_FAILED_LOGINS=0`.
2. Create a file with the correct permissions to hold the logging information.
   a. Create the `authlog` file in the `/var/adm` directory.
   b. Set read and write permissions for the `root` user on the `authlog` file.
   c. Change group membership to `sys` on the `authlog` file.
3. Edit the `syslog.conf` file to log failed password attempts.
   a. Type the `auth.notice` entry into the `syslog.conf` file.
   b. Refresh the `system-log` service.
4. Verify that the log works.

To capture all failed login attempts in a `syslog` file, you must edit the `SYSLOG` and `SYSLOG_FAILED_LOGINS` values in the `/etc/default/login` file. To do this, use the `vi /etc/default/login` command. When you are in the text editor, set `SYSLOG` to `YES` (`SYSLOG=YES`) and `SYSLOG_FAILED_LOGINS` to `0`.

The second step is to create a file with the correct permissions to hold the logging information; this is similar to what you did to the `loginlog` file. First, you create the `authlog` file in the `/var/adm` directory by using the `touch /var/adm/authlog` command. Then, on the file, you set read and write permissions for the `root` user by using the `chmod 600 /var/adm/authlog` command. After that, change the group membership to `sys` on the file by using the `chgrp sys /var/adm/authlog` command.

The third step is to edit the `syslog.conf` file to log failed password attempts. This step is so that the `syslogd` daemon can recognize the configuration and send notices to this destination. To do this, you use the `vi /etc/syslog.conf` command. When you are in the editor, add the following entry to the `syslog.conf` file:

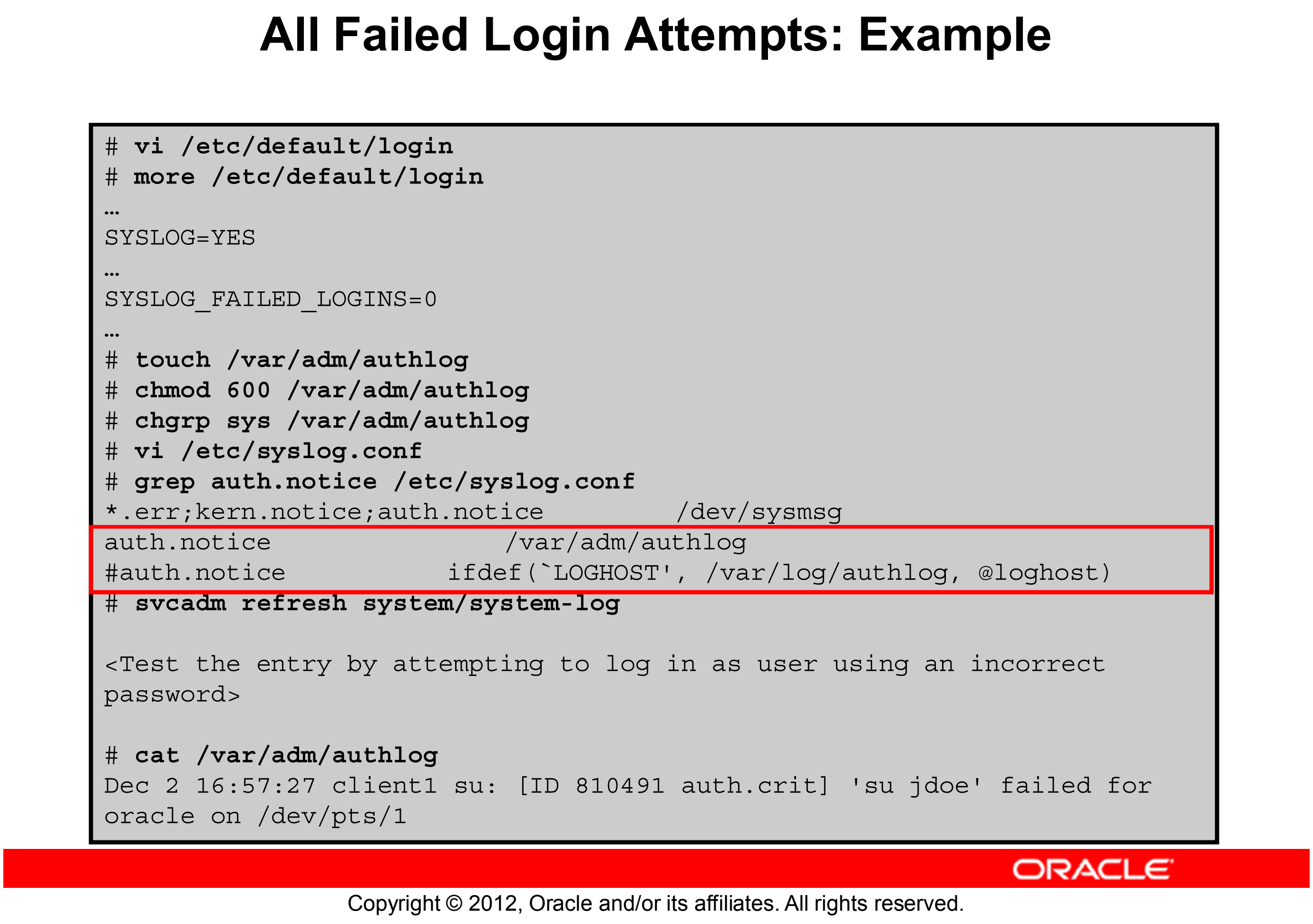
```
auth.notice <Press Tab>  /var/adm/authlog
```

**Note:** Fields on the same line in `syslog.conf` are separated by tabs. `syslog.conf` is covered in more detail in the lesson titled “Performing Basic System Monitoring and Troubleshooting.”

Next, refresh the `system-log` service by using the `svcadm refresh system/system-log` command to make the changes effective.

The final step is to verify that the log works. As before, you can log in to the system as a user and attempt to log in with the wrong password. Then, as the superuser, you can display the `/var/adm/authlog` file.

Be sure to monitor the `/var/adm/authlog` file on a regular basis.



# All Failed Login Attempts: Example

```
# vi /etc/default/login
# more /etc/default/login
…
SYSLOG=YES
…
SYSLOG_FAILED_LOGINS=0
…
# touch /var/adm/authlog
# chmod 600 /var/adm/authlog
# chgrp sys /var/adm/authlog
# vi /etc/syslog.conf
# grep auth.notice /etc/syslog.conf
*.err;kern.notice;auth.notice         /dev/sysmsg
auth.notice               /var/adm/authlog
#auth.notice          ifdef(`LOGHOST', /var/log/authlog, @loghost)
# svcadm refresh system/system-log

<Test the entry by attempting to log in as user using an incorrect
password>

# cat /var/adm/authlog
Dec 2 16:57:27 client1 su: [ID 810491 auth.crit] 'su jdoe' failed for
oracle on /dev/pts/1
```

The slide shows the commands that are used to configure the system to monitor all failed login attempts.

**Note:** Some output has been omitted to save space.

# Changing the Password Algorithm

1. View available password encrypting algorithms in the `/etc/security/crypt.conf` file and determine which algorithm you want to use.
2. Using a text editor, change the password algorithm in the `/etc/security/policy.conf` file by:
   a. Commenting out the current default entry
   b. Specifying a different encryption algorithm from the list of available algorithms

Requiring a username and password is the first defense in controlling access to the system. Having the password encrypted is another layer of protection. To provide the strongest password encryption possible for your site, you can specify the algorithm for password encryption. To do this, view the password algorithms that are currently supported by the system (`CRYPT_ALGORITHMS_ALLOW=1,2a,md5,5,6`) and select the algorithm that you want to specify in the `/etc/security/policy.conf` file. Next, using a text editor, change the password algorithm in the `/etc/security/policy.conf` file by commenting out the current default entry, and then specifying a different encryption algorithm from the list of available algorithms found in the `/etc/security/crypt.conf` file.

**Note:** You might want to comment the file to explain your choice.

The change to the password algorithm will be evident when a new password is used.

**Note:** The algorithm that you select is determined by the level of security that your company requires. A complex algorithm ensures greater security.

# Changing the Password Algorithm: Example

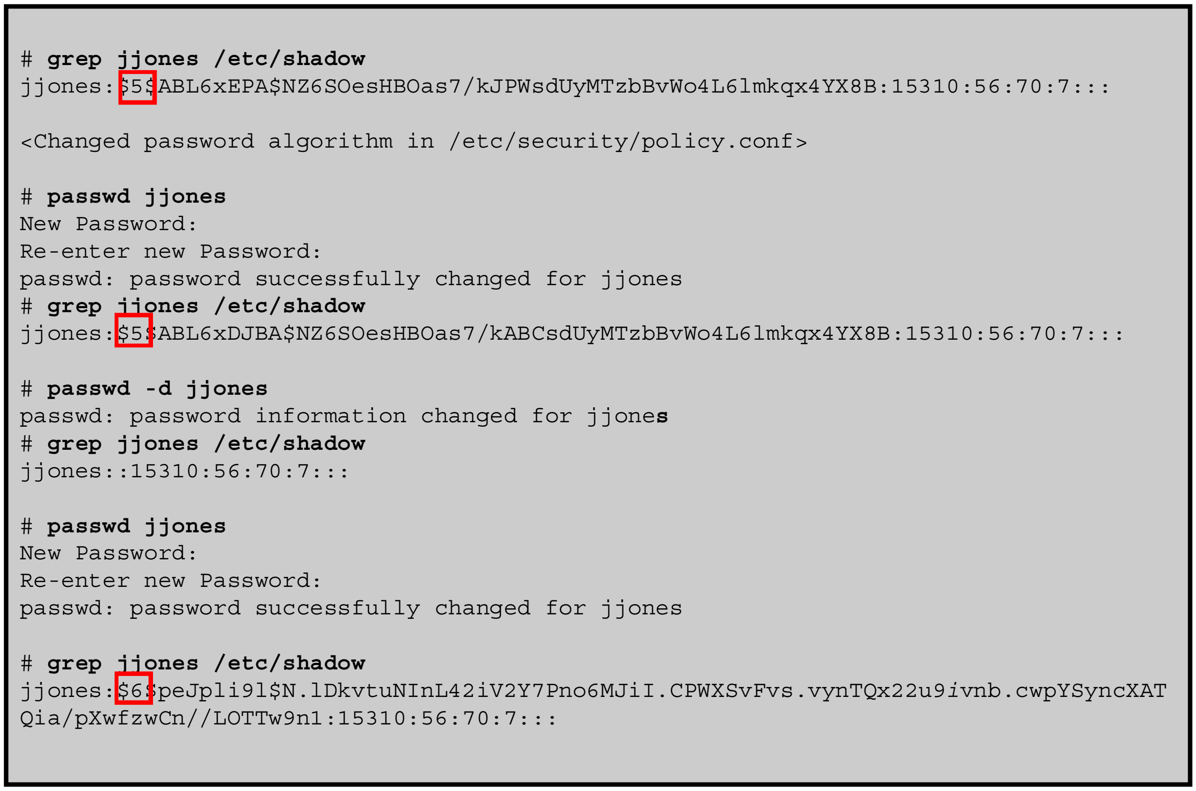
```
# cat /etc/security/crypt.conf
#
#ident   "%Z%%M%  %I%      %E% SMI"
#
# The algorithm name __unix__ is reserved.

1        crypt_bsdmd5.so.1
2a       crypt_bsdbf.so.1
md5      crypt_sunmd5.so.1
5        crypt_sha256.so.1
6        crypt_sha512.so.1…
# vi /etc/security/policy.conf
#
CRYPT_ALGORITHMS_ALLOW=1,2a,md5,5,6
#
# Passwords previously encrypted with SHA256 will be encrypted with
SHA512
# when users change their passwords.
#
#CRYPT_DEFAULT=5
CRYPT_DEFAULT=6
```

ORACLE

The slide shows the commands that are used to change a password algorithm. In the example, you first view `/etc/security/crypt.conf` to see what password algorithms are available. You then edit the `/etc/security/policy.conf` file by commenting out the current default value of `5`, which is the `crypt_sha256` algorithm, and specifying the `crypt_sha12` algorithm as the default. You have also added a comment to explain what change was made.

# Verifying the Password Algorithm Change



```
# grep jjones /etc/shadow
jjones:$5$ABL6xEPA$NZ6SOesHBOas7/kJPWsdUyMTzbBvWo4L6lmkqx4YX8B:15310:56:70:7:::

<Changed password algorithm in /etc/security/policy.conf>

# passwd jjones
New Password:
Re-enter new Password:
passwd: password successfully changed for jjones
# grep jjones /etc/shadow
jjones:$5$ABL6xDJBA$NZ6SOesHBOas7/kABCsdUyMTzbBvWo4L6lmkqx4YX8B:15310:56:70:7:::

# passwd -d jjones
passwd: password information changed for jjones
# grep jjones /etc/shadow
jjones::15310:56:70:7:::

# passwd jjones
New Password:
Re-enter new Password:
passwd: password successfully changed for jjones

# grep jjones /etc/shadow
jjones:$6$peJpli9l$N.lDkvtuNInL42iV2Y7Pno6MJiI.CPWXSvFvs.vynTQx22u9ivnb.cwpYSyncXAT
Qia/pXwfzwCn//LOTTw9n1:15310:56:70:7:::
```

You can verify that the password algorithm change has taken effect by viewing an entry in the /etc/shadow file. In the first command, you are looking at jjones's password algorithm before the change is made to the password algorithm in the /etc/security/policy.conf file. In the password field, the second character tells you the algorithm being used, which in this case, is 5 (crypt_sha256).

Next, you changed the password for jjones. When you view the password again in the /etc/shadow file, you can see that the hashing has changed but the password algorithm has not changed. It is still 5.

Next, you deleted jjones's password, confirmed that it was deleted, and then created a new password for jjones. If, after changing the password, you go into the /etc/shadow file and look at the entry for jjones, you can see that the number in the field for the password algorithm is now 6 (crypt_sha12).

# Monitoring Who Is Using the `su` Command

- By default, `su` logging is enabled in `/var/adm/sulog`.
- The `SULOG=/var/adm/sulog` entry in `/etc/default/su` enables `su` logging.

To monitor `su` logging, use `more /var/adm/sulog`.

```
# more /var/adm/sulog
SU 12/01 10:26 - pts/0 jjones-root
SU 12/01 10:59 + pts/0 jjones-root
SU 12/02 11:11 + pts/0 root-omai
SU 12/02 14:56 - pts/0 jdoe-root
SU 12/02 14:57 + pts/0 jdoe-root
```

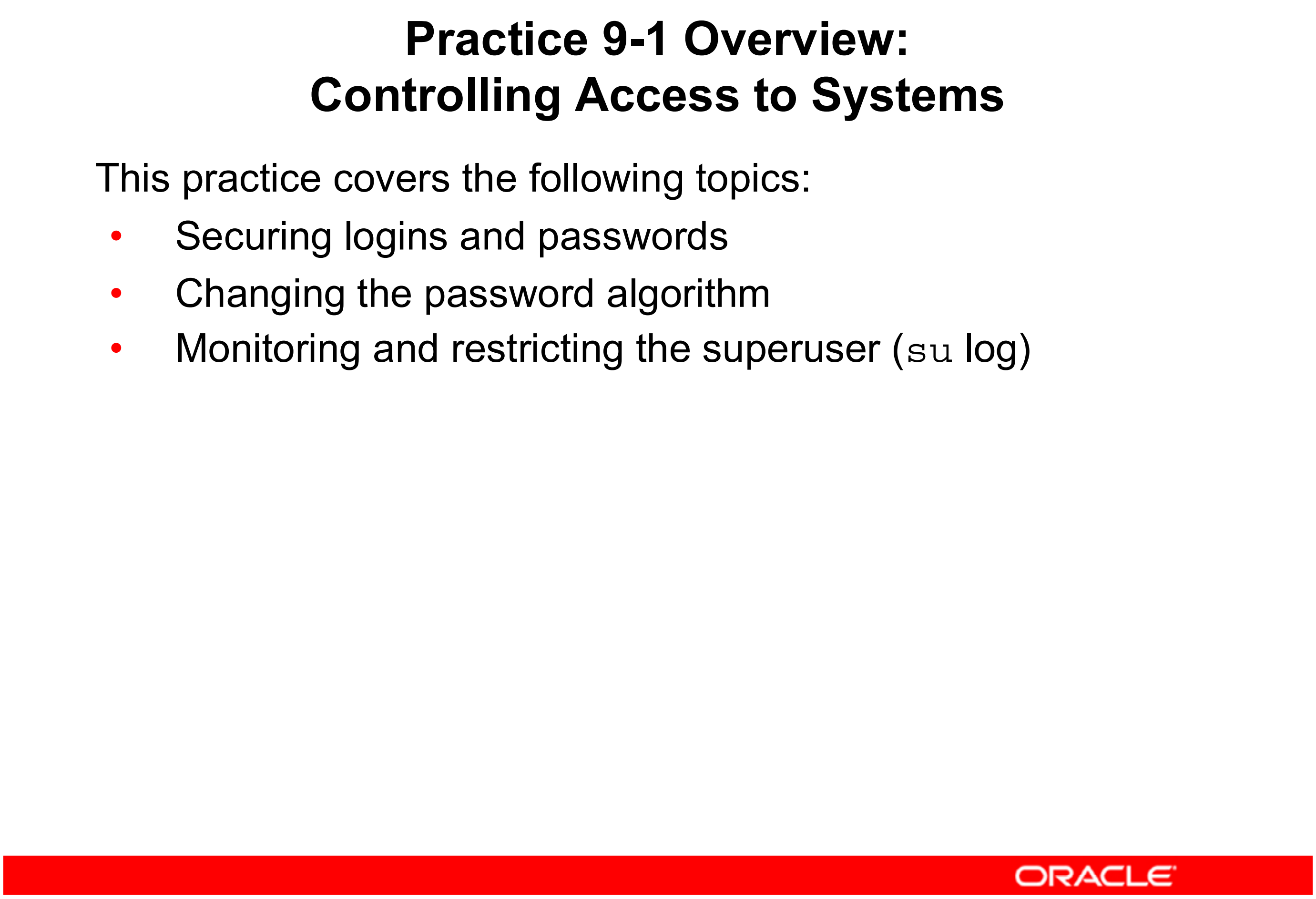
As discussed earlier, it is important to ensure that the `su` command is not being used to break into the system. By default, `su` logging is enabled in the `/var/adm/sulog` file through the following entry in the `/etc/default/su` file:

```
SULOG=/var/adm/sulog
```

The system scans the `sulog` file on a regular basis. You can monitor the contents of this file by using the `more /var/adm/sulog` command, as shown in the example.

The entries display the following information:

- Date and time that the command was entered
- If the attempt was successful. A plus sign (`+`) indicates a successful attempt. A minus sign (`-`) indicates an unsuccessful attempt.
- The port from which the command was issued
- The name of the user and the name of the switched identity



# Practice 9-1 Overview: Controlling Access to Systems

This practice covers the following topics:

- Securing logins and passwords
- Changing the password algorithm
- Monitoring and restricting the superuser (`su` log)

In the practices for this lesson, you will perform the following tasks:

- **Practice 9-1:** Controlling access to systems
- **Practice 9-2:** Controlling access to files
- **Practice 9-3:** Configuring and using Secure Shell

You will find Practice 9-1 in your *Activity Guide*. It should take about one hour to complete these practices.

# Lesson Agenda

- Planning for System and File Access Control
- Controlling Access to Systems
- Controlling Access to Files
- Configuring and Using Secure Shell

## Controlling Access to Files

- Protecting Files with Basic UNIX Permissions
- Protecting Against Programs with Security Risk

In this section, you are shown how to use basic UNIX permissions to control user, group, and superuser access to files. You are also shown how to protect files against programs that pose a security risk.

# Protecting Files with Basic UNIX Permissions

- Displaying file permissions
- Changing file ownership
- Changing group ownership of a file
- Changing file permissions in symbolic mode
- Changing file permissions in absolute mode
- Setting special file permissions in absolute mode

In the subsequent slides, you are shown how to protect files with basic UNIX permissions. You first learn to display the file permissions, and then look at how to change file ownership and group ownership of a file. Next, you learn how to change file permissions in both symbolic and absolute modes, as well as how to set special file permissions in absolute mode.

# Displaying File Permissions

To display file permissions for all the files in a directory, use `ls -la`.

```
# cd /sbin
# ls -la
total 4960
drwxr-xr-x   2 root     sys           64 Nov 18 11:57 ./
drwxr-xr-x  39 root     root          41 Nov 18 15:20 ../
-r-xr-xr-x   1 root     bin        21492 Oct 20 20:55 autopush*
-r-xr-xr-x   1 root     bin        33680 Oct 20 11:36 beadm*
```

To display the permissions for a directory, use `ls -ld`.

```
# cd..
# ls -ld sbin
drwxr-xr-x   4 root     bin          456 Nov 18 04:23 sbin
```

To display the file information that includes user ownership, group ownership, and file permissions for all the files in a directory, use the `ls` command with the `-l` and `-a` options. The example shows a partial list of the files in the `/sbin` directory. Each line of the output displays the following information:

- **Type of file:** For example, `d`, which as you know indicates a directory; `l` for a symbolic link; or `-` for a regular text file or a program
- **Permissions:** For example, `r-xr-xr-x` (which means the file owner, group, and other have read and execute permissions)
- **Number of hard links:** For example, 1
- **Owner of the file:** For example, `root`
- **Group of the file:** For example, `bin`
- **Size of the file, in bytes:** For example, `184360`
- **File creation or last change date:** For example, `Aug 1 20:55`
- **Name of the file:** For example, `bootadm`

To display permissions for a directory, use the `ls - ld` command, as shown in the second example.

# Changing File Ownership

1. Display the permissions on a file by using `ls -l` *`filename`*.
2. Change the owner of the file by using `chown` *`loginname`* *`filename`*.
3. Verify that the owner of the file has changed by using `ls -l` *`filename`*.

```
# ls -l test-file
-rw-r--r--   1 mhatter  staff   112640 Dec 2 10:49 test-file
# chown omai test-file
# ls -l test-file
-rw-r--r--   1 omai     staff   112640 Dec 2 08:50 test-file
```

There are several reasons why the ownership of a file may need to be changed. For example, if someone who was the owner of a critical file has left the company, the ownership of that file must be transferred to someone who has been designated as the new owner. As someone who has authority to make this kind of change, you may be asked to perform the task.

To change the ownership of a file, you must first determine who owns the file. To do this, you use the `ls -l` command followed by the name of the file for which you want to change the ownership. Next, you use the `chown` command followed by the name of the person to whom you want to change the ownership, and the file name. The last step is to verify that the owner of the file has changed. You do this by using the `ls -l` *`filename`* command as you did in the first step.

In the example, you are changing the ownership of the file called `test-file` from `mhatter` to `omai`.

# Changing Group Ownership of a File

1. Display the permissions on a file by using `ls -l` *`filename`*.
2. Change the group ownership of the file by using `chgrp groupname` *`filename`*.
3. Verify that the group ownership of the file has changed by using `ls -l` *`filename`*.

```
# ls -l test-file
-rw-r--r--   1 omai     staff    112640 Dec 6 08:50 test-file
# chgrp itsupport test-file
# ls -l test-file
-rw-r--r--   1 omai  itsupport   112640 Dec 6 08:50 test-file
```

Similar to the need to change the ownership of a file from one owner to another, there may be times when the group ownership of a file needs to be transferred to a different group. For example, there may have been organizational changes that shifted the responsibility of one group to another and, as a result, the files that were owned by the first group must now be assigned to a different group.

The steps for changing the group ownership of a file are very similar to those of changing the file ownership from one user to another.

To begin, display the permissions on the file by using `ls -l filename`. Make a note of the group name. Next, to change the group ownership of the file, use the `chgrp` command followed by the name of the group that you want to change the ownership of the file to and the file name. To verify that the group ownership of the file has changed, use the `ls -l` *`filename`* command as you did in the first step.

In the example, you are changing the group ownership of the file called `test-file` from `staff` to `itsupport`.

# Changing File Permissions in Symbolic Mode

1. Display the permissions on a file by using `ls -l` *`filename`*.
2. Change the file permissions by using `chmod` *`who operator permissions filename`*.
3. Verify that the permissions of the file have changed by using `ls -l` *`filename`*.

```
# ls -l test-file
-rw-r--r--   1 omai itsupport   112640 Dec 6 08:50 test-file
# chmod g+wx test-file
# ls -l test-file
-rw-rwxr--   1 omai itsupport   112640 Dec 6 09:00 test-file
# chmod u-w test-file
# ls -l test-file
-r--rwxr--   1 omai itsupport   112640 Dec 6 09:05 test-file
```

To change file permissions in symbolic mode, you must first determine what the current file permissions are. To do this, use the `ls -l` command followed by the name of the file. Next, use the `chmod` command followed by whose permissions are to be changed, the operation that is to be performed (*`operator`*), what permissions are to be changed (*`permissions`*), and the file name (*`filename`*). The last step is to verify that the permissions of the file have changed. You do this by using the `ls -l` *`filename`* command as you did in first step.

In the example, you are changing the permissions on the file called `test-file` to give the `itsupport` group write and execute permissions. To do this, you use the `chmod` command with the `g` symbol for group followed by a plus sign (`+`) to indicate that you want to add write (`w`) and execute (`x`) to the group. You then remove the write permissions from the user. To do this, you use the `u` symbol for user followed by the minus sign (`-`) and a `w` to indicate that you want to remove the write permissions on this file from the user.

# Changing File Permissions in Absolute Mode

1. Display the permissions on a file by using `ls -l` *`filename`*.
2. Change the file permissions by using `chmod` *`nnn`* *`filename`*.
3. Verify that the permissions of the file have changed by using `ls -l` *`filename`*.

```
# ls -l test-file
-rw-r--r--   1 omai itsupport   112640 Dec 7 08:50 test-file
# chmod 674 test-file
# ls -l test-file
-rw-rwxr--   1 omai itsupport   112640 Dec 7 09:10 test-file
# chmod 474 test-file
# ls -l test-file
-r--rwxr--   1 omai itsupport   112640 Dec 7 09:15 test-file
```

Now you will look at how to perform the same file permission changes in absolute mode. To change file permissions in absolute mode, you first need to determine what the current file permissions are. To do this, use the `ls -l` command followed by the name of the file. Next, use the `chmod` command followed by octal values (*`nnn`*) that represent the permissions for the file owner, file group, and others, in that order. The last step is to verify that the permissions of the file have changed. You do this by using the `ls -l` *`filename`* command as you did in first step.

In the example, you are changing the permissions on the file called `test-file` to give the `itsupport` group write and execute permissions. In absolute mode, the current permissions would be `644` (read and write; read-only; read-only), and you want to change them to `674` (read and write; read, write, and execute; read-only). To do this, you use the `chmod` command followed by `674` and the file name. To remove the write permissions from the user, you use `474.`

**Note:** Refer to the slide titled "Setting File Permissions in Absolute Mode" earlier in this lesson for a list of the absolute mode octal values.

**Note:** The mode that you use is up to you; use the mode that you are most comfortable with.

# Setting Special File Permissions in Absolute Mode

1. Display the permissions on a file by using `ls -l` *`filename`*.
2. Change the special file permissions by using `chmod` *`nnnn`* *`filename`*.
3. Verify that the permissions of the file have changed by using `ls -l` *`filename`*.

```
# ls -l test-file
-rw-r--r--   1 omai  itsupport   112640 Dec  8  09:50 test-file
# chmod 4644 test-file
# ls -l test-file
-rwsr--r--   1 omai  itsupport   112640 Dec  8  10:10 test-file
```

To change special permissions in absolute mode, you use the octal value at the extreme left to set the special permissions on the file.

**Note:** There are four octal values (*`nnnn`*) that are used with special file permissions. For a list of the octal values that are used for special file permissions, refer to the slide titled "Setting Special File Permissions in Symbolic or Absolute Mode" earlier in this lesson.

Go back to the test-file example. Assume that you want to set the `setuid` permission on this file. The current permissions are `644` (read and write; read-only; read-only). You might recall that the octal value of `4` is used to set the `setuid` permission. Given this, to set the `setuid` permission on the `test-file`, you use the `chmod` command followed by `4644` and the file name.

If you wanted to set the `setgid` permission on this file instead, you would have used the following command:

```
    # chmod 2644 test-file
```

In this case, the output for `ls -l test-file` would have displayed the "s" in the group permissions as follows:

```
    -rw-r-sr--   1 omai  itsupport   112640   Dec  8 10:10   test-file
```

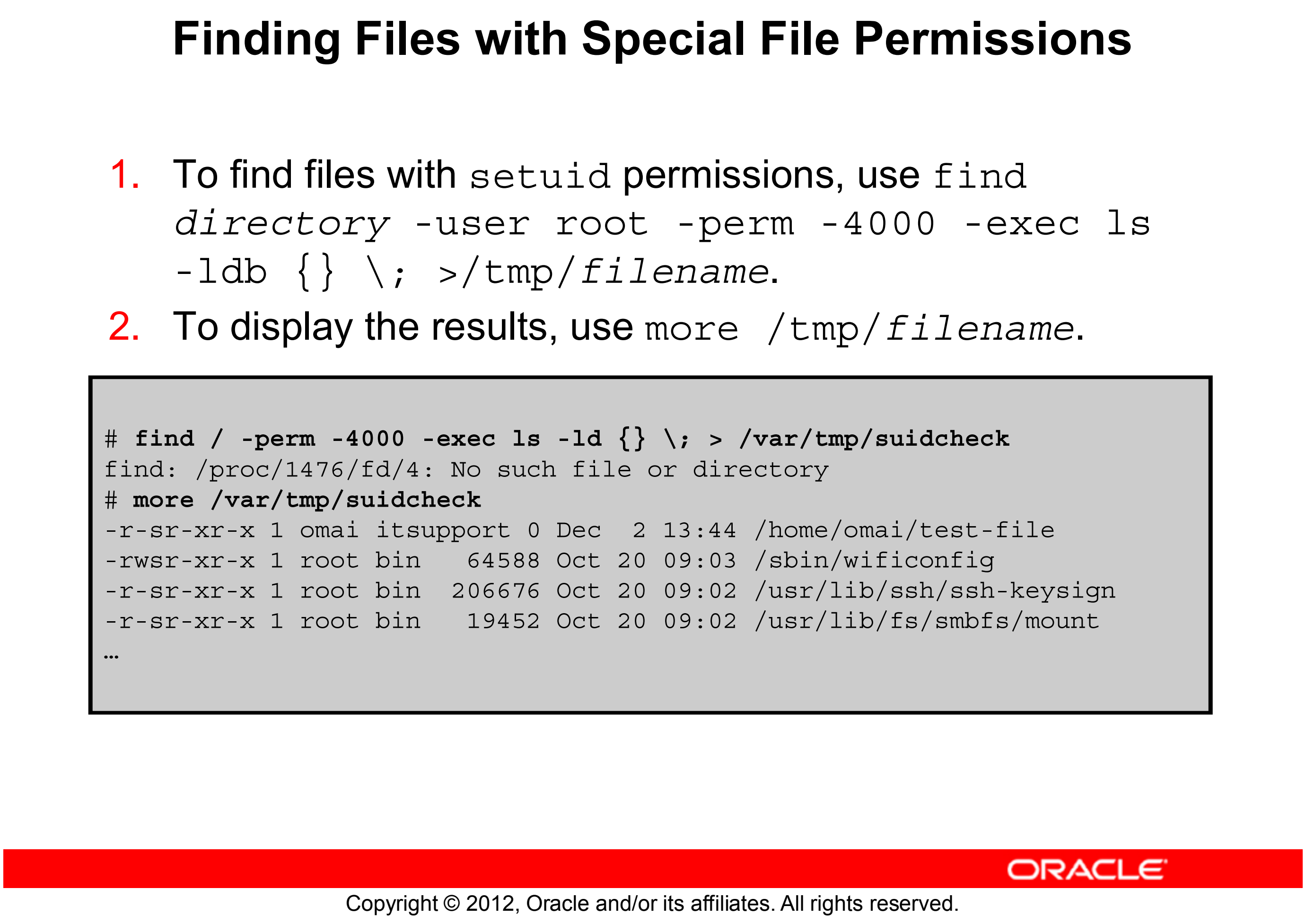
If this is a critical file and you want to ensure that no other user deletes it, you could set the sticky bit permission on the file by using the `chmod 1644 test-file` command.

The output for `ls -l test-file` would display a “t” at the end of the permission set as follows:

```
    -rw-r--r-t  1 omai itsupport   112640 Dec  8 10:10    test-file
```

# Protecting Against Programs with Security Risk

- Finding files with special file permissions
- Disabling programs from using executable stacks



# Finding Files with Special File Permissions

1. To find files with setuid permissions, use find *directory* -user root -perm -4000 -exec ls -ldb {} \; >/tmp/*filename*.
2. To display the results, use more /tmp/*filename*.

```
# find / -perm -4000 -exec ls -ld {} \; > /var/tmp/suidcheck
find: /proc/1476/fd/4: No such file or directory
# more /var/tmp/suidcheck
-r-sr-xr-x 1 omai itsupport 0 Dec  2 13:44 /home/omai/test-file
-rwsr-xr-x 1 root bin   64588 Oct 20 09:03 /sbin/wificonfig
-r-sr-xr-x 1 root bin  206676 Oct 20 09:02 /usr/lib/ssh/ssh-keysign
-r-sr-xr-x 1 root bin   19452 Oct 20 09:02 /usr/lib/fs/smbfs/mount
…
```

As already discussed, the setuid and setgid permissions enable ordinary users to gain superuser capabilities and it is important to ensure that no unauthorized use of the setuid and setgid permissions on programs is occurring. To find files with setuid permissions, use the find command followed by the directory name and -user root -perm -4000 -exec ls -ld {} \; >/tmp/filename. By specifying a directory name, the find command checks all mounted paths, which can be root (/), sys, bin, or mail. The rest of the command can be broken down as follows:

- **-user root:** Displays only files owned by root
- **-perm -4000:** Displays only files with permissions set to 4000, which corresponds to the octal value 4 that is used to set the setuid file permission
- **-exec ls -ld:** Displays the output of the find command in ls -ldb format:
  - **-l:** Long format listing
  - **-d:** List only the directory name, not its contents
- **{}:** Placeholder for the command output
- **\;:** Signifies the end of the command

- **`>`:** Indicates that the output of the command should be sent to the specified file
- **`/tmp/`*`filename`*:** File that contains the results of the `find` command

In the example, you are looking for `setuid` permissions in the `root` directory and have specified that the results of the search be put in the `/tmp/suidcheck` file. Using the `more /var/tmp/suidcheck` command, you can see all the files that have the `setuid` permission set. You can now look for suspicious executable files that are granting ownership to a user rather than to `root` or `bin`.

# Disabling Programs from Using Executable Stacks

1. Save a copy of the `/etc/system` file.
2. Edit the `/etc/system` file and add the following system directives:
   ```
   set noexec_user_stack=1
   set noexec_user_stack_log=0
   ```
3. Reboot the system by using `init 6`.

```
# vi /etc/system
# cat /etc/system
set noexec_user_stack=1
set noexec_user_stack_log=0
# init 6
```

A number of security bugs are related to default executable stacks when their permissions are set to read, write, and execute. Although stacks with execute permissions are allowed, most programs can function correctly without using executable stacks. The `noexec_user_stack` variable in the `/etc/system` file enables you to specify whether stack mappings are executable. By default, this variable is set to zero. If the variable is set to a non-zero value, the system marks the stack of every process in the system as readable and writable, but not executable.

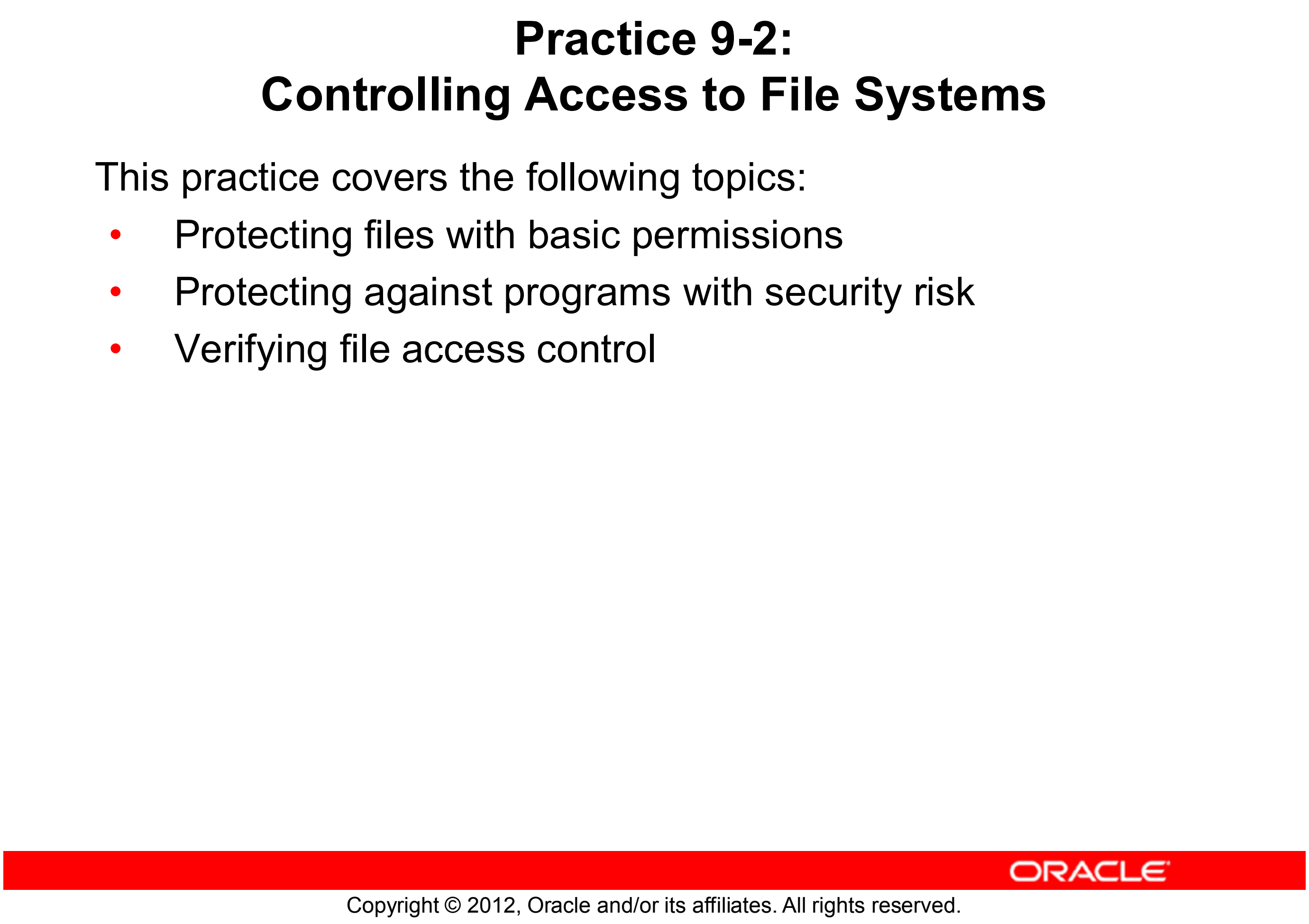
To disable programs from using executable stacks, you must modify the `/etc/system` file to add the following system directives:

```
set noexec_user_stack=1
set noexec_user_stack_log=0
```

The first directive is a security measure and tells the Solaris kernel not to provide an executable stack while executing user programs. If the executable stack is made available, the program in the stack has the ability to write to other buffers, thereby consuming memory resources. The second entry is a directive to the kernel not to log any messages when a program attempts to execute code on their stack.

**Note:** It is a best practice to save a copy of the `/etc/system` file before you edit it. To do this, use the `cp` command.

After you have added the system directives to the `/etc/system` file, you can reboot the system to make the changes take effect.



# Practice 9-2: Controlling Access to File Systems

This practice covers the following topics:

- Protecting files with basic permissions
- Protecting against programs with security risk
- Verifying file access control

This practice should take about one hour to complete.

# Lesson Agenda

- Planning for System and File Access Control
- Controlling Access to Systems
- Controlling Access to Files
- Configuring and Using Secure Shell

ORACLE

# Configuring and Using Secure Shell

- Configuring Secure Shell
- Using Secure Shell

# Configuring Secure Shell

- Enabling host-based authentication
- Configuring Secure Shell

# Enabling Host-Based Authentication

## Server side

```
# grep HostBasedAuthentication /etc/ssh/ssh_config
HostBasedAuthentication yes
# cat /etc/ssh/shosts.equiv
client-host
# svcadm restart ssh
```

## Client side

```
# grep HostBasedAuthentication /etc/ssh/ssh_config
HostBasedAuthentication yes
# cat /etc/ssh/shosts.equiv
server-host
# svcadm restart ssh
```

To configure host-based authentication for Secure Shell, you must enable Secure Shell on both the server and the client systems.

**Note:** You must have the required security attributes to perform this task.

The steps for enabling host-based authentication are as follows:

1. On the server and client sides, type the following entry in the `/etc/ssh/ssh_config` client configuration file: `HostbasedAuthentication yes`
2. On the server side, add the client as an entry to the server's `/etc/ssh/shosts.equiv` file and on the client side, add the server as an entry to the clients `/etc/ssh/shosts.equiv` file. This step puts the public key for each host on the other host.
3. On both the server and client sides, restart the `ssh` service to make the changes effective.

# Verifying That Users Have Access on Both Hosts

Server side

```
# grep jjones /etc/passwd
jjones:x:1003:110:joe jones:/home/jjones:/bin/sh
```

Client side

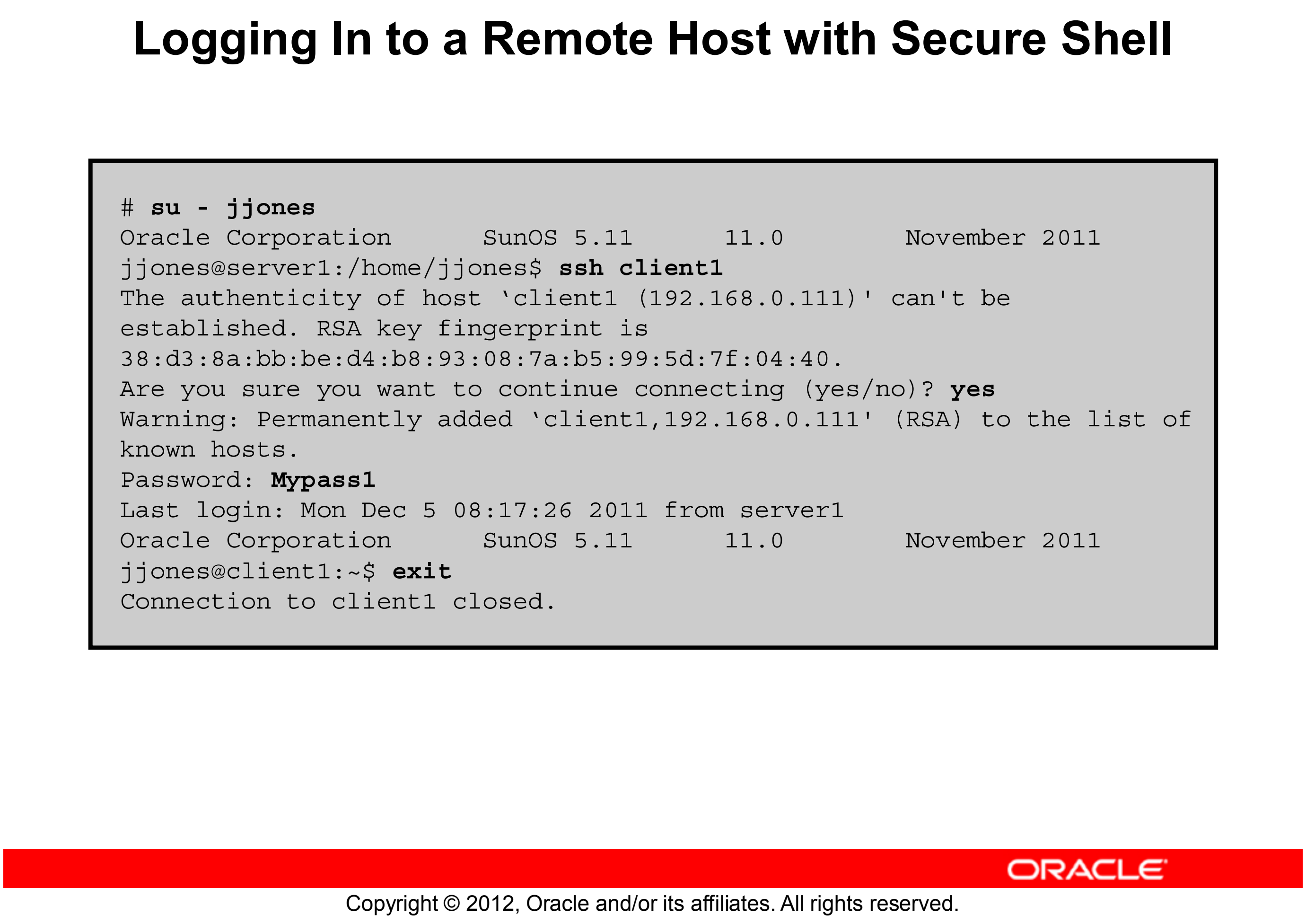
```
# grep jjones /etc/passwd
jjones:x:1003:110:joe jones:/home/jjones:/bin/sh
```

Next, you must verify that the users have access on both hosts. To do this, use the `grep` command followed by the user's login name and `/etc/passwd`, as shown in both examples in the slide.

# Configuring Secure Shell

- Logging in to a remote host with Secure Shell
- Generating the public/private RSA key pair
- Copying the RSA public key to the remote host
- Verifying Secure Shell access
- Generating the public/private DSA key pair
- Copying the DSA public key to the remote host
- Verifying the authentication process

After you have enabled host-based authentication, the next task is to configure the Secure Shell. As part of this task, you log in to the remote host with Secure Shell, generate the public/private RSA key pair, and then copy the public key onto the remote host. After you have verified that the RSA key pair is functioning correctly, you generate the DSA key pair, and then copy the DSA public key to the remote host. Your final step is to verify the authentication process.



# Logging In to a Remote Host with Secure Shell

```
# su - jjones
Oracle Corporation      SunOS 5.11      11.0        November 2011
jjones@server1:/home/jjones$ ssh client1
The authenticity of host ‘client1 (192.168.0.111)' can't be
established. RSA key fingerprint is
38:d3:8a:bb:be:d4:b8:93:08:7a:b5:99:5d:7f:04:40.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added ‘client1,192.168.0.111' (RSA) to the list of
known hosts.
Password: Mypass1
Last login: Mon Dec 5 08:17:26 2011 from server1
Oracle Corporation      SunOS 5.11      11.0        November 2011
jjones@client1:~$ exit
Connection to client1 closed.
```

From the server you will log in to the server, su to a user, and then as the user, you will log in to the remote host (or client) machine with Secure Shell. After logging in successfully, you will exit the client and return to the server.

In the example, you su to user jjones. Then, as jjones, you use the ssh command to log in to the remote host client1. The system then prompts you to verify the authenticity of the remote host key by asking you whether you want to continue connecting, and you respond with a yes.

**Note:** The system administrator is responsible for updating the global /etc/ssh/ssh_known_hosts file. An updated ssh_known_hosts file prevents this prompt from appearing.

You then receive confirmation that this host has been permanently added to the list of known hosts.

You then log in to client1 by using jjones’s user account password.

To close the Secure Shell connection, you exit client1.

# Generating the Public/Private RSA Key Pair

```
jjones@server1:/home/jjones$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jjones/.ssh/id_rsa): Press
Enter Key
Enter passphrase (empty for no passphrase): <passphrase>
Enter same passphrase again: <passphrase>
Your identification has been saved in /home/jjones/.ssh/id_rsa.
Your public key has been saved in /home/jjones/.ssh/id_rsa.pub.
The key fingerprint is:
51:28:86:f9:3b:55:d3:bf:eb:a9:5d:af:0d:f5:2a:8f jjones@server1
jjones@server1:/home/jjones$ ls .ssh
id_rsa  id_rsa.pub  known_hosts
```

Next, still as the user, you will generate the public/private RSA key pair that will be used by Secure Shell. To do this, use the command `ssh-keygen -t rsa`, as shown in the example. The system then generates the public/private RSA key pair and creates the `/etc/home/`*`loginname`*`/ .shh/id_rsa` file in which to save the key. You confirm the file creation by pressing the Enter key. Next, you are prompted to enter and confirm a passphrase for using your key. This passphrase (or identification) is used for encrypting your private key and is saved in the `.shh/id_rsa` file along with your public key.

**Note:** You should not use a null entry for your passphrase. Good passphrases are 10–30 characters long, are not simple sentences or otherwise easy to guess, and contain a mix of uppercase and lowercase letters, numbers, and non-alphanumeric characters. The passphrase is not displayed when you type it in. The passphrase can be changed later by using the `-p` option. For information about key generation commands and options, see the `ssh-keygen`(1) man page.

You are then presented with the key fingerprint.

To verify that the path to the key file is correct, you can use the `ls .ssh` command. You should see the following files: `id_rsa`, `id_rsa.pub`, and `known_hosts`. If you see these files listed, you have successfully created a public/private key pair.

# Copying the RSA Public Key to the Remote Host

```
jjones@server1:/home/jjones$ scp .ssh/id rsa.pub \
jjones@client1:id_rsa.pub
Password: <password>
id_rsa.pub       100% |*****************************|   398       00:00
jjones@server1:/home/jjones$ ssh client1
Password: <password>
Last login: Mon Dec 5 08:19:04 2011 from server1
Oracle Corporation      SunOS 5.11      11.0       November 2011
jjones@client1:~$ ls
crmindex    id_rsa.pub
jjones@client1:~$ mkdir -p .ssh
jjones@client1:~$ ls
crmindex    id_rsa.pub
jjones@client1:~$ cat ./id_rsa.pub >> .ssh/authorized_keys
jjones@client1:~$ rm ./id_rsa.pub
```

The next step is to copy the local host's public key to the remote host and store it in the user's `.ssh` directory. To copy the local host's public key to the remote host, use the `scp` command followed by `.ssh/id rsa.pub` *`loginname@remotehost`*`:id_rsa.pub`.

**Note:** The `.pub` indicates the public key.

You are then prompted to provide the user's login password. The copy process begins and the progress of the copy is presented by a meter. The progress meter displays:

- The file name
- The percentage of the file that has been transferred
- A series of asterisks that indicate the percentage of the file that has been transferred
- The quantity of data transferred
- The estimated time of arrival, or ETA, of the complete file (that is, the remaining amount of time)

After the copy is completed, you `ssh` to the client system and enter the user's password. Then, run `ls` to ensure that the `id_rsa.pub` file is present. Next, run the `mkdr -p .ssh` command to create the `.ssh` directory, and then run the `ls` command again. Now you must place the `id_rsa.pub` file in the `.ssh/authorized_keys` file. To do this, you run the `cat ./id rsa.pub >> .ssh/authorized_keys` command. This public key will be used by the client host to authenticate your incoming `ssh` connection.

The final step is to run the `rm ./id rsa.pub` command.

# Verifying That the RSA Public Key Is Functioning

```
jjones@client1:~$ exit
Connection to client1 closed.
jjones@server1:/home/jjones$ ssh client1
Enter passphrase for key '/home/jjones/.ssh/id_rsa': <passphrase>
Last login: Mon Dec 5 08:21:32 2011 from server1
jjones@client1:~$ exit
Connection to client1 closed.
```

The next step is to verify that the public key is functioning. To do this, you exit the client system, and then `ssh` to the client side from the server side by using the `ssh` command followed by the client system's name. If the public key is functioning, you should be prompted to enter the passphrase.

# Generating the Public/Private DSA Key Pair

```
jjones@server1:/home/jjones$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/jjones/.ssh/id_dsa): Press
Enter Key
Enter passphrase (empty for no passphrase): <passphrase>
Enter same passphrase again: <passphrase>
Your identification has been saved in /home/jjones/.ssh/id_dsa.
Your public key has been saved in /home/jjones/.ssh/id_dsa.pub.
The key fingerprint is:
7a:b8:cb:f8:33:e5:fb:02:a5:c3:b2:53:cc:75:90:9e jjones@server1
jjones@server1:/home/jjones$ ls -a .ssh
id_dsa.pub  id_rsa  id_rsa.pub  known_hosts
```

The next step is to generate the public/private DSA key pair by following the same approach that you used to generate the RSA key pair. However, this time, instead of specifying `rsa`, you specify `dsa`, as shown in the example.

# Copying the DSA Public Key to the Remote Host

```
jjones@server1:/home/jjones$ scp ./.ssh/id_dsa.pub \
jjones@client1:id_dsa.pub
Enter passphrase for key '/home/jjones/.ssh/id_rsa': <passphrase>
id_dsa.pub       100% |*****************************|   606       00:00
jjones@server1:/home/jjones$ ssh client1
Enter passphrase for key '/home/jjones/.ssh/id_rsa': <passphrase>
Last login: Mon Dec 5 08:23:05 2011 from server1
Oracle Corporation      SunOS 5.11      11.0        November 2011
jjones@client1:~$ ls
crmindex    id_dsa.pub
jjones@client1:~$ cat ./id_dsa.pub >> .ssh/authorized_keys
jjones@client1:~$ rm ./id_dsa.pub
jjones@client1:~$ exit
Connection to client1 closed.
```

ORACLE

By following the same approach that you used for the RSA public key, you copy the DSA key to the remote host, as shown in the example.

You have successfully created the RSA and DSA key pairs. The private keys are on the server side and you have copied and stored the public keys on the remote system (the client system) for authentication.

# Verifying the Authentication Process

```
jjones@server1:/home/jjones$ ssh client1
Enter passphrase for key '/home/jjones/.ssh/id_rsa': <Press Enter Key>
Enter passphrase for key '/home/jjones/.ssh/id_dsa': <passphrase>
Last login: Mon Dec 5 08:25:16 2011 from server1
Oracle Corporation      SunOS 5.11      11.0        November 2011
jjones@server1:/home/jjones$ exit
Logout
Connection to client1 is closed.
```

The final step is to verify the authentication process. To do this, you `ssh` to the client, where you are prompted for the RSA and DSA passphrases. If you have set up everything correctly, if you provide an incorrect passphrase for the RSA key when prompted to do so and a correct passphrase for the DSA key, you should be connected, as shown in the example.

# Using the Secure Shell

- Reducing password prompts
- Locking and unlocking the authentication agent

# Reducing Password Prompts

```
jjones@server1: ~$ env | grep SSH
SSH_AGENT_PID=2943
SSH_AUTH_SOCK=/tmp/ssh-XXXXJqaWVf/agent.2942
jjones@server1: ~$ ssh-add
Enter passphrase for /home/jjones/.ssh/id_rsa: <passphrase>
Identity added: /home/jjones/.ssh/id_rsa (/home/jjones/.ssh/id_rsa)
Identity added: /home/jjones/.ssh/id_dsa (/home/jjones/.ssh/id_dsa)
jjones@server1:~$ ssh-add -l
2048 51:28:86:f9:3b:55:d3:bf:eb:a9:5d:af:0d:f5:2a:8f
/home/jjones/.ssh/id_rsa (RSA)
1024 7a:b8:cb:f8:33:e5:fb:02:a5:c3:b2:53:cc:75:90:9e
/home/jjones/.ssh/id_dsa (DSA)
jjones@server1: ~$ ssh client1
Last login: Mon Dec 5 08:26:22 2011 from server1
Oracle Corporation      SunOS 5.11      11.0        November 2011
jjones@client1:~$ exit
Connection to client1 closed.
```

If you do not want to type your passphrase and your password to use the Secure Shell, you can use the agent daemon. You might do this to save time. To begin, you start the daemon at the beginning of the session. Then, you store your private keys with the agent daemon by using the ssh-add command. If you have different accounts on different hosts, you add the keys that you need for the session.

To start the agent daemon, use the env | grep SSH command. As you can see, the environmental variables are populated. Next, you add your private key to the agent daemon by using the ssh-add command. Use the ssh-add -l command to list the identities and confirm that they are available with the authentication agent. To verify that the passphrase does not appear, you start a Secure Shell session. Notice in the example that the prompt for the passphrase did not appear.

# Locking and Unlocking the Authentication Agent

```
jjones@server1:~$ ssh-add -x
Enter lock password: <password>
Again:
Agent locked.
jjones@server1:~$ ssh client1
Enter passphrase for key '/home/jjones/.ssh/id_rsa': <passphrase>
Last login: Mon Dec 5 08:27:14 2011 from server1
Oracle Corporation      SunOS 5.11      11.0        November 2011
jjones@server1:~$ exit
Connection to client1 closed.
```

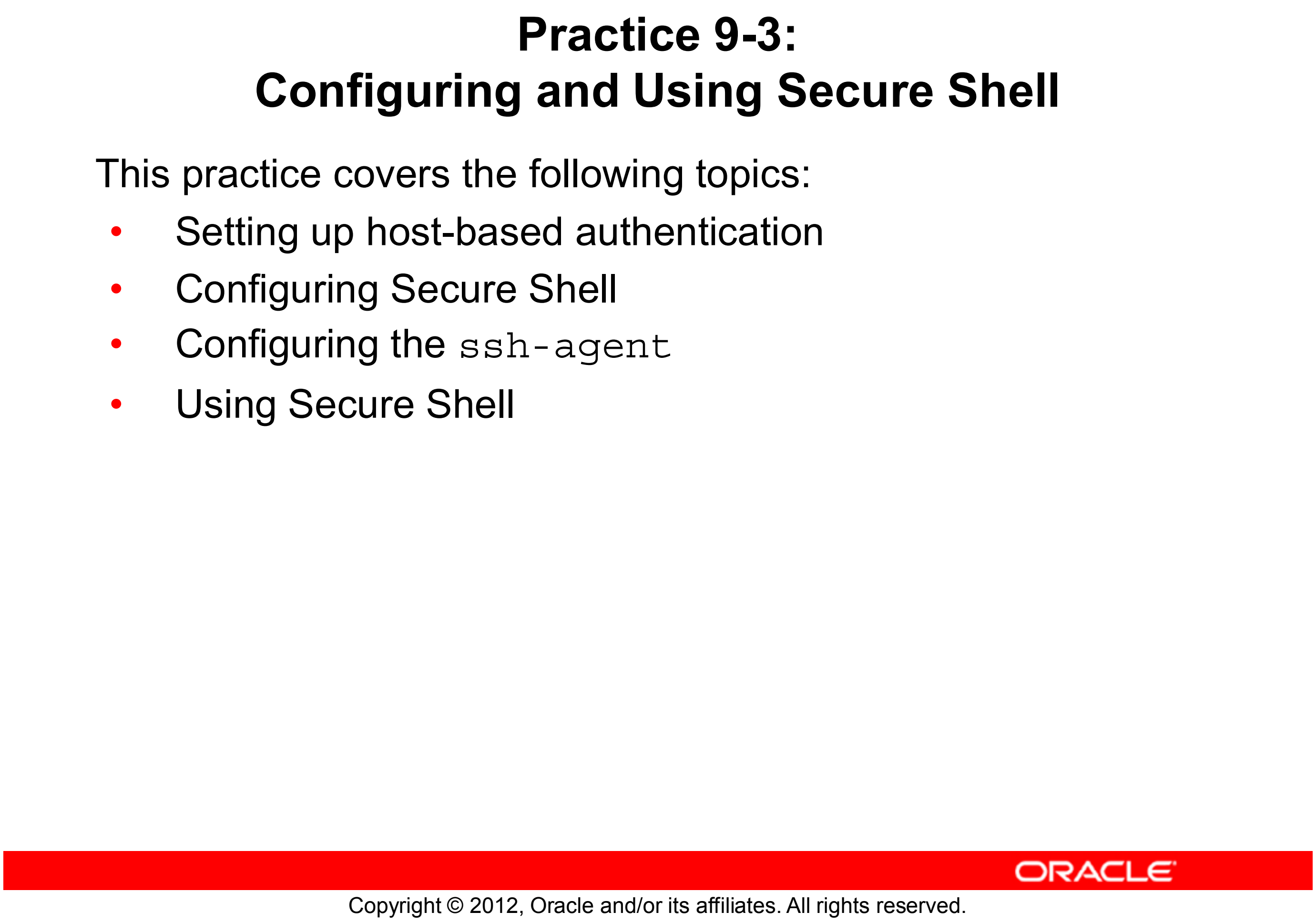
```
jjones@server1:~$ ssh-add -X
Enter lock password: <password>
Agent unlocked.
jjones@server1:~$ ssh client1
Last login: Mon Dec 5 08:27:36 2011 from server1
Oracle Corporation      SunOS 5.11      11.0       November 2011
Connection to client1 closed.
```

You can reinstate the requirement for a passphrase or remove it by locking and unlocking the authentication agent.

It is possible to lock the authentication agent, in which case, you are again prompted for the passphrase.

To lock the agent and reinstate the requirement for a passphrase, use the `ssh-add -x` command, as shown in the first example. You will notice that after the agent is locked, you are again prompted for the passphrase.

To unlock the agent and remove the requirement for a passphrase, use the `ssh-add -X` command, as shown in the second example. Here you can see that when the agent is unlocked, you are no longer prompted for a passphrase.

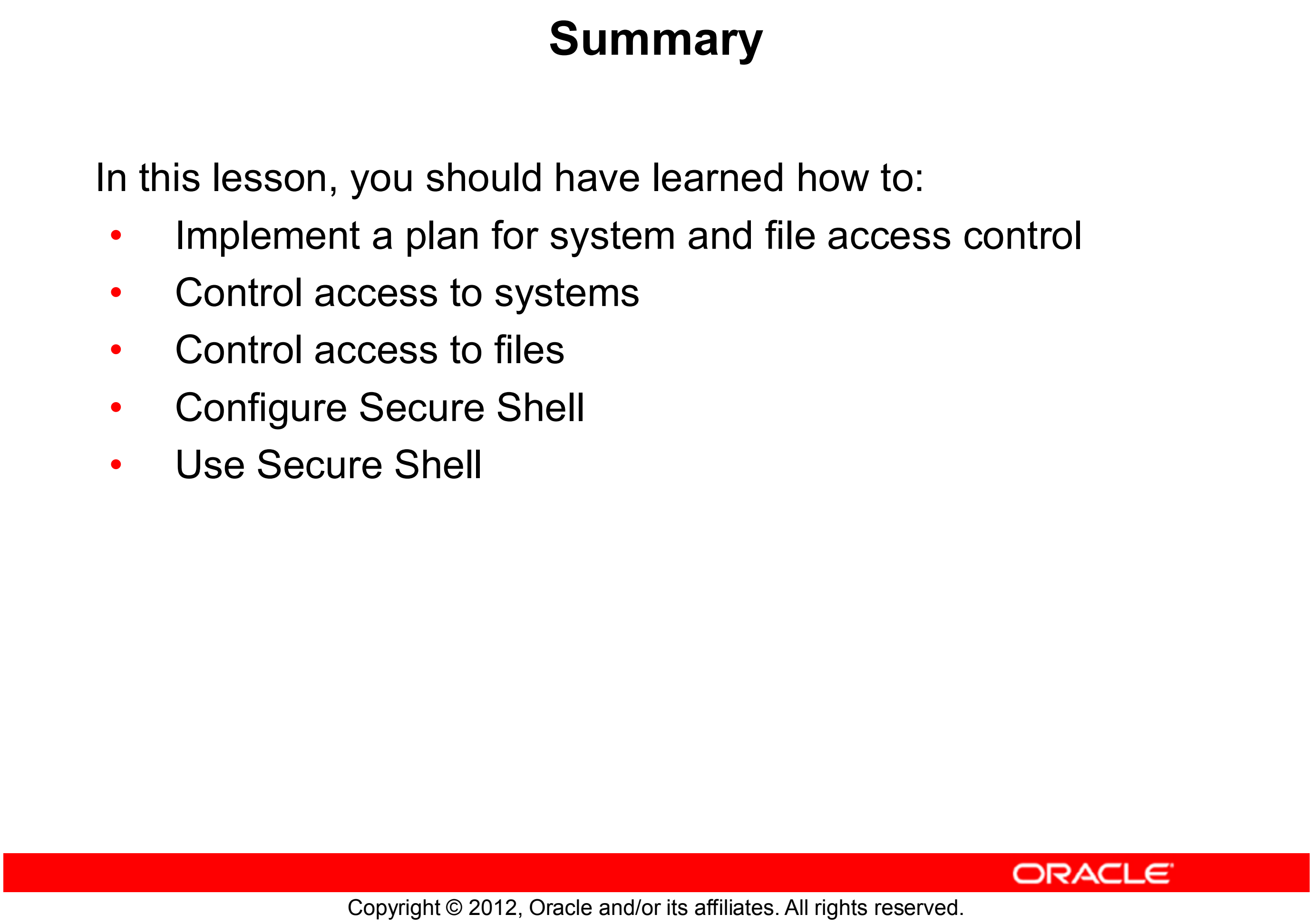


# Practice 9-3: Configuring and Using Secure Shell

This practice covers the following topics:

- Setting up host-based authentication
- Configuring Secure Shell
- Configuring the `ssh-agent`
- Using Secure Shell

This practice should take about one hour to complete.



# Summary

In this lesson, you should have learned how to:

- Implement a plan for system and file access control
- Control access to systems
- Control access to files
- Configure Secure Shell
- Use Secure Shell

In this lesson, you were shown how to control user access to a system and to files. You also learned how to configure and use Secure Shell.

10

# Managing System Processes and Scheduling System Tasks

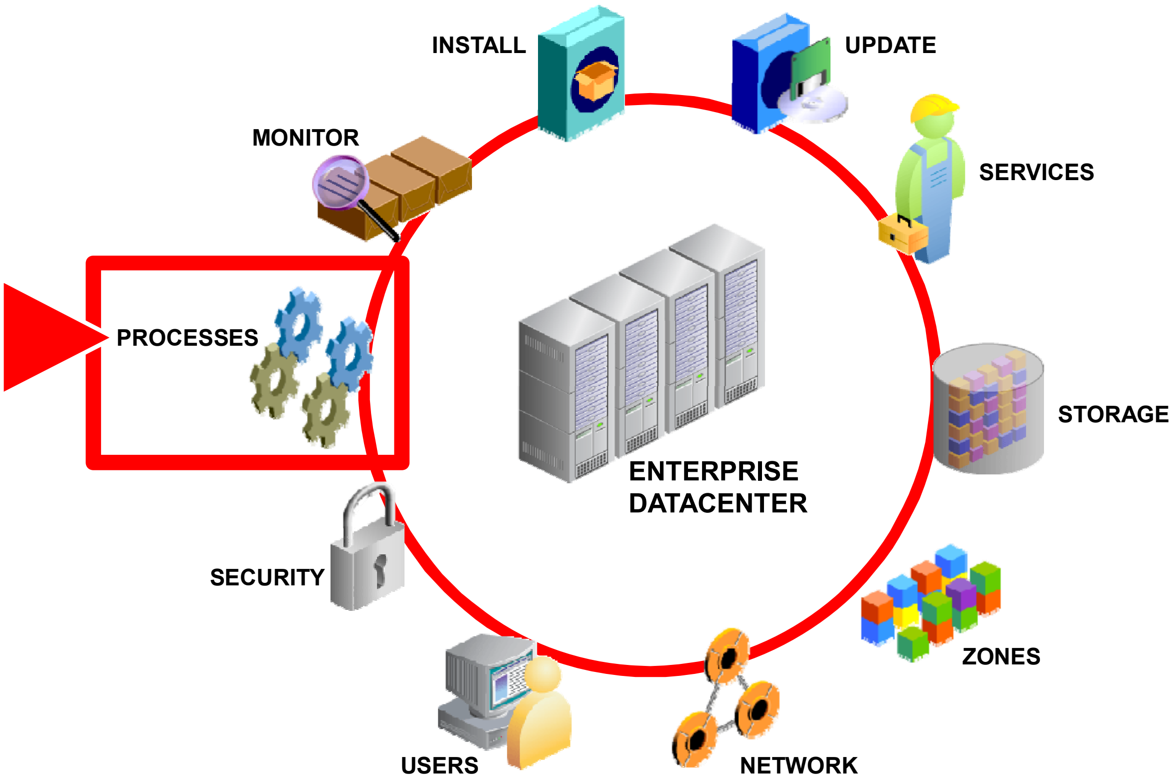
# Objectives

After completing this lesson, you should be able to:

- Plan for system processes management
- Manage system processes
- Schedule system administration tasks

In this lesson, you are introduced to system processes and you learn how to manage them in accordance with a plan. You also learn how to schedule system tasks by using the `crontab` file and how to administer `crontab` files.

# Workflow Orientation



Before you start the lesson, orient yourself to where you are in the job workflow. In the lesson titled “Controlling Access to Systems and Files,” you learned how to secure system and file access for both users and groups. In this lesson, you learn how to manage the system and user processes that the Oracle Solaris 11 operating system uses to run business functions. Your responsibility is to control and monitor these processes to ensure that they run smoothly and that they do not hang or consume too many system resources, such as CPU, memory, and disk space. You also learn how to schedule routine system administration tasks (specifically, how to schedule the more repetitive tasks).

# Lesson Agenda

- Planning for System Processes Management
- Managing System Processes
- Scheduling System Administration Tasks

# Planning for System Processes Management

The system processes management plan ensures that you can:

- Determine what processes are running on the system
- Determine what state a process is in
- Determine which processes are using the greatest percentage of system resources
- Control processes
- Terminate unwanted processes
- Schedule routine tasks

Unlike the previous plans that your company presented you with, the system processes management plan is more about ensuring that you are ready to manage processes on an Oracle Solaris 11 system.

In short, the plan is designed to bring you up to speed on how to perform basic system process management tasks, such as being able to determine the processes that are running on the system, the state the process is in, and the system resource that the process is utilizing. By the time you implement the plan, you should know how to control processes and terminate unwanted processes. And because your time is valuable, the plan also covers how to schedule administrative processes or tasks that recur on a regular basis.

In this section, you are introduced to what processes are, how they are identified by the operating system, and how you can interact with them.

# System Processes

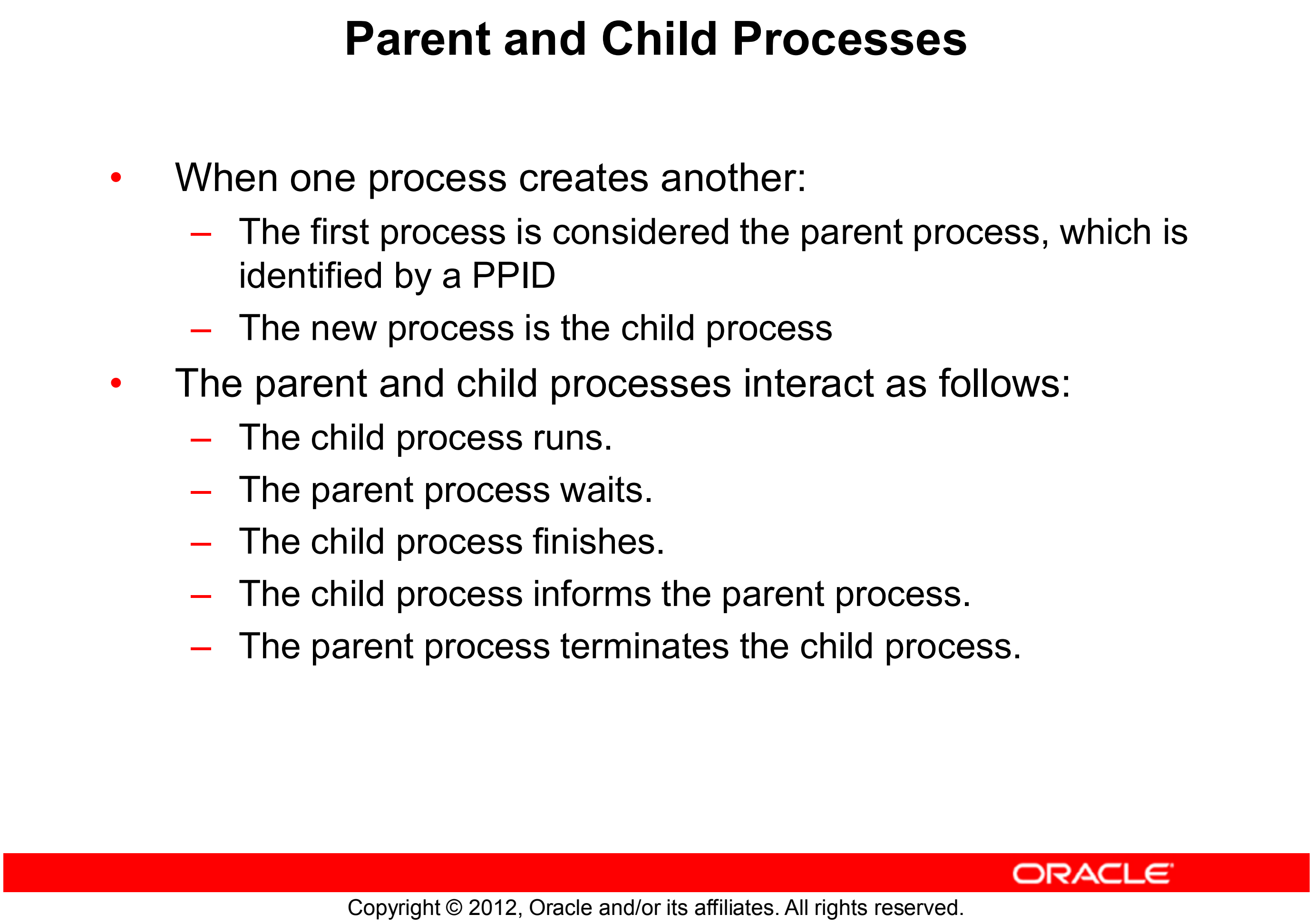
Processes

- Any program that is running on the system
- Assigned a unique process identification (PID) number:
  - Used by the kernel to track, control, and manage a process
  - Displayed by using the `ps` or `pgrep` command

Every program that you run in the Oracle Solaris 11 OS creates a process. When you log in and start the shell, you start a process. When you run a command or when you open an application, you start a process.

The system starts processes called daemons. Daemons are processes that run in the background and provide services. For instance, the desktop login daemon (`dtlogin`) provides a graphical prompt that you use to log in to the operating system.

Every process has a unique process identification number (PID), which the kernel uses to track, control, and manage the process. You can use the `ps` command to view the PID associated with each process that is currently running on the system. If you know the process name, you can use the `pgrep` command to find out the process ID. You learn how to use both the `ps` and `pgrep` commands in the subsequent slides.



# Parent and Child Processes

- When one process creates another:
  - The first process is considered the parent process, which is identified by a PPID
  - The new process is the child process
- The parent and child processes interact as follows:
  - The child process runs.
  - The parent process waits.
  - The child process finishes.
  - The child process informs the parent process.
  - The parent process terminates the child process.

When one process creates another, the first process is considered to be the parent of the new process. The new process is called the child process.

**Note:** The parent process is identified by a parent process ID number (PPID).

While the child process runs, the parent process waits. When the child finishes its task, it informs the parent process. The parent process then terminates the child process. If the parent process is an interactive shell, a prompt appears, indicating that it is ready for a new command.

# Viewing the Parent/Child Process Relationship

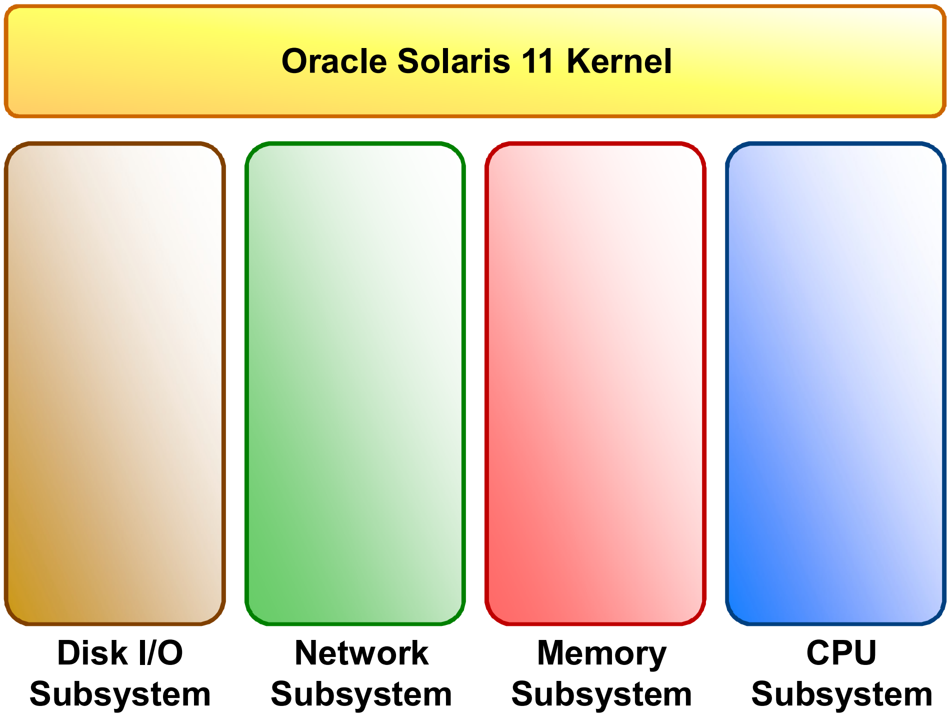
To view the parent/child process relationship, use `ptree` *`pid`*.

```
# ps -ef
   UID    PID  PPID   C    STIME TTY          TIME CMD
---
---
oracle    914   838   0 15:22:32 ?            0:01 gnome-panel
---
---

# ptree 914
710   /usr/sbin/gdm-binary
  716   /usr/lib/gdm-simple-slave --display-id /org/gnome/DisplayManager/Display1
    811   /usr/lib/gdm-session-worker
      838   gnome-session
        914   gnome-panel
```

When you know the PID of a particular process, you can view the parent/child process relationship with the `ptree` utility. To view the processes' tree, use the `ptree` command followed by the process ID. The output from the command contains the specified PIDs or users, with child processes indented from their respective parent processes. In this example, you are looking at the `gnome-panel` process (PID `914`). To find out what the PID is for this process, you ran the `ps` command with the `-ef` option. You then ran the `ptree` command with PID `914`. As you can see, this process is the child of the `gnome-session` process (PID `838`), which is the child of the `gdm-session-worker`, and so on up the tree.

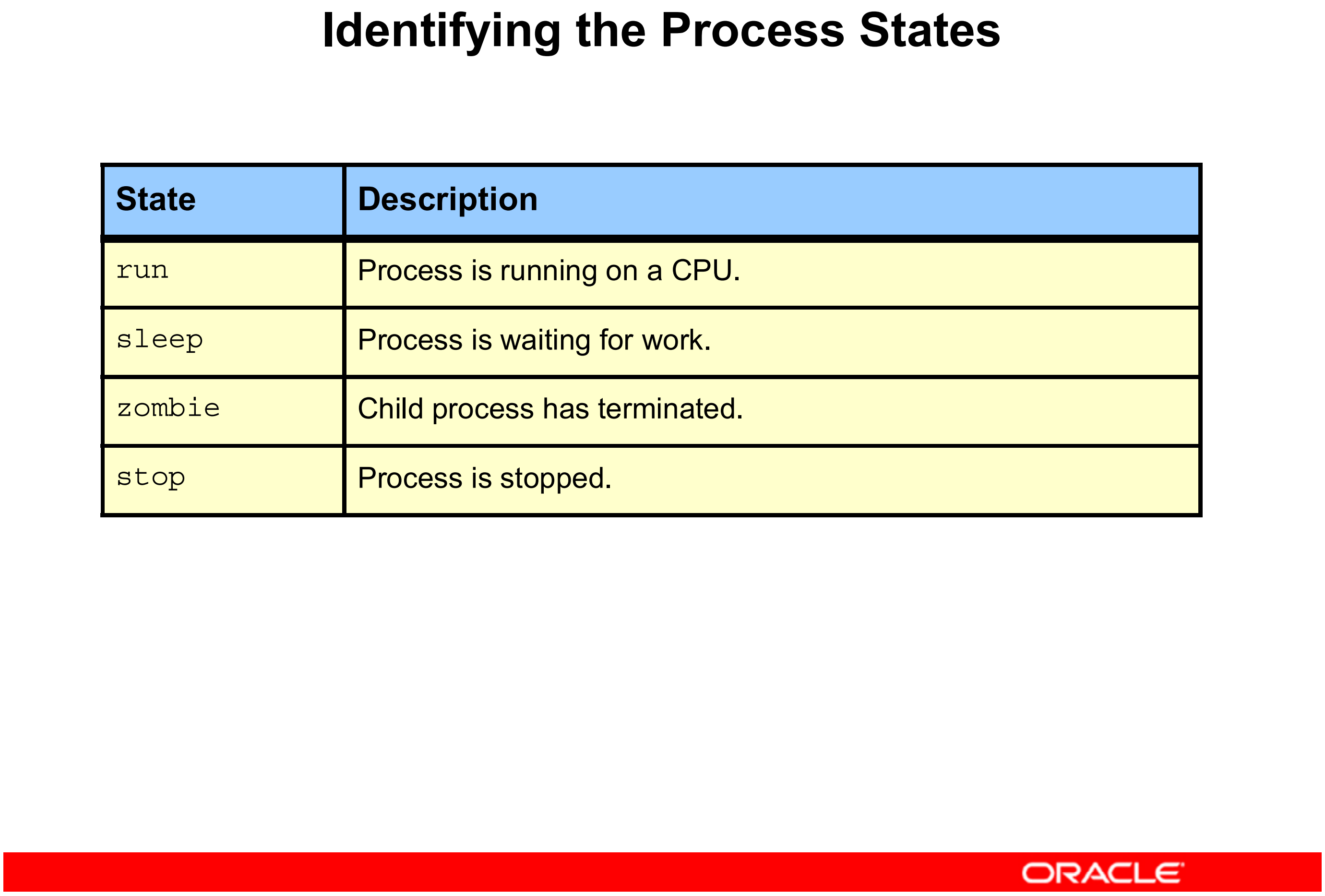
# Identifying the Process Subsystems



Each time you boot a system, execute a command, or start an application, the system activates one or more processes. As processes run, they use disk, network, memory, and CPU subsystem resources. Each of these subsystems plays a vital role in the workings of a system and in support of the business applications that are being run on that system:

- **Disk I/O subsystem:** Controls disk utilization and resourcing, as well as file system performance
- **Network subsystem:** Controls the throughput and directional flow of data between systems over a network connection
- **Memory subsystem:** Controls the utilization and allocation of physical, virtual, and shared memory
- **CPU subsystem:** Controls CPU resources, loading, and scheduling

If not monitored and controlled, processes can consume too much of your system resources, causing the system to run slowly and in some cases, even halt. The Oracle Solaris 11 kernel collects performance-relevant statistics on each of these subsystems, including process information. You can view and use this information to assess the impact that the processes are having on the subsystem resources.



## Identifying the Process States

| State | Description |
|---|---|
| `run` | Process is running on a CPU. |
| `sleep` | Process is waiting for work. |
| `zombie` | Child process has terminated. |
| `stop` | Process is stopped. |

A process can be in one of the following states:

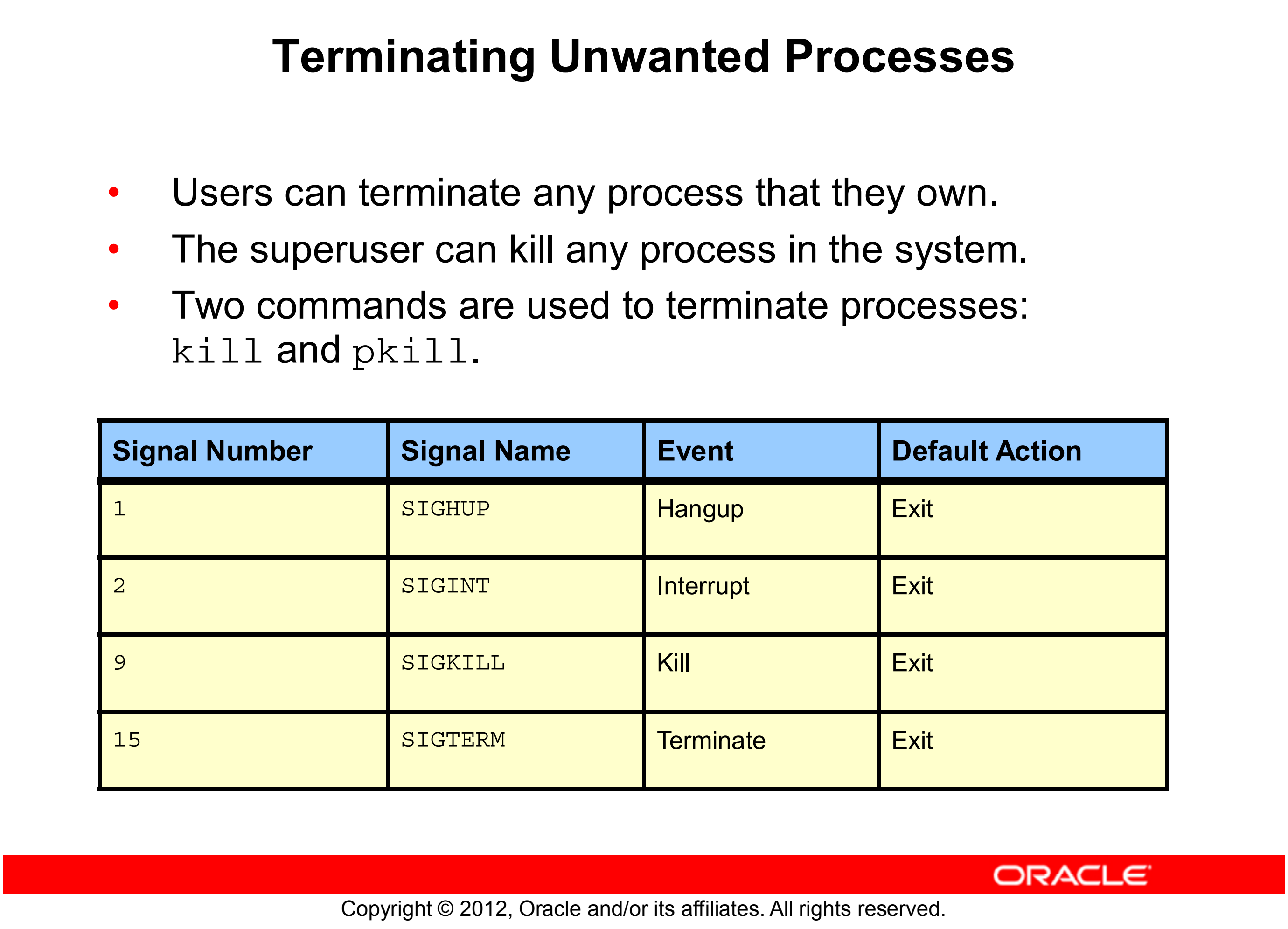
- **`run`**: The process is in the run queue and running on a CPU.
- **`sleep`**: The process is waiting for work.
- **`zombie`**: The child process has terminated but is awaiting acknowledgment from the parent process before it can be removed from the system.
- **`stop`**: The process is stopped.

# Managing and Controlling Processes

| Command | Description |
|---|---|
| `ptree` | Displays the process trees for the specified process ID |
| `ps` | Displays detailed information about the active processes on the system |
| `pgrep` | Displays information about a process based on specific criteria |
| `prstat` | Displays statistics for active processes on a system |
| `pstop` | Stops each process |
| `prun` | Starts each process |

There are many commands that you can use to manage processes. In this lesson, the focus is on the commands that are presented in the table in the slide. You have already learned to use the `ptree` command. In the slides that follow, you learn how to display information about processes by using the `ps` and `pgrep` commands and how to check the status of active processes on the system by using the `prstat` command. You also learn how to control processes by using the `pstop` and `prun` commands.

**Note:** For a full list of commands that you can use to manage processes, see the Oracle Solaris system administration documentation.



## Terminating Unwanted Processes

- Users can terminate any process that they own.
- The superuser can kill any process in the system.
- Two commands are used to terminate processes: `kill` and `pkill`.

| Signal Number | Signal Name | Event | Default Action |
|---|---|---|---|
| 1 | SIGHUP | Hangup | Exit |
| 2 | SIGINT | Interrupt | Exit |
| 9 | SIGKILL | Kill | Exit |
| 15 | SIGTERM | Terminate | Exit |

There might be times when you must terminate an unwanted process. The process might be in an endless loop, or you might have started a large job that you want to stop before it is completed. You can kill (stop) any process that you own. The superuser can kill any process in the system; however, there are processes that should not be terminated (for example, the `init` process). Killing these types of processes can result in a system crash.

There are two commands that you can use to terminate one or more processes: `kill` and `pkill`. The primary difference between the two commands is that the `kill` command requires a PID whereas the `pkill` command uses a process name.

The `kill` and `pkill` commands send signals to processes directing them to terminate. Each signal has a number, name, and an associated event. The default action for all the signals is that the process exits.

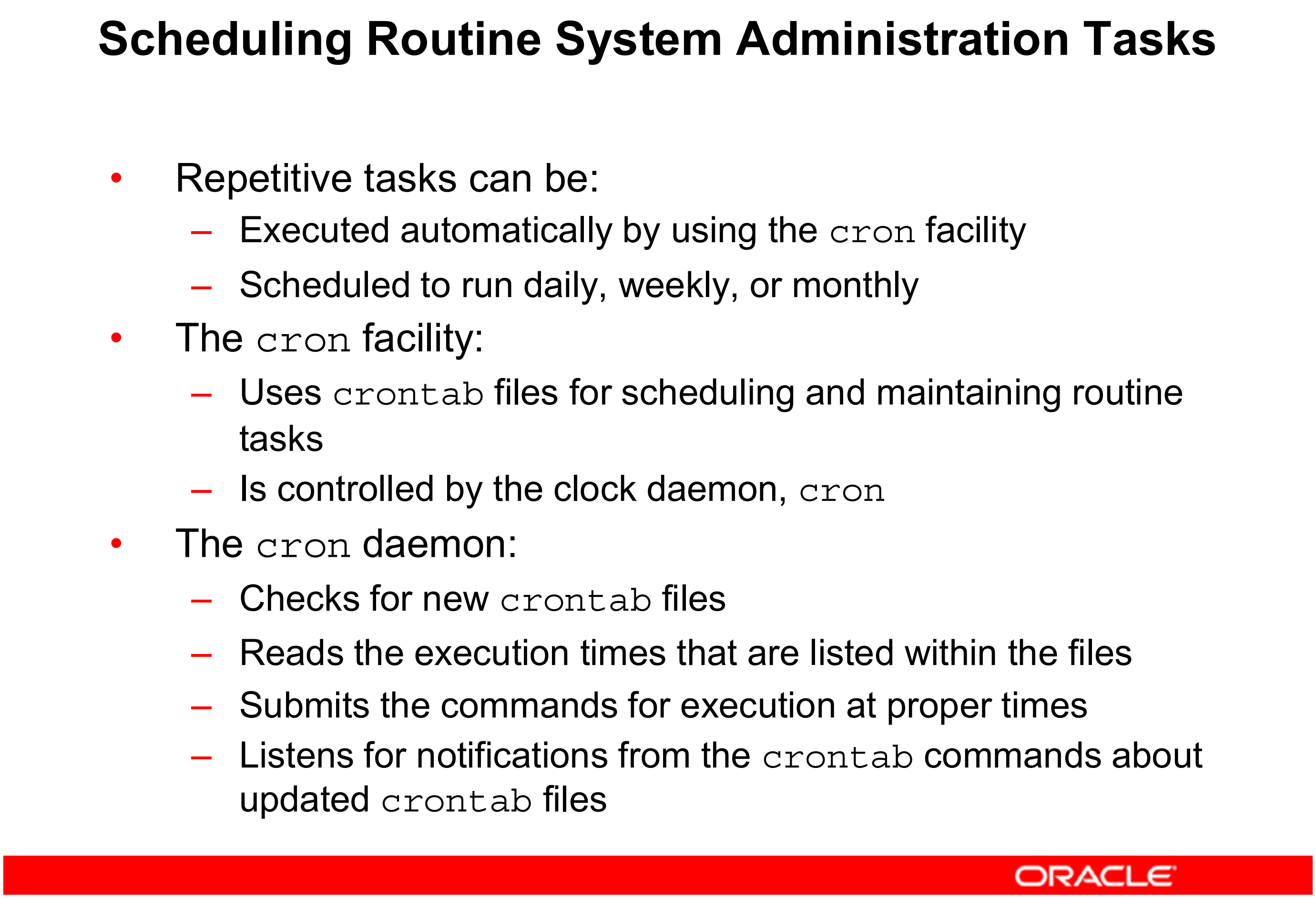
**Note:** The table in the slide contains a subset of the commands that can be used with the `kill` and `pkill` commands. To see a complete list of signals that the `kill` command can send, execute the `kill -l` command or refer to the man page for `signal`:

```
    # man -s3head signal
```

The definitions for the signals presented in the table in the slide are as follows:

- **1, SIGHUP:** Hangup signal to cause a telephone line or terminal connection to be dropped. For certain daemons, such as inetd and in.named, a hangup signal causes the daemon to reread its configuration file.
- **2, SIGINT:** Interrupt signal from your keyboard, usually from a Ctrl + C key combination
- **9, SIGKILL:** A signal to kill a process. A process cannot ignore this signal. The process is terminated instantly with no opportunity to perform an orderly shutdown. Because of this, the -9 signal should not be used to kill certain processes, such as a database process or an LDAP server process. The result is that data might be lost.
- **15, SIGTERM:** A signal to terminate a process in an orderly manner. Some processes ignore this signal.

You learn how to use these signals with the kill and pkill commands later in this lesson.



## Scheduling Routine System Administration Tasks

- Repetitive tasks can be:
  - Executed automatically by using the `cron` facility
  - Scheduled to run daily, weekly, or monthly
- The `cron` facility:
  - Uses `crontab` files for scheduling and maintaining routine tasks
  - Is controlled by the clock daemon, `cron`
- The `cron` daemon:
  - Checks for new `crontab` files
  - Reads the execution times that are listed within the files
  - Submits the commands for execution at proper times
  - Listens for notifications from the `crontab` commands about updated `crontab` files

Most administrators have many system tasks that occur on a regularly basis, such as:

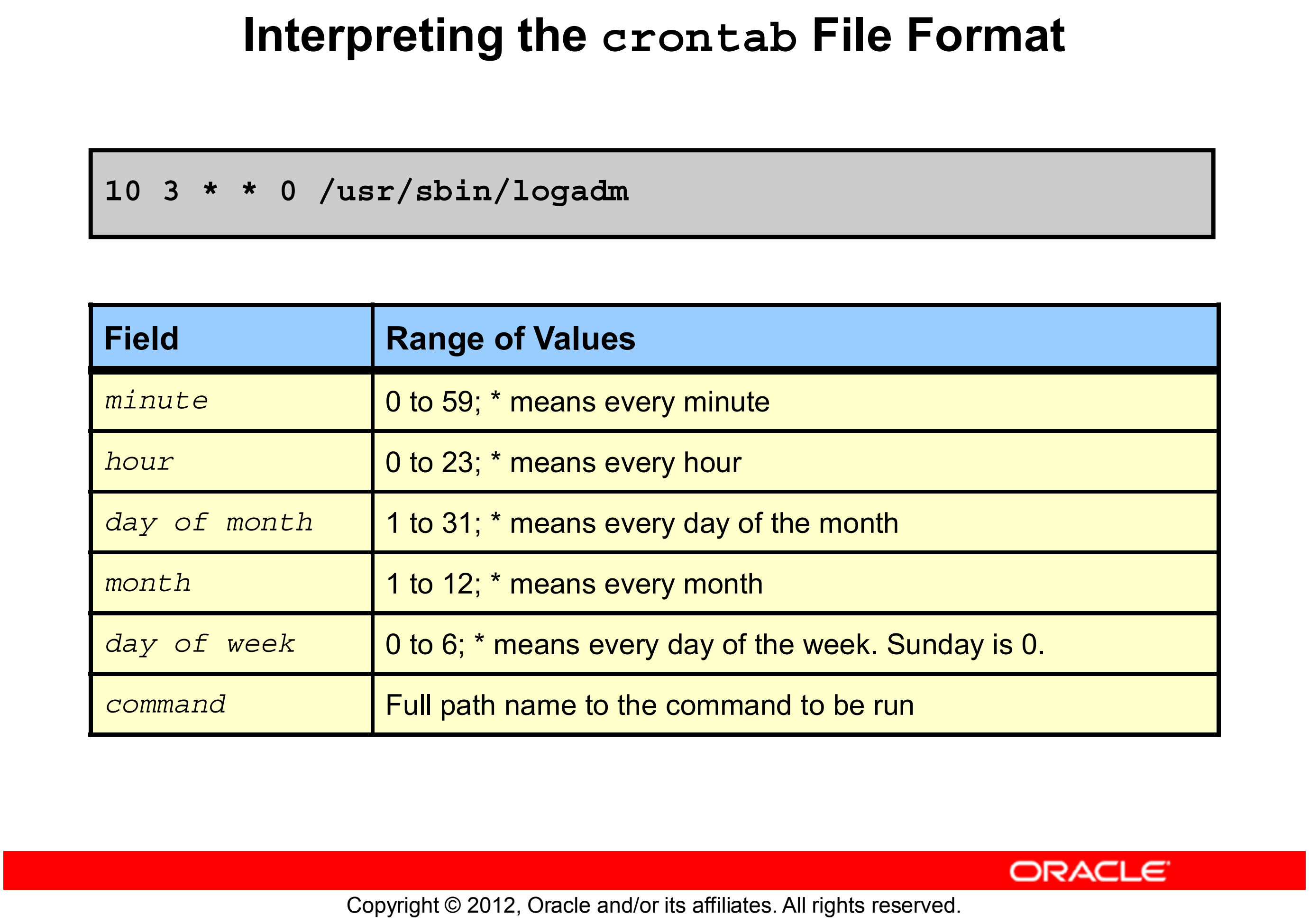
- Removing files that are more than a few days old from temporary directories
- Taking snapshots of the system
- Running system backups

Administrators, as well as users with appropriate privileges, can set up routine tasks to execute automatically on a daily, weekly, or monthly basis by using the `cron` facility.

**Note:** Administrators can create `crontab` files for any user, whereas non-administrative users can create only their own `crontab` files.

The `cron` facility uses `crontab` files for scheduling the tasks. Users create, edit, and manage their routine tasks with these files.

The `cron` facility is controlled by the clock daemon, `cron`, which is responsible for managing the jobs that are submitted through the `crontab` files. The `cron` daemon, which starts at system boot and runs continuously in the background, performs the tasks shown in the slide at system startup.



# Interpreting the `crontab` File Format

```
10 3 * * 0 /usr/sbin/logadm
```

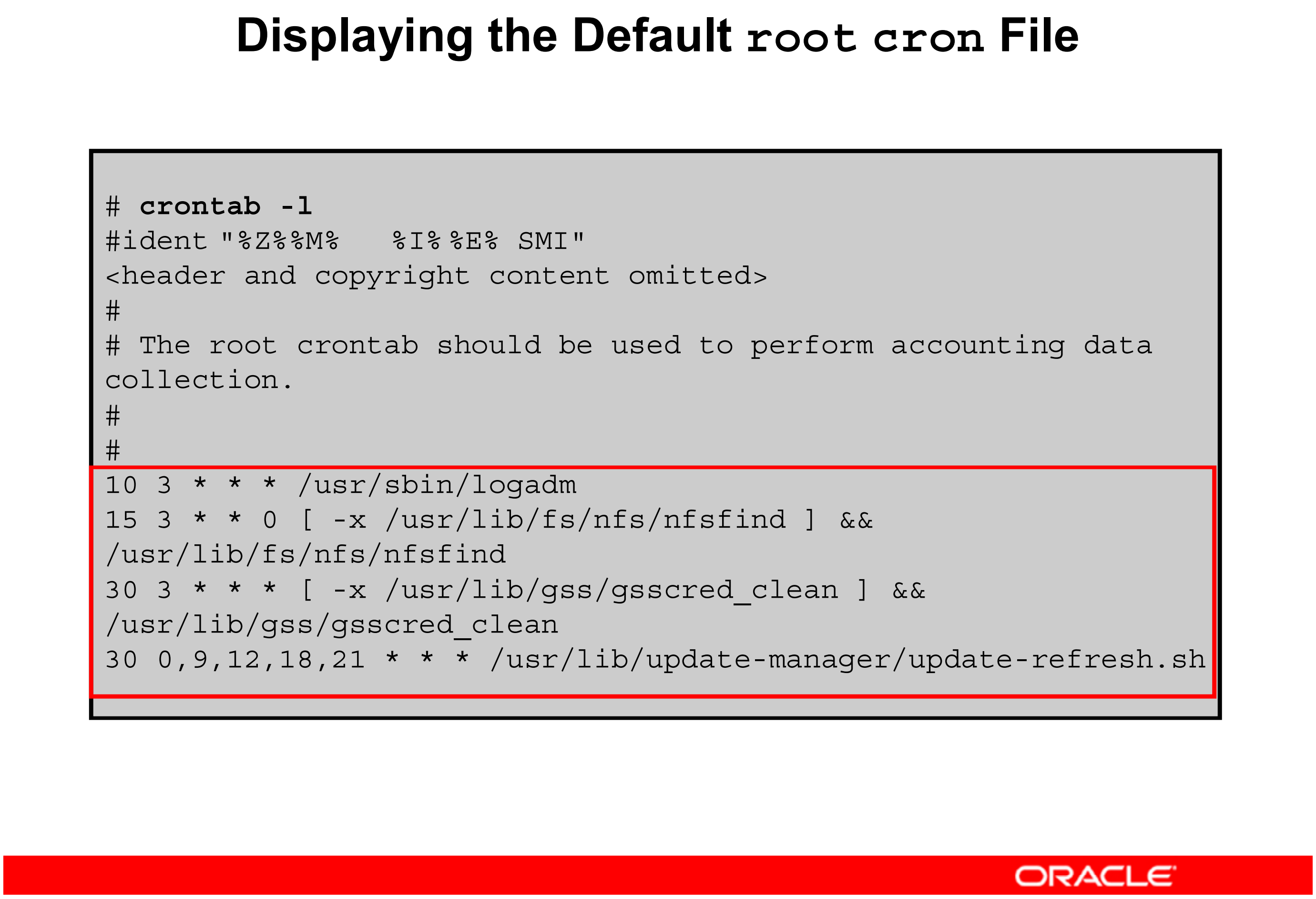
| Field | Range of Values |
|---|---|
| *minute* | 0 to 59; * means every minute |
| *hour* | 0 to 23; * means every hour |
| *day of month* | 1 to 31; * means every day of the month |
| *month* | 1 to 12; * means every month |
| *day of week* | 0 to 6; * means every day of the week. Sunday is 0. |
| *command* | Full path name to the command to be run |

An example of the `crontab` file entry is shown in the slide. The fields, from left to right, and their value ranges are presented in the table that appears below the example.

A `crontab` file entry consists of one line with six fields. The fields are separated by spaces or tabs. The first five fields provide the timing information (that is, the minute, hour, day, month, and day of the week) for the command that is to be scheduled. The last field is the full path to the command.

The entry presented in the example schedules the `logadm` command to be run at 3:10 AM every day of every month on Sunday.

You learn how to create, edit, and manage a `crontab` file later in this lesson.



# Displaying the Default `root cron` File

```
# crontab -l
#ident "%Z%%M%   %I% %E% SMI"
<header and copyright content omitted>
#
# The root crontab should be used to perform accounting data
collection.
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 [ -x /usr/lib/fs/nfs/nfsfind ] &&
/usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] &&
/usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
```

A default `root cron` file is located in the `/var/spool/cron/crontabs` directory that you can display by using the `crontab -l` command. A portion of the file is shown in the slide.

**Note:** Users use the same `crontab -l` command to view the contents of their own `crontab` files. As an administrator, you can view the contents of any regular user's `crontab` file by executing the command:

```
# crontab -l username
```

In the example, you can see that four routine system tasks have already been scheduled. They are as follows:

- **`logadm`:** This command starts the log rotation tool that is used to check the corresponding log file to see if it should be rotated. This task is scheduled to run at 3:10 AM every day.
- **`nfsfind`:** This command starts the NFS find tool that is used to locate NFS. This task is scheduled to run on Sundays at 3:15 AM.

- `gsscred_clean`: This command instructs the system to check for and remove duplicate entries in the Generic Security Service table. This task is scheduled to run at 3:30 AM every day.
- `update-refresh.sh`: This command tells the system to refresh the Update Manager application. This task is scheduled to run on the half hour at midnight, 9 AM, 12 noon, 6 PM, and 9 PM every day.

If you have a new routine task that you want to schedule, you can schedule it by adding a new entry to this file.

# Introducing the `crontab` Files

- The files are maintained in `/var/spool/cron/crontabs`.
- Access to the files is controlled through:
  - `/etc/cron.d/cron.allow`
  - `/etc/cron.d/cron.deny`
- Only specified users are permitted to perform `crontab` tasks based on the access files, as follows:
  - If the `cron.allow` file exists, only the users listed in this file can create, edit, display, or remove the `crontab` files.
  - If the `cron.allow` file does not exist, all users, except the users listed in the `cron.deny` file, can create, edit, display, or remove the `crontab` files.
  - If neither file exists, only the superuser can run the `crontab` command.

All `crontab` files are maintained in the `/var/spool/cron/crontabs` directory and are stored as the login name of the user that created the `cron` job. You can control access to the `crontab` files through two files in the `/etc/cron.d` directory:

`/etc/cron.d/cron.allow` and `/etc/cron.d/cron.deny`.

These files permit only specified users to perform the `crontab` tasks, such as creating, editing, displaying, or removing their own `crontab` files.

The Oracle Solaris OS provides a default `cron.deny` file. The file consists of a list of usernames, one per line, of users who are not allowed to use `cron`. The `/etc/cron.d/cron.allow` file does not exist by default, so all users (except those listed in the `cron.deny` file) can access their `crontab` files. By creating a `cron.allow` file, you can list only those users who can access `crontab` commands. The file consists of a list of usernames, one per line.

The interaction between the `cron.allow` and `cron.deny` files follows the rules shown in the slide.

# Introducing the Default `cron.deny` File

```
# cat /etc/cron.d/cron.deny
daemon
bin
nuucp
listen
nobody
noaccess
```

An example of the default `cron.deny` file is shown in the slide. Here you can see the list of users, one per line, who are not allowed to use `cron`. As an administrator, you can edit this file to add other usernames that should be denied access to the `crontab` command. You learn how to do this later in this lesson.

# Implementing the System Process Management Plan

Your assignment is to:

- Determine what processes are running on the system
- Determine what state the processes are in
- Determine which processes are using the greatest percentage of system resources
- Terminate unwanted processes
- Schedule routine tasks
- Administer `crontab` files

As part of the system processes management plan, you have been tasked with learning how system processes are managed in the Oracle Solaris 11 operating system. This assignment includes determining what processes are running on the system, what state each process is in, and which processes are using the greatest percentage of system resources. You also practice how to terminate unwanted processes and schedule routine tasks by using `crontab` files. Your final task is to learn how to control access to the `crontab` files.

# Quiz

What state is a parent process in when it is waiting for an event to complete?

a. `run`

b. `sleep`

c. `zombie`

d. `stop`

**Answer: b**

# Quiz

When used with `kill` or `pkill`, which signal terminates a process instantly with no opportunity to perform an orderly shutdown?

a. `1, SIGHUP`
b. `2, SIGINT`
c. `9, SIGKILL`
d. `15, SIGTERM`

**Answer: c**

# Quiz

If the `cron.allow` file does not exist, all users (except the users listed in the `cron.deny` file) can create, edit, display, or remove the `crontab` files.

a. True

b. False

**Answer: a**

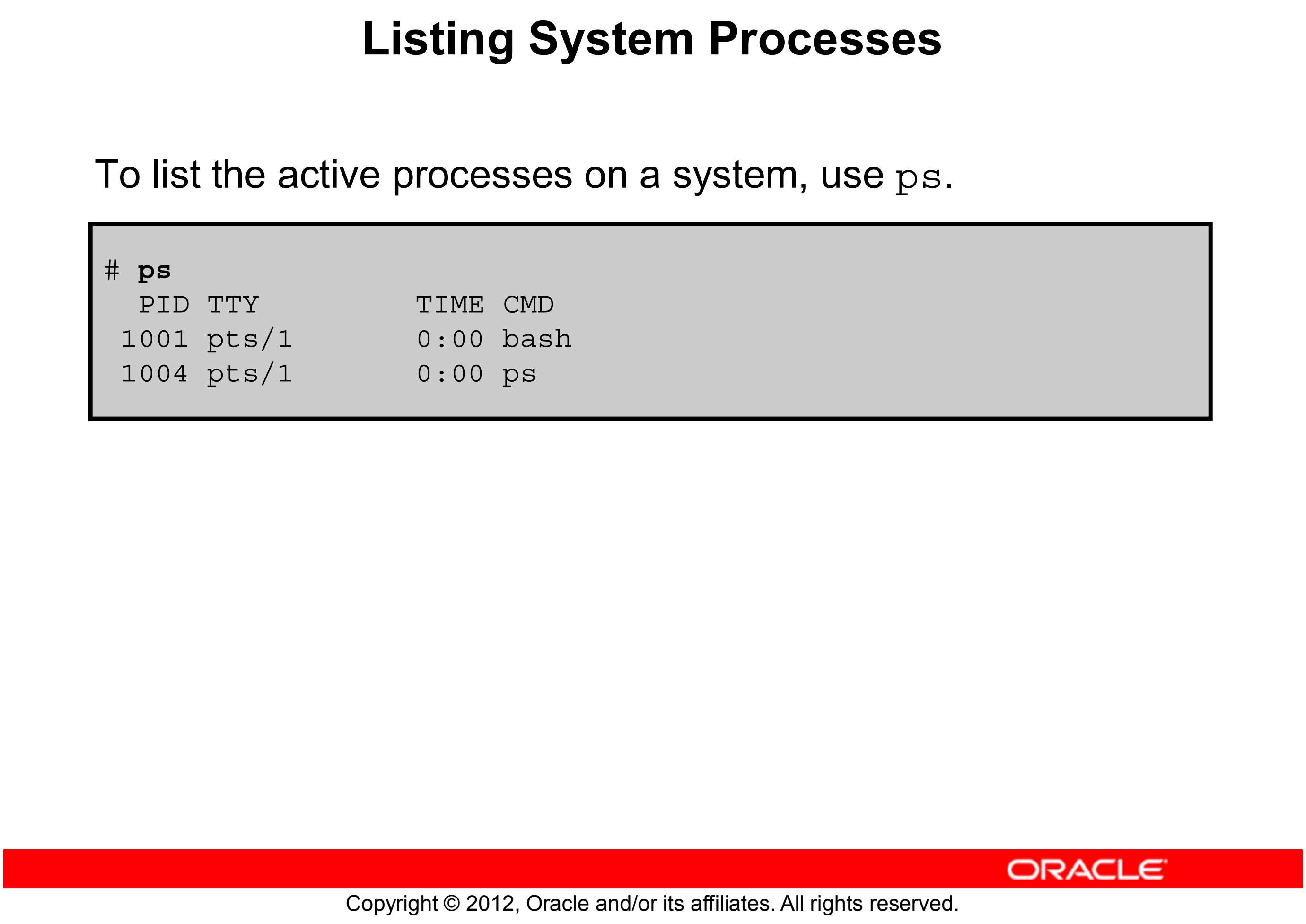
# Lesson Agenda

- Planning for System Processes Management
- Managing System Processes
- Scheduling System Administration Tasks

# Managing System Processes

- Listing system processes
- Displaying information about processes
- Displaying active process statistics
- Stopping and starting a system process
- Killing a process

In the subsequent slides, you learn how to list, temporarily stop, restart, and kill system processes. You are also shown how to display process information and statistics.



# Listing System Processes

To list the active processes on a system, use `ps`.

```
# ps
  PID TTY          TIME CMD
 1001 pts/1        0:00 bash
 1004 pts/1        0:00 ps
```

To manage the processes on the system, you must know what processes are running. To list the processes that are currently running on the system (that is, the active processes), use the `ps` command. This command, without any options, displays the process ID (`PID`), terminal identifier (`tty`), cumulative execution time (`TIME`), and the command name (`CMD`).

In the example in the slide, the default process is the current shell, which, in this case, is `bash`. The second line that shows `ps` under the `CMD` column is the `echo` command; it is not a `ps` process running.

To see additional process information, you can use a variety of options, most common of which are:

- **`-a`:** Print information about all the processes that are most frequently requested, except process group leaders and processes that are not associated with a terminal.
- **`-e`:** Print information about every process that is now running.
- **`-f`:** Generate a full listing.

- **`-l`:** Generate a long listing.
- ***`-o format`*:** Write information according to the format specification given in *`format`*. Multiple `-o` options can be specified; the format specification is interpreted as the space-character-separated concatenation of all format option arguments.

For a full list of options, see the `ps`(1) man page.

# Listing System Processes

To generate a full listing of every currently running process, use `ps -ef`.

```
# ps -ef
     UID   PID  PPID   C    STIME TTY         TIME CMD
    root     0     0   0 06:50:42 ?           0:02 sched
    root     5     0   0 06:50:40 ?           0:02 zpool-rpool
    root     1     0   0 06:50:43 ?           0:00 /sbin/init
---
---
---
```

To generate a more detailed list of every process that is currently running on the system, you can use the `ps` command with the `-e` and `-f` options. This command displays the following information:

- **`UID`:** Effective user ID number of the process. Examples include `root`, `netadm`, `dladm`, and `daemon`.
- **`PID`:** Process ID of the process

  **Note:** As you will see shortly, you need the PID to kill a process.
- **`PPID`:** Process ID of the parent process
- **`C`:** Processor utilization for scheduling (obsolete)
- **`STIME`:** Starting time of the process, given in hours, minutes, and seconds. A process that begins more than 24 hours before the `ps` inquiry is executed is given in months and days.
- **`TTY`:** Controlling terminal for the process. The message `?` is printed when there is no controlling terminal.
- **`CMD`:** Command name

The example, which presents a partial output for the command, shows several processes belonging to root and associated with the following commands: `sched`, `zpool-rpool`, and `/sbin/init`.

# Displaying Information About Processes

To display the PID of a particular process, use `pgrep` *`process`*.

```
# pgrep sched
0
```

```
# pgrep -l manager
  968 updatemanagerno
  973 nwam-manager
```

To display more information about a particular process, you can use the `pgrep` command. The `pgrep` command looks through the currently running processes and lists the process IDs that match the selection criteria that you have specified. For example, if you know the process name and need to find out what the PID is for the process, you can do so by using the `pgrep` *`processname`* command, as shown in the first example in the slide. The second example in the slide shows the results of using the `grep -l` command to perform a search on a partial process name (`manager`).

Here you can see that two processes with the keyword `manager` are running on the system: the Update Manager process (PID `968`) and the NWAM Manager process (PID 973).

For information about the different selection criteria that you can use with the `pgrep` command, see the `pgrep`(1) man page.

# Displaying Active Process Statistics

To display statistical information about running processes, use `prstat`.

```
# prstat
   PID USERNAME  SIZE   RSS STATE   PRI NICE      TIME  CPU PROCESS/NLWP
   920 oracle    159M  140M run      59    0   0:02:25 1.4% java/20
   184 root     8976K 3204K cpu0     59    0   0:00:00 0.5% prstat/1
   982 oracle     88M   19M run      59    0   0:00:19 0.2% gnome-terminal/2
   693 oracle     63M   51M run      58    0   0:00:29 0.2% Xorg/3
   949 oracle     27M   13M sleep    59    0   0:00:07 0.0% nwam-manager/2
    11 root       11M   10M sleep    59    0   0:00:35 0.0% svc.configd/26
   956 oracle     12M 5672K run      59    0   0:00:07 0.0% xscreensaver/1
   938 oracle     51M   32M sleep    12   19   0:00:02 0.0% updatemanagerno/1
   915 oracle     27M   16M sleep    59    0   0:00:01 0.0% metacity/1
   990 root     8660K 2600K run      32    0   0:00:00 0.0% bash/1
   921 oracle     87M   17M sleep    59    0   0:00:01 0.0% gnome-power-man/1
<output omitted>
Total: 92 processes, 451 lwps, load averages: 0.10, 0.11, 0.09
```

To display dynamic, statistical information about process size, state, and percentage of CPU usage, use the `prstat` command, as shown in the example in the slide.

**Note:** The `prstat` utility iteratively examines all active processes on the system and reports statistics based on the selected output mode and sort order.

The command displays the following information:

- **`PID`:** Process ID of the process
- **`USERNAME`:** Login name or UID of the owner of the process
- **`SIZE`:** Total virtual memory size of the process
- **`RSS`:** Resident set size of the process, which represents the physical memory being used by the process, in kilobytes (`K`), megabytes (`M`), or gigabytes (`G`)

- **STATE:** State of the process
  - **cpu*N*:** 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.
- **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:** The value that is 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.
- **TIME:** Cumulative execution time for the process, given in hours, minutes, and seconds
- **CPU:** Percentage of recent CPU time used by the process
- **PROCESS/NLWP:** Name of the process/the 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 that are 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.

The `Total` line at the bottom of the list of processes identifies the total number of processes, the total number of lightweight processes (`lwps`), and the CPU load averages. The averages are based on one-, five-, and 15-minute intervals. By using the `prstat` command, you can monitor the processes to ensure that they are not using up the CPU capacity.

**Note:** If you do not specify an option, the `prstat` command examines all processes and reports statistics sorted by CPU usage.

Now take a closer look at the `gnome-terminal/2` and `updatemanagerno/1` processes as examples of how to interpret the `prstat` command output. The `gnome-terminal/2` (PID `982)` is owned by the `oracle` user. It has a total virtual memory size of `88M` and a resident set size of `19M`. The process is running and has a priority of `59` with no nice value calculation. At the time the example was taken, the process had been running for `19` seconds and was using `0.2%` of the CPU's capacity.

The `updatemanagerno/1` process (PID `938`) is also owned by the `oracle` user. It has a total virtual memory size of `51M` and a resident set size of `32M`. At the time the example was taken, the process was sleeping. It has a priority of `12` with a nice value of `19`. It ran for two seconds and was utilizing no CPU capacity.

# Displaying Active Process Statistics

```
# prstat -s cpu 20 3
   PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU PROCESS/NLWP
   933 oracle     67M   50M sleep   59    0   0:05:54 1.7% java/20
   995 oracle     89M   20M sleep   58    0   0:00:25 1.6% gnome-terminal/2
   703 oracle     59M   48M sleep   59    0   0:00:42 1.4% Xorg/3
  1207 root     8912K 3148K cpu0    54    0   0:00:00 0.4% prstat/1
   736 zfssnap    20M 5056K sleep   59    0   0:01:49 0.4% time-sliderd/2
   944 oracle     12M 5568K sleep   59    0   0:00:07 0.2% xscreensaver/1
<output omitted>
Total: 92 processes, 453 lwps, load averages: 0.14, 0.11, 0.09
```

```
# prstat -s rss 20 3
   PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU PROCESS/NLWP
   933 oracle     67M   50M sleep   59    0   0:06:06 1.6% java/20
   703 oracle     59M   48M sleep   59    0   0:00:45 1.7% Xorg/3
   931 oracle    108M   39M sleep   49    0   0:00:03 0.1% nautilus/1
   968 oracle     54M   32M sleep   12   19   0:00:05 0.0% updatemanagerno/1
   928 oracle     93M   25M sleep   59    0   0:00:01 0.0% gnome-panel/1
   601 root       37M   24M sleep   59    0   0:00:05 0.0% fmd/27
   946 oracle     93M   21M sleep   59    0   0:00:04 0.0% isapython2.6/1
   934 oracle     36M   20M sleep   59    0   0:00:00 0.0% isapython2.6/1
<output omitted>
Total: 92 processes, 453 lwps, load averages: 0.15, 0.12, 0.10
```

You can use the `prstat` command options to target certain statistical information that you might be particularly interested in, such as which process has the highest CPU usage. To see this particular statistic, use the `prstat` command with the `-s` option followed by `cpu` and a specified time frame, such as every 20 seconds, 3 times, as shown in the example.

**Note:** The `-s` option sorts the output in descending order by the specified *key*, which can be CPU usage (`cpu`), priority (`pri`), resident set size (`rss`), process image (`size`), or execution time (`time`). Only one key can be specified. To see the same type of output but in ascending order, use the `-S` option.

In the example, the `java/20` process is using the most CPU at `1.7%`.

To see which processes are using the most memory, you could use the same command but replace `cpu` with `rss`, as shown in the second example. Here you see that again the `java/20` process is using the most memory resource.

For a full list of the `prstat` command options, see the `prstat` man page.

## Stopping and Starting a System Process

1. Using `pgrep` *`process`*, obtain the process ID of the process that you want to control.
2. Temporarily stop the process by using `pstop` *`pid`*.
3. Verify that the process has stopped by using `ps -ef | grep` *`pid`*.
4. Restart the process by using `prun` *`pid`*.
5. Verify that the process has restarted by using `ps -ef | grep` *`pid`*.

As was discussed earlier, because processes use system resources, it is important that they are monitored and kept under control. To control the processes, you might have to clear hung processes, terminate other processes, and stop or restart processes. To temporarily stop and then restart a process, you perform the steps shown in the slide.

**Note for step 2:** Note the time that you stopped the process. You need this information for the next step.

**Note for step 3:** To verify that the process has stopped, check whether the elapsed time on the extreme right is incrementing. To do this check, run the `ps -ef | grep` *`pid`* command twice in succession.

**Note for step 5:** To verify that the process has restarted, check whether the elapsed time on the extreme right is incrementing. To do this check, run the `ps -ef | grep` *`pid`* command twice in succession as you did in step 3.

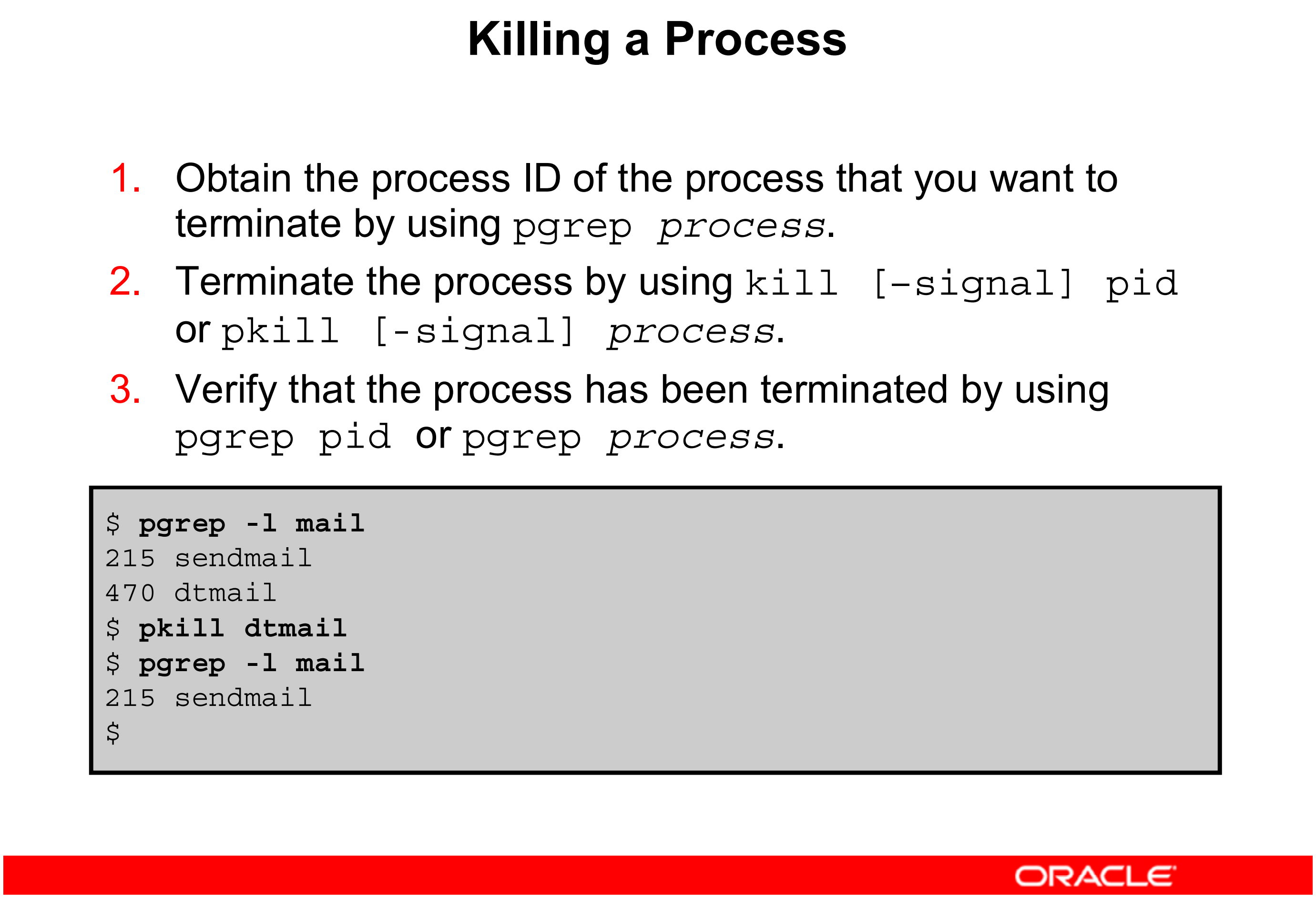
# Stopping and Starting a System Process: Example

```
# pgrep rptpgm
3366
# pstop 3366
# ps -ef | grep 3366
root   3366  2864  47 16:09:54 pts/2   0:48 dd if=/dev/zero of=/dev/null
# ps -ef | grep 3366
root   3366  2864  47 16:09:54 pts/2   0:48 dd if=/dev/zero of=/dev/null

# prun 3366
# ps -ef | grep 3366
root   3366  2864  47 16:10:17 pts/2   0:52 dd if=/dev/zero of=/dev/null
# ps -ef | grep 3366
root   3366  2864  47 16:10:20 pts/2   1:01 dd if=/dev/zero of=/dev/null
```

In the example shown in the slide, you have a process running called rptpgm (for report program) that takes a very long time to execute and is consuming resources that you currently need for other jobs. You have decided to temporarily stop this process to allow the other shorter and more important jobs to complete. The first step is to identify the PID for the process by using the pgrep command. It is 3366. Next, you temporarily stop the process by using the pstop command. To determine whether the process has stopped running, you run the ps -ef command with the PID twice and check the elapsed time to see whether it is incrementing. If it is not, the process has been stopped.

To restart the process, you use the prun command with the PID, and then verify that the process is running by again running the ps -ef command twice and checking the elapsed time. The time should now be incrementing.



## Killing a Process

1. Obtain the process ID of the process that you want to terminate by using `pgrep` *`process`*.
2. Terminate the process by using `kill [-signal] pid` or `pkill [-signal]` *`process`*.
3. Verify that the process has been terminated by using `pgrep pid` or `pgrep` *`process`*.

```
$ pgrep -l mail
215 sendmail
470 dtmail
$ pkill dtmail
$ pgrep -l mail
215 sendmail
$
```

You can use the steps shown in the slide to terminate an unwanted process.

When using the `pkill` command to terminate a process, first try using the command by itself, without including a signal option. Wait a few minutes to see whether the process terminates before using the `pkill` command with the `-9` signal. As discussed earlier, using the `-9` signal (`SIGTERM`) with the `pkill` command ensures that the process terminates promptly; however, it should be used with caution. The syntax for killing a process with the `-9` signal is as follows:

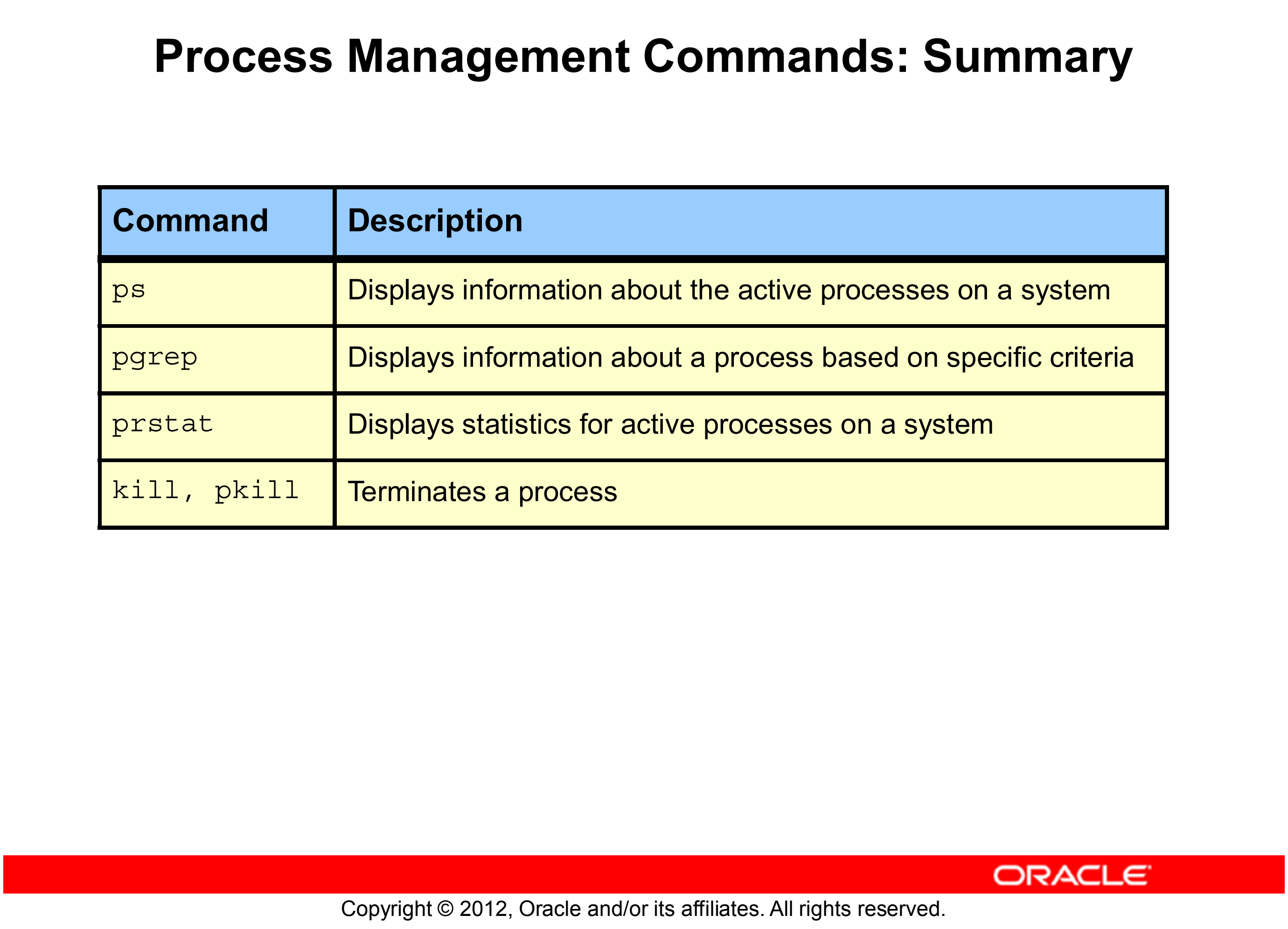
```
$ pkill -9 process
```

**Note for step 2:** When no signal is included in the `pkill` command line syntax, the default signal that is used is -15 (`SIGKILL`).

**Note for step 3:** The process that you terminated should no longer be listed in the output of the `pgrep` command.

You can terminate more than one process at the same time by using the following syntax:

```
# kill [-signal]pid pid pid
# pkill [-signal] process process process
```



# Process Management Commands: Summary

| Command | Description |
|---|---|
| `ps` | Displays information about the active processes on a system |
| `pgrep` | Displays information about a process based on specific criteria |
| `prstat` | Displays statistics for active processes on a system |
| `kill, pkill` | Terminates a process |

The table shown in the slide summarizes the commands that you can use to list, control, and kill processes, as well as to display process information.

# Practice 10-1 Overview: Managing System Processes

This practice covers the following topics:

- Listing system processes
- Verifying process status
- Terminating a process
- Controlling a process

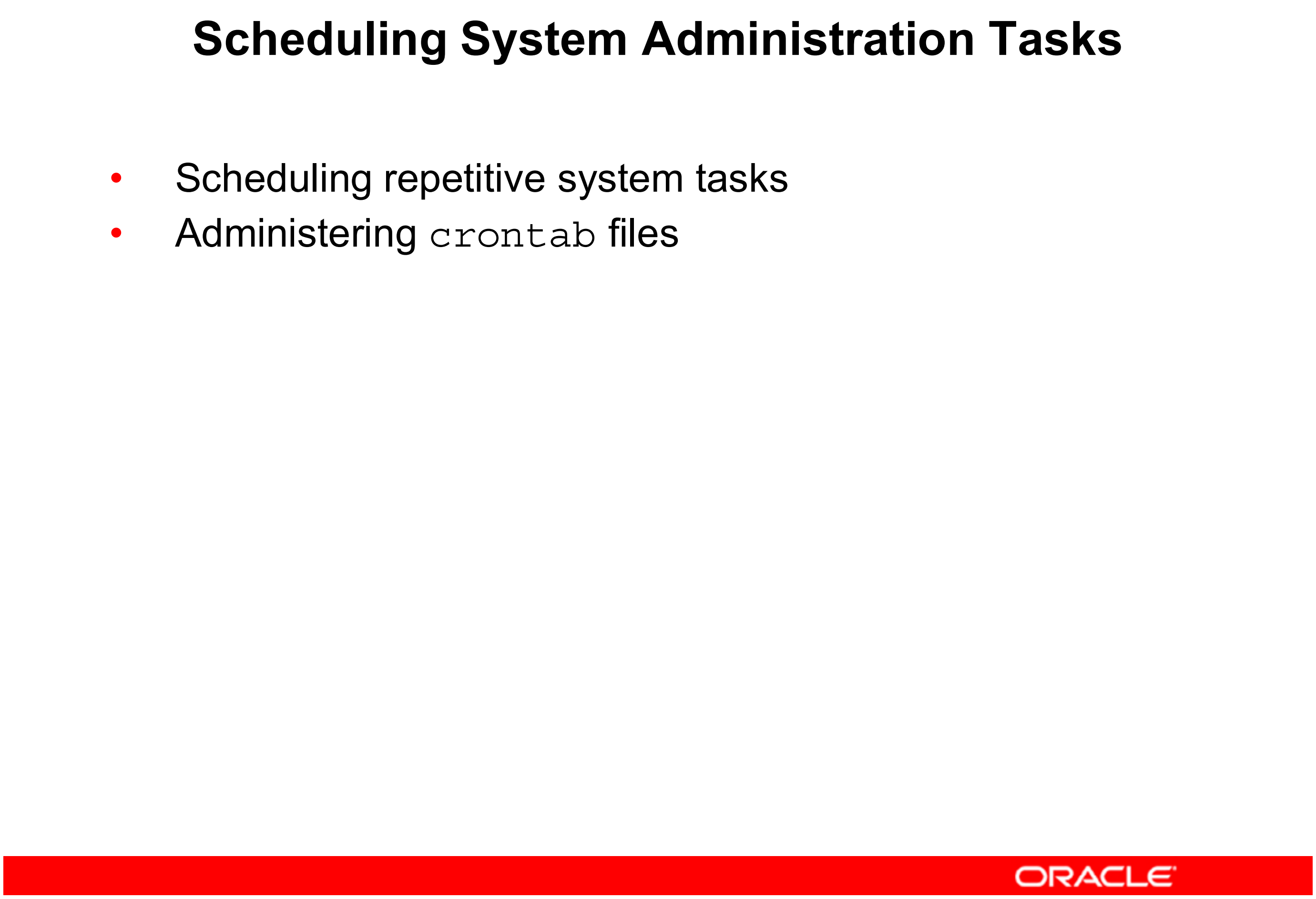
In the practices for this lesson, you perform the following tasks:

- **Practice 10-1:** Managing system processes
- **Practice 10-2:** Scheduling system tasks

You will find Practice 10-1 in your *Activity Guide*. It should take about 30 minutes to complete.

# Lesson Agenda

- Planning for System Processes Management
- Managing System Processes
- Scheduling System Administration Tasks



## Scheduling System Administration Tasks

- Scheduling repetitive system tasks
- Administering `crontab` files

In the subsequent slides, you learn to schedule repetitive system tasks and to administer the `crontab` files that are used to schedule the tasks.

**Note:** You can schedule an automatic one-time execution of a command by using the `at` command. Because this command is not used very often, you are not taught how to use it in this course. To learn more about the `at` command, see the *Oracle Solaris Administration: Common Tasks* guide.

## Scheduling Repetitive System Tasks

1. Set up `vi` as the default editor by using `EDITOR=vi`.
2. Create a new `crontab` file by using `crontab -e` [*username*].
3. Verify that your `crontab` file changes by using `crontab -l` [*username*].
4. Verify that the `crontab` file exists by using `ls -l /var/spool/cron/crontabs`.

To schedule a task, you create or edit a `crontab` file. The simplest way to create a `crontab` file is to use the `crontab -e` command to edit the existing administrator's `cron` file. This command invokes the text editor that has been set for your system environment.

**Note:** The default editor for your system environment is defined in the `EDITOR` environment variable. If this variable has not been set, the `crontab` command uses the default editor, `ed`.

The steps for setting up `vi` as the default editor and creating a new `crontab` file are shown in the slide.

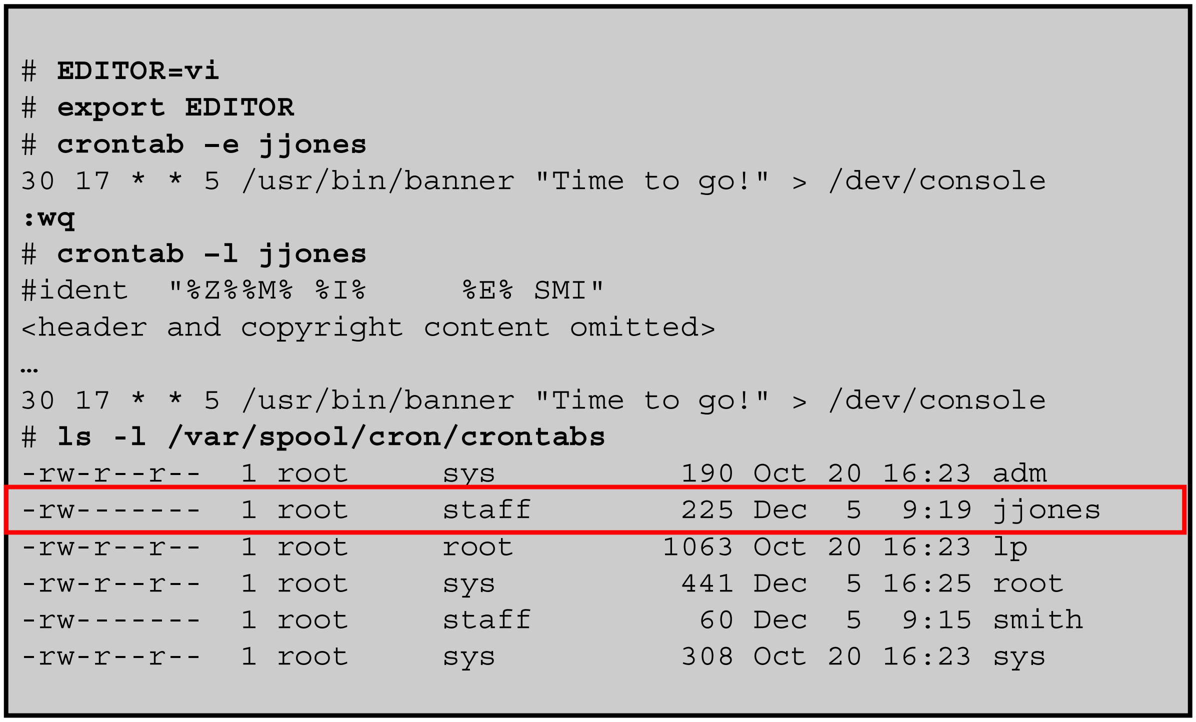
**Note for step 2:** If you are creating a crontab file for another user, you would specify the user's name as part of the `crontab -e` command (for example: `crontab -e jjones`). Follow these guidelines for using special characters in the `crontab` time fields:

- Use a space to separate each field.
- Use a comma to separate multiple values.
- Use a hyphen to designate a range of values.
- Use an asterisk as a wildcard to include all possible values.
- Use a comment mark (`#`) at the beginning of a line to indicate a comment or blank line.

When you have finished creating the new file, it is placed in the `/var/spool/cron/crontabs` directory.

**Note:** If users do not redirect the standard output and standard errors of their commands in the `crontab` file, any generated output or errors are mailed electronically to the user.

# Scheduling Repetitive System Tasks: Example



```
# EDITOR=vi
# export EDITOR
# crontab -e jjones
30 17 * * 5 /usr/bin/banner "Time to go!" > /dev/console
:wq
# crontab -l jjones
#ident  "%Z%%M% %I%     %E% SMI"
<header and copyright content omitted>
…
30 17 * * 5 /usr/bin/banner "Time to go!" > /dev/console
# ls -l /var/spool/cron/crontabs
-rw-r--r--  1 root     sys          190 Oct 20 16:23 adm
-rw-------  1 root     staff        225 Dec  5  9:19 jjones
-rw-r--r--  1 root     root        1063 Oct 20 16:23 lp
-rw-r--r--  1 root     sys          441 Dec  5 16:25 root
-rw-------  1 root     staff         60 Dec  5  9:15 smith
-rw-r--r--  1 root     sys          308 Oct 20 16:23 sys
```

In the example in the slide, you have set up the `vi` editor as the default editor and have created a new `crontab` file for the user `jjones`. This “`Time to go!`” reminder is scheduled to run every Friday at 5:30 PM and appears in the console window. You verified the change by displaying the `crontab` file with the `crontab -l` command. The final step is to verify that the `crontab` file exists in the `/var/spool/cron/crontabs` directory, which it does.

# Administering `crontab` Files

- Removing a `crontab` file
- Denying `crontab` command access
- Limiting `crontab` command access to specified users

# Removing a `crontab` File

To remove a `crontab` file, use `crontab -r` *`username`*.

```
# crontab -r jjones
```

To verify that the `crontab` file has been removed, use `ls -l /var/spool/cron/crontabs`.

```
# ls -l /var/spool/cron/crontabs
-rw-r--r--  1 root     sys          190 Oct 20 16:23 adm
-rw-r--r--  1 root     root        1063 Oct 20 16:23 lp
-rw-r--r--  1 root     sys          441 Dec  5 16:25 root
-rw-------  1 root     staff         60 Dec  5  9:15 smith
-rw-r--r--  1 root     sys          308 Oct 20 16:23 sys
```

The correct way to remove a `crontab` file is to use the `crontab -r` command. This command removes a user's `crontab` file from the `crontab` directory. Typical users can remove only their own `crontab` file. The superuser can delete any user's `crontab` file.

**Caution:** If you accidentally type the `crontab` command with no option, you can press the interrupt character for your editor. This character allows you to quit without saving changes. If you save the change and exit the file, the existing `crontab` file is overwritten with an empty file.

To verify that you have removed the file, run the `ls -l /var/spool/cron/crontabs` command. The `crontab` file for that user should no longer be listed.

# Denying `crontab` Command Access

1. Change directories to `/etc/cron.d`.
2. Using the `vi` text editor, add an entry to the `cron.deny` file for each user.
3. Verify that the users are listed in the file.

```
# cd /etc/cron.d
/etc/cron.d# vi cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jjones
/etc/cron.d# grep jjones cron.deny
jjones
```

To deny one or more users access to the `crontab` command, you add the user's or users' name to the `cron.deny` file. The steps to complete this task are shown in the slide.

**Note for step 2:** Be sure to enter only one user per line.

In the example, you add the user `jjones` to `cron.deny` file. You then verify that `jjones` is included in the file.

# Limiting `crontab` Access to Specified Users

1. Change directories `to/etc/cron.d`.
2. Using the `vi` text editor, create the `cron.allow` file and add an entry for each additional user.
3. Verify that `root` and the other users are listed in the file by using `cat cron.allow`.

```
# cd /etc/cron.d
/etc/cron.d# vi cron.allow
omai
jsmith
tbone
/etc/cron.d# cat cron.allow
omai
jsmith
tbone
```

To limit `crontab` command access to specific users, you create a `cron.allow` file and add the list of users to the file. The steps to complete this task are shown in the slide.

**Note for step 2:** Be sure to add only one username per line.

Remember from an earlier discussion about `crontab` file access that now that a `cron.allow` file exists, only the users listed in this file can create, edit, display, or remove the `crontab` files.

**Note:** If by chance a user's name is in both the `cron.deny` and `cron.allow` files, the user will be able to access the `crontab` command.

# Practice 10-2: Scheduling System Tasks

This practice covers the following topics:

- Scheduling a repetitive task with the `cron` utility
- Scheduling a user task as a superuser

This practice should take about 30 minutes to complete.

# Summary

In this lesson, you should have learned to:

- Implement a plan for system processes management
- Manage system processes
- Schedule system tasks

In this lesson, you learned how to manage system processes in accordance with a plan, as well as how to schedule repetitive system tasks by using the `crontab` file.

11

# Performing Basic System Monitoring and Troubleshooting

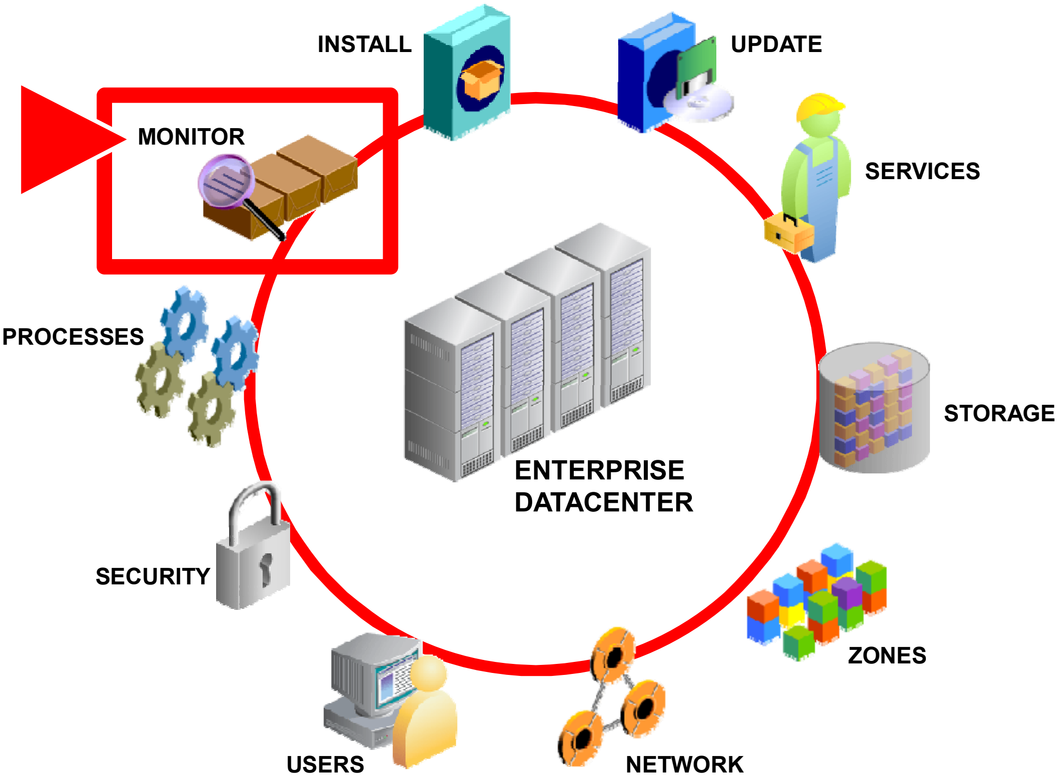
# Objectives

After completing this lesson, you should be able to:

- Monitor system logs
- Identify a crash dump file
- Identify a core dump file
- Troubleshoot a script execution issue
- Troubleshoot a software update failure
- Troubleshoot a network connectivity issue
- Troubleshoot a directory access issue
- Troubleshoot a default shell issue

In this lesson, you are introduced to system logs and shown how to monitor them. You are also introduced to crash and core dump files. This course concludes by you learning how to perform basic troubleshooting tasks in the following areas: search paths, software updates, network connectivity, and directory access.

# Workflow Orientation



Before you start the lesson, orient yourself to where you are in the job workflow. You have reached the end of the workflow. You have successfully performed all major administrative tasks: installation and software updates, as well as services, data storage management, zones, networking, and user management. You have also put system and file access controls in place and are managing the system processes. In this last lesson, you draw upon all that you have learned so far. In the practice exercises, you encounter basic issues that may arise when you update software or manage the network, users, or processes, and you are asked to resolve those issues.

# Lesson Agenda

- Monitoring System Logs
- Introducing Core Files
- Introducing Crash Dump Files
- Introducing Core Dump Files

# Monitoring System Logs

System messages are stored in the /var/adm directory.

```
client1:/var/adm# ls
acct   exacct   log       pool   sm.bin      streams   utmpx
aculog lastlog   messages   sa    spellhist   sulog     wtmpx
```

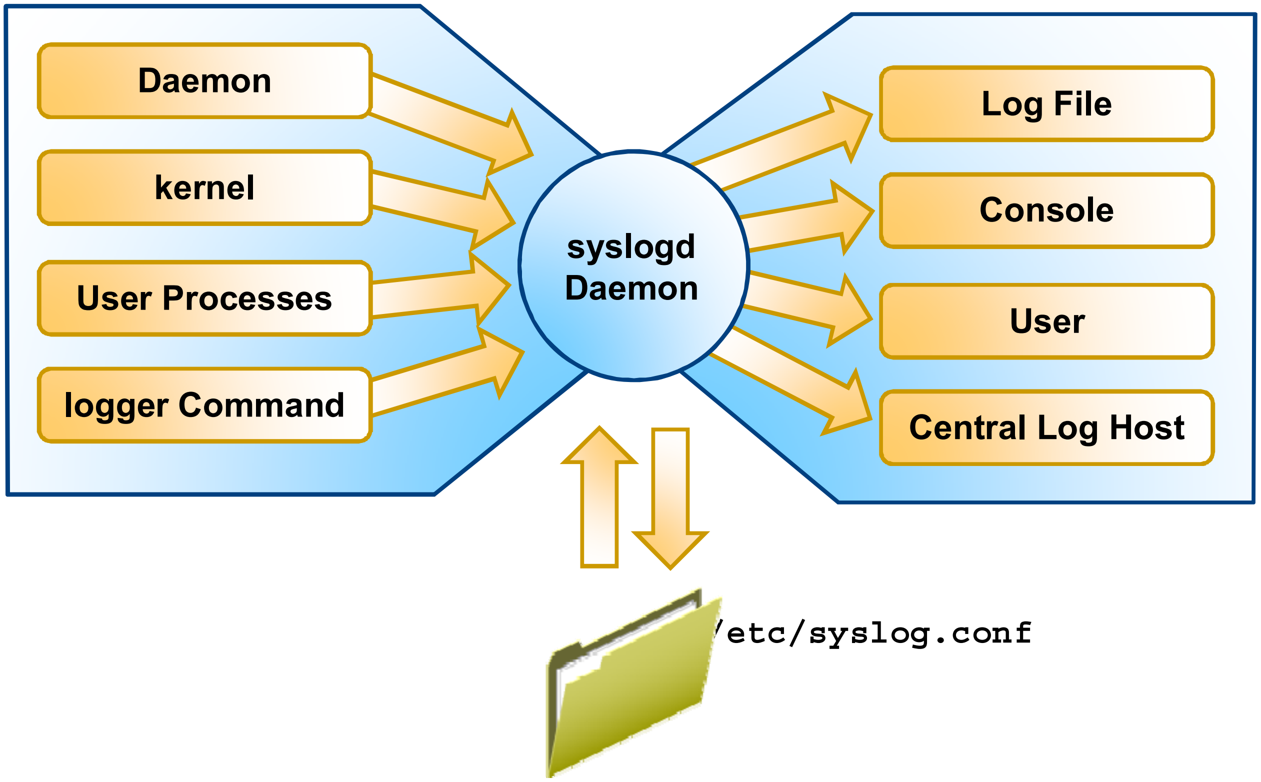
- The /var/adm directory contains several message files:
  - Most recent messages: /var/adm/messages
  - Oldest messages: messages.3
- Message files are rotated about every 10 days:
  - messages.0 is renamed to messages.1.
  - messages.1 is renamed to messages.2.
  - messages.2 is renamed to messages.3.

When the operating system detects a problem with the system, it records the issue in the form of a system message, which, by default, is displayed on a system console or stored in the /var/adm directory. The /var/adm directory contains several message files. The most recent messages are in the /var/adm/messages file (and in messages.*), and the oldest messages are in the messages.3 file. After a period of time (usually every 10 days), a new messages file is created. The messages.0 file is renamed messages.1; messages.1 is renamed messages.2; and messages.2 is renamed messages.3. The current /var/adm/messages.3 file is deleted.

**Note:** System log files are rotated by the logadm command from an entry in the default root crontab file. You may recall seeing this entry when you were learning about the root cron file in the lesson titled “Controlling Access to Systems and Files.”

In this lesson, the focus is on the file that contains the most recent messages. Before you take a look at the actual system messages, spend some time in looking at how the system messages are generated.

# syslogd Daemon



Oracle Solaris 11 system messaging and system logs are controlled by the syslog system messaging facility, which consists of the `syslog` function, the `syslogd` daemon, and input from the `/etc/syslog.conf` file. The `syslog` function sends messages generated by the kernel and system utilities and applications to the `syslogd` daemon, as shown in the graphic on the left in the slide.

**Note:** The `logger` command enables system administrators to send messages to the `syslogd` daemon. A system administrator can write administrative shell scripts that report the status of backups or other functions by using the `logger` command.

As illustrated by the remainder of the graphic, the `syslogd` daemon is then responsible for:

- Writing messages to the system log file
- Writing messages to the system console
- Forwarding messages to a list of users
- Forwarding messages to a centralized log host
- Sending and receiving information from the `/etc/syslog.conf` file

What system process sends the messages to the `syslogd` daemon and where the `syslogd` daemon sends the messages are determined by the configuration of the `/etc/syslog.conf` file. Now you can take a closer look at this file.

# /etc/syslog.conf File

An /etc/syslog.conf file configuration entry consists of:

- The selector field, which contains two components:
  - *Facility* – Category of system process that can generate messages
  - *facility.level* – Severity or importance of a message
- The action field, which determines where the message is sent

```
*.err                    /var/adm/messages
```

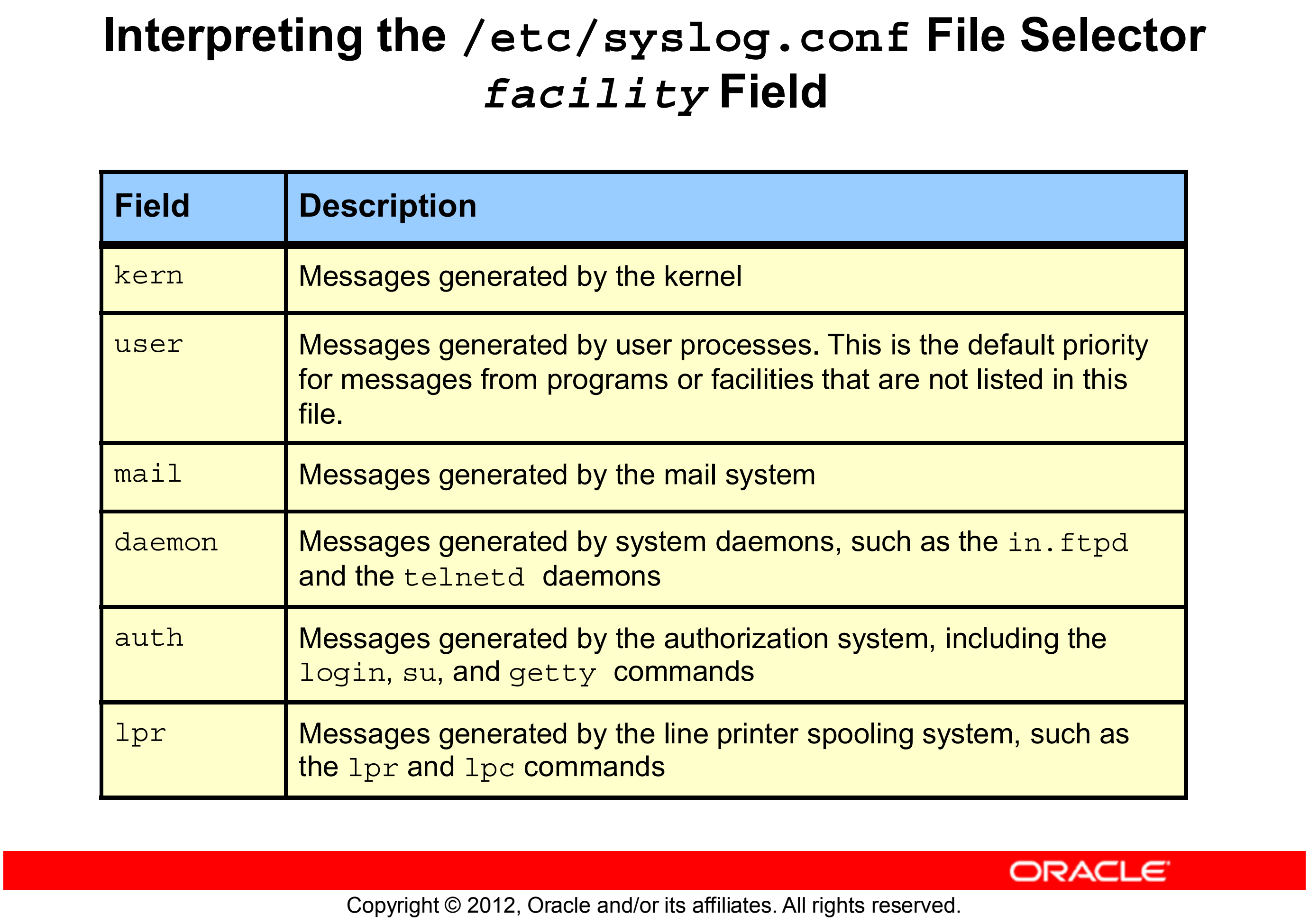
A configuration entry in the /etc/syslog.conf file consists of two, tab-separated fields: selector and action.

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.

In the example /etc/syslog.conf file entry in the slide, error messages for all facilities are sent to the /var/adm/messages file. The asterisk (*) is the facility and indicates in this case that all facilities can generate messages; the err field is the level or severity of the message; and /var/adm/messages is where the message is sent (that is, the action).

You will now take a closer look at the selector and action fields, beginning with the selector field.



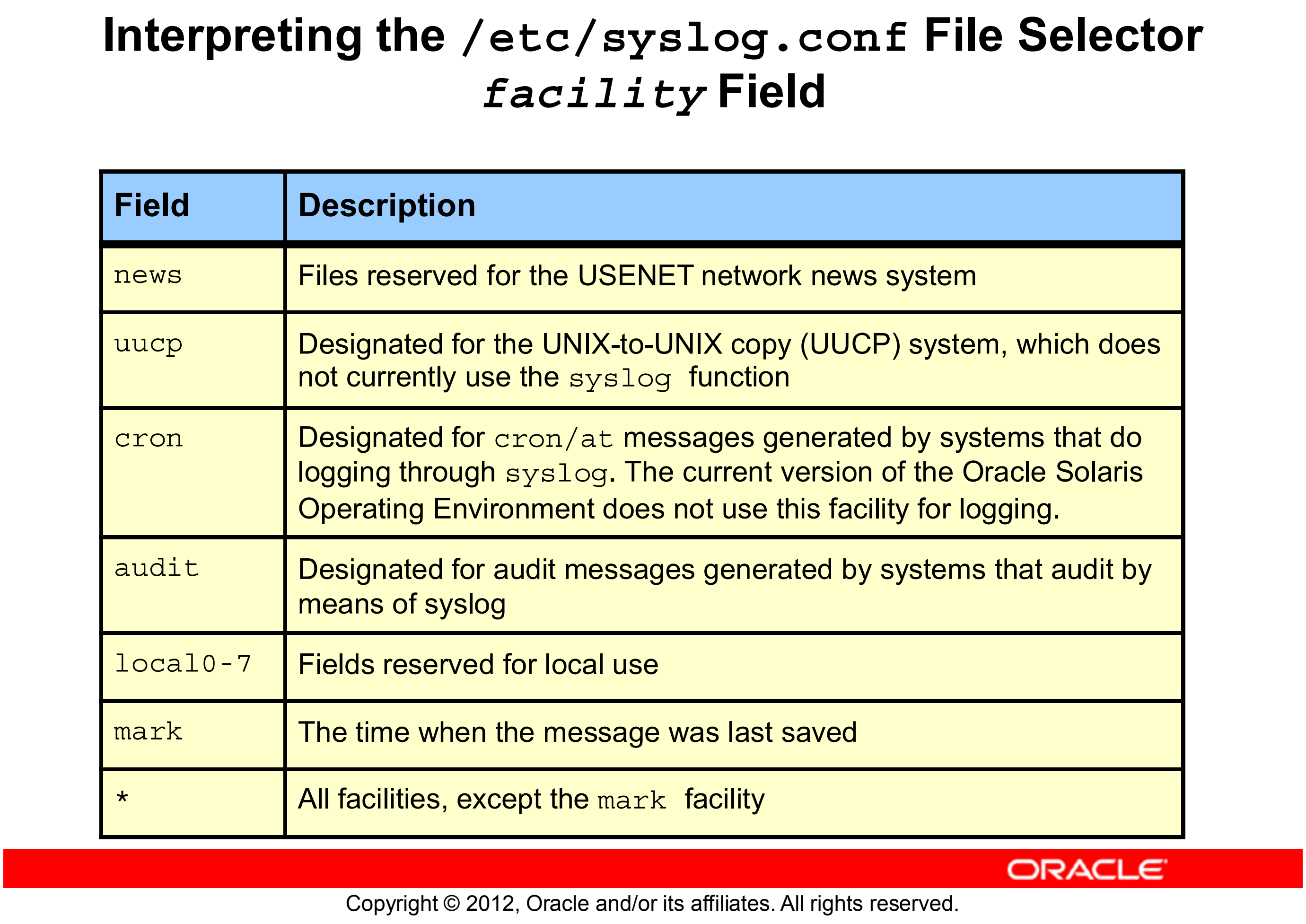
# Interpreting the `/etc/syslog.conf` File Selector *`facility`* Field

| Field | Description |
|---|---|
| `kern` | Messages generated by the kernel |
| `user` | Messages generated by user processes. This is the default priority for messages from programs or facilities that are not listed in this file. |
| `mail` | Messages generated by the mail system |
| `daemon` | Messages generated by system daemons, such as the `in.ftpd` and the `telnetd` daemons |
| `auth` | Messages generated by the authorization system, including the `login`, `su`, and `getty` commands |
| `lpr` | Messages generated by the line printer spooling system, such as the `lpr` and `lpc` commands |

The selector field is a semicolon-separated list of priority specifications in the following format:

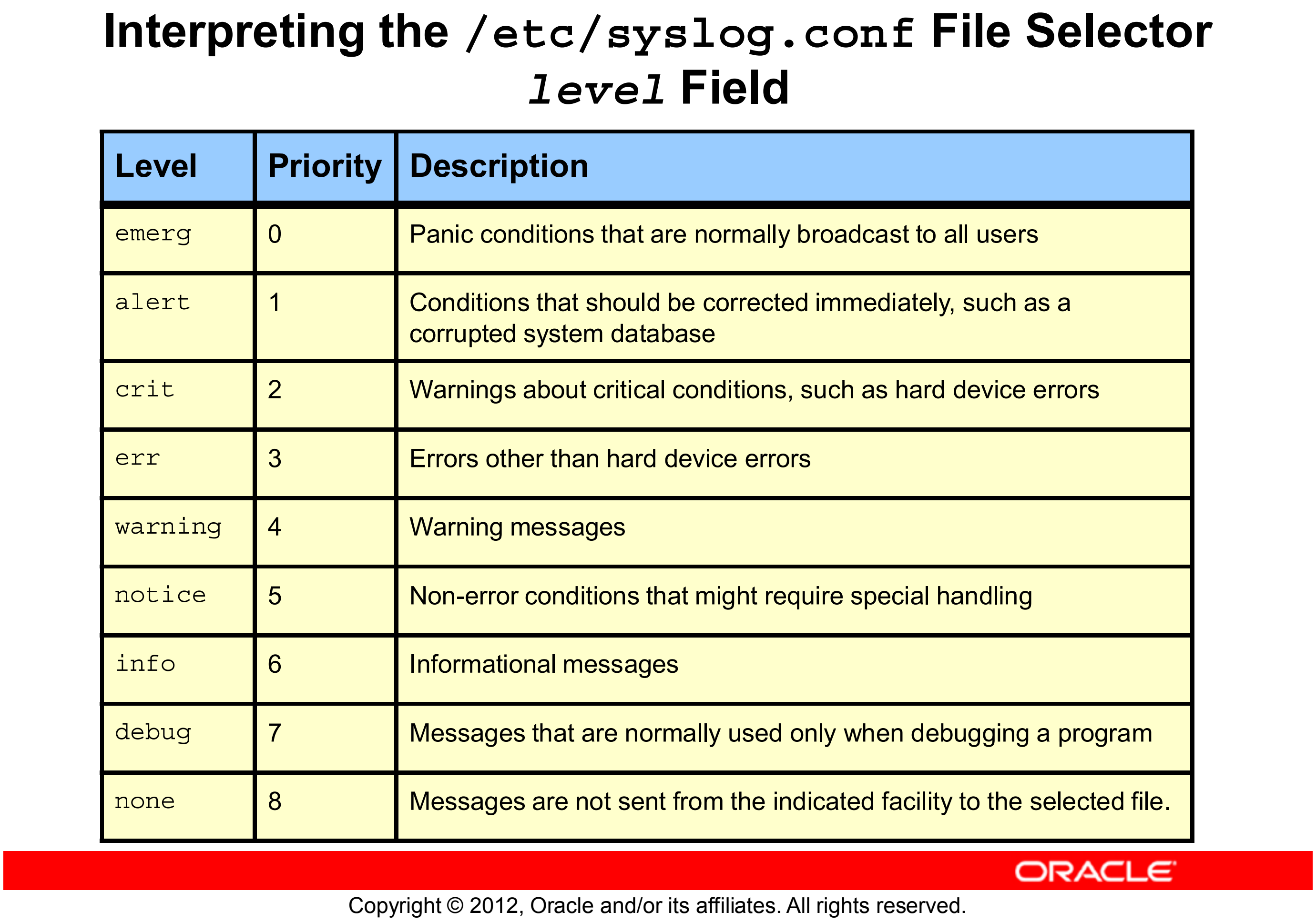
```
facility.level;facility.level
```

The tables shown in this slide and the following slide display the values that the selector *`facility`* field can contain.



# Interpreting the `/etc/syslog.conf` File Selector *`facility`* Field

| Field | Description |
|---|---|
| `news` | Files reserved for the USENET network news system |
| `uucp` | Designated for the UNIX-to-UNIX copy (UUCP) system, which does not currently use the `syslog` function |
| `cron` | Designated for `cron/at` messages generated by systems that do logging through `syslog`. The current version of the Oracle Solaris Operating Environment does not use this facility for logging. |
| `audit` | Designated for audit messages generated by systems that audit by means of syslog |
| `local0-7` | Fields reserved for local use |
| `mark` | The time when the message was last saved |
| `*` | All facilities, except the `mark` facility |



# Interpreting the `/etc/syslog.conf` File Selector *`level`* Field

| Level | Priority | Description |
|---|---|---|
| `emerg` | 0 | Panic conditions that are normally broadcast to all users |
| `alert` | 1 | Conditions that should be corrected immediately, such as a corrupted system database |
| `crit` | 2 | Warnings about critical conditions, such as hard device errors |
| `err` | 3 | Errors other than hard device errors |
| `warning` | 4 | Warning messages |
| `notice` | 5 | Non-error conditions that might require special handling |
| `info` | 6 | Informational messages |
| `debug` | 7 | Messages that are normally used only when debugging a program |
| `none` | 8 | Messages are not sent from the indicated facility to the selected file. |

The table shown in the slide displays the levels of severity for a message in descending order of severity. Each level includes all the levels above it (that is, those of a higher severity).

**Note:** Not all levels of severity are implemented for all facilities in the same way. For more information, refer to the online man pages.

# Interpreting the `/etc/syslog.conf` File Action Field

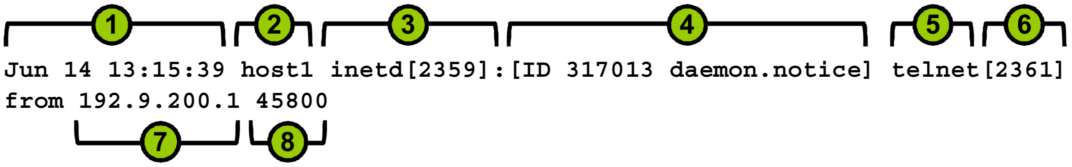
| Field | Description |
|---|---|
| `/pathname` | This indicates the full path name to the targeted file. |
| `@host` | The `@` sign denotes that messages must be forwarded to a remote host. Messages are forwarded to the `syslogd` daemon on the remote host. |
| `user1,`<br>`user2` | The `user1` and `user2` entries receive messages if they are logged in. |
| * | All logged in users receive messages. |

The table in the slide displays the action field entry options.

**Note:** The `/etc/syslog.conf` file can be customized to capture additional error messages that are generated by various system processes. How to configure system messaging is covered in the Oracle *Solaris 11 Advanced System Administration* course. Also, see the Oracle Solaris system administration documentation.

# Monitoring a `syslog` File in Real Time

To view messages sent to the `/var/adm/messages` file, use `tail -f /var/adm/messages`.



```
Jun 14 13:15:39 host1 inetd[2359]:[ID 317013 daemon.notice] telnet[2361]
from 192.9.200.1 45800
```

| Number | Field | Result |
|---|---|---|
| 1 | Date/time | Jun 14 13:15:39 |
| 2 | Local host name | host1 |
| 3 | Process name/PID number | inetd[2359] |
| 4 | MsgID number/selector facility.level | [ID 317013 daemon.notice] |
| 5 | Incoming request | telnet |
| 6 | PPID number | [2361] |
| 7 | IP address | 192.9.200.1 |
| 8 | Port number | 45800 |

By default, the `/etc/syslog.conf` file directs many system process messages to the `/var/adm/messages` files.

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 being written to the file by the `syslogd` daemon.

The graphic in the slide shows a sample log entry that has been generated by a `telnet` request to the `host1` system from the IP address `192.9.200.1` on port `45800`.

The table below the log entry lists each field in this figure and its corresponding result.

To exit the `/var/adm/messages` file, press Ctrl + C.

# Interpreting System Messages

```
# tail -f /var/adm/messages
Aug 10 05:35:53 server1 named[472]: [ID 873579 daemon.notice] running
Aug 10 05:36:24 server1 mac: [ID 736570 kern.info] NOTICE: net1
unregistered
#
```

```
# tail -f /var/adm/messages
Aug 10 05:40:03 client1 genunix: [ID 936769 kern.info] fssnap0 is
/pseudo/fssnap@0
Aug 10 05:40:07 client1 gnome-session[784]: [ID 702911 daemon.warning]
WARNING: IceListenForConnections returned 2 non-local listeners:
inet/client1:39166,inet6/client1:38708
#
```

Shown in the slide is a selection of system messages taken from the `/var/adm/messages` file. The first example at the top is from a system called `server1` and the second example at the bottom is from a system called `client1`. Take a few minutes to look at each message and see whether you can identify the following:

- The process name/PID number
- The message ID number
- The facility that generated the message (for example, the kernel, a system daemon, or the `syslogd` daemon)
- Level of severity for the message (for example, emergency, error, warning, notice, or information)
- The problem

An interpretation of each message is as follows:

- The first message in the example at the top tells you that the named daemon (PID `472`) is running. The message ID number is `873579`. The system daemon is the facility that generated the message and the level of severity for the message is "notice only." No action is required.
- The second message in the example at the top is an informational message from the kernel relating to the MAC address that informs you that the network interface `net1` is unregistered. Unregistered means that the network interface has been configured on a temporary basis. The message ID number is `736570`. `NOTICE` in this message is a part of the `info` level of severity.
- The first message in the example at the bottom (message ID number `936769`) is another informational message from the kernel that tells you that the general part of the UNIX kernel (`genunix`) created a snapshot called `fssnap0` of the file system and that the snapshot is located in `/pseudo/fssnapshot@0`.
- The second message in this example, which is a warning message, has been generated by a system daemon to let you know that when the Gnome session was being established (PID `784`), the daemon detected two "listeners" on the connection: `inet/client1:39166,inet6/client1:38708`.

# Lesson Agenda

- Monitoring System Logs
- Introducing Core Files
- Introducing Crash Dump Files
- Introducing Core Dump Files

ORACLE

# What Is a Core File?

- A file generated in response to fatal errors.
  - For the kernel, the file is referred to as a “crash dump.”
  - For an application or process, the file is referred to as a “core dump.”
- A fatal kernel error:
  - Results in a system crash (“panic”)
  - Generates a core file that contains a snapshot of the kernel’s memory space when the error occurred
- A fatal application or process error:
  - Results in abnormal termination
  - Generates a core file that contains a snapshot of the process’s memory space when the error occurred

When software encounters a fatal error, a core file is generated. The core file is a file that represents the memory image of the software that encountered the fatal error. For fatal errors associated with the kernel, the resulting core file is referred to as a “crash dump.” For fatal errors associated with an application or process, this file is often referred to as a “core dump.”

A fatal error in the kernel results in a system crash (typically a “panic”). The kernel core image is a snapshot of the kernel’s memory space when the error occurred. This file contains information about the kernel core, name list, and symbol table information.

For an application or process, a fatal error results in abnormal termination. The core file that is generated when the application or process encounters a fatal error is a snapshot of what the memory space of the application or process looked like when the error occurred. The file contains information about the task name, task owner, priority, and instruction queue in execution at the time that the core file was created.

# Core File Generation: Advantages and Disadvantages

- Advantages
  - Core files are used to analyze and determine the root cause of a crash or core dump so that the problem can be resolved.
- Disadvantages
  - Core files take up a large amount of disk space.
  - Core files must be managed.

Core files are generated so that a crash dump analysis (the kernel) or core dump analysis (user processes) can be performed, the root cause of the error can be determined, and corrective action can be taken to resolve the problem. Often, the only possible way to debug a problem that has caused a fatal software crash is through core file analysis. Usually, the analysis is performed by a support engineer.

The only drawback to generating core files is that these files can take up a large amount of disk space. Consequently, they must be managed. This is usually the system administrator's job.

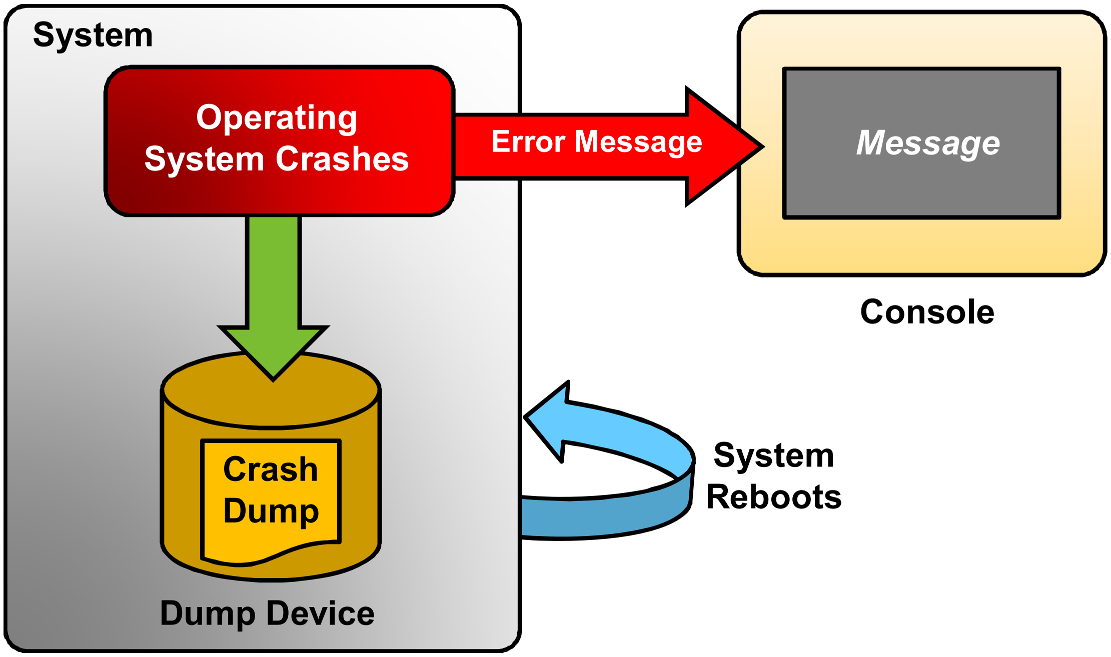
**Note:** How to manage core files is covered in the *Oracle Solaris 11 Advanced System Administration* course.

**Note:** In addition to storing large files that contain messages, the `/var/adm` directory also stores crash dumps and other data. This means that this directory can consume large amounts of disk space. To keep the `/var/adm` directory from growing too large, and to ensure that future crash dumps can be saved, the system administrator is responsible for removing unnecessary files periodically.

# Lesson Agenda

- Monitoring System Logs
- Introducing Core Files
- Introducing Crash Dump Files
- Introducing Core Dump Files

# Crash Dump Process: Overview



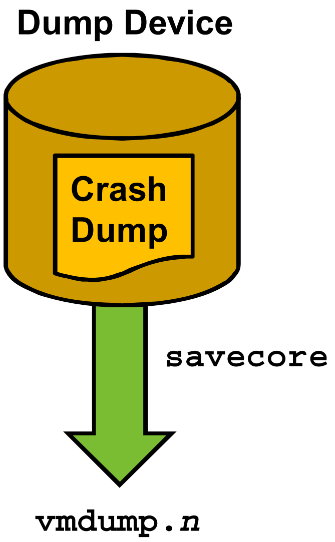
If the Oracle Solaris OS kernel encounters a problem that might endanger the integrity of data or when the kernel encounters an unexpected hardware fault, the panic routine is executed. Despite its name, a system panic is a well-controlled event where the memory contents are copied to a disk partition that is defined as a dump device. Whatever the cause, the crash dump itself provides valuable information to help your support engineer to diagnose the problem.

When a fatal operating system error occurs, the operating system prints a message to the console, describing the error. The operating system then generates a crash dump file by writing some of the contents of the physical memory to a predetermined dump device, which must be a local disk slice.

**Note:** A dump device is usually disk space that is reserved to store system crash dump information. By default, a system's dump device is configured to be a swap slice. If possible, an alternate disk partition should be configured as a dedicated dump device to provide increased reliability for crash dumps and faster reboot time after a system failure.

After the operating system has written the crash dump file to the dump device, the system reboots. The crash dump file is saved for future analysis to help determine the cause of the fatal error.

# How and Where Crash Dump Files Are Saved



Crash dump files are saved in a predetermined directory, which, by default, is `/var/crash/`*`hostname`*. In previous releases, crash dump files were overwritten when a system rebooted, unless you manually enabled the system to save the images of the physical memory in a crash dump file. Now, the saving of crash dump files is enabled by default.

When an operating system crashes, the `savecore` command is automatically executed during a boot. The `savecore` command retrieves the crash dump from the dump device, and then writes the crash dump information in a compressed format to the `vmdump.`*`n`* file.

**Note:** *n* is an integer that identifies the crash dump.

Later, the `savecore` command can be invoked on the same system or another system to expand the compressed crash dump to a pair of files that together form the saved crash dump. These two files are:

- **`/var/crash/`*`nodename`*`/vmcore.`*`X`*:** Contains the kernel core information

  **Note:** *nodename* is the name returned by `uname -n`, and `X` is an integer that identifies the dump.
- **`/var/crash/nodename/unix.`*`X`*:** Contains the name list and symbol table information

# Crash Dump: Example

```
# savecore -L
dumping to /dev/dsk/c7t5d0s3, offset 65536, content: kernel
 0:04 100% done
100% done: 103879 pages dumped, dump succeeded
savecore: System dump time: Thu Aug 18 10:23:31 2011

savecore: Saving compressed system crash dump in /var/crash/s11-
desktop/vmdump.0
savecore: Decompress the crash dump with
'savecore -vf /var/crash/s11-desktop/vmdump.0'
```

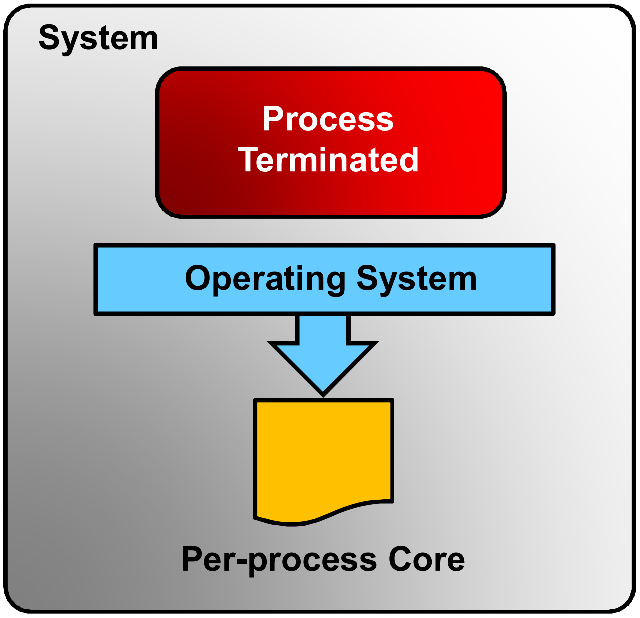
In this example, the `savecore` utility retrieves the crash dump from the dump device, and then writes the crash dump information in a compressed format to the `vmdump.0` file.

# Lesson Agenda

- Monitoring System Logs
- Introducing Core Files
- Introducing Crash Dump Files
- Introducing Core Dump Files

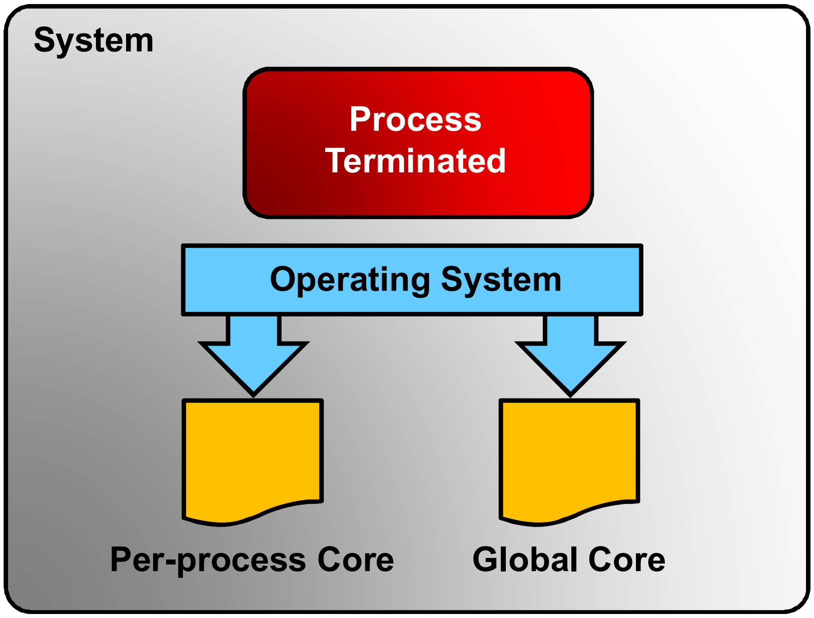
You now look at core dump files, what they are, and when they occur.

# Core Dump Process: Overview



When a process receives a specific signal and terminates, the system generates a minimum of one core dump, which is known as the per-process core file, and stops the process.

## How and Where Core Dump Files Are Saved



Depending on the system options in effect, when a process terminates abnormally, one or two core dump files can be generated.

**Note:** If the application or process is running in a local zone, a third core file is created in the global zone's location.

By default, a core file is generated in the current working directory. This file is known as the per-process core file. 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 second file known as the global core file is generated if the global core file path is enabled. The global core file path defaults to `core` and is disabled by default. If 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.

**Note:** The system administrator can modify the core dump configuration to control which core dump paths are enabled. This task is covered in the *Oracle Solaris 11 Advanced System Administration* course.

# Core Dump: Example

```
core:  ELF 32-bit LSB core file 80386 Version 1, from 'bash'
```

In this example, the system has created a core file for the bash process in the “current directory,” which is the current directory at the time of dump creation.

# Quiz

In which directory are the system messages stored?

a. `/var/lib`

b. `/var/tmp`

c. `/var/adm`

d. `/var/log`

**Answer: c**

# Quiz

What is the facility in the following syslog entry?

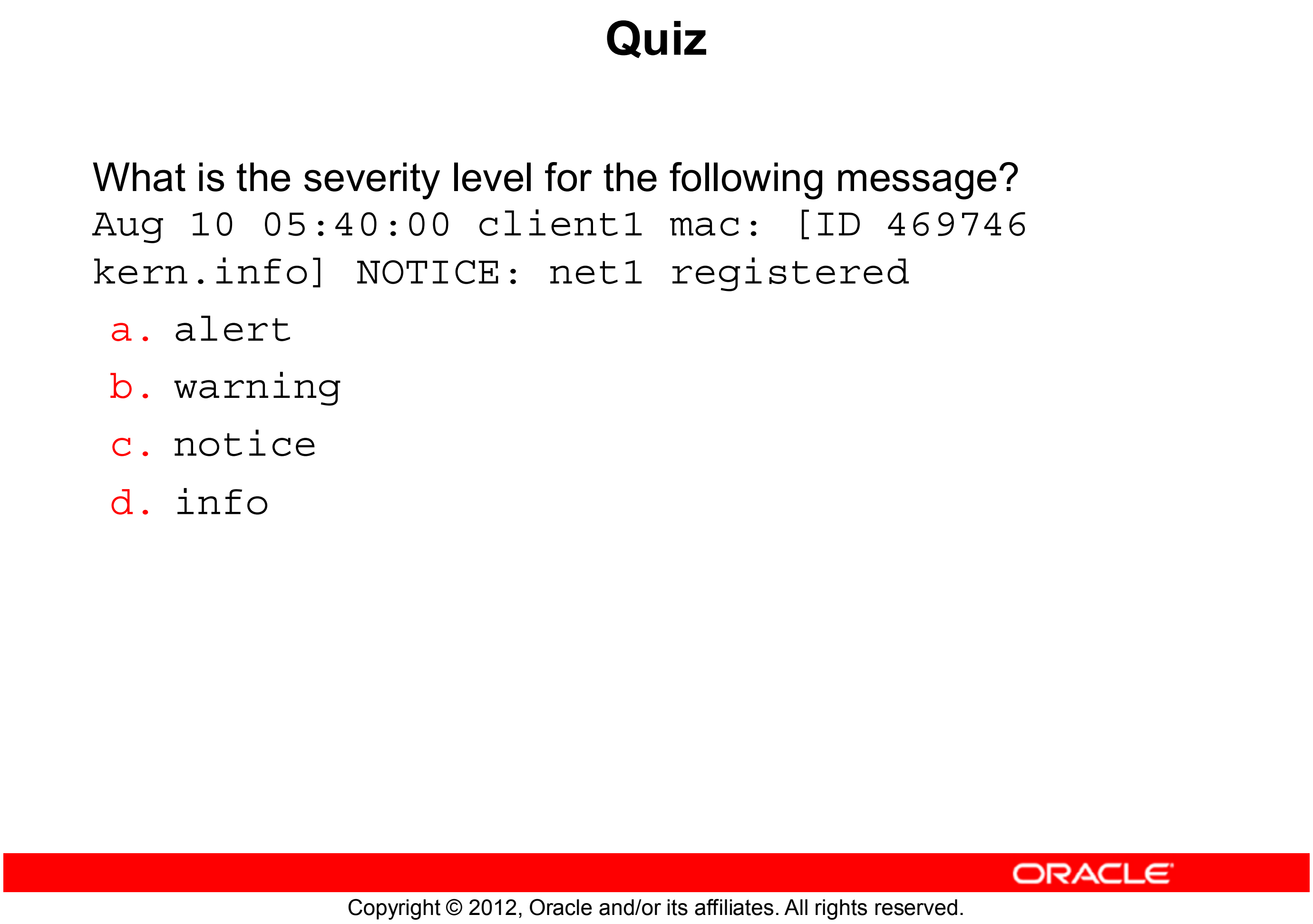
```
Aug 10 05:40:03 client1 genunix: [ID 936769
kern.info] fssnap0 is /pseudo/fssnap@0
```

a. genunix

b. kern

c. info

**Answer: b**



# Quiz

What is the severity level for the following message?

```
Aug 10 05:40:00 client1 mac: [ID 469746
kern.info] NOTICE: net1 registered
```

a. alert
b. warning
c. notice
d. info

**Answer: d**

# Quiz

A core dump is generated when the system experiences a fatal error.

a. True

b. False

**Answer: b**

# Quiz

What is the result of a fatal application or process error?

a. System panic

b. Abnormal termination of the application or process

c. A core dump being created on the designated dump device

ORACLE

**Answer: b, c**

# Quiz

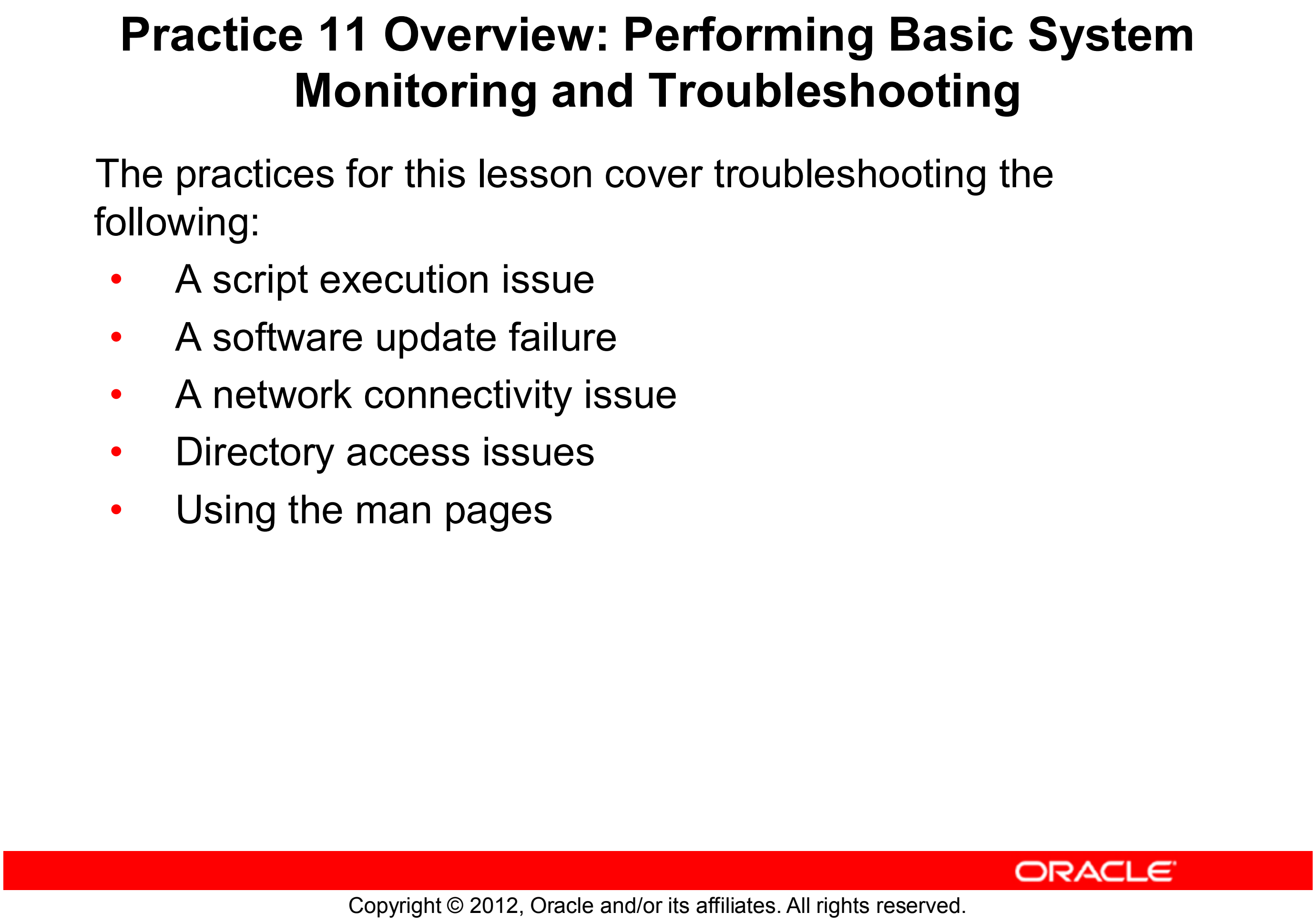
The global core file path is enabled by default.

a. True

b. False

ORACLE

**Answer: b**



## Practice 11 Overview: Performing Basic System Monitoring and Troubleshooting

The practices for this lesson cover troubleshooting the following:

- A script execution issue
- A software update failure
- A network connectivity issue
- Directory access issues
- Using the man pages

In the practices for this lesson, you are presented with five tasks (listed in the slide) that are designed to reinforce the concepts presented in the lecture portion of this lesson.

All the practices are in your *Activity Guide*. It should take about 1.5 hours to complete all five of them.

# Summary

In this lesson, you should have learned to:

- Monitor system logs
- Identify a crash dump file
- Identify a core dump file
- Troubleshoot a script execution issue
- Troubleshoot a software update failure
- Troubleshoot a network connectivity issue
- Troubleshoot a directory access issue
- Troubleshoot a default shell issue

In this lesson, you were introduced to system logs and shown how to monitor system messages. You were also introduced to the crash and core dump files. In addition, you learned how to troubleshoot a number of common system issues that had to do with software updates, network connectivity, and directory access.