ORACLE UNIVERSITY




# Oracle Solaris 11 System Administration

Student Guide – Volume II
D72896GC40
Edition 4.0 | September 2014 | D88133

Learn more from Oracle University at **oracle.com/education/**

ORACLE

**Author**

Vijetha M Malkai

**Technical Contributors and Reviewers**

Muhammad Aseel Khan
Rajesh Rajasekharan
Gary Riseborough
David Maxwell

**Editors**

Vijayalakshmi Narasimhan
Smita Kommini

**Graphic Designers**

Maheshwari Krishnamurthy
James Hans

**Publishers**

Nita Brozowski
Syed Imtiaz Ali

# Contents

**7 Administering the Network**

7

# Administering the Network

# Objectives

After completing this lesson, you should be able to:

- Explain some of the basic networking concepts
- Administer a datalink configuration
- Administer a network interface
- Administer a profile-based network configuration
- Configure a virtual network
- Verify the network operations
- Manage resources on the network

# Workflow Orientation




Before you start the lesson, orient yourself to where you are in the job workflow. You have successfully installed the operating system, tested the system boot and shutdown, worked with SMF services, and set up and administered the data storage environment.

Now, you will be introduced to 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 the hosts to ensure that communications continue uninterrupted.

In this lesson, you learn how to configure both physical as well as a virtual network.

# Lesson Agenda

- **Reviewing Networking Fundamentals**
- Administering Datalink Configuration
- Administering the Network Interface
- Administering Profile-Based Network Configuration
- Configuring a Virtual Network
- Verifying Network Operation
- Managing Resources on the Virtual Network

# Importance of Network Administration

It is important to administer the network in the Oracle Solaris 11 OS to address the following requirements:

- IP addressing scheme
- Network interfaces
- Datalinks
- Network configuration profiles
- Virtual networks
- Network resources



The Oracle Solaris 11 network architecture is significantly different from the previous releases of Oracle Solaris. Not only has its implementation changed, but also the names of the network interfaces, commands, and methods of administering and configuring them. These changes were introduced to bring in a more consistent and integrated experience to network administration, particularly as administrators add more complex configurations, including link aggregation, bridging, load balancing, or virtual networks. In addition to the traditional fixed networking configuration, Oracle Solaris 11 introduced automatic network configuration through network profiles.

Becoming familiar with the TCP/IP protocol architecture model will help you to administer your network in a more orderly manner. In this lesson, you learn about this model, as well as how the IP addressing schemes, network interfaces, and datalinks support this model.

Some of the advanced concepts related to elastic virtual switch (EVS) and network high-availability such as load balancing, IP multipathing (IPMP), and VRRP are taught in the *Oracle Solaris 11 Advanced System Administration* course, which is a follow-on to this one. For detailed information about Oracle Solaris 11 network administration, you can take up the specialty training on *Oracle Solaris 11 Network Administration*.

# TCP/IP Protocol Architecture Model

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



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. It 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 bottom-most 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.

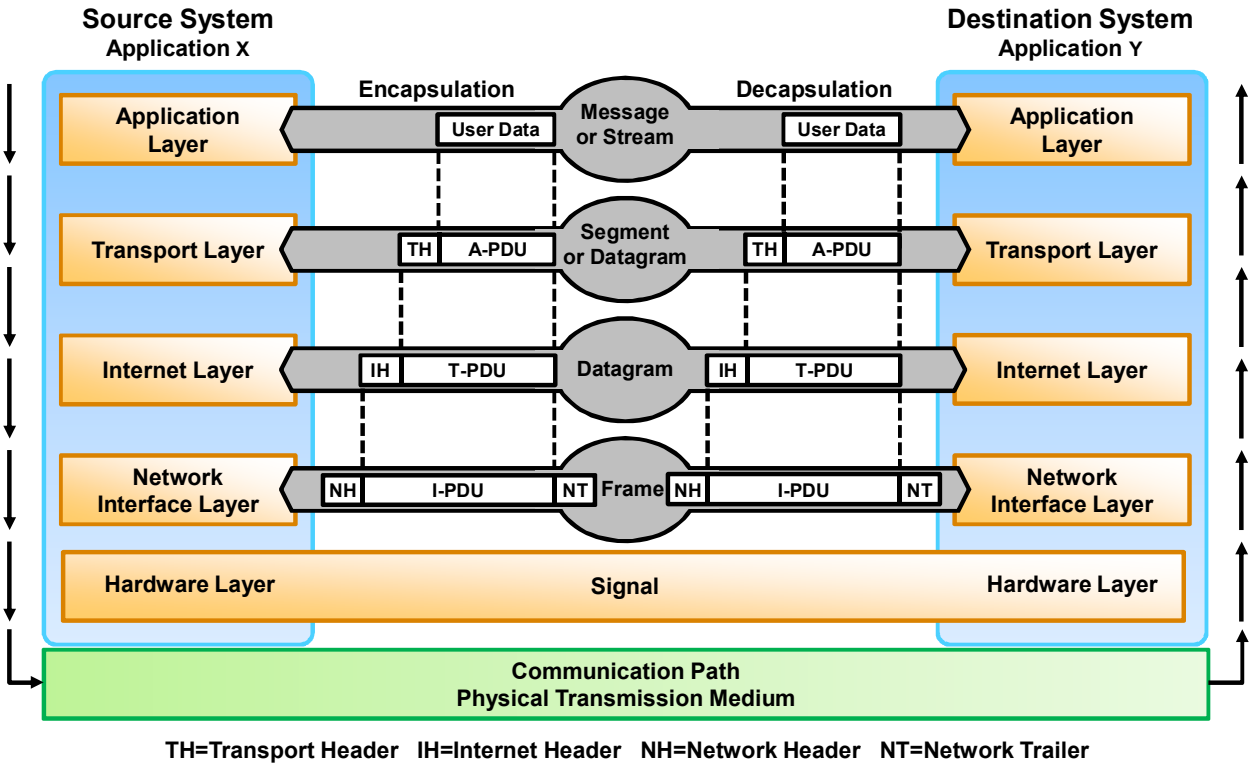
You now look at each of the TCP/IP layers, beginning with the physical layer and working your way up to the application layer.

- **Network layer:** Specifies the protocols and hardware required to send data through the network. Details include how bits are represented by the hardware devices that interface directly with a physical network medium, such as coaxial cable, optical fiber, or twisted-pair copper wire.
  The 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. The network layer also identifies the network protocol type of the packet, in this instance TCP/IP, for example, 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. It 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 whether 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.

# How TCP/IP Handles Data Communications



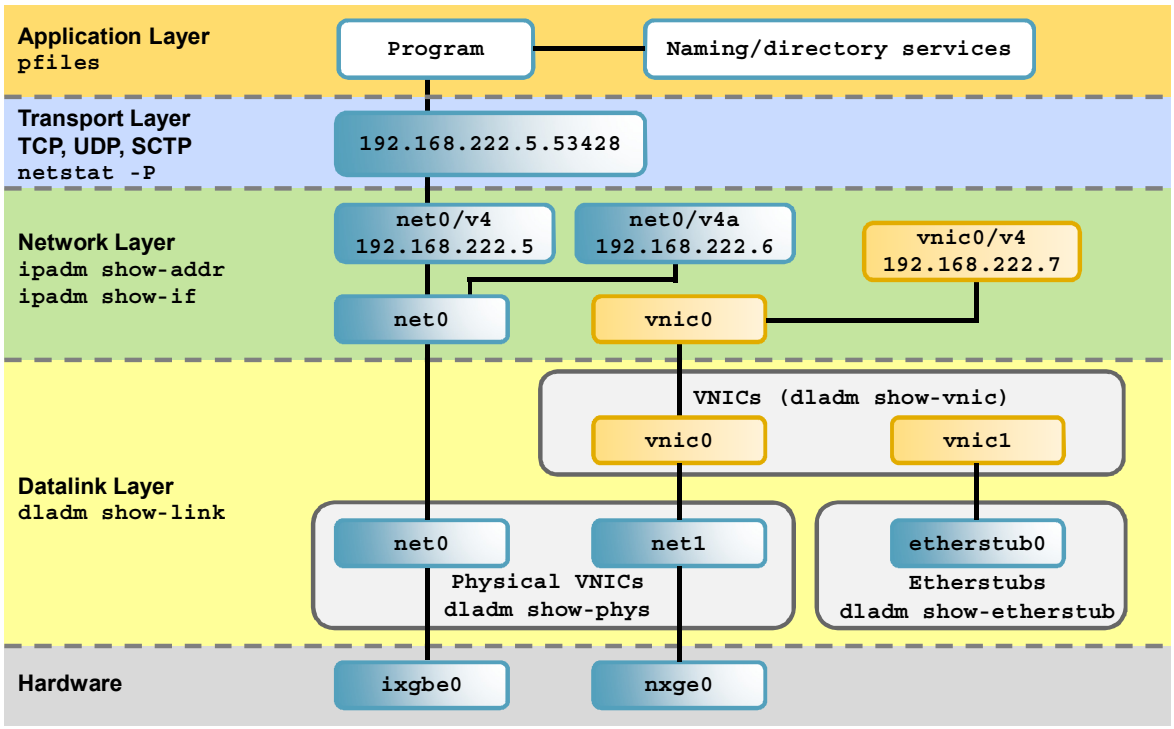



When a user issues a command that uses a TCP/IP Application layer protocol, a series of events is initiated. The user's command or message passes through the TCP/IP protocol stack on the local system. The command or message then passes across the network media to the protocols on the remote system. The protocols at each layer on the sending host add information to the original data. The graphic in the slide illustrates this process. This is a peer-to-peer communication, where one layer on a system communicates with a corresponding layer on another system. For example, the Application layer on the source system interacts with the Application layer on the destination system.

As data passes down through each layer in the stack, it is encapsulated. During encapsulation, header information is added, which helps the destination system to direct the data to the appropriate protocol. Trailer information is added at the final layer.

The data that arrives at a destination system is decapsulated. During decapsulation, data travels up through the layers. At each layer, headers and trailers are removed before the data is passed up to the next layer.

# Oracle Solaris 11 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 11 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 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`.

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, whether the IP stack is functioning properly.

For more information about key Oracle Solaris network administration features, refer to http://docs.oracle.com/cd/E36784_01/html/E37473/gnogg.html#scrolltoc and http://docs.oracle.com/cd/E36784_01/html/E37473/desstack-1.html#scrolltoc.

The naming and directory services such as NFS, DNS, and LDAP are covered in the follow-on course *Oracle Solaris 11 Advanced System Administration.*

# Configuring a Host for TCP/IP

Network configuration checklist:

- ☑ IP addresses
- ☑ Netmask
- ☑ Domain name
- ☑ Name service
- ☑ Default router



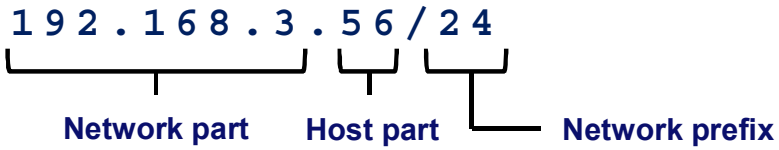
After a network is physically in place, the network configuration process involves configuring the network interfaces and associated IP addresses. The daemons and services that implement the TCP/IP protocol are made available to the system based on this configuration or acquired from the network configuration server, known as *network client mode*.

A typical TCP/IP network configuration requires the following information:

- IP address of each network interface on every system. The address scheme can be IP version 4 (IPv4) or IP version 6 (IPv6) and it may include subnet addressing.
- Netmask in use on each system's network and subnetmask, if applicable
- Domain name for your network, such as oracle.com
- Name service or directory service that your network uses, such as NIS, LDAP, or DNS
- Default router addresses

# 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
- The component parts of an IPv4 address include:
  - Network part
  - Host part
  - Network prefix






IPv4 addresses are part of 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.

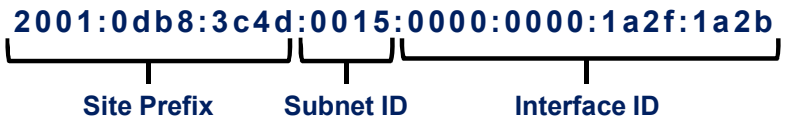
An IPv4 address is composed of the following component parts:

- 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.

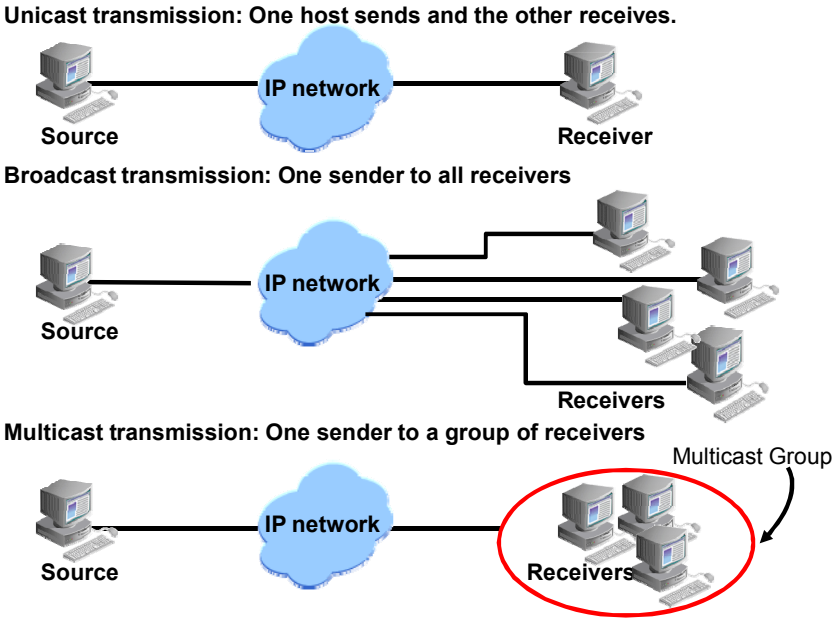
- Interface ID (64 bits), which is either automatically configured from the interface's MAC address or manually configured in EUI-64 format (also referred to as a token)

**Note:** Oracle Solaris supports IPv4 and IPv6 addressing on the same host by using 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

For each type of data transmission, there is an associated IP addressing type:

- Unicast
- Broadcast
- Multicast



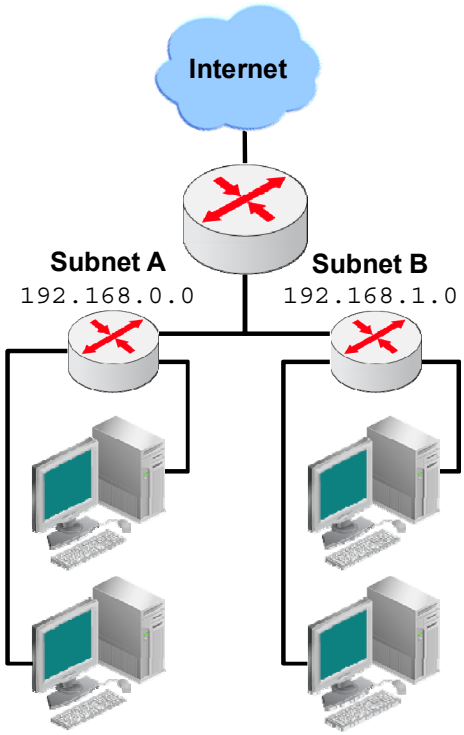



NICs are usually configured to listen for three types of messages: messages sent to a specific address, messages sent to a specific group of interfaces, and messages sent to all interfaces on a specific subnet. For each type of message transmission, there is an associated IP addressing type:

- **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 a 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`).

# Network Configuration Modes

The network configuration modes refer to the ability of the system to automatically adjust to changes in the current network environment and not to whether static or fixed IP addresses can be configured in these modes.

The following network configuration modes are supported in Oracle Solaris 11:

- Fixed
- Reactive



Fixed mode means that the instantiated configuration on the system is persistent, regardless of any changes in network conditions. When such changes occur, such as the addition of interfaces, you must reconfigure the network for the system to adapt to the new environment. When using fixed mode, your system is configured by following the same set of network configuration commands every time. Corporate servers most often use this configuration mode due to a relatively stable network environment. When using the fixed network configuration mode, you use the `dladm` and `ipadm` commands to manage the various aspects of network configuration.

Reactive mode, on the other hand, is when the network is configured automatically in response to current network conditions. This mode is primarily used for laptop computers and notebook PCs, and in situations where network conditions might change.

In reactive mode, a network daemon (`nwamd`) monitors the state of the system's network interfaces. The network daemon adjusts the network configuration dynamically when conditions change. For example, a notebook PC might be physically attached to the corporate network, or it might not be physically attached. When it is physically attached, you most likely would disable the notebook's wireless interface. Also, it is most often desirable to have the wireless interface automatically enabled when the Ethernet cable is detached from the notebook.

# Oracle Solaris 11
# Network Administration Commands

| Command | Description |
|---|---|
| `dladm` | Used to administer datalinks. It helps in managing physical interfaces (Ethernet, wireless, and InfiniBand), virtual networking features (Etherstubs, VNICs, and IP tunnels), switch features (link aggregations, VLANs, VXLANs, and bridging technologies), and device characteristics (speed, duplexing, priority, and feature negotiation). |
| `ipadm` | Used to administer IP interfaces and IP addresses |
| `netcfg` | Used to manage various types of profiles, for example, NCPs and location profiles |
| `netadm` | Used to enable and disable profiles and display information about profiles and their states |

# Administering the Network

As part of network administration, you will now learn how to:

- Administer datalink configuration
- Administer the network interface
- Administer profile-based network configuration
- Configure a virtual network
- Verify network operation
- Manage resources on the virtual network


ORACLE

# 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

- Reviewing Networking Fundamentals
- **Administering Datalink Configuration**
- Administering the Network Interface
- Administering Profile-Based Network Configuration
- Configuring a Virtual Network
- Verifying Network Operation
- Managing Resources on the Virtual Network

# Datalink Configuration in Oracle Solaris11

- Administrators create IP interfaces on top of datalinks.
- Each datalink represents a link object in the second layer of the Open Systems Interconnection (OSI) model.
- Datalinks can represent many different Layer 2 entities such as physical network devices (termed physical links), aggregations of physical datalinks, virtual network interface cards (VNICs), and so on.



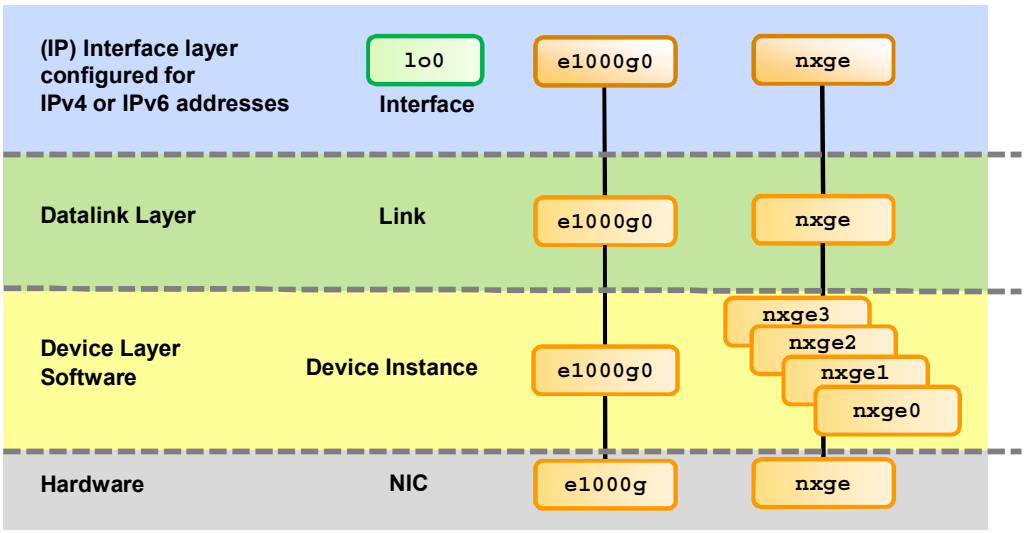
In this section, you mostly learn about physical links or links that represent network devices.

Link names are either automatically assigned when the associated link object is automatically created, or you can explicitly assign link names when you create the datalinks. Physical links (those that are associated with physical network devices) are created automatically when devices are added or when an Oracle Solaris system first boots after installation. By default, datalinks are assigned names that are prefixed by `net` and suffixed by a number that reflects the physical location of the datalink in the system. For example, the first onboard network device `e1000g0` would be assigned the name `net0`, the next `e1000g1` device would be assigned the name `net1`, and so on. You can assign arbitrary names to datalinks that you explicitly create, for example, link aggregations. Also, you can explicitly rename the default-assigned `net`*N* name of a datalink, if desired.

In this Oracle Solaris release, the naming of physical datalinks is no longer tied to the underlying hardware that is associated with the network device. By default, such devices are assigned the generic name "`net`" and a suffix that reflects the device's physical location in the system.

# Determining Datalink Availability

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






As 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 the Physical Links That 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 in 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 the Datalinks That Are Available

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

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



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`)
- 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 a 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`.
- 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.

For information about the `dladm` command and its subcommands, refer to http://docs.oracle.com/cd/E36784_01/html/E37475/gfrtp.html#scrolltoc and the `dladm` (1M) man page.

# Verifying That the Network Service Is Running

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

```
# svcs network/physical
online         3:33:46 svc:/network/physical:upgrade
online         3:33:53 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.

# Quiz

Which utility is used to create virtual switches and VNICs?

a. `lnkadm`
b. `dladm`
c. `vniccfg`
d. `dlcfg`



**Answer: b**

# Lesson Agenda

- Reviewing Networking Fundamentals
- Administering Datalink Configuration
- **Administering the Network Interface**
- Administering Profile-Based Network Configuration
- Configuring a Virtual Network
- Verifying Network Operation
- Managing Resources on the Virtual Network

# Administering a Network Interface

- Displaying network interface configuration information
- Displaying network interface IP address information
- Configuring a physical network interface manually
- Taking down a network interface
- Bringing up a network interface
- Deleting a physical network interface manually



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 a 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, and cover 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           192.168.0.100/24
net1/v4       static   ok           192.168.0.201/24
net2/v4       static   ok           192.168.0.202/24
lo0/v6        static   ok           ::1/128
net0/v6       addrconf ok           fe80::a00:27ff:fe68:6f2d/10
```



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.

# Configuring a Physical Network Interface Manually

1. Check the current status of the `network/physical:default` service by using `svcs network/physical`. If the service is not up and running, enable it by using `svcadm enable network/physical:default`.
2. Create a network interface by using `ipadm create-ip` *`interface`*.
3. Specify the IP address by using `ipadm create-addr -T static -a` *`addrobj`*.
4. Verify the network interface configuration by using `ipadm show-if`.
5. Verify the IP address information by using `ipadm show-addr`.

ORACLE



To configure a physical network interface manually (as opposed to having it done automatically for you by the reactive network configuration), you complete the steps listed in the slide.

**Notes for step 2:** The `ipadm` command is used to configure and manage IP network interfaces, addresses, and TCP/IP protocol properties. 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.

**Notes for step 3:** The `ipadm create-addr -T` command, followed by the address type (the `-a` option) helps you in specifying the IP address to configure on the interface, 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. The `create-addr` subcommand with the `-T static -a` option 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 the source or destination of or for outbound and inbound packets.

**Notes for step 4:** The `show-if` subcommand displays network interface configuration information, either for all the network interfaces configured on the system, including the ones that are only in the persistent configuration, or for the specified network interface.

# Configuring a Physical Network Interface Manually: Example

```
# svcs network/physical
STATE          STIME    FMRI
online         9:34:40 svc:/network/physical:default
# ipadm create-ip net0
# ipadm create-addr -T static -a 192.168.0.100/24 net0/v4add1
# ipadm show-if
IFNAME     CLASS    STATE    ACTIVE OVER
lo0        loopback ok       yes    --
net0       ip       ok       yes    --
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4add1       static   ok           192.168.0.100/24
lo0/v6            static   ok           ::1/128
```



The example in the slide presents the steps for configuring a physical network interface manually.

1. You check if the `network/physical:default` service is online and find that it is.
2. You then create the network interface `net0` and specify the IP address.
3. To verify that your network interface is working, use the `ipadm show-if` command. Here you can see that `net0` is in the `ok` state and active.
4. Verify the IP address for the new network interface by using the `ipadm show-addr` command.

# 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           192.168.0.100/24
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
net0/v6           addrconf disabled     ::
```



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          192.168.0.100/24
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
net0/v6           addrconf disabled    fe80::a00:27ff:fe68:6f2d/10
```



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 a Physical Network Interface Manually

1. Delete the IP address by using `ipadm delete-addr` *`addrobj`*.
2. Delete the network interface by using `ipadm delete-ip` *`interface`*.
3. Verify that the network interface is deleted by using `ipadm show-if`.
4. Verify that the IP address information is deleted by using `ipadm show-addr`.



To delete a physical network interface manually, you complete the steps listed in the slide.

**Notes for step 1:** The `delete-addr` subcommand deletes all the addresses identified for the specified interface. It also removes these addresses from the persistent data store. This means that these addresses will not be instantiated on reboot.

If the address object is a DHCP-controlled address, `delete-addr` removes the address from the system without notifying the DHCP server, and records the current lease for later use.

**Notes for step 2:** The `delete-ip` subcommand deletes the interface from active configuration. All the addresses configured on the interface are deleted. Further, all persistent information related to the interface is removed from the persistent data store and, therefore, the interface is not instantiated on reboot. To disable an interface from active configuration (rather than delete the interface), you can use the `disable-if` subcommand.

**Note:** If you use the `ipadm delete-ip` *`interface`* command first, you do not need to use the `ipadm delete-addr` *`addrobj`* command because the former automatically removes all IP addresses associated with the specified interface.

# Deleting a Physical Network Interface Manually: Example

```
# ipadm delete-addr 192.168.0.100/24 net0/v4add1
# ipadm delete-ip net0
# ipadm show-if
IFNAME     CLASS    STATE    ACTIVE OVER
lo0        loopback ok       yes    --
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
lo0/v6            static   ok           ::1/128
```



The example in the slide presents the steps for deleting a physical network interface manually. In this case, you first delete the IP address associated with the network interface `net0`, and then the interface itself. To verify that the network interface is deleted, use the `ipadm show-if` command. Here, you see that `net0` is no longer part of the configuration. The final step is to verify that the IP address is deleted as well, by using the `ipadm show-addr` command.

# 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



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

- **Practice 7-1:** Manually configuring the network interface
- **Practice 7-2**: Administering profile-based network configuration
- **Practice 7-3**: Creating a virtual network
- **Practice 7-4:** Verifying network operation
- **Practice 7-5**: Managing the virtual network data flow

You will find Practice 7-1 in your *Activity Guide*. It should take about 30 minutes to complete the practice.

# Lesson Agenda

- Reviewing Networking Fundamentals
- Administering Datalink Configuration
- Administering the Network Interface
- **Administering Profile-Based Network Configuration**
- Configuring a Virtual Network
- Verifying Network Operation
- Managing Resources on the Virtual Network

# Profile-Based Network Configuration

- Provides a predetermined set of system-defined profiles
- Provides capabilities for creating various types of user-defined profiles
- Provides the following profile types:
  - Network Configuration Profiles (NCPs)
  - Network Configuration Units (NCUs)
  - Location profiles
  - External Network Modifiers (ENMs)
  - Known WLANs



Oracle Solaris 11 provides a predetermined set of system-defined profiles, as well as the capability for creating various types of user-defined profiles that contain the properties and activation conditions that you specify. User-defined profiles can be used as needed to simplify the basic configuration of the datalinks and IP addresses on your system, as well as to define more complex system-wide network configurations, for example, naming services, IPFilter, and IP Security (IPsec) configurations.

Profile-based network configuration enables you to define multiple alternative configurations, each identified by a single profile [referred to as a network configuration profile (NCP)]. For example, you could create a profile named `office` for a notebook PC that configures the system with static IP addresses and DNS server locations. An alternate `home` profile might use DHCP to acquire this information. A single command enables you to switch from one profile to another profile in a matter of seconds. The various types of profiles that you can enable support two possible network configuration modes: fixed and reactive. The default network configuration mode is determined by whichever profile (NCP) is currently active on your system. If you are unsure of which profile is currently active on your system, use the `netadm list` command to display this information.

The following profile types are used for profile-based network configuration:

- **Network Configuration Profiles (NCPs):** An NCP is the principal profile type that is used to specify the configuration of network datalinks and IP interfaces. NCPs are configured with property values that specify how the network is configured when that particular NCP is activated on the system. NCPs can be reactive or fixed. You can have multiple reactive NCPs configured, but Oracle Solaris 11 currently only supports one fixed NCP named `DefaultFixed`.
  The `Automatic` NCP represents all the links and interfaces that are currently in the system. The content of the `Automatic` NCP changes if network devices are added or removed. The `Automatic` NCP provides access to a profile that utilizes DHCP and address auto configuration, which makes it possible to obtain IP addresses for the system. This NCP also implements a link selection policy that favors wired links over wireless links. If the specification of an alternate IP configuration policy or an alternate link selection policy is required, you would need to create another NCP on your system. You cannot delete the `Automatic` NCP. You can copy this NCP and make changes to the copy.
- **Network Configuration Units (NCUs):** The individual configuration information (or properties) that defines an NCP is configured within NCUs. An NCU can represent a physical link or an interface, and contains properties that specify the configuration for that link or interface. There are two types of NCUs:
  - **Link NCUs:** Link NCUs, for example, physical devices, are Layer 2 entities in the Open Systems Interconnection (OSI) model. Link NCUs represent data links. There are several different classes of data links:
    - Physical links (Ethernet or WiFi)
    - Tunnels
    - Aggregations
    - Virtual local area networks (VLANs)
    - Virtual network interface cards (VNICs)
  - **Interface NCUs:** Interface NCUs, specifically, IP interfaces, are Layer 3 entities in the OSI model. They represent the following IP layer classes:
    - IP interfaces
    - IPMP interfaces
    - VNI interfaces
- **Location Profiles:** They specify the system-wide network configuration, for example, naming services, domain, IPFilter configuration, and IPsec configuration. By default, three location profiles are predefined by the system:
  - **`DefaultFixed`:** The `DefaultFixed` Location is enabled whenever the `DefaultFixed` NCP is active. The `DefaultFixed` Location cannot be directly modified by the using the `netcfg` command.
  - **`Automatic`:** The `Automatic` Location is activated if there are networks available but no other location profile supersedes it. You can modify the `Automatic` Location by using the `netcfg` command.
  - **`NoNet`:** The `NoNet` Location has very specific activation conditions. This Location is applied by the system to a stand-alone system when no local interfaces have an assigned IP address. You can modify the `NoNet` Location by using the `netcfg` command.

- **External Network Modifiers (ENMs):** An ENM is a profile that manages applications that are responsible for creating a network configuration that is external to the system's network configuration, for example, a VPN application.
- **Known Wireless Local Area Networks (WLANs):** A Known WLAN is a profile type that stores information about the wireless networks that are discovered by your system.

For more information about profiles types and their descriptions, refer to http://docs.oracle.com/cd/E36784_01/html/E37475/gneee.html#scrolltoc.

# Reactive Network Configuration Mode

- A reactive network configuration automatically configures Ethernet and Wi-Fi connections.
- The primary focus of a reactive network configuration is mobility.
- A reactive network configuration automatically manages network configuration by storing information in the form of *profiles* in the system.
- You use the `netcfg` and `netadm` commands to create and customize new profiles.



The system's network configuration is organized into various types of profiles. These profiles are either reactive or fixed. Reactive network configuration mode means that the system automatically adapts to any changes in network conditions and network configuration without requiring any manual reconfiguration. For example, if your wired network interface becomes unplugged or if a new wireless network becomes available, the system adapts accordingly.

With the primary focus on mobility, the reactive network configuration policy in Oracle Solaris enables the system's configuration to change dynamically in response to different network events or at your request. This type of network configuration works best for notebook PCs and in situations where network conditions change often. When using reactive network configuration, basic Ethernet and WiFi configuration of a system is automatically performed. The system automatically connects to a wired or wireless network at startup and notifications about the status of the currently active network connection are displayed on the desktop. Reactive profiles are configured with properties that determine the conditions under which the profile is enabled. These properties enable the profile to be applied dynamically to the system by the network management daemon, `nwamd`, as needed.

You use two commands to administer network configuration when you use reactive mode:

- The `netcfg` command for making network configuration changes to profiles
- The `netadm` command for displaying information about all the profiles on a system, and for enabling and disabling profiles.

# How Reactive Network Profiles Work

- The system provides the `Automatic` NCP and the location profile as the default reactive profiles.
- The automatic or reactive network configuration is triggered by an event or activity.
- The profiles perform a basic configuration of your wired or wireless network automatically, without any user interaction.



The system provides the `Automatic` NCP and the location profile as the default reactive profiles. These profiles perform a basic configuration of your wired or wireless networking automatically, without any user interaction. The only time that you are required to interact with reactive networking is if you are prompted by the system for more information, for example, to provide a security key or password for a wireless network. The automatic or reactive network configuration is triggered by the following events and activities:

- Connecting or disconnecting an Ethernet cable
- Connecting or disconnecting a WLAN card
- Booting a system when a wired interface, a wireless interface, or both, are available
- Resuming from suspend when a wired interface, a wireless interface, or both, is available (if supported)
- Acquiring or losing a DHCP lease

To perform basic configuration of your network automatically, the `Automatic` NCP implements the following basic policy:

- Configure all available (connected) Ethernet interfaces by using DHCP.

- If no Ethernet interfaces are connected, or if none can obtain an IP address, enable one wireless interface to automatically connect to the best available WLAN from the Known WLAN list. Alternatively, wait for the user to select a wireless network to connect to.
- Until at least one IPv4 address is obtained, the `NoNet` location remains active. This location profile provides a strict set of IP Filter rules that pass only data that is relevant to the IP address acquisition (DHCP and IPv6 `autoconf` messages). All the properties of the `NoNet` location, with the exception of the activation conditions, can be modified.
- When at least one IPv4 address has been assigned to one of the system's interfaces, the `Automatic` location is enabled. This location profile has no IP Filter or IPsec rules. The location profile applies the DNS configuration data that is obtained from the DHCP server. As with the `NoNet` location, all the properties of the `Automatic` location, with the exception of its activation conditions, can be modified.
- The `NoNet` location is always applied when the system has no IPv4 addresses assigned to it. When at least one IPv4 address is assigned, the system selects the location profile with the activation rules that best match the current network conditions. In the absence of a better match, the system falls back to the `Automatic` location.

# Interaction of Reactive Networking with Other Oracle Solaris Networking Technologies

- Network virtualization
  - Virtual machines: Oracle VM Server for SPARC (formerly Logical Domains) and Oracle VM VirtualBox
  - Oracle Solaris zones and stack instances
- Dynamic Reconfiguration (DR) and NCPs
- Fixed network configuration mode commands



Consider the following when using reactive network configuration with other Oracle Solaris technologies:

- **Virtual machines: Oracle VM Server for SPARC (formerly Logical Domains) and Oracle VM VirtualBox:** Reactive network profiles are supported in both Oracle Solaris hosts and guests. A reactive network configuration manages only the interfaces that belong to the specified virtual machines, and does not interfere with other virtual machines.
- **Oracle Solaris zones and stack instances:** Reactive network profiles work in the global zone or in an exclusive stack, non-global zone.
  **Note:** Reactive network profiles cannot be used in a shared stack zone, because the network configuration for shared stack zones is always managed in the global zone.
- **Dynamic Reconfiguration (DR) and NCPs:** The system's network configuration supports dynamic reconfiguration (DR) and hot-plug features only on systems that support these capabilities. You can use these features to add or remove a device if the active NCP on the system is either reactive (`Automatic` or any user-defined reactive profile) or fixed (`DefaultFixed`). However, the behavior of the system varies depending on the active profile.

- **Fixed network configuration mode commands:** You can use the `ipadm` and `dladm` commands to view the current network configuration and to modify the currently active NCP, when the active NCP is `DefaultFixed` or a user-defined NCP that is managed by reactive networking.
  **Note:** When a reactive NCP is active, the links and interfaces that are created with these commands are assigned implicit activation conditions, such that they depend on their underlying link or interface. For example, if `dladm` is used to create a VNIC, that VNIC NCU has an implicit dependency on its underlying link.

# `netcfg` Command

| `netcfg` Subcommand | Description |
|---|---|
| `create` | Create an in-memory profile of the specified type and name. |
| `select` *`object-type`* | Select the profiles that are available at the current scope level and move into that object's scope. |
| `walkprop` | Walk each property associated with the current profile. For each property, the name and current value are displayed, and a prompt is given to allow the user to change the current value. |
| `set` *`prop-name=value1`* | Set the current (in-memory) value of the specified property. If the process is performed in non-interactive mode, the change is also committed to persistent storage. |
| `list` | List all profiles, property-value pairs, and resources that exist at the current or specified scope. |
| `verify` | Verify that the current in-memory object has a valid configuration. |
| `commit` | Commit the current profile to persistent storage. |
| `end` | End the current profile specification, and move to the next higher scope. |
| `exit` | Exit the `netcfg` session. The current profile is verified and committed before ending. |
| `destroy` | Remove the specified profile from memory and persistent storage. |

ORACLE



This slide shows the `netcfg` subcommands. The `netcfg` command is used to create and modify network configuration profiles. Using the `netcfg` command, you can perform the following tasks:

- Create or destroy a user-defined profile.
- Open an existing profile for viewing and/or editing.
- List all the profiles that exist on a system and their property values.
- List all the property values and resources for a specified profile.
- Display each property that is associated with a profile.
- Set or modify one or all the properties of a specified profile.
- Export the current configuration for a user-defined profile to standard output or a file.
- Delete any changes that were made to a profile and revert to the previous configuration for that profile.
- Verify that a profile has a valid configuration.

# `netadm` Command

| `netadm` Subcommand | Description |
|---|---|
| `enable` | Enable the specified profile. If the profile name is not unique, the profile type must be specified to identify the profile that is to be enabled. |
| `disable` | Disable the specified profile. If the profile name is not unique, the profile type must be specified to identify the profile that is to be disabled. |
| `list` | List all available profiles and their current state. If a profile is specified by name, list only the current state of that profile. |
| `show-events` | Listen for a stream of events from the NWAM daemon and display them. |
| `scan-wifi` | Initiate a wireless scan on link *linkname*. |
| `select-wifi` | Select a wireless network to connect to, from the scan results on link *linkname*. You may be prompted for selection, WiFi key, and so forth, if necessary. |
| `help` | Display a usage message with short descriptions for each subcommand. |



The `netadm` command is used to administer network profiles and interact with the NWAM daemon.

The subcommands that are supported by the `netadm` command are shown in this slide.

# SMF Network Services

In Oracle Solaris 11, network configuration is implemented by multiple SMF services as follows:

| Service | Description |
|---|---|
| `svc:/network/loopback:default` | Creates the IPv4 and IPv6 loopback interfaces |
| `svc:/network/netcfg:default` | Manages the network configuration repository, with its primary function being to start the `netcfgd` daemon. This service is a prerequisite for the `svc:/network/physical:default` service. |
| `svc:/network/physical:default` | Brings up links and plumbs IP interfaces. This service starts the network management daemon, `nwamd`. |
| `svc:/network/location:default` | Enables the location profile that is selected by the `nwamd` daemon. This service is dependent on the `svc:/network/physical:default` service. |



**Note:** The `svc:/network/location:default` service has a property that stores the current location profile. Do not directly manipulate this property. Rather, use the CLI or the network administration GUI to make changes.

# Configuring a Reactive Network

This section covers the following topics:

- Configuring network configuration profile
- Creating a location profile
- Listing a location profile
- Modifying profiles
- Listing reactive network profiles
- Enabling and disabling profiles
- Displaying profile states
- Querying profile information
- Creating a backup of a profile
- Removing reactive network profiles

ORACLE

# Creating a Network Configuration Profile

To create an NCP, use the `netcfg` utility.

```
# netcfg
netcfg> create ncp my_profile
netcfg:ncp:my_profile> create ncu phys net1
Created ncu 'net1'. Walking properties …
activation-mode (manual) [manual|prioritized]> manual
mac-address> <ENTER>
autopush> <ENTER>
mtu> <ENTER>
netcfg:ncp:my_profile:ncu:net1> list
ncu:net1
type                 link
class                phys
parent               "my_profile"
activation-mode      manual
enabled              true
netcfg:ncp:my_profile:ncu:net1> end
Committed changes
netcfg:ncp:my_profile> list
ncp:my_profile
        management-type      reactive
NCUs:
       phys  net1
netcfg:ncp:my_profile> exit
```



Using the interactive `netcfg` tool, you can create a Network Configuration Profile (NCP) and any Network Configuration Units (NCUs) within it.

**Note:** Recall that NCUs are containers that store all the individual configuration objects that make up an NCP. Each object correlates to an individual link or interface in the system.

When creating the NCU, the system interactively walks you through the creation process via setting the profile properties. You can use the defaults by pressing the Enter key, or enter the desired configuration for each step. In this example, the activation mode is set to `manual` by entering it when prompted and all the default link properties are selected by pressing the Enter key. When you have finished, you can list the NCU to display the configuration.

Note that after the `end` command commits the changes to the NCU, you can enter another `list` command at the profile level to list all the NCUs contained within the profile.

# Creating a Location Profile

Use the `netcfg` utility as follows:

```
# netcfg
netcfg> create loc office
Created loc 'office'.  Walking properties ...
activation-mode (manual) [manual|conditional-any|conditional-all]> conditional-all
conditions> "system-domain is mydomain.com"
nameservices (dns) [dns|files|nis|ldap]> dns
nameservices-config-file ("/etc/nsswitch.dns")> <ENTER>
dns-nameservice-configsrc (dhcp) [manual|dhcp]> manual
dns-nameservice-domain> "mydomain.com"
dns-nameservice-servers> "192.168.0.100"
dns-nameservice-search> <ENTER>
dns-nameservice-sortlist> <ENTER>
dns-nameservice-options> <ENTER>
nfsv4-domain> <ENTER>
ipfilter-config-file> <ENTER>
ipfilter-v6-config-file> <ENTER>
ipnat-config-file> <ENTER>
ippool-config-file> <ENTER>
ike-config-file> <ENTER>
ipsecpolicy-config-file> <ENTER>
netcfg:loc:office> list
```

ORACLE



You can also create a location profile in interactive mode by using `netcfg`. Again, the system walks you through setting the properties of your location profile, and then enables you to list them. In the example in the slide, a location profile called `office` is created. When entering the `conditional-all` property, the next prompt asks you to state the conditions. In this case, the system domain is set to the domain name. When a name service (such as DNS) is selected, the properties for that name service appear so that you can set them. Again, you can accept the default setting by pressing Enter or you can enter the desired setting.

**Note:** The output continues in the next slide.

# Listing a Location Profile

```
netcfg:loc:office> list
loc:office
          activation-mode                 conditional-all
          conditions                      "system-domain is mydomain.com"
          enabled                         false
          nameservices                    dns
          nameservices-config-file        "/etc/nsswitch.dns"
          dns-nameservice-configsrc       manual
          dns-nameservice-domain          "mydomain.com"
          dns-nameservice-servers         "192.168.0.100"
netcfg:loc:office> verify
All properties verified
netcfg:loc:office> commit
Committed changes
netcfg:loc:office> end
netcfg> exit
```



The `list` command (at the location profile level within the `netcfg` command) lists the properties of the `office` location profile that was created in the previous slide. The `verify` command verifies all the properties and the `commit` command commits the changes. The location profile creation process is complete after you exit the command.

# Modifying Profiles

```
# netcfg
netcfg> select ncp my_profile
netcfg:ncp:my_profile> select ncu net1
netcfg:ncp:my_profile:ncu:net1> list
ncu:net1
              type                    link
              class                   phys
              parent                  "my_profile"
              activation-mode         manual
              enabled                 true
netcfg:ncp:my_profile:ncu:net1> set activation-mode=prioritized
netcfg:ncp:my_profile:ncu:net1> list
ncu:net1
              type                    link
              class                   phys
              parent                  "my_profile"
              activation-mode         prioritized
              enabled                 true
netcfg:ncp:my_profile:ncu:net1> commit
Committed changes
netcfg:ncp:my_profile:ncu:net1> end
netcfg:ncp:my_profile> exit
```



To modify a profile, use the `netcfg` utility. First, select the profile, and then the NCU. You can then set a different property by using the `set` subcommand and the syntax of *`property=value`*.

In the example in the slide, the activation mode is changed from `manual` to `prioritized`. Some properties (such as `type`, `class`, and `enabled`) are read-only and cannot be modified.

# Listing Reactive Network Profiles

Use the `netcfg` utility to list all the NCPs and locations:

```
# netcfg list
NCPs:
     Automatic
     DefaultFixed
     my_profile
Locations:
     aces
     Automatic
     NoNet
     DefaultFixed
     office
```



You can use the `netcfg list` command to list all current NCPs and location profiles in the system, which includes system-defined profiles and locations such as `Automatic`, `NoNet`, and `User`. Any custom NCPs and locations that were created also appear, such as the `my_profile` profile and the `office` location.

# Enabling and Disabling Reactive Network Profiles

Use the `netadm` utility to enable and disable an NCP or location profile.

- To enable newly created profiles:

```
# netadm enable office
Enabling loc 'office'
# netadm enable my_profile
Enabling ncp 'my_profile'
```

- To disable newly created profiles:

```
# netadm disable office
Disabling loc 'office'
# netadm enable -p ncp Automatic
Enabling ncp 'Automatic'
```



After the reactive network profiles are created and verified, you can use the `netadm enable` command to enable the profiles. When enabling or disabling profiles, if the profile name is not unique, the profile type (NCU/loc/NCP) must be specified with the `-p` option. To disable a location profile, use the `netadm disable` command. To disable an NCP, enable another one in its place. You cannot disable an NCP with the `netadm disable` command.

Profiles are also automatically enabled according to the policies set, or when an event occurs such as switching from an Ethernet cable to a wireless connection.

# Displaying Profile States

To list the reactive network profiles and their current states, use the `netadm` utility.

```
# netadm list
TYPE        PROFILE        STATE
ncp         Automatic      disabled
ncp         DefaultFixed   disabled
ncp         my_profile     online
ncu:phys    net0           online
ncu:ip      net0           online
loc         office         online
loc         Automatic      offline
loc         NoNet          offline
loc         DefaultFixed   offline
```



The states reported are `online`, `offline`, `disabled`, `initialized`, or `uninitialized`.

# Displaying Profiles and Their Auxiliary States

To list the reactive network profiles and their auxiliary states, use `netadm list -x`.

```
# netadm list -x
TYPE      PROFILE      STATE     AUXILIARY STATE
ncp       Automatic    disabled  disabled by administrator
ncp       DefaultFixed disabled  disabled by administrator
ncp       my_profile   online    active
ncu:phys  net0         online    interface/link is up
ncu:ip    net0         online    interface/link is up
loc       office       online    active
loc       Automatic    offline   conditions for activation are unmet
Loc       DefaultFixed offline   conditions for activation are unmet
loc       NoNet        offline   conditions for activation are unmet
```

# Creating a Backup of a Profile

To create a backup of a reactive network profile, use `netcfg export -f profile`.

```
# netcfg export -f oracle_ncp_backup ncp my_profile
# ls *backup
oracle_ncp_backup
```



To create a backup of a profile, use the `netcfg export -f` command followed by the name of the backup file and the profile. In the example, a backup called `oracle_ncp_backup` is being created for the `my_profile` profile. You can verify that the backup has been created by using the `ls *backup` command. The backup is listed.

# Removing Reactive Network Profiles

To remove a profile, use `netcfg destroy`.

```
# netcfg destroy loc office
# netcfg destroy ncp my_profile
```

# Practice 7-2 Overview: Administering Profile-Based Network Configuration

This practice covers the following topics:

- Assessing the current reactive network configuration
- Creating and deploying a reactive network profile



Practice 7-2 should take you about 40 minutes to complete.

# Lesson Agenda

- Reviewing Networking Fundamentals
- Administering Datalink Configuration
- Administering the Network Interface
- Administering Profile-Based Network Configuration
- **Configuring a Virtual Network**
- Verifying Network Operation
- Managing Resources on the Virtual Network

# Network Virtualization and Virtual Networks

- Network virtualization:
  - Is the process of combining hardware network resources and software network resources
  - Provides efficient, controlled, and secure sharing of network resources
- Virtual networks:
  - **External networks:** Several local networks that are administered by software as a single entity
  - **Internal networks:** One system that uses virtual machines or zones that are configured over at least one pseudo network interface
    - A special type of internal virtual network is the **private virtual network**, which is a virtual network on a system that cannot be accessed by external networks.

ORACLE



Network virtualization is the process of combining hardware network resources and software network resources into a single administrative unit. This single administrative unit is known as a virtual network. The goal of network virtualization is to provide systems and users with efficient, controlled, and secure sharing of networking resources.

The end product of network virtualization is the virtual network. Virtual networks are classified into two broad types: external and internal. External virtual networks consist of several local networks that are administered by software as a single entity. The building blocks of classic external virtual networks are switch hardware and VLAN software technology. Examples of external virtual networks include large corporate networks and data centers.

An internal virtual network consists of one system that uses virtual machines or zones that are configured over at least one pseudo network interface. These containers can communicate with each other as though they are on the same local network, thus providing a virtual network on a single host. The building blocks of a virtual network are virtual network interface cards or virtual NICs (VNICs) and virtual switches. Oracle Solaris network virtualization provides the internal virtual network solution, which is covered in this course.

A special type of internal virtual network is the *private virtual network*. Private virtual networks are different from virtual private networks (VPNs). VPN creates a secure point-to-point link between two endpoint systems. The private virtual network is a virtual network on a system that cannot be accessed by external networks. The isolation of this internal network from other external networks is achieved by configuring VNICs over a pseudo NIC called an etherstub.

Network virtualization is optimized when it is combined with network resource management. You can provide systems and users with controlled sharing of the hardware and software networking resources, thus increasing the efficiency of the virtual networking processes. The features of network virtualization with network resource management help you to manage flow control, improve system performance, and configure the network utilization needed to achieve OS virtualization, utility computing, and server consolidation.

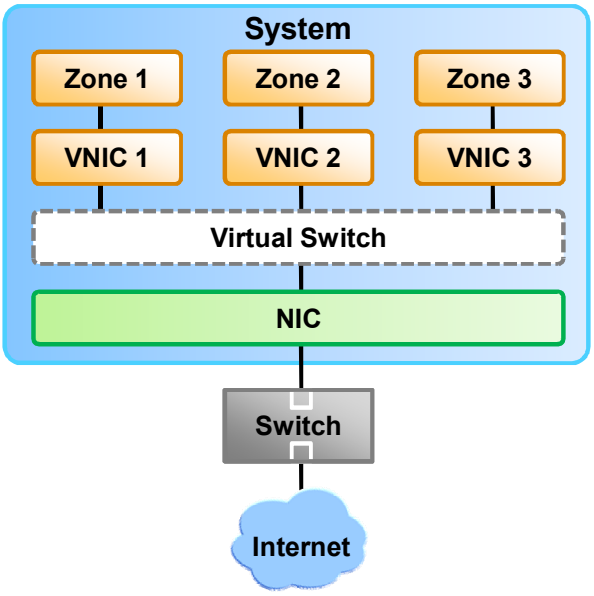
Traditional network isolation methods, such as VLANs, are not adequate to support virtualization in large data centers. Because cloud environments are tightly coupled with the underlying physical networks, virtual machines cannot be migrated between physical servers that belong to different physical Layer 2 networks. Oracle Solaris 11.2 introduces support for the virtual extensible local area network (VXLAN) technology that addresses such virtualization issues in a large virtualized data center or cloud environment. *Virtual eXtensible Area Network* (VXLAN) is an L2 and L3 technology that works by overlaying a datalink (L2) network on top of an IP (L3) network. VXLANs address the 4K limitation that is imposed when using VLANs. Typically, VXLANs are used in a cloud infrastructure to isolate multiple virtual networks. You can manage VXLANs by using the Elastic Virtual Switch (EVS) feature, which is covered in detail in the follow-on course *Oracle Solaris 11 Advanced System Administration*. For more information about VXLAN and EVS, refer to http://docs.oracle.com/cd/E36784_01/html/E36813/vxlan.html#NWVIRvxlan and http://docs.oracle.com/cd/E36784_01/html/E36813/gnrgr.html#scrolltoc.

You learn more about network resource management in the next section.

# Virtual Network Components

A virtual network has the following components:

- Virtual Network Interface Card (VNIC)
- Virtual switch
- Etherstub
- Zone




ORACLE



An internal virtual network that is built on Oracle Solaris consists of the following components:

- At least one network interface card (NIC)
- A virtual NIC (VNIC), which is configured on top of the network interface. The VNIC is a virtual network device with the same datalink interface as the physical interface.
- A virtual switch, which is configured at the same time as the first VNIC on the interface. The virtual switch provides the same connectivity between the VNICs on a virtual network as that provided by the switch hardware for the systems that are connected to a switch's ports.
- A container, such as a zone or virtual machine, which is configured on top of the VNIC

The graphic in the slide shows these components and how they fit together on a single system. The single system has one NIC. The NIC is configured with three VNICs. Each VNIC supports a single zone. Therefore, Zone 1, Zone 2, and Zone 3 are configured over VNIC 1, VNIC 2, and VNIC 3, respectively. The three VNICs are virtually connected to one virtual switch. This switch provides the connection between the VNICs and the physical NIC on which the VNICs are built. The physical interface provides the system with its external network connection. You learn about the Oracle Solaris 11 Zones feature in the lesson titled "Administering Oracle Solaris Zones."

Alternatively, you can create a virtual network based on an etherstub. Etherstubs are purely software and do not require a network interface as the basis for the virtual network. In this lesson, you learn how to create a virtual network by using an etherstub.

# Creating a Virtual Network

This section covers the following topics:

- Creating a virtual network switch
- Creating the virtual network interfaces
- Displaying the virtual network configuration

# Creating a Virtual Network Switch

To create an etherstub, use `dladm create-etherstub` *`etherstub`*.

```
# dladm create-etherstub stub0
```

To verify the creation of the etherstub, use `dladm show link`.

```
# dladm show-link
LINK        CLASS     MTU    STATE    OVER
net0        phys      1500   up       --
net1        phys      1500   up       --
net2        phys      1500   up       --
net3        phys      1500   up       --
stub0       etherstub 9000   unknown  --
```

ORACLE



An etherstub can be used instead of a physical NIC to create VNICs. VNICs that are created on an etherstub appear to be connected through a virtual switch, allowing complete virtual networks to be built without physical hardware. The VNICs over an etherstub become independent of the physical NICs in the system. You can use etherstubs to isolate the virtual network from the rest of the virtual networks in the system, as well as the external network to which the system is connected.

You cannot use an etherstub by itself. Instead, you use VNICs with an etherstub to create the private or isolated virtual networks. You can create as many etherstubs as you require. You can also create as many VNICs over each etherstub as required.

To create an etherstub, use the `dladm create-etherstub` command followed by the etherstub name. In the example in the slide, you are creating the etherstub `stub0`.

To confirm the creation of the etherstub, you can use the `dladm show-link` command, as shown in the example in the slide. Here, you can see that `stub0` has been created and that its current state is unknown.

# Creating the Virtual Network Interfaces

To create a VNIC and attach it to the etherstub, use `dladm create-vnic -l` *`etherstub vnic`*.

```
# dladm create-vnic -l stub0 vnic0
# dladm create-vnic -l stub0 vnic1
# dladm create-vnic -l stub0 vnic2
```



After you have created the etherstub, you can create the VNICs and attach them to the etherstub by using the `dladm create-vnic` command followed by the `-l` option, the etherstub name, and the VNIC name, as shown in the first example in the slide. The `-l` option precedes the link, which can be either a physical link or an etherstub.

**Note**

- `vnic0` is required for the virtual switch. The other VNICs (`vnic1` and `vnic2`) are for use with the zones that will be created.
- You can use the `dladm delete-vnic` command followed by the VNIC name to delete the VNICs.

# Displaying the Virtual Network Configuration

To display the virtual network configuration, use `dladm show-vnic`.

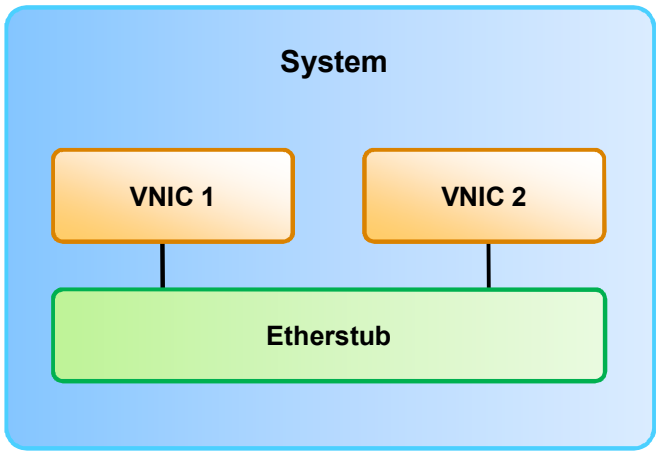
```
# dladm show-vnic
LINK        OVER        SPEED  MACADDRESS        MACADDRTYPE   VID
vnic0       stub0       40000  2:8:20:70:d0:f8   random        0
vnic1       stub0       40000  2:8:20:80:65:0    random        0
vnic2       stub0       40000  2:8:20:1f:c5:bd   random        0
```



To verify that the VNICs are created and to display the virtual network configuration, you can use the `dladm show-vnic` command, as shown in the example in the slide. The `dladm show-vnic` command is used to show the VNIC configuration information for all VNICs, all VNICs on a link, or only a specified *`vnic-link`*. The output for this command displays the name of the link (`LINK`), the name of the physical link over which the VNIC is configured (`OVER`), the maximum speed of the VNIC [in megabits per second (`SPEED`)], the MAC address of the VNIC (`MACADDRESS`), the MAC address type of the VNIC (`MACADDRTYPE`) that can be either a random address assigned to the VNIC (`random`) or a factory MAC address used by the VNIC (`factory`), and the VLAN identifier (VID). The etherstub or virtual switch uses the VLAN identifier to determine the interface to send a data packet to.

In this example, all the VNICs have been configured over etherstub `stub0`. Currently, no data is passing through the links, so speed is not being recorded. MAC addresses are present for each VNIC and they have all been randomly assigned. There is one VLAN and it is identified with VID `0`.

# The Virtual Network Configuration So Far






The graphic in the slide illustrates what the virtual network configuration looks like so far. There is the etherstub, and two VNICs connected to the switch.

Now that you have created the network, you are ready to configure your zones on top of this network. You look at how to do this in the lesson titled “Administering Oracle Solaris Zones.”

# Quiz

Which utility is used to create virtual switches and VNICs?

a. `lnkadm`
b. `dladm`
c. `vniccfg`
d. `dlcfg`



**Answer: b**

# Quiz

A VNIC is a virtual network device with the same datalink interface as a physical interface.

a. True
b. False



**Answer: a**

# Quiz

In which order is a virtual network created?

a. Virtual switch, VNICs, zones
b. Zones, VNICs, virtual switch
c. VNICs, virtual switch, zones



**Answer: a**

# Quiz

You have created an etherstub called `stub2`. You now want to create `vnic1` and attach it to `stub2`. Which set of commands would you use to do this?

a. `# dladm create-vnic1`
b. `# dladm create-vnic -l vnic1`
c. `# dladm create-vnic -l stub2 vnic0`
d. `# dladm create-vnic -l stub2 vnic1`



**Answer: d**

# Practice 7-3 Overview: Creating a Virtual Network

This practice covers the following topics:

- Creating a virtual network switch
- Creating the virtual network interfaces
- Displaying the virtual network configuration



This practice should take about 15 minutes to complete.

# Lesson Agenda

- Reviewing Networking Fundamentals
- Administering Datalink Configuration
- Administering the Network Interface
- Administering Profile-Based Network Configuration
- Configuring a Virtual Network
- **Verifying Network Operation**
- Managing Resources on the Virtual Network

# Verifying Network Operation

- Examining the status of all network interfaces
- Checking network interface traffic status
- Verifying the status of network interfaces
- Checking the routing table
- Viewing user and process information
- Viewing statistics on IP, TCP, and UDP traffic
- Checking network connectivity and response times
- Capturing packets from the network

# 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
net3/v4           static   ok       192.168.0.203/24
lo0/v6            static   ok       ::1/128
net0/v6           addrconf ok       fe80::a00:27ff:fe68:6f2d/10
```

ORACLE



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 in the slide, you again see the IPv4 and IPv6 loopback interfaces and the four `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 four `netX` interfaces. All these are examples of the Classless Inter-Domain Routing (CIDR) format of the IPv4 address 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    --
net3       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, thus 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 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 (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 in the slide, 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 statistics for the interface. On the right, you have the total number of packets sent and received by this interface.

# Verifying the Status of Network Interfaces

To display the status of the network interfaces, use the `netstat -i` command.

```
# netstat -i

Name  Mtu  Net/Dest      Address        Ipkts  Ierrs Opkts  Oerrs Collis Queue
lo0   8232 loopback      localhost      845037 0     845037 0     0      0
net0  1500 server1       server1        87805  0     126771 0     0      0

...
(output truncated)
```



The main utility for verifying network statistics is `netstat`. The `netstat -i` command shows the state of the interfaces that are used for IP traffic. The output includes the names of the physical interfaces, counts for the input and output packets (`Ipkts` and `Opkts`), plus additional information, such as counts for input and output errors (`Ierrs` and `Oerrs`) and collisions (`Collis`). You can study these stats to determine the health of the network.

# Checking the Routing Table

To view the known routes, use the `netstat -r` command.

```
# netstat -r
Routing Table:  IPv4
Destination   Gateway     Flags   Ref   Use    Interface
-----------   --------    -----   ---   ----   ---------
localhost     localhost   UH      2     2817   lo0
192.168.0.0   server1     U       4     14293  net0
. . .
(output truncated)
```



The `netstat -r` command shows the routing tables for either IPv4 or IPv6. In the example in the slide, the `UH` flags mean that the route is up through a host, as opposed to `UG`, which is through a gateway. The `Ref` column shows the current number of routes that share the same link layer, and the `Use` column indicates the number of packets sent.

**Note:** If you set `DEFAULT_IP=VERSION4` in the `/etc/default/inet_type` file, the IPv6 statistics are omitted from the `netstat` displays.

# Viewing User and Process Information

To list the user, process ID, and the program that originally created the network endpoint or controls it now, use the `netstat -u` command.

```
# netstat -nauv
UDP: IPv4
   Local Address        Remote Address      User     Pid     State       Command
-------------------- -------------------- -------- ------ ---------- ----------------
      *.*                                  root         79 Unbound    /lib/inet/in.mpathd
      *.*                                  root         79 Unbound    /lib/inet/in.mpathd
      *.*                                  netadm      308 Unbound    /lib/inet/nwamd
      *.*                                  netadm      308 Unbound    /lib/inet/nwamd
      *.631                                root        430 Idle       /usr/sbin/cupsd -C
/etc/cups/cupsd.conf
127.0.0.1.53                               root        443 Idle       /usr/sbin/named
192.168.0.100.53                           root        443 Idle       /usr/sbin/named
      *.111                                daemon      539 Idle       /usr/sbin/rpcbind
      *.*                                  daemon      539 Unbound    /usr/sbin/rpcbind
      *.52951                              daemon      539 Idle       /usr/sbin/rpcbind
      *.111                                daemon      539 Idle       /usr/sbin/rpcbind
      *.*                                  daemon      539 Unbound    /usr/sbin/rpcbind
      *.36871                              daemon      539 Idle       /usr/sbin/rpcbind
      *.*                                  root        585 Unbound    /usr/lib/inet/in.ndpd
      *.520                                root        782 Idle       /usr/sbin/in.routed
      *.68                                 root        787 Idle       /sbin/dhcpagent
      *.546                                root        787 Idle       /sbin/dhcpagent
. . .
(output truncated)
```



Starting with Oracle Solaris 11.2 release, the `netstat` command provides the `-u` option to view information about the processes and users in the `netstat` output. In the slide example:

- **`-a`:** Displays the state of all sockets, all routing table entries, or all interfaces, both physical and logical
- **`-n`:** Displays network addresses as numbers. `netstat` normally displays addresses as symbols.
- **`-v`:** Provides verbose information

In the example, the output includes details of both IPv4 and IPv6, and all active UNIX domain sockets.

# Viewing Statistics on IP Traffic

To gather and report statistics on IP traffic based on the selected output mode and sort order, use the `ipstat` command.

```
# ipstat -l 5
SOURCE                    DEST                      PROTO    INT         BYTES
s11-server1.mydomain.com  s11-desktop.mydomain.com  UDP      net0        39.0
s11-desktop.mydomain.com  s11-server1.mydomain.com  UDP      net0        28.0
Total: bytes in: 39.0  bytes out: 28.0
```



Starting with Oracle Solaris 11.2 release, you can use the `ipstat`(1M) command to report statistics on IP traffic. `ipstat` provides options to gather and report statistics only on IP traffic that matches the specified source or destination address, interface, and higher layer protocol. For more information, refer to the `ipstat`(1M) man page.

The following list defines the column headings and meanings of an `ipstat` report:

- **SOURCE:** The source IP address or host name associated with the traffic
- **DEST:** The destination IP address or host name associated with the traffic
- **PROTO:** The higher layer protocol associated with the traffic (TCP, UDP, SCTP)
- **IFNAME:** The name of the interface associated with the traffic
- **BYTES:** The rate of IP traffic over the sampling interval. In regular output, the rate is reported in bytes (no suffix), kilobytes (K), megabytes (M), gigabytes (G), terabytes (T), or petabytes (P) per second. In machine-parsable output, the rate is given in bytes per second. The `-u` option can be used to specify a fixed unit for this number.

# Viewing Statistics on TCP and UDP Traffic

To gather and report statistics on TCP and UDP traffic based on the selected output mode and sort order, use the `tcpstat` command.

```
# tcpstat -l 5
ZONE          PID PROTO   SADDR               SPORT DADDR               DPORT   BYTES
global        795 UDP     s11-server1.mydo       53 s11-desktop.mydo   42857    20.0
global        795 UDP     s11-desktop.mydo    42857 s11-server1.mydo       53     9.0
global        795 UDP     s11-desktop.mydo    59127 s11-server1.mydo       53     7.0
global        795 UDP     s11-desktop.mydo    38509 s11-server1.mydo       53     7.0
global        795 UDP     s11-server1.mydo       53 s11-desktop.mydo   59127     7.0
Total: bytes in: 34.0  bytes out: 23.0
```



Starting with Oracle Solaris 11.2 release, you can use the `tcpstat`(1M) command to report statistics on TCP and UDP traffic. `tcpstat` provides options to gather and report statistics only on traffic that matches the specified source or destination address, interface, process ID, source or destination port, and zone name. For more information, refer to the `tcpstat`(1M) man page.

The following list defines the column headings and meanings of a `tcpstat` report:

- **ZONE:** The name of the zone associated with the network traffic
- **PID:** The process ID associated with the network traffic
- **PROTO:** The protocol associated with the network traffic
- **SADDR:** The source IP address or host name associated with the network traffic
- **SPORT:** The source port associated with the network traffic
- **DADDR:** The destination IP address or host name associated with the network traffic
- **DPORT:** The destination port associated with the network traffic
- **BYTES:** The rate of network traffic over the sampling interval. In regular output, the rate is reported in bytes (no suffix), kilobytes (K), megabytes (M), gigabytes (G), terabytes (T), or petabytes (P) per second. In machine-parsable output, the rate is given in bytes per second. The `-u` option can be used to specify a fixed unit for this number.

# Checking Network Connectivity and Response Times

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

```
# 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
round-trip (ms) min/avg/max/stdev = 1.639/0.724/1.143/0.649
```



To check 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.

# Capturing Packets from the Network

To capture packets, use `snoop`.

```
# snoop -v
Using device net0 (promiscuous mode)
ETHER:  ----- Ether Header -----
ETHER:  Packet 1 arrived at 13:52:2.50694
ETHER:  Packet size = 106 bytes
ETHER:  Destination = 0:7:e9:24:45:93, PCS Computer Systems GmbH
ETHER:  Source      = 0:3:ba:45:a6:d4,
ETHER:  Ethertype = 0800 (IP)
. . . . .
IP:   ----- IP Header -----
IP:   Version = 4
IP:   Header length = 20 bytes
IP:   Type of service = 0x00
IP:         xxx. .... = 0 (precedence)
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
. . . . .
^C
```



The `snoop` command is a useful troubleshooting or informational tool. It captures packets from a datalink or an IP interface and displays their content. If a datalink or an IP interface is not specified on the command line, `snoop` will pick a datalink to use based on the ones that have been configured for IP traffic. It can display packets in a single-line summary form or in verbose multiline forms. The output mode runs until a Ctrl + C character is entered. The captured packets can also be saved to a file by using `snoop`.

The example in the slide shows a truncated output that uses the multiline verbose mode.

# 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 7-4 Overview: Verifying Network Operation

This practice covers the following topics:

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



This practice should take about 20 minutes to complete.

# Lesson Agenda

- Reviewing Networking Fundamentals
- Administering Datalink Configuration
- Administering the Network Interface
- Administering Profile-Based Network Configuration
- Configuring a Virtual Network
- Verifying Network Operation
- **Managing Resources on the Virtual Network**

# Network Resource Management: Overview

- Network resource management is the process of managing and allocating resources for networking processes.
- It is comparable to creating dedicated lanes for traffic.
- You can assign resources differently depending on the amount of network traffic that is being processed.
- It helps in increasing the system's efficiency when processing packets.



Network resource management is the process of managing and allocating resources for networking processes. You can assign resources differently depending on the amount of network traffic that is being processed. By managing and allocating resources according to actual need, you increase the system's efficiency when processing packets.

In Oracle Solaris, quality of service (QoS) is obtained more easily and dynamically by managing network resources. Network resource management is comparable to creating dedicated lanes for traffic. When you combine different resources to provide to the specific types of network packets, those resources form a network lane for those packets. Resources can be assigned differently for each network lane. For example, you can allocate more resources to a lane where network traffic is the heaviest. By configuring network lanes where resources are distributed according to actual need, you increase the system's efficiency in processing network packets. For more information about network lanes, refer to http://docs.oracle.com/cd/E36784_01/html/E36813/gjzbf.html#scrolltoc.

The following network resources are used to increase the system's efficiency in processing packets:

- **Bandwidth:** You can limit the bandwidth of the datalink according to the need of the networking processes supported by the datalink.
- **Priority:** You can prioritize the order in which packets are processed. Latency is reduced for packets with higher priority because they are processed ahead of the other packets.
- **NIC rings:** If a NIC supports ring allocation, its transmit and receive rings can be dedicated for use by datalinks. For more information, refer to http://docs.oracle.com/cd/E36784_01/html/E36813/gjxje.html#scrolltoc.
- **CPU pools:** Pools of CPUs are created and associated with specific zones. These pools can be further assigned to datalinks to manage the network processes of their associated zones. For more information, refer to http://docs.oracle.com/cd/E36784_01/html/E36813/gjcjq.html#scrolltoc.
- **CPUs:** On a system with multiple CPUs, you can dedicate a given number of CPUs for specific network processing. For more information, refer to http://docs.oracle.com/cd/E36784_01/html/E36813/gndgu.html#scrolltoc.

### Benefits of Network Resource Management

By using network resource management, you can isolate, prioritize, track, and control data traffic on an individual system without the complex QoS rule definitions. Network resource management is helpful for the following tasks:

- Provisioning the network
- Establishing service-level agreements
- Billing clients
- Diagnosing security problems

# Methods of Managing Network Resources

The network resources on a system can be managed in one of the following ways:

- **Datalink properties:** Improves the system's efficiency in processing packets
- **Flows:** Controls how resources are used to process network packets



Managing network resources by setting datalink properties improves the system's efficiency in processing packets. You can set datalink properties when you create the link. Alternatively, you can set datalink properties later, for example, after studying resource usage over time and determining how to better allocate the resource. By setting datalink properties that pertain to network resources, you can decide how much of a given resource can be used for the networking processes. The procedures for allocating resources apply to the virtual network, as well as the physical network. For more information about datalink properties and how to configure them, refer to http://docs.oracle.com/cd/E36784_01/html/E36813/gnbdn.html#scrolltoc.

A flow is a customized way of categorizing network packets based on a single attribute or a combination of attributes. The attributes that serve as the basis for creating flows are derived from the information in a network packet's header. After setting datalink properties for network resource management, flows can be used to further control how resources are used to process network packets. Flows alone can also be used to manage network resources without setting datalink properties.

Using flows for managing resources involves the following steps:

- Creating a flow based on a single attribute or a combination of attributes
- Customizing a flow's use of resources by setting properties that pertain to network resources. Currently, only bandwidth and priority properties can be set.

# Managing Virtual Network Resources by Using Flows

Flows:

- Are created on a per-VNIC basis
- Are used to categorize network packets
- Define and isolate packets with similar characteristics
- Can be assigned specific resources

Bandwidth is assigned based on the usage policy for the system.



Resource management for a virtual network involves creating flows on a per-VNIC basis. A flow is a customized way of categorizing network packets to further control how resources are used to process these packets. These flows define and isolate packets with similar characteristics, such as the port number or IP address of the sending host. Packets that share an attribute constitute a flow and are labeled with a specific flow name. Specific resources can then be assigned to the flow. You assign bandwidth based on the usage policy for the system.

# Managing Resources on the Virtual Network

This section covers the following topics:

- Determining the configured VNIC states
- Creating and adding a flow
- Displaying flow controls
- Creating flows and selecting flow properties
- Setting flow properties
- Displaying flow control properties
- Setting a priority property

# Determining the Configured VNIC States

To determine the current state of the VNICs on the system, use `dladm show-link`.

```
# dladm show-link
net0                phys       1500   up        --
stub0               etherstub  9000   unknown   --
net3                phys       1500   up        --
net1                phys       1500   up        --
net2                phys       1500   up        --
vnic0               vnic       9000   up        stub0
vnic1               vnic       9000   up        stub0
vnic2               vnic       9000   up        stub0
```



Before you create a flow, you want to determine the current state of the VNICs that you want to create the flow for. To do this, use `dladm show-link`, as shown in the example in the slide. As you can see, the VNICs that you created earlier are up.

# Creating and Adding a Flow

1. Create a new VNIC by using `dladm create-vnic -l` *`etherstub vnic`*.
2. Select the attribute on which you want to base the flow.
3. Determine how you want to customize the flow's use of the network resources.
4. Add the VNIC as a flow by using `flowadm add-flow -l link -a attribute=value` *`flow`*.

```
# dladm create-vnic -l stub0 vnic3
# flowadm add-flow -l vnic3 -a transport=tcp,local_port=80 http1
```



To create a flow, you first create a new VNIC. You then select the attribute that you want to base the flow on, and then determine how you want to customize the flow's use of resources by selecting the bandwidth and priority settings for the network resource.

Next, you add the VNIC as a flow by using the `flowadm add-flow` command followed by the `-l` option, the link name, the `-a` option that specifies the attribute value, the attribute value, and the flow name.

In the example in the slide, you have created a new VNIC, `vnic3`, to add as a flow. The flow is based on the TCP transport protocol, which is the attribute. You have defined transport as `tcp` and assigned it to port `80`. The name of the flow is `http1`.

# Displaying Flow Controls

To display the flow controls that are currently configured in the system, use `flowadm show-flow`.

```
# flowadm show-flow
FLOW    LINK     PROTO LADDR          LPORT RADDR              RPORT DSFLD
http1   vnic3    tcp   --             80    --                 --    --
```



To display the flow controls that are currently configured in the system, use the `flowadm show-flow` command, as shown in the example in the slide. As you can see, there is only one flow that is currently configured in the system, and that is the flow that you just created.

# Creating Flows and Selecting Flow Properties

- Flows are created according to attributes.
- Attributes are classifications that are used to organize network packets into a flow.
- Flows use properties to control resources:
  - **`maxbw`:** Maximum amount of a link's bandwidth that packets identified with this flow can use
  - **`priority`:** Priority given to the packets in a flow:
    - Options: `high`, `medium`, or `low`
    - Default: `medium`



Flows are created according to the attribute that you determined for each flow. An attribute is a classification that you use to organize network packets into a flow. For example, an IP address or a transport protocol, such as TCP, can be used as an attribute. When you create a flow, you identify an attribute as well as a name for the flow.

Flows also have properties that are used to control resources. Currently, there are only two flow properties that can be set:

- **`maxbw`:** The maximum amount of a link's bandwidth that packets identified with this flow can use. The value you set must be within the allowed range of values for the link's bandwidth.
- **`priority`:** The priority given to the packets in this flow. The possible values are `high`, `medium`, and `low`; `medium` is the default value.

# Setting Flow Properties

To set a flow property, use `flowadm set-flowprop -p property=value` *`flow`*.

```
# flowadm set-flowprop -p maxbw=100M http1
```



To set a flow property, use the `flowadm set-flowprop` command followed by the `-p` option to specify the properties that you want to use to control the resources, property values, and flow name.

In the example in the slide, the maximum bandwidth is set to 100 MB per second.

**Note:** The value that you set for the bandwidth must be within the allowed range of values for the link's bandwidth. To display the possible range of values for a link's bandwidth, check the `POSSIBLE` field in the output that is generated by the following command: `dladm show-linkprop -p maxbw` *`link`*.

# Displaying Flow Control Properties

To display a flow's control properties, use `flowadm show-flowprop flow`.

```
# flowadm show-flowprop http1
FLOW      PROPERTY      PERM VALUE     DEFAULT    POSSIBLE
http1     maxbw         rw   100       --         --
http1     priority      rw   medium    medium     low,medium,high
http1     hwflow        r-   off       --         on,off
```



If you want to see the control properties that a flow has, you can do so by using the `flowadm show-flowprop` command, as shown in the example in the slide.

# Setting a Priority Property

To set a link property, use `dladm set-linkprop -p property=high` *`vnic`*.

```
# dladm set-linkprop -p priority=high vnic1
```

To view the priority property for a link, use `dladm show-linkproperty -p priority` *`vnic`*.

```
# dladm show-linkprop -p priority vnic1
LINK     PROPERTY     PERM VALUE   EFFECTIVE   DEFAULT   POSSIBLE
vnic1    priority     rw   high    high        medium    low,medium,high
```

ORACLE



After you set the flow properties, you can also set a priority property on the link. To do so, use the `dladm set-linkprop` command followed by the `-p` option to specify the priority value, the priority value itself, and the name of the VNIC, as shown in the first example in the slide. The possible priority values are low, medium, and high. In this example, the link priority for `vnic1` is set to high.

To view the priority property for a link, you can use the `dladm show-linkprop` command followed by the `-p` priority subcommand and the name of the VNIC, as shown in the second example.

# Quiz

Which two properties do flows use to control resources?

1. `speed` and `mtu`
2. `maxbw` and `priority`
3. `flowctrl` and `threshold`



**Answer: b**

# Practices 7-5 Overview: Managing the Virtual Network Data Flow

In this practice, you manage resources on the virtual network by using data flows.



Each practice should take about 15 minutes to complete.

# Summary

In this lesson, you should have learned how to:

- Describe some of the basic networking concepts
- Administer a datalink configuration
- Administer a network interface
- Administer a profile-based network configuration
- Configure a virtual network
- Verify the network operations
- Manage resources on the network

# 8

# Administering Oracle Solaris Zones

# Objectives

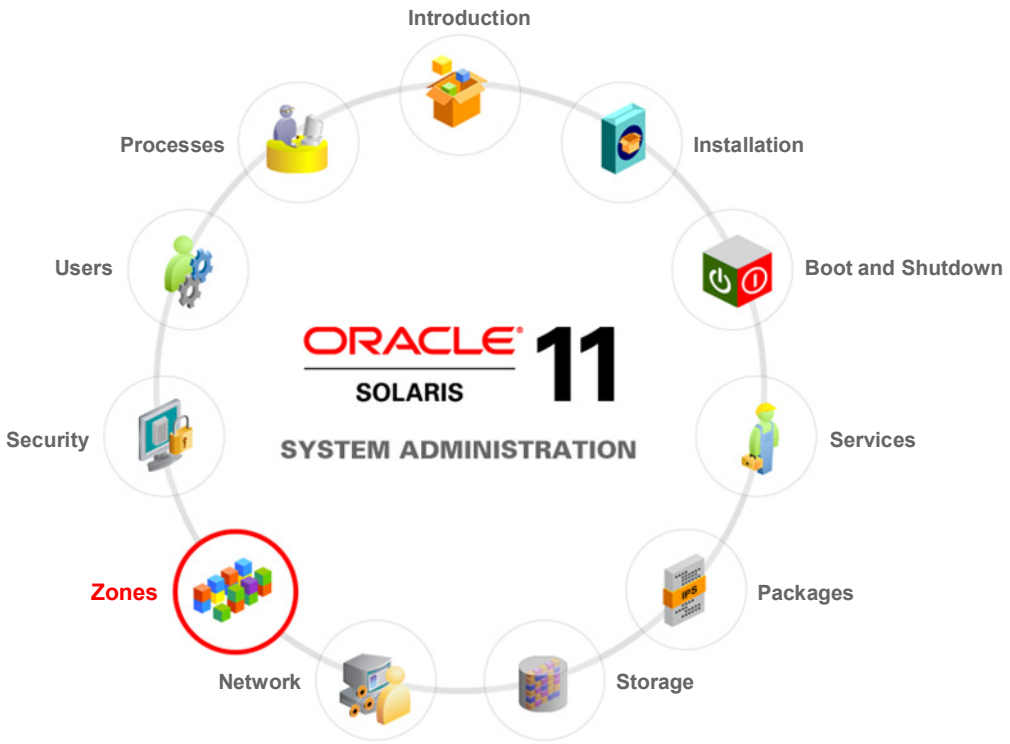
After completing this lesson, you should be able to:

- Explain the fundamentals of Oracle Solaris zones
- Configure an Oracle Solaris zone
- Determine an Oracle Solaris zone configuration



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 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 as to where you are in the job workflow. You have successfully installed the operating system, tested the system's boot and shutdown functionality as well as SMF services, and set up and administered the package repository, data storage environment, and network. Now, you will be introduced to zones virtualization. In today's enterprise data center, 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

- **Introducing Oracle Solaris Zones**
- Configuring an Oracle Solaris Zone
- Determining an Oracle Solaris Zone Configuration



In this lesson, you are introduced to the Oracle Solaris 11 zones virtualization technology. This lesson also teaches you how to configure a zone.

# Oracle Solaris 11 Virtualization Technologies

- Virtualization technologies provide solutions to constantly changing business conditions.
- Data centers are using virtualization technologies to:
  - Consolidate applications and data onto fewer servers
  - Provide better flexibility for managing workloads
  - Support legacy applications on newer systems
  - Provision systems faster
  - Overcome scalability constraints
- The Oracle Solaris 11 virtualization technologies include:
  - Server virtualization
  - Desktop virtualization
  - Integrated solutions

ORACLE





Business depends on the applications and services that are provided by corporate data centers to cope with constantly changing business conditions. As users demand greater functionality, applications are becoming increasingly sophisticated. These changes are in turn placing burdens on the underlying computing infrastructure. To compensate, organizations spend a significant portion of the IT budget on capacity expansion to build on existing infrastructure and meet service-level agreements.

Over time, the influx of servers results in a sprawling, complex network of systems that consume valuable data center floor space, create excessive power and cooling demands, and are costly and difficult to manage. Therefore, your company may have been concerned about the increasing cost and complexity of managing the numerous systems in its data center, and may want to consolidate many of its applications onto fewer, more scalable servers by using the virtualization technologies.

**Solutions Through Virtualization Technologies**

Virtualization provides the ability to get more work done with fewer resources. Virtualization is fast becoming a necessity. Data centers are using virtualization technologies to:

- Consolidate applications and data onto fewer servers
- Gain the ability to move workloads to systems with available resources on an as-needed basis

- Support legacy applications on newer systems
- Provision systems faster
- Overcome scalability constraints

Now storage and desktop virtualization mechanisms are taking virtualization to a new level, helping to optimize the entire data center infrastructure. Although hardware platform advances have made it possible to deliver significant capacity and performance improvements every 12 to 18 months, virtualization has its challenges. The ability to capitalize on Moore's Law and take advantage of greater processor and thread density in systems allows more virtual environments to be placed on a server—and each one must be maintained. In addition, virtualization density can increase application licensing costs and introduce performance overhead and security challenges if not performed well. An integrated virtualization strategy that uses the right technology in the right place is needed to optimize the data center and gain greater efficiency and improved flexibility at less cost.

# Server Virtualization

- Oracle Solaris Zones
  - Provides isolated runtime environments for individual applications by using flexible, software-defined boundaries
- Oracle VM Server for SPARC
  - Is built for Oracle servers with chip multithreading (CMT) technology
  - Is tightly integrated with the hardware
  - Reduces the overhead typically associated with software-based solutions
- Dynamic Domains
  - Are available on Oracle's Sun SPARC Enterprise M-Series servers
  - Divide a single machine into multiple electrically isolated partitions for efficient workload isolation

ORACLE



**Oracle Solaris Zones**

An integral part of the Oracle Solaris 11 operating system, Oracle Solaris Zones provision many secure, isolated runtime environments for individual applications by using flexible, software-defined boundaries. All containers run under a single operating system kernel, enabling fine-grained control over the rights and resources within a consolidated server without increasing the number of OS instances to manage. Oracle Solaris 10 applications and their environments can run in zones on Oracle Solaris 11, giving organizations access to the latest hardware and OS advancements without impacting investments in applications. In addition, applications can be managed independently of each other. Companies can place one application in each virtual server to maintain isolation, if desired, while simultaneously sharing hardware resources.

### Oracle VM Server for SPARC

Built for Oracle servers with chip multithreading (CMT) technology, Oracle VM Server for SPARC (previously called Sun Logical Domains) provides a full virtual machine that runs an independent operating system instance and contains a wide range of virtualized devices. A hypervisor that largely resides in a chip on the server is tightly integrated with the hardware, enabling virtual machines to take advantage of underlying system advancements and reduce the overhead typically associated with software-based solutions. Unlike solutions from other vendors that do not permit add-on networking or cryptographic devices to be partitioned, shared, or abstracted, Oracle VM Server for SPARC supports virtualized CPU, memory, storage, I/O, console, cryptographic devices, and redundant I/O paths, to make maximum use of platform resources.

### Dynamic Domains

Available on Oracle's Sun SPARC Enterprise M-Series servers, the Dynamic Domains technology enables a single system to be divided into multiple electrically isolated partitions for efficient workload isolation. Each domain runs its own instance of Oracle Solaris 11 (or different versions of the operating system) on dedicated hardware. A high-performance system, network, and I/O architecture eliminates overhead, and delivers bare-metal performance to applications. Hardware and software failures are contained within a domain, thereby increasing availability and providing a reliable, secure platform for running multiple applications simultaneously. These hard partitions also support the physical insertion or removal of system boards from a running domain without stopping the server or operating system.

# Desktop Virtualization

- Oracle Secure Global Desktop Software
  - Provides secure access to centralized, server-hosted Windows, UNIX, mainframe, and midrange applications from a variety of clients, including Windows PCs, Mac OS X systems, Oracle Solaris workstations, Linux PCs, thin clients, and more
- Oracle VM VirtualBox
  - Is an open-source solution that allows systems to run multiple environments at the same time to get the most flexibility and utilization

ORACLE



**Oracle Secure Global Desktop Software**

This software delivers secure access to centralized, server-hosted Windows, UNIX, mainframe, and midrange applications from a variety of clients, including Windows PCs, Mac OS X systems, Oracle Solaris workstations, Linux PCs, thin clients, and more. Access to full-screen desktop environments is provided, allowing administrators to use a single solution to provide access to server-based applications and server-hosted desktop environments.

**Oracle VM VirtualBox**

Supporting an extensive range of host and guest operating systems, the open-source Oracle VM VirtualBox solution lets client systems run multiple environments at the same time to get the most flexibility and utilization out of systems. It provides high-performance support for a large number of virtual appliances that are available in Open Virtualization Format (OVF), multiplatform application development and testing, 2D and 3D graphics acceleration, as well as the ability to teleport a running virtual machine between hosts without interruption.

# Integrated Solutions

Oracle Enterprise Manager provides a comprehensive management solution for:

- Managing virtual machines
- Operating systems
- Software



## Managing Virtual Environments

### Oracle Enterprise Manager

Oracle Enterprise Manager provides a comprehensive management solution for managing virtual machines, and the operating systems and software inside them, from a single product. With the Oracle VM Management Pack, Oracle Enterprise Manager can perform end-to-end monitoring, configuration management, and life cycle automation of virtual machines to ensure that you capture and maximize the benefits of virtualization.

# Oracle Solaris 11 Zones 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
- Support installing and running Oracle Solaris Zones on shared storage

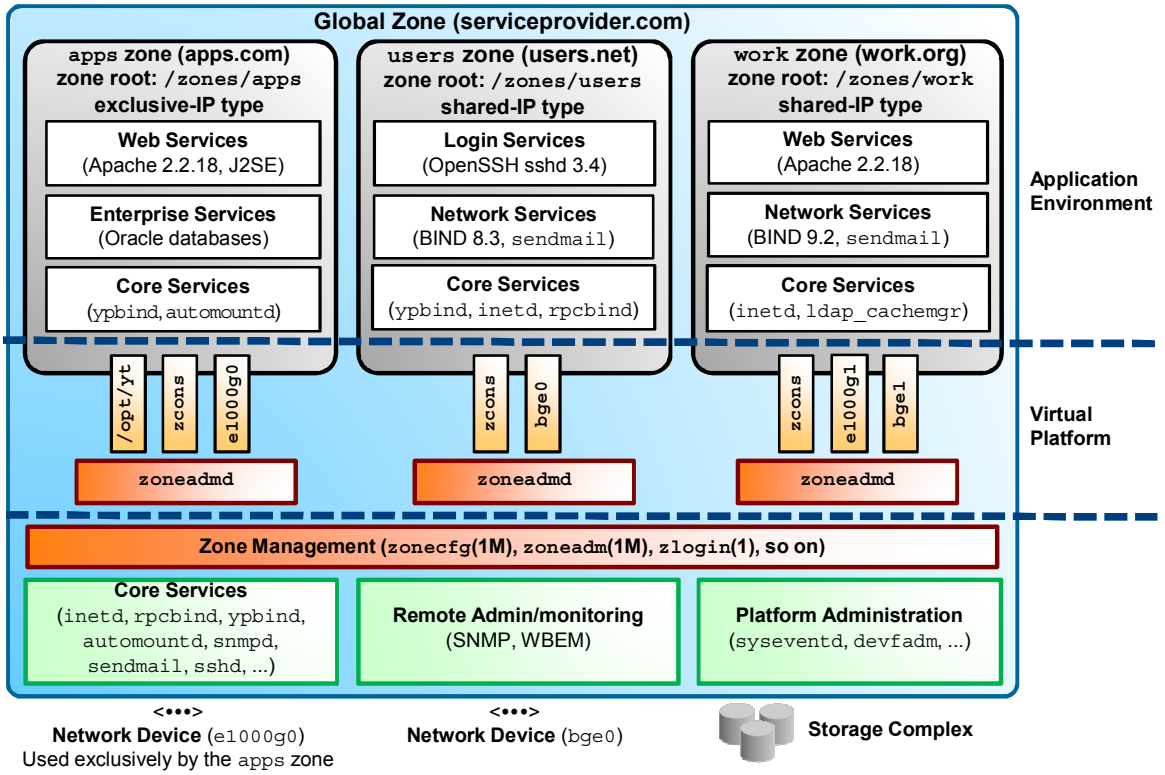


Oracle Solaris 11 Zones provide an isolated and secure environment for running applications. Zones are virtualized operating system environments, each created within a single instance of the Oracle Solaris operating system. When you create a zone, you produce an application execution environment in which 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 that is running with root role 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.

To meet business requirements, your Oracle Solaris 11 implementation team decides to implement the Oracle Solaris 11 Zones technology. As a system administrator, you should put together an extensive implementation plan for the Oracle Solaris zones implementation 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 should include how many zones you want to configure based on business application needs, as well as how to allocate system resources to each zone.

# When to Use Zones





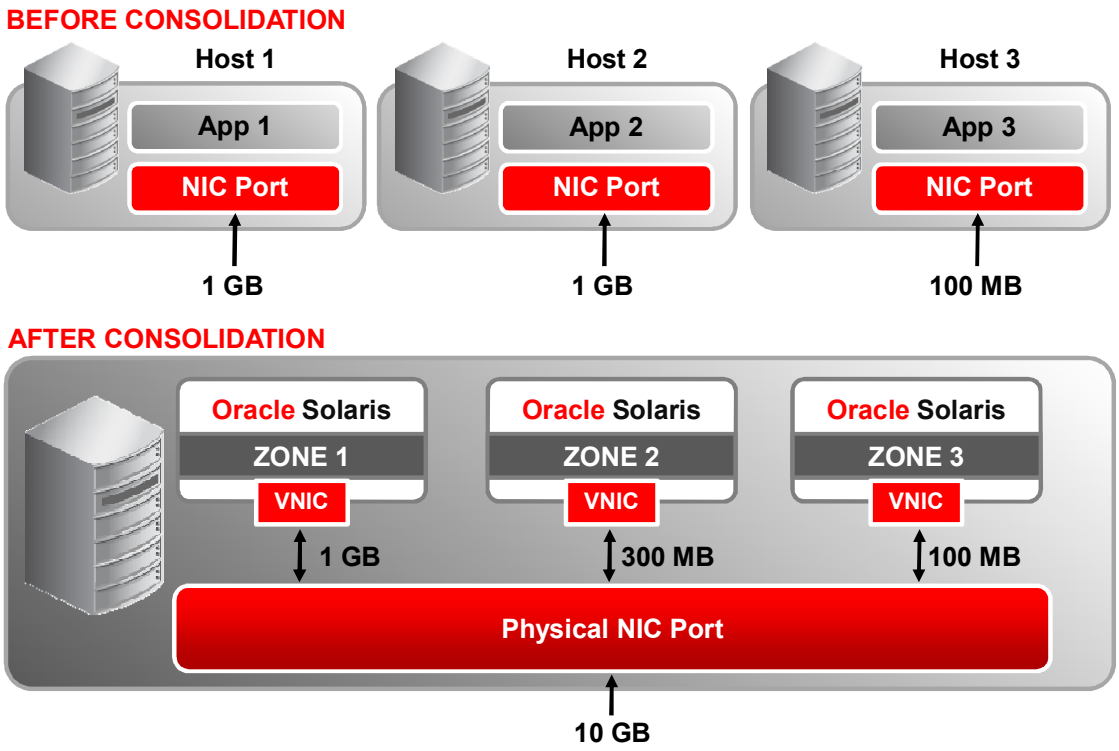

Zones are ideal for environments that consolidate a number of applications on a single server. The cost and complexity of managing numerous machines make it advantageous to consolidate several applications on larger, more scalable servers.

The figure in the slide illustrates the zones server consolidation example. It shows a system with three zones. Each of the zones—`apps`, `users`, and `work`—is running a workload that is 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.

Zones enable more efficient resource utilization on your system. Dynamic resource reallocation permits unused resources to be shifted to other zones as needed. Fault and security isolation means 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.

Zones allow you to delegate some administrative functions while maintaining overall system security.

# Network Virtualization with Zones






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 that are 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.

# 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 following:
  - Total resource requirement of the application software that is 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 requirement of the application software that is 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.

# Zone Types

- A 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, which has a dual function. It is both the default zone for the system and the zone that is used for system-wide 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 each other. 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 the several zones that are running within a system, or allocated on a per-zone basis by using the resource management features available in the operating system.

# Characteristics of the Global Zone and Non-Global Zones

| Global Zone | Non-Global Zone |
|---|---|
| Is assigned ID `0` by the system | Is assigned a zone ID by the system when the zone is booted |
| Provides a single instance of the Oracle Solaris kernel that is bootable and running on the system | Shares operations under the Oracle Solaris kernel that is booted from the global zone |
| Contains a complete installation of the Oracle Solaris system software packages | Contains an installed subset of the complete Oracle Solaris operating system software packages |
| Can contain additional software packages or additional software, directories, files, and other data that are not installed through packages | Can contain additional software, directories, files, and other data created on the non-global zone that are not installed through packages; can also contain additional installed software packages |

ORACLE

# Characteristics of the Global Zone and Non-Global Zones

| Global Zone | Non-Global Zone |
|---|---|
| Provides a complete and consistent product database that contains information about all the software components installed in the global zone | Provides a complete and consistent product database that contains information about all the software components installed in the zone |
| Holds configuration information that is specific only to the global zone, such as the global zone host name and file system table | Has configuration information that is specific only to that non-global zone, such as the non-global zone host name and file system table |
| Is the only zone that is aware of all device file systems, and non-global zones along with their configurations | Is not aware of the existence of any other zones |
| Is the only zone from which a non-global zone can be configured, installed, managed, or uninstalled | Cannot install, manage, or uninstall other zones, including itself |

ORACLE

# Branded Zones

- Provide an extension of Oracle Solaris zones
- Contain operating environments that are 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 in 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 that is 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.

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

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

# Immutable (Read-Only) Zone

- A zone with a read-only root is called an Immutable Zone.
- It preserves a zone's integrity by using a read-only root file system.
- It blocks modifications to system binaries or system configurations.
- The `file-mac-profile` property:
  - Is used to configure a read-only root
  - Is set by using the `zonecfg` utility
  - Is not set by default



A zone with a read-only zone root is called an *Immutable Zone*. An Oracle Solaris Immutable Zone preserves the zone's configuration by implementing read-only root file systems for non-global zones. This zone extends the zone's secure runtime boundary by adding additional restrictions to the runtime environment. Unless performed as specific maintenance operations, modifications to system binaries or system configurations are blocked.

The mandatory write access control (MWAC) kernel policy is used to enforce file system write privilege through the `zonecfg file-mac-profile` property. Because the global zone is not subject to the MWAC policy, it can write to a non-global zone's file system for installation, image updates, and maintenance. The MWAC policy is downloaded when the zone enters the ready state. It is enabled at zone boot. To perform post-installation assembly and configuration, a temporary writable root file system boot sequence is used. Modifications to the zone's MWAC configuration take effect only after a zone restart.

For more information about the `zonecfg file-mac-profile` property, refer to http://docs.oracle.com/cd/E36784_01/html/E37628/zmfp.html#scrolltoc.

## Immutable Global Zones

Starting with Oracle Solaris 11.2 release, immutable global zones support has been added to extend the immutable zone implementation to the global zone. If a system is configured to have an immutable global zone, files in the root file system are read-only. A Trusted Path login is provided to allow maintenance tasks, such as performing system updates.

Configuration of the global zone is performed through the `zonecfg` command `file-mac-profile`. If the system uses DHCP to set network interfaces, `flexible-configuration` must be selected.

The `rpool` dataset is restricted but you can add an unrestricted sub-data-set by using `add dataset`. An immutable global zone can run zones only in unrestricted datasets. All the children of an unrestricted dataset are also unrestricted. After committing the zone configuration, the `zonecfg boot` information is written and the boot archive is updated. Reboot the system to boot with an immutable global zone.

**Note:** You can maintain the global zone by using the Trusted Path access. Trusted Path is available only on the console, so ensure that the console is accessible through the ILOM, a serial connection, or through the graphical console. After a system is configured as an immutable global zone, use the break sequence on the console to access the Trusted Path console. Log in and assume the root role. When a package update is performed, the first boot of the immutable global zone is read-write. The system needs these permissions to perform the required self-assembly steps. When the self-assembly steps have been performed, the system reboots. In this second boot, the system becomes immutable again.

To understand how to enable immutable global zones, refer to the notes section of the slide titled “Administering Immutable Global Zones” in the later part of this lesson.

# Zone Network Interfaces

- 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.

ORACLE



Basic communication between zones is accomplished by giving each zone Internet Protocol (IP) network connectivity. An application that is 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.

It is now time to test the zones functionality within Oracle Solaris 11. Your task is to configure a zone and review its configuration. In the topics that follow, you 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

- Getting Started with Oracle Solaris Zones
- **Configuring an Oracle Solaris Zone**
- Determining an Oracle Solaris Zone Configuration

# Planning for Non-Global Zone Configuration

Before configuring your non-global zone, its important to do the following:

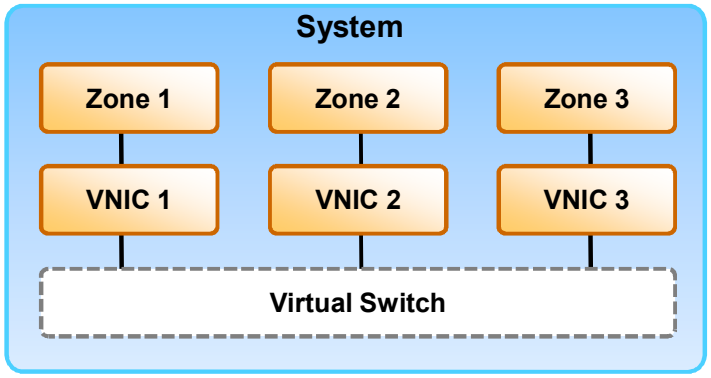
- Evaluate the applications running on your system to determine the applications that you want to run in a zone.
- Assess the availability of disk space to hold the files that are unique in the zone.
- Decide the naming convention you want to follow for your zone.
- Determine the zone path.
- Determine the type of zone (shared or exclusive) you want to set up. Note that exclusive-IP is the default type for zones.
- Determine the file system that you want to mount in the zone.
- Determine the network interface that should be made available in the zone.

# Planning for a Virtual Network and Zones

- Identify the virtual network configuration:
  - Virtual switch or etherstub
  - Number of VNICs and name assignments
- Identify the zone configuration:
  - Number of zones
  - Zone configuration details
  - Zone and VNIC assignments

# Configuring Zones by Using VNICs

1. Create the virtual switch or etherstub.
2. Create the VNICs.
3. Configure the zones to use the VNICs.






To configure zones to use a virtual network, perform the following steps:

1. Create your virtual network by creating the virtual switch or etherstub.
2. Create the VNICs over the switch or etherstub.
3. After you have the VNICs created, configure your zones to use the VNICs.

**Note:** Starting from Oracle Solaris 11.2 release, you can create VNICs directly in a non-global zone from a global zone by specifying the link as zone or link. This method creates the VNIC directly in the namespace of the non-global zone. The `-t` option is used to specify that the VNIC is temporary. Temporary VNICs persist until the next reboot of the zone. The global zone and other non-global zones can also have VNICs with the same name. VNICs can be created only temporarily by using this method. In addition to temporarily creating VNICs, you can also temporarily create VLANs and IP over InfiniBand (IPoIB) partitions. Refer to the `dladm`(1M) man page for complete instructions.

The following example shows how to create a VNIC named `vnic1` in a non-global zone from the global zone.

```
globalzone# zoneadm -z zone1 boot
globalzone# dladm create-vnic -t -l net0 zone1/vnic1
globalzone# dladm show-link -Z
LINK         ZONE      CLASS   MTU    STATE   OVER
net0         global    phys    1500   up      --
zone1/vnic1 zone1      vnic    1500   down    net0
```

The following example shows the output of the `dladm show-link` command from `zone1`.

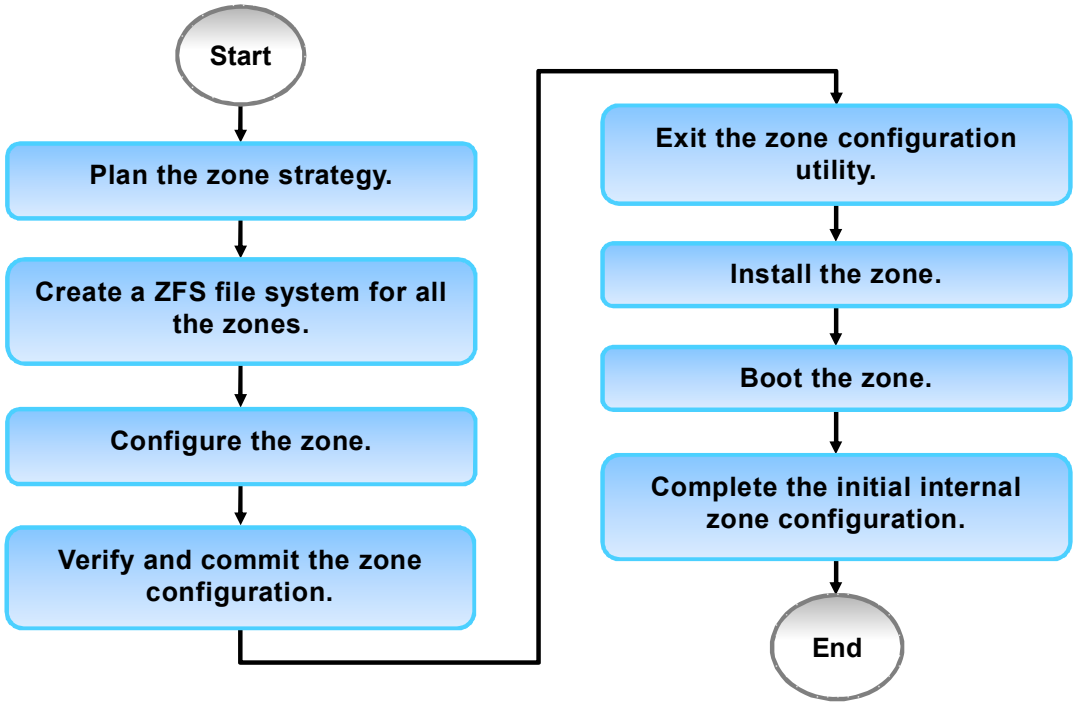
```
zone1# dladm show-link
LINK         CLASS     MTU     STATE   OVER
vnic1         vnic     1500    down    ?
```

The following example shows how to create a VLAN named `vlan3` in a non-global zone from a global zone.

```
globalzone# dladm create-vlan -t -l net0 -v 3 zone1/vlan3
```

**Note:** The `-v` option specifies the VLAN ID of the VLAN over the Ethernet link.

# Non-Global Zone Configuration Process: Overview




ORACLE





It is important to understand the steps of configuring a zone. The process of configuring a non-global zone begins with planning the zone strategy as listed in the previous slide. You will be studying each step in detail in this section.

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 in the private file systems of the zone.

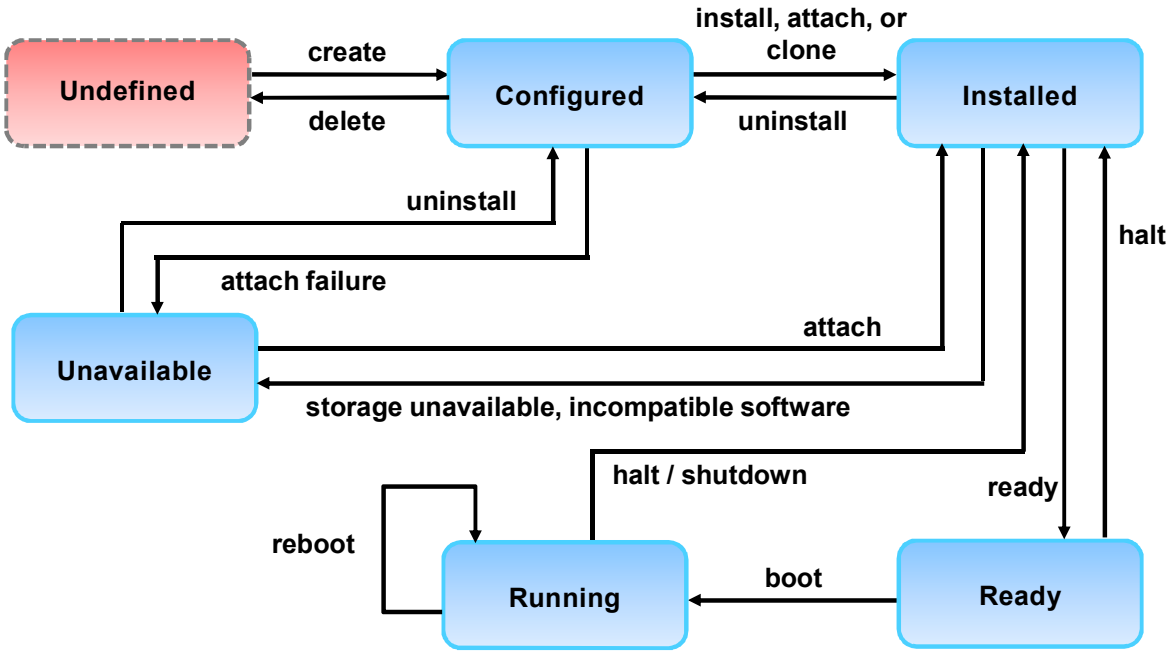
Starting with Oracle Solaris 11.2, you can rename an Oracle Solaris Zone or Oracle Solaris 10 Zone in the configured or installed state by using the `zoneadm rename` command. However, before the `rename` subcommand is used, the installed zone must be halted. In order to rename an already configured zone, use:

```
# zoneadm -z myzone rename newzone
```

On renaming the zone, you can rename the zone path accordingly by using the `zoneadm`(1M) `move` subcommand. This will result in renaming the corresponding ZFS datasets that make up the zone path, as well as renaming the `zonepath` property in the zone configuration itself. Example:

```
# zoneadm -z newzone move /path/newzone
```

# Non-Global Zone States






Before you start configuring a zone, 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 installation process will stop in this state.

- **Unavailable:** Indicates that the zone is installed, but cannot be verified, made ready, booted, attached, or moved. A zone enters the unavailable state at the following times:
  - When the zone's storage is unavailable and `svc:/system/zones:default` begins, such as during system boot
  - When the zone's storage is unavailable
  - When archive-based installations fail after successful archive extraction
  - When the zone's software is incompatible with the global zone's software
- **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.

# Planning the Zone Strategy

- Virtual network configuration: Two VNICs `vnic1` and `vnic2`
- Two zones: `hrzone` and `itzone`
- Zone paths: `/zones/hrzone` and `/zones/itzone`
- IP type: Exclusive-IP
- VNIC to zone association: `vnic1` for `hrzone`; `vnic2` for `itzone`



Suppose that you have been tasked with creating two zones over a virtual network. Your strategy is to create the virtual network first, which you have already done, and then create the zones. As part of your zones configuration planning, you have identified the following information:

- **Zone names:** The zone name must be unique. You use the names `hrzone` and `itzone` to create your zones.
- **Zone paths:** Each zone requires a path to its root directory that is relative to the global zone's root directory. You are creating a file system called `zones` as part of `rpool`, and then you create two other file systems under zones, one to contain `hrzone` and one to contain `itzone`. The two zone paths should look like the following, respectively: `/zones/hrzone` and `/zones/itzone`.
- **IP type:** To use VNICs, a zone must be configured as an exclusive IP zone.
- **Specific VNIC to be associated with the zone:** You use `vnic1` for `hrzone` and `vnic2` for `itzone`.

Now that you know what your zone strategy is, your next step is to create the ZFS file system structure for your zones.

# Creating a ZFS File System for Zones in `rpool`

To create a ZFS file system for zones in `rpool`, use the following command:

```
# zfs create -o mountpoint=/zones rpool/zones
```

To verify that the file system exists and that it has been mounted, use the following command:

```
# zfs list rpool/zones
NAME           USED  AVAIL  REFER  MOUNTPOINT
rpool/zones     31K  22.6G    31K  /zones
```

ORACLE



The first ZFS file system that you want to create in `rpool` is a file system that will contain all the individual zones' file systems. Typically, this file system is called `zones`. To create this file system, use the `zfs create` command with the `-o` option (to specify the `mountpoint` property), followed by the `mountpoint` property value (`mountpoint=/zones`) and the file system name (`rpool/zones`), as shown in the first example in the slide.

You can then verify that the file system has been created and mounted by using the `zfs list` command followed by the file system name, as shown in the second example.

You create the zone-specific file system during zone configuration.

# Configuring the Zone

To configure a zone, use `zonecfg -z zonename`.

```
# zonecfg -z hrzone
hrzone: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:hrzone> create
create: Using system default template 'SYSdefault'
zonecfg:hrzone> set zonepath=/zones/hrzone
zonecfg:hrzone> set autoboot=true
zonecfg:hrzone> add net
zonecfg:hrzone:net> set physical=vnic1
zonecfg:hrzone:net> end
zonecfg:hrzone> verify
zonecfg:hrzone> commit
zonecfg:hrzone> exit
```



1. The `zonecfg` command is used to create the zone configuration. You must be in the `root` role or have the appropriate rights profile to configure a zone. To perform the configuration, use the `zonecfg` command with the `-z` option to specify the name of the zone followed by the zone name, as shown in the example in the slide. If this is the first time that you have configured this zone, you see the following system message: "`No such zone configured. Use 'create' to begin configuring a new zone.`"
2. Enter `create`. This enables you to create the new zone configuration by setting specific properties, such as the zone path, the IP type, and the network type.
3. Set the zone path by using the `set zonepath` command followed by the zone path, for example, `/zones/hrzone`. The zone must reside on a ZFS dataset. The ZFS dataset is created automatically when the zone is installed or attached. If a ZFS dataset cannot be created, the zone will not install or attach.
   **Note:** If the parent directory of the zone path exists, it must be the mount point of a mounted dataset as shown in the previous slide.
4. Set `autoboot` to `true` by using `set autoboot=true`. This setting indicates that the zone is automatically booted when the global zone is booted. The default value is `false`.
   **Note:** For the zones to autoboot, the zones service `svc:/system/zones:default` must also be enabled. This service is enabled by default.

5. In the default zone template, `SYSdefault`, the IP type is set to `exclusive` by default. To modify the default IP type and set it to `shared`, use the `set ip-type=shared` command.
6. Specify that you want to add a network interface to the zone by using the `add net` command. Notice in the example that the `zonecfg` prompt for the zone that you are creating has been modified to include "`net`": `zonecfg:hrzone:net`. Here, you can set the network `physical` property to specify the VNIC that you want this zone to use by using `set physical=` followed by the VNIC name (for example, `set physical=vnic1`).
7. To stop working on the zone's network configuration, enter the `end` command. You have completed the zone configuration.
8. Use the `verify` command, as shown in the example in the slide, after you complete your zone configuration to verify that all the required information is present. If all the required information is not present, the system will notify you, in which case you will need to review your configuration to determine what is missing. If no messages are displayed, you can continue to the next step, which is to commit the configuration. The `commit` command takes the configuration from memory and puts it into permanent storage.
9. After committing the zone configuration, you can exit the session by using the `exit` command.

For information about zone components and resources, refer to http://docs.oracle.com/cd/E36784_01/html/E36848/z.config.ov-3.html#scrolltoc, and the `zoneadm`(1M) and `zonecfg`(1M) man pages.

An example of how to create a zone with the `shared` IP type is as follows:

```
# zonecfg -z hrzone
hrzone: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:hrzone> create
create: Using system default template 'SYSdefault'
zonecfg:hrzone> set zonepath=/zones/hrzone
zonecfg:hrzone> set autoboot=true
zonecfg:hrzone> set ip-type=shared
zonecfg:hrzone> remove anet
zonecfg:hrzone> add net
zonecfg:hrzone:net> set physical=net0
zonecfg:hrzone:net> set address=192.168.0.10
zonecfg:hrzone:net> end
zonecfg:hrzone> info
zonename: hrzone
zonepath: /zones/hrzone
brand: solaris
autoboot: true
bootargs:
file-mac-profile:
pool:
limitpriv:
scheduling-class:
ip-type: shared
```

```
 hostid:
fs-allowed:
net:
        address: 192.168.0.10
        allowed-address not specified
        configure-allowed-address: true
        physical: net0
        defrouter not specified
zonecfg:hrzone> verify
zonecfg:hrzone> commit
zonecfg:hrzone> exit
```

Notice in the example that the `zonecfg` prompt for the zone that you are creating has been modified to include remove `anet` and `add net`. The `anet` resource represents the automatic creation of a network resource for an exclusive-IP zone. When `zonecfg` creates a zone using the default `SYSdefault` template, an `anet` resource with several properties is automatically included in the zone configuration. Because you are creating a shared-IP zone, you need to remove the `anet` resource.

**Note:** Starting with Oracle Solaris 11.2, Oracle Solaris Zones have also been enhanced to take advantage of zone template properties. This allows simplified zone configuration. Default configuration values are populated when zones are created, cloned, and migrated.

Starting with Oracle Solaris 11.1, you can configure, install, and run Oracle Solaris Zones that are hosted directly on arbitrary storage device objects, such as Fibre Channel or iSCSI targets. You can specify and configure the path to the device directly by using the `zonecfg`(1M) command. The zone is then automatically encapsulated into its own `zpool`. The aim is to simplify deployment, administration, and migration of Oracle Solaris Zones.

Starting with Oracle Solaris 11.2, you can create datalinks in non-global zones from the global zone. This feature allows administrators to dynamically create VNICs, VLANs, and IP-over-InfiniBand partitions directly in the non-global zone's namespace from the global zone. Link names are specified as "`<zonename>/<linkname>`" and the links are created directly in the specified non-global zone.

The following example shows how to create a VNIC `v1` in non-global zones `zone1` and `zone2` from the global zone. `zone1/net0` and `zone2/net0` are automatically created VNICs for `zone1` and `zone2`, respectively.

```
# dladm create-vnic -t -l net1 zone1/v1
# dladm create-vnic -t -l net1 zone2/v1
# dladm show-link -Z
LINK         ZONE          CLASS       MTU    STATE      OVER
net1         global        phys        1500   unknown    --
net0         global        phys        1500   up         --
zone1/net0   zone1         vnic        1500   up         net0
zone2/net0   zone2         vnic        1500   up         net0
zone1/v1     zone1         vnic        1500   up         net1
zone2/v1     zone2         vnic        1500   up         net1
```

# Verifying That a Zone Is in `configured` State

To list all configured and running zones in the system, use `zoneadm list -cv`.

```
# zoneadm list -cv
  ID NAME STATUS           PATH                BRAND     IP
   0 global   running      /                   solaris   shared
   - hrzone   configured   /zones/hrzone       solaris   excl
   - itzone   configured   /zones/itzone       solaris   excl
```



You are now ready to install the zone. But before you do that, it is a good idea to confirm that the zone is in the `configured` state. You can use the `zoneadm list -cv` command to see all configured and running zones in a system, as shown in the example in the slide. Both the zones that you have created, `hrzone` and `itzone`, have a status of `configured`.

This slide shows the output of `zoneadm list -cv`. In this system, three zones are running: global zone, `hrzone`, and `itzone`. 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 paths for the non-global zones are `/zones/hrzone` and `/zones/itzone`. The brand for both zones is `solaris`, which is the default zone brand in the Oracle Solaris 11 release. Both the global zone and `hrzone` are set up as shared-IP, whereas `itzone` is set up as exclusive-IP.

You can now install the configured zones.

# Installing the Zone

To install a zone, use `zoneadm -z <zone_name> install`.

```
# zoneadm -z hrzone install
The following ZFS file system(s) have been created:
    rpool/zones/hrzone
Progress being logged to /var/log/zones/zoneadm.20131037T065334Z.hrzone.install
       Image: Preparing at /zones/hrzone/root.

AI Manifest: /tmp/manifest.xml.fXai_f
SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
  Zonename: hrzone
Installation: Starting ...

           Creating IPS image
Startup linked: 1/1 done
           Installing packages from:
               solaris
                  origin: http://s11-server1.mydomain.com/
DOWNLOAD                PKGS         FILES       XFER (MB)     SPEED
Completed            255/255   51394/51394     341.6/341.6    2.8M/s
...
...
Log saved in non-global zone as
/zones/hrzone/root/var/log/zones/zoneadm.20131027T235442Z.hrzone.install
```



After you have completed configuring the zone, you are ready to install it. To install a zone, use the `zoneadm -z <zone_name> install` command as shown in the slide example.

The installation process automatically creates a ZFS file system (dataset) for the zone path when the zone is installed. If the file system cannot be created, the zone is not installed. The installation process also verifies the specified publisher, and downloads the zone installation packages from IPS. This process normally takes about three to five minutes per zone.

Observe in the output of the zone installation that SC Profile and AI Manifest are used to install the zone. AI Manifest is the Automated Install Manifest that describes the software and other configuration information that are used to install the zone. This instance represents a zone default AI manifest. A custom manifest can be created and used to define the software and other configuration information that will be used for the zone. This custom manifest can be passed as an option to the `zoneadm` command when the zone is installed.

SC Profile is a system configuration profile. In the default instance, this points to the `/usr/share/auto_install/sc_profiles/enable_sci.xml` profile, which starts an interactive system configuration when the zone is booted.

**Note:** In Oracle Solaris 11, to perform a hands-free configuration, an SC profile (XML file) can be created from an installed zone and provided as an option to the `zoneadm` command when installing another zone. The profile is applied to the zone after the zone is installed and is used to configure the zone.

To create the system configuration profile, you need to first log in to the zone, and then perform the following:

- For an exclusive-IP, use `sysconfig create-profile -o /<path>/sysconf.xml`.
- For a shared-IP, use `sysconfig create-profile -o /<path>/sysconf.xml -g location,identity,naming_services,users`.

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

# Booting the Zone

To list all running and installed zones on the system, use `zoneadm list -iv`.

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

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

```
# zoneadm -z hrzone boot
# zoneadm -z itzone boot
# zoneadm list -v
  ID NAME     STATUS       PATH              BRAND     IP
   0 global   running      /                 solaris   shared
   1 hrzone   running      /zones/hrzone     solaris   excl
   2 itzone   running      /zones/itzone     solaris   excl
```



The next step is to boot the zone. But, before you do that, it is a good idea to confirm whether the zone is in the `installed` state. You can use the `zoneadm list -iv` command to see all the running and installed zones in a system, as shown in the first example in the slide. As you can see, both `hrzone` and `itzone` have a status of `installed`.

You can now boot the installed zones. To boot a zone, use the `zoneadm -z` command followed by the zone name and the `boot` subcommand, as shown in the second example.

To verify that a zone is in the `running` state, you can run the `zoneadm list -v` command, as shown in the second part of the second example. Note that the two non-global zones now have assigned IDs.

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.

# Logging In to a Zone

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

```
# zlogin -C hrzone
[Connected to zone 'hrzone' console]
```



To log in to a zone, use the `zlogin` command followed by the zone name, as shown in the example in the slide. The `zlogin -C` option connects to the zone console. Using the `zlogin` command with the `-C` option starts the SCI Tool if the configuration has not been performed.

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.

# Gathering Information for the System Configuration Tool

- Computer Name: `hrzone`
- DNS Name service: `Do not configure DNS`
- Alternate Name Service: `None`
- Time Zone, Region, and Location: *Use your specific location.*
- Locale Language and Territory: *Use your specific locale.*
- Users, username, and password



After you have logged in to the zone, you need to complete the zone's configuration. Gather the information that is required to complete the zone's configuration. The system configuration profile for the zone utilizes the System Configuration Interactive tool (SCI or `sysconfig`, for short). The SCI tool helps you to specify the default locale and time zone, the zone's `root` password, a naming service, and other aspects of the application environment, to include (but not limited to) the following:

- The computer name of the zone (for example, `hrzone`)
- IP address of the zone, which is based on the IP address of the zone's VNIC
- Netmask of the IP address

Most of the information is supplied by selecting from a list of choices. Typically, the default options are enough unless your system configuration requires otherwise. After you have supplied the required information for the zone, the zone is restarted.

This slide presents a sample of the type of information that you need to complete the system configuration profile.

# Checking the Virtual Network Configuration in a Zone

To display the network interface address information for a zone, log in to the zone, and then use `ipadm show-addr`.

```
# zlogin hrzone
[Connected to zone 'hrzone' pts/2]
Oracle Corporation      SunOS 5.12      11.2     June 2014
root@hrzone:~# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
vnic1/v4          static   ok           192.168.1.100/24
lo0/v6            static   ok           ::1/128
vnic1/v6          addrconf ok           fe80::8:20ff:fe43:7986/10
```



Now you learn how to check the virtual network configuration in a zone. First, you need to log in to the zone. Log in to `hrzone`. To do this, use the `zlogin` command again followed by the zone name, as shown in the example in the slide. After you are logged in, you can use the `ipadm show-addr` command to see the network interface address information for the zone. Here, you can see the IP address assignment of `192.168.1.100` that you made for the `vnic1` network interface while creating the system configuration profile. You can also see the type and state of the interface.

# 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 a 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 in the slide. To exit a zone from a virtual console, use the tilde (`~`) character and a period, as shown in the second example.

# Halting a Zone

To halt a zone, run `zoneadm -z <zone_name> halt`.

```
global# zoneadm -z hrzone 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
 2   itzone    running      /zones/itzone     solaris    excl
 -   hrzone    installed    /zones/hrzone     solaris    excl
```



The `zoneadm halt` command is used to terminate all the processes running in a zone and remove the virtual platform. The zone is then brought back to the installed state. All processes are killed, devices are unconfigured, network interfaces are destroyed, file systems are unmounted, and the kernel data structures are destroyed. The `halt` command does not run any shutdown scripts within the zone.

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 in the slide. 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.

# Shutting Down a Non-Global Zone

To shut down a zone, use `zoneadm -z <zone_name> shutdown`.

```
global# zoneadm -z hrzone shutdown
```



There are various reasons that you might be directed to shut down a zone. For example, another zone needs to complete a database update before the zone that 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, as shown in the example in the slide.

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

Alternatively, for instructions on how to perform the same procedure by using the `zlogin` command, refer to http://docs.oracle.com/cd/E36784_01/html/html/E37628/z.login.task-25.html#scrolltoc.

# Administering Immutable Zones

- Setting a strict Immutable Zone

```
zonecfg:hrzone> set file-mac-profile=strict
```

- Setting a fixed-configuration Immutable Zone

```
zonecfg:itzone> set file-mac-profile=fixed-configuration
```

- Setting a flexible-configuration Immutable Zone

```
zonecfg:userszone> set file-mac-profile=flexible-configuration
```

- Displaying zone properties

```
# zoneadm list -p
0:global:running:/::solaris:shared:-:none
1:hrzone:running:/zones/hrzone:<UUID>:solaris:excl:R:strict
2:itzone:running:/zones/itzone:<UUID>:solaris:excl:R:fixed-configuration
3:userszone:running:/zones/userszone:<UUID>:solaris:shared:R:flexible-
    configuration
```

ORACLE



This slide shows examples of configuring and viewing Immutable Zones.

- **strict:** Permits a read-only file system, and disallows exceptions
  - IPS packages cannot be installed.
  - Persistently enabled SMF services are fixed.
  - SMF manifests cannot be added from the default locations.
  - Logging and auditing configuration files are fixed. Data can be logged only remotely.
- **fixed-configuration:** Permits updates to `/var/*` directories, except directories that contain system configuration components
  - IPS packages, including new packages, cannot be installed.
  - Persistently enabled SMF services are fixed.
  - SMF manifests cannot be added from the default locations.
  - Logging and auditing configuration files can be local. `syslog` and audit configuration are fixed.

- **flexible-configuration:** Permits modification of files in `/etc/*` directories, changes to root's home directory, and updates to `/var/*` directories. This configuration provides the closest functionality to the Oracle Solaris 10 native sparse-root zone.
  - IPS packages, including new packages, cannot be installed.
  - Persistently enabled SMF services are fixed.
  - SMF manifests cannot be added from the default locations.
  - Logging and auditing configuration files can be local. `syslog` and audit configuration can be changed.
- **none:** Is a standard, read/write, non-global zone, with no additional protection beyond the existing zones boundaries. Setting the value to `none` is equivalent to not setting the `file-mac-profile` property.

**Note:** To enable immutable global zones, use:

```
# zonecfg -z global set file-mac-profile=fixed-configuration
```

# Booting Immutable Zones

You can temporarily override the zone restrictions in the read-only root file system by booting the zone with the `-w` option.

```
# zoneadm -z hrzone boot  -w
```



The `zoneadm boot` subcommand provides two options that allow the global zone administrator to manually boot a read-only zone with either a writable root file system or with a transient writable root file system.

**Note:** The zone is in writable mode only until the next reboot occurs.

- `-w` manually boots the zone with a writable root file system.
- `-W` manually boots the zone with a transient writable root file system. The system is rebooted automatically when the self-assembly-complete milestone is reached. The reboot places the zone under the control of the MWAC policy again. This option is permitted when the zone has an MWAC policy of `none`.

# Delegating Zone Administration

Delegate zones administration to different users.

- The `auth` property:
  - `login (solaris.zone.login)`
  - `manage (solaris.zone.manage)`
  - `clone (solaris.zone.clonefrom)`
- The `admin` zone property:

```
zonecfg:zone1> add admin
zonecfg:zone1:admin> set user=oracle
zonecfg:zone1:admin> set auths=login,manage
zonecfg:zone1:admin> end
```



With Oracle Solaris 11, you can delegate common zone administration tasks for specific zones to different administrators by using role-based access control (RBAC). With delegated administration, for each zone, a user or set of users may be identified with permissions to log in, manage, or clone that zone. These specific authorizations that are associated with the `auth` property are interpreted by the appropriate commands running in the global zone to allow access at the correct authorization level to the correct user.

The `admin` zone property defines the username and the authorizations for that user for a given zone (as shown in the example in the slide).

# Quiz

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

a. True
b. False



**Answer: a**

# Quiz

After you have run the `zonecfg -z` *zonename* command, which command would you use to start the configuration of a new zone?

a. `add zone`
b. `begin`
c. `create`
d. `start`



**Answer: c**

# Quiz

To use VNICs, as which IP type must a zone be configured?

a. Shared-IP
b. Exclusive-IP
c. Either shared or exclusive



**Answer: b**

# Quiz

You have created the configuration for a new zone. What is the next step?

a. Boot the new zone.
b. Commit the configuration.
c. Exit the configuration.
d. Verify the configuration.



**Answer: d**

# 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 8-1 Overview: Configuring Zones

This practice covers the following topics:

- Configuring three zones to use VNICs
- Displaying the zone configuration, including the interfaces



This practice should take about 45 minutes to complete.

# Lesson Agenda

- Getting Started with Oracle Solaris Zones
- Configuring an Oracle Solaris Zone
- **Determining an Oracle Solaris Zone Configuration**

# Determining an Oracle Solaris Zone Configuration

- Displaying the status of zones
- Displaying a zone configuration
- Displaying zone network information
- Determining a zone's resource utilization
- Determining a zone's kernel file system statistics

# Displaying the Status of Zones

The `zoneadm list` subcommand helps in verifying the status of all the zones running in the system.

`list` options include the following:

- `-c` displays all the configured zones in the system.
- `-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.

```
# zoneadm list -cv
ID NAME            STATUS      PATH            BRAND    IP
 0 global          running     /               solaris  shared
 2 itzone          running     /zones/itzone   solaris  excl
 - hrzone          installed   /zones/hrzone   solaris  excl
```



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

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

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

You also have the choice of combining options and even using all the three options presented in the slide, 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.

# Displaying a Zone Configuration

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

```
# zonecfg -z finzone info
zonename: finzone
zonepath: /zones/finzone
brand: solaris
autoboot: true
bootargs:
file-mac-profile:
pool:
limitpriv:
scheduling-class:
ip-type: shared
hostid:
fs-allowed:

<output continued on next slide>
```



After you have finished your zone configuration, it is a good practice to review it before you install the zone. To display a zone configuration, use the `zonecfg -z` command followed by the zone name and the `info` subcommand, as shown in the slide. Verify that you have set the zone path, IP type, and network interface properties correctly.

You must be the global administrator in the global zone, or a user with the correct rights profile, to display a zone's configuration.

The first part of the output in the slide displays the zone name (`finzone`), the zone path (`/zones/finzone`), 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 previous slides, 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/finzone
        special: rpool/finzone
        raw not specified
        type: lofs
        options: []
net:
        address: 192.168.0.20
        allowed-address not specified
        configure-allowed-address: true
        physical: net0
        defrouter not specified
rctl:
        name: zone.max-lwps
        value: (priv=privileged,limit=500,action=deny)
```



The next section of the output (shown in the slide) displays the zone's file system information (`fs`), such as the directory location, the zone's network information, such as the IP address (`192.168.0.20`) and NIC (`net0`), the zone attribute information, and the resource control settings (`rctl`).

# 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/zoneadmd-v4   static   ok       127.0.0.1/8
net0/v4           static   ok       192.168.0.100/24
net0/zoneadmd-v4  static   ok       192.168.0.10/24
lo0/v6            static   ok       ::1/128
lo0/zoneadmd-v6   static   ok       ::1/128
net0/v6           addrconf ok       fe80::a00:27ff:fe68:6f2d/10
```



For network interface address information, you can use the `ipadm show-addr` command, as shown in the slide. You may recall this command from when you were verifying the network interface information during the OS installation verification task. As you can see, it is more difficult to directly associate the zone with its IP address in 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
SUMMARYInterval: 3, Duration: 0:00:15
SUMMARY                   Cpus/Online: 1/1   PhysMem: 1023M  VirtMem: 2047M
                     ---CPU----  --PhysMem-- --VirtMem-- --PhysNet--
                ZONE  USED %PART  USED %USED  USED %USED PBYTE %PUSE
             [total]  1.00  100%  658M 64.3%  839M 41.0%  1431 0.00%
            [system]  0.18 18.9%  373M 36.5%  521M 25.4%     -     -
           choczone  0.68 68.8% 44.0M 4.30% 49.6M 2.42%     0 0.00%
              global  0.11 11.0%  133M 13.0%  167M 8.16%  1431 0.00%
                  QA  0.00 0.40% 53.5M 5.23% 50.3M 2.46%     0 0.00%
         grandmazone  0.00 0.81% 53.3M 5.21% 51.4M 2.51%     0 0.00%
...
...
...
(output truncated)
```



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 the 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 in the slide, you have selected one interval with a duration of five 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 to 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 in the slide, you are monitoring 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 is partitioned (that is, if multiple processor sets have been created on the system by dividing the CPU), usage of the partition is displayed.

To the right of CPU usage, the physical and virtual memory usage by the system and by each zone is displayed. Physical memory refers to RAM on the machine, and virtual memory refers to RAM plus system swap space. Memory usage shows how much memory is being used and the percentage of memory being used. To the right of 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 about resource capping, refer to http://docs.oracle.com/cd/E36784_01/html/E36848/gejkz.html#VLZONgejkz.

Now take a closer look at the percentages in each area to see what determinations you can make about resource utilization for this system. Begin with 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. A total of 2047 MB of physical RAM is 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. The non-global zones are using 2.64%, 2.84%, and 2.76% of the physical memory, respectively, 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, refer to http://docs.oracle.com/cd/E36784_01/html/E37628/gklcu.html#scrolltoc.

# Determining a Zone's Kernel File System Statistics

```
# fsstat -z s10 -z s10u9 zfs tmpfs
  new  name  name  attr attr lookup rddir  read  read write write
 file remov  chng   get  set    ops   ops   ops bytes   ops bytes
   93    82     6  163K  110   507K   148 69.7K 67.9M 4.62K 13.7M zfs:s10
  248   237   158  188K  101   612K   283 70.6K 68.6M 4.71K 15.2M zfs:s10u9
12.0K 1.90K 10.1K 35.4K   12  60.3K     4 25.7K 29.8M 36.6K 31.0M tmpfs:s10
12.0K 1.90K 10.1K 35.6K   14  60.2K     2 28.4K 32.1M 36.5K 30.9M tmpfs:S10u9
# fsstat -A -Z zfs tmpfs
  new  name  name  attr  attr lookup rddir  read  read write write
 file remov  chng   get   set    ops   ops   ops bytes   ops bytes
 360K 1.79K 20.2K 4.20M 1.02M  25.0M  145K 5.42M 2.00G 1.07M 8.10g zfs
 359K 1.48K 20.1K 4.04M 1.02M  24.5M  144K 5.31M 1.88G 1.06M 8.08G zfs:global
   93    82     6 74.8K   107   250K   144 54.8K 60.5M 4.61K 13.7M zfs:s10
  248   237   158 90.2K   101   336K   283 53.0K 58.3M 4.71K 15.2M zfs:s10u9
60.0K 41.9K 17.7K  410K   515   216K   426 1022K 1.02G  343K  330M tmpfs
49.4K 38.1K 11.0K  366K   489   172K   420  968K  979M  283K  273M tmpfs:global
5.28K 1.90K 3.36K 21.9K    12  21.7K     4 25.7K 29.8M 29.9K 28.3M tmpfs:s10
5.25K 1.90K 3.34K 22.1K    14  21.6K     2 28.4K 32.1M 29.8K 28.2M tmpfs:s10u9
```



The `fsstat` utility collects and prints `kstats` per zone, including aggregations. By default, the utility reports an aggregate of all running zones. A `per-fstype kstat` is produced for each zone. The global zone `kstat` reports its exclusive activity. The global zone can see the `kstats` of all the zones on the system. Non-global zones see only the `kstats` associated with the zone in which the utility is run. A non-global zone cannot monitor file system activity in other zones.

- `-z`: Reports file system activity per zone. Multiple `-z` options can be used to monitor activity in selected zones.
- `-A`: Reports aggregate file system activity for the specified `fstypes` across all zones. If neither the `-z` nor `-Z` option is used, this is the default behavior. This option displays the aggregate for the specified `fstypes` across all zones when used with either the `-z` or the `-Z` option.
- `-Z`: Reports file system activity in all zones on the system. If used with the `-z` option, this option has no effect. If it is used only to monitor `mountpoints` and not `fstypes`, this option has no effect.

# 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 8-2 Overview: Determining an Oracle Solaris Zone's Configuration

This practice covers the following topics:

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



This practice should take you about 30 minutes to complete.

# Summary

In this lesson, you should have learned how to:

- Explain the fundamentals of Oracle Solaris zones
- Configure an Oracle Solaris zone
- Determine an Oracle Solaris zone configuration



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 a zone's configuration and resource utilization. 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.

9

# Controlling Access to Systems and Files

# Objectives

After completing this lesson, you should be able to:

- Establish system and file access control
- Control access to systems
- Control access to files
- Secure access to a remote host



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 as 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.

# Importance of System and File Access Control

It is important to control access to systems and files to prevent:

- Unauthorized user access
- Intruders gaining remote access





Every company has a vested interest in ensuring that its business data remains confidential and private. In the workplace, all computers that are connected to a server can be thought of as one large multifaceted system. You are responsible for the security of this larger system. You need to defend the network from outsiders who are trying to gain access. You also need to ensure the integrity of data on the computers within the network. At the file level, Oracle Solaris provides standard security features that you can use to protect files, directories, and devices. At the system and network levels, the security issues are mostly the same. The first line of security defence is to control access to your system.

To control access to your system, you must maintain the physical security of your computing environment. For instance, a system that is logged in and left unattended is vulnerable to unauthorized access. An intruder can gain access to the operating system and to the network. The computer's surroundings and the computer hardware must be physically protected from unauthorized access. Remote logins offer a tempting avenue for intruders.

Oracle Solaris 11 provides a number of system security features that help your company keep its data secure.

# Implementing System and File Access Control

As part of implementing system and file access control, you will learn how to:

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


ORACLE



Based on your company's data protection needs and requirements and an existing security policy that outlines the organization's security guidelines, you need to develop 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. You will also set up and test Secure Shell by using the host-based authentication method.

# Lesson Agenda

- **Controlling Access to Systems**
- Controlling Access to Files
- Securing Access to Remote Host

# Controlling Access to Systems

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

- Securing logins and passwords
- Changing the password algorithm



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

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.

# 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 through:
  - Encryption
  - Placement in a separate file from username and other 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 are 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 is 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.

Passwords are initially created and encrypted when you set up a user account, which you learn about in the next lesson titled “Administering User Accounts.” 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. The `/etc/passwd` file is available to any user who can log in to a system, whereas only the user with the root role (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 OpenSolaris default is SHA256 based algorithm. To revert to
# the policy present in Solaris releases set CRYPT_DEFAULT=__unix__,
# which is not listed in crypt.conf(4) since it is internal to libc.
#
CRYPT_DEFAULT=5
#
```

ORACLE



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

As a system administrator, you can specify the 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 using 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.

# Controlling and Monitoring System Activities

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, use `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.

There are two common ways to access a system: by using a conventional user login or by using the root role login. The `sudo` command can be used to enable a user to run administrative commands without using the `root` role account. In the next lesson titled "Administering User Accounts," you learn more about user administration.

**Note:** The system identifies and tracks logins as being from either the original or real user ID (UID) or 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 `root` access and monitor who has been using the `su` command, especially those users who are trying to gain `root` access. In the next section, you are shown how to do this.

# 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
- Changing the password algorithm
- Verifying the password algorithm change
- 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 users that 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 determined in part by your company's security policies, as well as by indications of suspicious activity or possible security breach attempts.

You 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    itadmin           110      joe jones
                        /export/home/jjones
                        /usr/bin/bash
                        PS 010170 -1 -1 -1
```



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 about the user and system logins that are known to the system. The default information consists of the following:

- Login ID
- User ID
- Primary group name
- Primary group ID
- Account field value

The `-x` option 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 in the slide. `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 in the slide. 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 requiring 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 privileges.

**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 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:Mon Nov 11 23:21:10 2013
jjones:/dev/pts/2:Mon Nov 11 23:21:21 2013
jjones:/dev/pts/2:Mon Nov 11 23:21:30 2013
jjones:/dev/pts/2:Mon Nov 11 23:21:40 2013
jjones:/dev/pts/2:Mon Nov 11 23:21:49 2013
```

ORACLE



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.

**Notes**

- You are setting file permissions in absolute mode.
- This procedure captures failed login attempts from terminal windows. However, this procedure will not capture failed logins from a desktop login attempt.

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. Make the `auth.notice` entry into the `syslog.conf` file.
   b. Refresh the `system-log` service.
4. Verify that the log works.



1. 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.
   a. To do this, use the `vi /etc/default/login` command.
   b. When you are in the text editor, set `SYSLOG` to `YES` (`SYSLOG=YES`) and `SYSLOG_FAILED_LOGINS` to `0`.
2. Create a file with the correct permissions to hold the logging information; this is similar to what you did to the `loginlog` file.
   a. Create the `authlog` file in the `/var/adm` directory by using the `touch /var/adm/authlog` command.
   b. On the file, set read and write permissions for the `root` user by using the `chmod 600 /var/adm/authlog` command.
   c. Change the group membership to `sys` on the file by using the `chgrp sys /var/adm/authlog` command.

3. Edit the `syslog.conf` file to log failed password attempts. This step is necessary so that the `syslogd` daemon can recognize the configuration and send notices to this destination.
   a. 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 follow-on course *Oracle Solaris 11 Advanced System Administration*.
   b. Refresh the `system-log` service by using the `svcadm refresh system/system-log` command to make the changes effective.
4. 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.

# Monitoring 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 the 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 a 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
```



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 was made in the `/etc/security/policy.conf` file. In the password field, the second character tells you the algorithm that is 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.

Then you deleted `jjones`'s password, confirmed that it was deleted, and then created a new password for `jjones`. After changing the password, go to 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 in the slide.

The entries display the following information:

- Date and time that the command was entered
- Whether 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

# 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**

# Practice 9-1 Overview: Controlling Access to Systems

This practice covers the following topics:

- Securing logins and passwords
- Changing the password algorithm



This practice covers 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 find Practice 9-1 in your *Activity Guide*. It should take about 45 minutes to complete the practices.

# Lesson Agenda

- Controlling Access to Systems
- **Controlling Access to Files**
- Securing Access to Remote Host

# Controlling Access to Files

To secure files and directories in Oracle Solaris 11, 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. |

ORACLE



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, refer to the http://docs.oracle.com/cd/E36784_01/html/E37122/secfile-37.html#scrolltoc and http://docs.oracle.com/cd/E36784_01/html/E36835/ftyxi.html#ZFSADMINftyxi sections.

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

In this section, you are shown how to use basic UNIX permissions to control user, group, and `root` role access to files. You are also shown how to protect files against programs that pose a security risk.

# 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:

- **owner:** 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
- **other:** 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 `root` role has access to all the 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 the class **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 class **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, the 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 user with the `root` role). 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. |

ORACLE



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 |

ORACLE



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 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.

# Protecting Files with Basic UNIX Permissions

- Displaying file permissions
- Changing file ownership
- Changing the 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 file permissions, and then look at how to change the 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   4 root     bin        454  Oct 28  05:10 .
drwxr-xr-x  33 root     sys         45  Oct 27  10:00 ..
-r-xr-xr-x   1 root     bin      12772  Oct 19  20:55 autopush*
lrwxrwxrwx   1 root     root        10  Oct 27  10:00 accept -> cupsaccept
...
```

To display the permissions for a directory, use `ls -ld`.

```
# cd ..
# ls -ld sbin
lrwxrwxrwx   1 root     root        10  Oct 27  10:03 sbin -> ./usr/sbin
```



To display 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 in the slide 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, `root`
- **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 Nov 2 10:49 test-file
# chown omai test-file
# ls -l test-file
-rw-r--r--   1 omai     staff   112640 Nov 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 the 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 Nov 6 08:50 test-file
# chgrp itadmin test-file
# ls -l test-file
-rw-r--r--   1 omai   itadmin  112640 Nov 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.

1. To begin, display the permissions on the file by using `ls -l filename`. Make a note of the group name.
2. 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.
3. 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 in the slide, you are changing the group ownership of the file called `test-file` from `staff` to `itadmin`.

# 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 itadmin   112640 Nov 6 08:50 test-file
# chmod g+wx test-file
# ls -l test-file
-rw-rwxr--   1 omai itadmin   112640 Nov 6 09:00 test-file
# chmod u-w test-file
# ls -l test-file
-r--rwxr--   1 omai itadmin   112640 Nov 6 09:05 test-file
```

ORACLE



To change file permissions in symbolic mode, you must first determine what the current file permissions are.

1. To do this, use the `ls -l` command followed by the name of the file.
2. Use the `chmod` command followed by whose permissions are to be changed, the operation that is to be performed (*`operator`*), the permissions that are to be changed (*`permissions`*), and the file name (*`filename`*).
3. Verify that the permissions of the file have changed. You do this by using the `ls -l` *`filename`* command as you did in the first step.

In the example in the slide, you are changing the permissions on the file `test-file` to give the `itsupport` group write and execute permissions.

1. 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.
2. 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 itadmin   112640 Nov 7 08:50 test-file
# chmod 674 test-file
# ls -l test-file
-rw-rwxr--   1 omai itadmin   112640 Nov 7 09:10 test-file
# chmod 474 test-file
# ls -l test-file
-r--rwxr--   1 omai itadmin   112640 Nov 7 09:15 test-file
```



Now you look at how to perform the same file permission changes in absolute mode.

To change file permissions in absolute mode:

1. Determine what the current file permissions are. To do this, use the `ls -l` command followed by the name of the file.
2. Use the `chmod` command followed by octal values (*`nnn`*) that represent the permissions for the file owner, file group, and others, in that order
3. Verify that the permissions of the file have changed. You do this by using the `ls -l` *`filename`* command as you did in the first step.

In the example in the slide, you are changing the permissions on the `test-file` 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). You want to now 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`.

**Notes**

- Refer to the slide titled “Setting File Permissions in Absolute Mode” earlier in this lesson for a list of the absolute mode octal values.
- 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 itadmin   112640 Nov 8 09:50 test-file
# chmod 4655 test-file
# ls -l test-file
-rwsr--r--   1 omai itadmin   112640 Nov 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 on this file are `644` (read and write; read-only; read-only). 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 “`l`” in the permissions as follows:

```
-rw-r-lr--   1 omai itadmin   112640      Nov  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 itadmin   112640   Nov  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 Sept 19 13:44 /home/omai/test-file
-rwsr-xr-x 1 root bin   64588 Sept 19 09:03 /sbin/wificonfig
-r-sr-xr-x 1 root bin  206676 Sept 19 09:02 /usr/lib/ssh/ssh-keysign
-r-sr-xr-x 1 root bin   19452 Sept 19 09:02 /usr/lib/fs/smbfs/mount
...
```



As already discussed, the `setuid` and `setgid` permissions enable ordinary users to gain `root` role 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 {} \; > /var/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`:** Is the long format listing
  - **`-d`:** Lists only the directory name, not its contents
- **`{}`:** Is the 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*:** Is the file that contains the results of the `find` command

In the example in the slide, you are looking for `setuid` permissions in the `root` directory and have specified that the results of the search should be sent to 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 the 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 nonzero 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 that 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 programs attempt 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.

# Quiz

Which command enables you to change permissions on a file that is 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**

# Practice 9-2 Overview: Controlling Access to File Systems

This practice covers the following topics:

- Protecting files with basic permissions
- Protecting against programs with security risk



This practice should take about 45 minutes to complete.

# Lesson Agenda

- Controlling Access to Systems
- Controlling Access to Files
- **Securing Access to Remote Host**

# Oracle Solaris Authentication Services

Oracle Solaris offers the following authentication services:

| Authentication Service | Description |
|---|---|
| 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



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. For more information, refer to http://docs.oracle.com/cd/E36784_01/html/E37126/auth-2.html#scrolltoc.
- **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 `ssh`. For more information, refer to http://docs.oracle.com/cd/E36784_01/html/E37126/pam-1.html#OSMKApam-1.
- **Simple Authentication and Security Layer (SASL):** A framework that provides authentication and security services to network protocols. For more information, refer to http://docs.oracle.com/cd/E36784_01/html/E37126/sasl-1.html#scrolltoc.

- **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. For more information, refer to http://docs.oracle.com/cd/E36784_01/html/E37126/kintro-1.html#scrolltoc.

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, only the Secure Shell topic is covered. The remaining authentication services are covered in detail in the specialty course *Oracle Solaris 11 Security Administration*.

# Secure Shell

- Is the default remote access control protocol on a newly installed Oracle Solaris 11 system
- 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

ORACLE



Secure Shell is the default remote access protocol on a newly installed Oracle Solaris system. Secure Shell in Oracle Solaris is built on top of the open source toolkit, OpenSSL, which implements the Secure Sockets Layer and Transport Layer Security. 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.

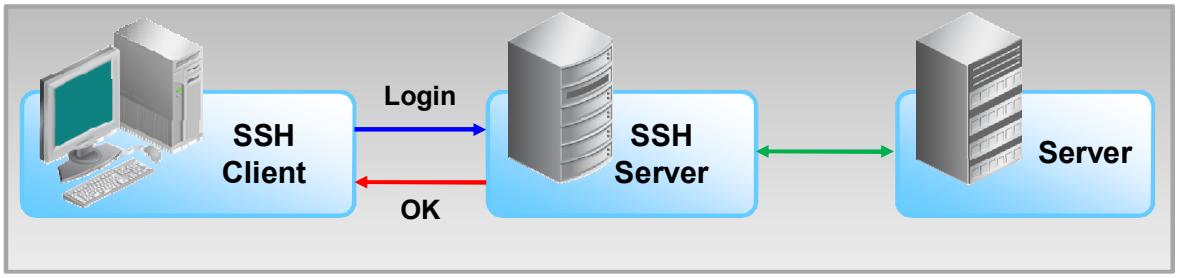
In Secure Shell, authentication is provided by the use of passwords, public keys, or both. All network traffic is encrypted. Thus, Secure Shell prevents a would-be intruder from being able to read an intercepted communication. Secure Shell also prevents an adversary from spoofing the system.

**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

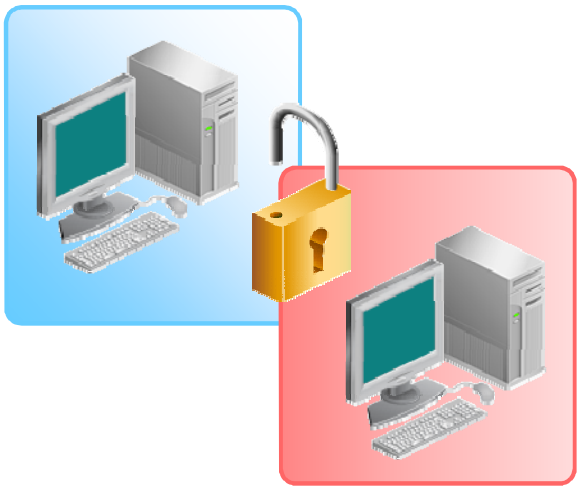
With Secure Shell, you can:

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



# 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.

On the server side, Secure Shell supports version 2 (v2) of the Secure Shell protocol. On the client side, in addition to v2, the client supports version 1 (v1).

# Secure Shell Protocol Version 2: Parts

| Protocol | Description |
|---|---|
| **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. |

ORACLE



**Note:** For a more detailed description of how Secure Shell authentication works, refer to http://docs.oracle.com/cd/E36784_01/html/E37125/sshuser-3.html#scrolltoc.

# Secure Shell Authentication Methods

| Method | Description |
|---|---|
| **GSS-API** | Uses credentials for GSS-API mechanisms |
| **Host-based authentication** | Uses host keys |
| **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; uses the client’s RSA and DSA public/private host keys to authenticate the client
- **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 and public key in the `/etc/ssh` directory<br>  • `ssh_host_rsa_key`<br>  • `ssh_host_rsa1_key`<br>  • `ssh_host_dsa_key`<br>• `HostbasedAuthentication yes` in the `/etc/ssh/sshd_config` directory<br>• Private key in `~/.ssh/id_rsa` or `~/.ssh/id_dsa`<br>• User's public key in `~/.ssh/id_rsa.pub` or `~/.ssh/id_dsa.pub` | • User account<br>• Local host public key in the `/etc/ssh` directory<br>• `HostbasedAuthentication yes` in `/etc/ssh/sshd_config`<br>• Client1 entry in `/etc/ssh/shosts.equiv`, `/etc/hosts.equiv`, `~/.rhosts`, or `~/.shosts`<br>• Client1 host name in `/etc/ssh/ssh_known_hosts` or `~/.ssh/known_hosts`<br>• `IgnoreRhosts no` in `/etc/ssh/sshd_config` |



You can 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. The directory `/etc/ssh` contains both private and public key files.

**On the Client1 machine:**

`HostbasedAuthentication` should be set to `yes` in the `/etc/ssh/sshd_config` directory.

**On the Server1 machine:**

- `HostbasedAuthentication` should be set to `yes` in the `/etc/ssh/sshd_config` directory.
- `/etc/ssh/ssh_known_hosts` should contain the `client1` host name along with the cryptographic algorithm, followed by the client's public key.
- The client machine's entry should be made in the `/etc/ssh/shosts.equiv` directory.

You can verify host-based authentication for SSH by trying to gain secure shell access to the `server1` machine as user X. You will notice that you are not prompted for a password as user X.

# Identifying the Secure Shell Defaults

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



The following are the key Secure Shell default settings:

- 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 communication 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 read its configuration files again, 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 communication 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 re-read its configuration files, you use the following command:

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

This simpler command also works:

```
# svcadm restart ssh
```

# Configuring Secure Shell

1. Verifying that users have access to both the client and the server.
2. Logging in to a remote host with Secure Shell.
3. Generating the public/private RSA key pair.
4. Copying the RSA public key to the remote host.
5. Verifying that the RSA public key is functioning.
6. Generating the public/private DSA key pair.
7. Copying the DSA public key to the remote host.
8. Verifying the authentication process.



After you have enabled host-based authentication, the next task is to configure Secure Shell.

# Verifying That Users Have Access to Both the Client and the Server

Server side

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

Client side

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



Next, you must verify that users have access to the client and the server. To do this, use the `grep` command followed by the user's login name and `/etc/passwd`, as shown in the two examples in the slide.

# Logging In to a Remote Host with Secure Shell

```
# su - jjones
Oracle Corporation      SunOS 5.11      11.2        June 2014
jjones@server1:$ 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: <password>
Last login: Tue Jul 29 08:17:26 2014 from server1
Oracle Corporation      SunOS 5.11      11.2        June 2014
jjones@client1:~$ exit
Connection to client1 closed.
```



The steps to log in to a remote host with Secure Shell are as follows:

1. Log in to the server system and `su` to a user, for example `jjones`, as shown in the slide example.
2. As the `jjones` user, use the `ssh` command to log in to the remote host (`client1`) machine. The system then prompts you to verify the authenticity of the remote host key by asking you whether you want to continue connecting.
3. Respond with a `yes`.
4. Log in to `client1` by using `jjones`'s user account password. You then receive confirmation that this host has been permanently added to the list of known hosts.
5. After logging in successfully, use the `exit` command to close the Secure Shell connection to `client1` and return to the server.

**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.

# Generating the Public/Private RSA Key Pair

```
jjones@server1:$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/export/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 /export/home/jjones/.ssh/id_rsa.
Your public key has been saved in /export/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:$ ls .ssh
id_rsa  id_rsa.pub
```



The steps to generate the public/private RSA key pair that will be used by Secure Shell are as follows:

1. As the `jjones` user, use the `ssh-keygen -t rsa` command, as shown in the example in the slide. The system generates the public/private RSA key pair and creates the `/export/home/`*`loginname`*`/ .shh/id_rsa` file in which the RSA key pair is saved.
2. Confirm file creation by pressing the Enter key.
3. Enter a passphrase for using your RSA 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. On entering a passphrase, you are presented with the key fingerprint.
4. Use the `ls .ssh` command to verify that the path to the key file is correct. 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:$ scp .ssh/id_rsa.pub jjones@client1:id_rsa.pub
Password: <password>
id_rsa.pub       100% |*****************************|   401       00:00
jjones@server1:$ ssh client1
Password: <password>
Last login: Tue July 29 08:19:04 2014 from server1
Oracle Corporation      SunOS 5.11      11.2       June 2014
jjones@client1:~$ ls
id_rsa.pub  local.cshrc  local.login  local.profile
jjones@client1:~$ mkdir -p .ssh
jjones@client1:~$ ls
id_rsa.pub  local.cshrc  local.login  local.profile
jjones@client1:~$ cat ./id_rsa.pub >> .ssh/authorized_keys
jjones@client1:~$ rm ./id_rsa.pub
```



The steps to copy the local host's public key to the remote host and store it in user `jjones`'s `.ssh` directory are as follows:

1. Use the `scp` command followed by `.ssh/id rsa.pub` *`loginname@remotehost`*`:id_rsa.pub` to copy the local host's public key to the remote host.
   **Note:** `.pub` indicates the public key.
2. Provide `jjones`'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 indicates 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)

3. After the copy is completed, use the `ssh` command to the client system and enter `jjones`'s password.
4. Run the `ls` command to ensure that the `id_rsa.pub` file is present.
5. Run the `mkdir -p .ssh` command to create the `.ssh` directory, and then run the `ls` command again.
6. Place the `id_rsa.pub` file in the `.ssh/authorized_keys` file. To do this, 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.
7. 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:~$ ssh client1
Enter passphrase for key '/export/home/jjones/.ssh/id_rsa': <passphrase>
Last login: Tue Jul 29 08:21:32 2014 from server1
jjones@client1:~$ exit
Connection to client1 closed.
```



The next step is to verify that the public key is functioning. To do this, 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:~$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/export/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 /export/home/jjones/.ssh/id_dsa.
Your public key has been saved in /export/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:~$ ls -a .ssh
.       id_dsa       id_rsa       known_hosts
..      Id_dsa.pub   id_rsa.pub
```



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 in the slide.

# Copying the DSA Public Key to the Remote Host

```
jjones@server1:~$ scp .ssh/id_dsa.pub jjones@client1:id_dsa.pub
Enter passphrase for key '/export/home/jjones/.ssh/id_rsa': <passphrase>
id_dsa.pub       100% |*****************************|   609       00:00
jjones@server1:~$ ssh client1
Enter passphrase for key '/export/home/jjones/.ssh/id_rsa': <passphrase>
Last login: Tue Jul 29 08:23:05 2014 from server1
Oracle Corporation      SunOS 5.11      11.2        June 2014
jjones@client1:~$ ls
id_dsa.pub local.cshrc  local.login  local.profile
jjones@client1:~$ cat ./id_dsa.pub >> .ssh/authorized_keys
jjones@client1:~$ rm ./id_dsa.pub
jjones@client1:~$ exit
Connection to client1 closed.
```



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 in the slide.

As you can see, 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:~$ ssh client1
Enter passphrase for key '/export/home/jjones/.ssh/id_rsa': <Press Enter
Key>
Enter passphrase for key '/export/home/jjones/.ssh/id_dsa': <passphrase>
Last login: Tue Jul 29 08:25:16 2014 from server1
Oracle Corporation      SunOS 5.11      11.2        June 2014
jjones@server1:~$ 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 in the slide.

# Using the Secure Shell

- Reducing password prompts
- Locking and unlocking the authentication agent

# Reducing Password Prompts

```
jjones@server1: ~$ eval `ssh-agent`
Agent pid 1886
jjones@server1: ~$ pgrep ssh-agent
1886
jjones@server1: ~$ env | grep SSH
SSH_AGENT_PID=1886
SSH_AUTH_SOCK=/tmp/ssh-XXXXJqaWVf/agent.1885
jjones@server1: ~$ ssh-add
Enter passphrase for /export/home/jjones/.ssh/id_rsa: <passphrase>
Identity added: /export/home/jjones/.ssh/id_rsa (/export/home/jjones/.ssh/id_rsa)
Identity added: /export/home/jjones/.ssh/id_dsa (/export/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 /export/home/jjones/.ssh/id_rsa
(RSA)
1024 7a:b8:cb:f8:33:e5:fb:02:a5:c3:b2:53:cc:75:90:9e /export/home/jjones/.ssh/id_dsa
(DSA)
jjones@server1: ~$ ssh client1
Last login: Tue Jul 29 08:26:22 2014 from server1
Oracle Corporation      SunOS 5.11      11.2        June 2014
jjones@client1:~$ exit
Connection to client1 closed.
```



If you do not want to type your passphrase and your password to use Secure Shell, you can use the `ssh-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. The steps to reduce password prompts are as follows:

1. Run the `eval `ssh-agent`` command to start the agent daemon.
2. Run the `env | grep SSH` command to verify that the environmental variables are populated.
3. Run the `ssh-add` command to add your private key to the agent daemon.
4. Run the `ssh-add -l` command to list the identities and confirm that they are available with the authentication agent.
5. To verify that the passphrase does not appear, 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: <password>
Agent locked.
jjones@server1:~$ ssh client1
Enter passphrase for key '/export/home/jjones/.ssh/id_rsa': <passphrase>
Last login: Tue Jul 29 08:27:14 2014 from server1
Oracle Corporation      SunOS 5.11      11.2        June 2014
jjones@server1:~$ exit
Connection to client1 closed.
```

```
jjones@server1:~$ ssh-add -X
Enter lock password: <password>
Agent unlocked.
jjones@server1:~$ ssh client1
Last login: Tue Jul 29 08:27:36 2014 from server1
Oracle Corporation      SunOS 5.11      11.2       June 2014
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 in the slide. 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 see that when the agent is unlocked, you are no longer prompted for a passphrase.

# Quiz

Secure Shell is an authentication service that _____.

a. Enables a user to securely access a remote host over an unsecure network
b. Provides authentication and security services to network protocols
c. Protects NFS mounts and a naming service



**Answer: a**

# Quiz

If you do not want to type your passphrase and your password to use Secure Shell, which of the following should you use?

a. `ssh-add`
b. `ssh-agent`
c. `ssh-keygen`



**Answer: b**

# Practice 9-3 Overview: Configuring Secure Shell

This practice covers the following topics:

- Setting up host-based authentication
- Verifying host-based authentication for SSH
- Configuring SSH for public key authentication
- Using SSH with no password prompt



This practice should take about one hour to complete.

# Summary

In this lesson, you should have learned how to:

- Establish system and file access control
- Control access to systems
- Control access to files
- Secure access to remote host



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

# Administering User Accounts

# Objectives

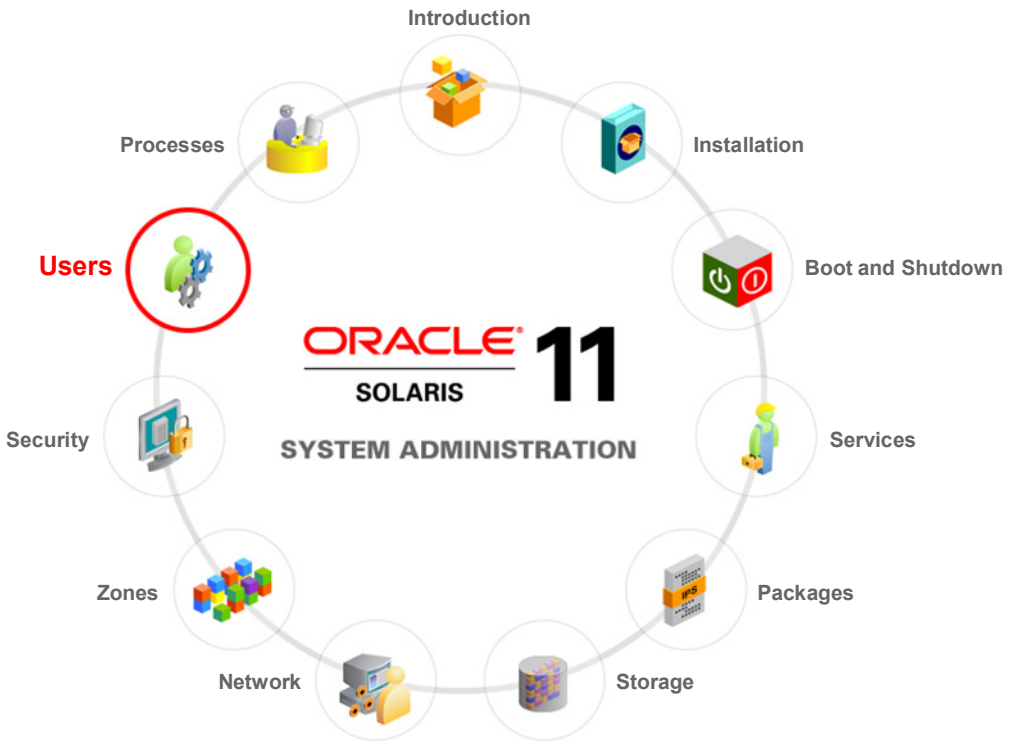
After completing this lesson, you should be able to:

- Get started with user administration
- Set up user accounts
- Manage user accounts
- Manage user initialization files
- Configure user disk quotas
- Use shell metacharacters



In this lesson, you are introduced to 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 as to where you are in the job workflow. So far you have successfully completed installing the operating system, managing system boot and shutdown, administering the SMF services and data storage environment, setting up IPS clients, and administering the network, zones, and system security. You 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, but also 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

- **Getting Started with User Administration**
- Setting Up User Accounts
- Maintaining User Accounts
- Managing User Initialization Files
- Configuring User Disk Quotas
- Using Shell Metacharacters

# Importance of User Administration

It is important to administer users to address the requirements of the user community, such as:

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



Your company's Oracle Solaris 11 implementation should include user administration. User administration usually involves a comprehensive approach to managing users and groups that includes setting up new accounts, maintaining accounts, and ensuring that users have access to the system and system resources that they need in order to do their work.

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

# Types of User Accounts

A user can have the following types of accounts:

| Account | Description |
|---|---|
| **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 |

ORACLE



A user can be identified in the system in multiple ways:

- As an individual user
- As a member of a group
- 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.

**Notes**

- 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.
- 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 |
|---|---|
| **Username** | 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 |

ORACLE



The user accounts that you will be setting up consist 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 with 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 more information about user account components, refer to http://docs.oracle.com/cd/E36784_01/html/E36818/userconcept-11407.html#ADUSRuserconcept-2. For additional guidelines on setting up user accounts, refer to http://docs.oracle.com/cd/E36784_01/html/E36818/userconcept-30.html#scrolltoc.

# 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 the 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. Because of 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. Because of 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 user 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:/root:/usr/bin/bash
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:/:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
dladm:x:15:65:Datalink Admin:/:
netadm:x:16:65:Network Admin:/:
netcfg:x:17:65:Network Configuration Admin:/:
smmsp:x:25:25:SendMail Message Submission Program:/:
gdm:x:50:50:GDM Reserved UID:/var/lib/gdm:
zfssnap:x:51:12:ZFS Automatic Snapshots Reserved UID:/:/usr/bin/pfsh
upnp:x:52:52:UPnP Server Reserved UID:/var/coherence:/bin/ksh
xvm:x:60:60:xVM User:/:
mysql:x:70:70:MySQL Reserved UID:/:
openldap:x:75:75:OpenLDAP User:/:
webservd:x:80:80:WebServer Reserved UID:/:
postgres:x:90:90:PostgreSQL Reserved UID:/:/usr/bin/pfksh
svctag:x:95:12:Service Tag UID:/:
unknown:x:96:96:Unknown Remote 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:/:
ikeuser:x:67:12:IKE Admin:/:
aiuser:x:61:61:AI User:/:
pkg5srv:x:97:97:pkg(5) server UID:/:
```



Although 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`:** Superuser account
- **`daemon`, `1`:** Umbrella system daemon associated with routine system tasks
- **`bin`, `2`:** Administrative daemon associated with running system binaries to perform some routine system tasks
- **`sys`, `3`:** Administrative daemon associated with system logging or updating files in temporary directories
- **`adm`, `4`:** Administrative daemon associated with system logging
- **`lp`, `71`:** Line printer daemon
- **`uucp`, `5`:** Daemon associated with the UNIX-to-UNIX Copy Program (UUCP) functions
- **`nuucp`, `6`:** Another daemon associated with the UUCP functions

- **dladm, 15:** Account reserved for datalink administration
- **netadm, 16:** Account reserved for network administration
- **netcfg, 17:** Account reserved for network configuration administration
- **smmsp, 25:** Daemon for the Sendmail message submission program
- **gdm, 50:** GNOME Display Manager daemon
- **zfssnap, 51:** Account reserved for automatic snapshots
- **upnp, 52:** Account reserved for the UPnP server
- **xvm, 60:** Account reserved for the xVM user
- **mysql, 70:** Account reserved for the MySQL user
- **openldap, 75:** Account reserved for the OpenLDAP user
- **webservd, 80:** Account reserved for WebServer access
- **postgres, 90:** Account reserved for PostgresSQL access
- **svctag, 95:** Service Tag Registry access
- **unknown, 96:** Account reserved for unmappable remote groups in NFSv4 ACLs
- **nobody, 60001:** Account 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
- **ikeuser, 67:** Account reserved for the IKE user
- **aiuser, 61:** Account reserved for the AI user
- **pkg5srv:** Account reserved for the `pkg`(5) depot server

# Interpreting an /etc/passwd File Entry

Each entry in the /etc/passwd file contains seven fields.

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

| Field | Description |
|---|---|
| *loginID* | Represents the user's login name |
| *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 this 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 mounted name of the user's home directory (done as a default by 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 Oracle Solaris OS:
  - Bash shell
  - Bourne shell
  - C shell
  - Korn shell
  - TC shell
  - Z shell

  The default shell for Oracle Solaris 11 is Bash.

# Interpreting the /etc/shadow File

```
root:$5$A9EW6h0R$B9cdXEPFGS8F2g4gEAWwlzUI40LBYUs7CRb9saMqx8XA:16283::::::
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::::::
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::::::
ikeuser:*LK*:15992::::::
aiuser:*LK*:15992::::::
pkg5srv:NP:15992::::::
```



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

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

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

| Field | Description |
|---|---|
| *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 |

ORACLE



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
#DICTIONMINWORDLENGTH=3
```



This slide presents the /etc/default/passwd file. You can set values for the following parameters in this file to control properties for all user passwords in 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 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:
netadm: 65:
mysql::70:
openldap::75:
webservd::80:
postgres::90:
unknown::96:
nobody::60001:
noaccess::60002:
nogroup::65534:
aiuser::61:
pkg5srv:97:
mlocate::95:
vboxsf::100:
```



This slide presents an example of the default entries in the /etc/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:** Superuser group
- **other, 1:** Optional group
- **bin, 2:** Administrative group associated with the running system binaries
- **sys, 3:** Administrative group associated with system logging or temporary directories
- **adm, 4:** Administrative group associated with system logging
- **uucp, 5:** Group associated with the uucp functions
- **mail, 6:** Electronic mail group
- **tty, 7:** Group associated with the tty devices
- **lp, 8:** Line printer group
- **nuucp, 9:** Group associated with the uucp functions

- **staff, 10:** General administrative group
- **daemon, 12:** Group associated with routine system tasks
- **sysadmin, 14:** Administrative group that is useful for system administrators
- **games, 20:** Group reserved for games
- **smmsp, 25:** Daemon for the Sendmail message submission program
- **gdm, 50:** Group reserved for the GNOME Display Manager daemon
- **upnp, 52:** Group associated with the UPnP server functions
- **xvm, 60:** Group reserved for xVM access
- **netadm, 65:** Group reserved for network administration
- **mysql, 70:** Group reserved for MySQL access
- **openldap, 75:** Group reserved for OpenLDAP access
- **webservd, 80:** Group reserved for WebServer access
- **postgres, 90:** Group reserved for PostgresSQL access
- **slocate, 95:** Group reserved for `slocate` indexing and query daemon. `slocate` is a secure version of the `locate` command.
- **unknown, 96:** Group reserved for unmappable remote groups in NFSv4 ACLs
- **nobody, 60001:** Group assigned for anonymous NFS access
- **noaccess, 60002:** Group assigned to a user or process that needs access to a system through some application but without actually logging in
- **nogroup, 65534:** Group that is assigned to a user who is not a member of a known group
- **aiuser, 61:** Group assigned for AI access
- **pkg5srv, 97:** Group assigned to `pkg`(5) depot server
- **mlocate, 95:** Group reserved for `mlocate` indexing and query daemon
- **vboxsf, 100:** Group assigned to access auto-mounted shared folders inside the VirtualBox environment

# Interpreting an /etc/group File Entry

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

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

| Field | Description |
|---|---|
| *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 this 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 User Administration

As part of user administration implementation, you will now learn how to:

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


ORACLE

# 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

- Getting Started with the User Administration
- **Setting Up User Accounts**
- Maintaining User Accounts
- Managing User Initialization Files
- Configuring User Disk Quotas
- Using Shell Metacharacters

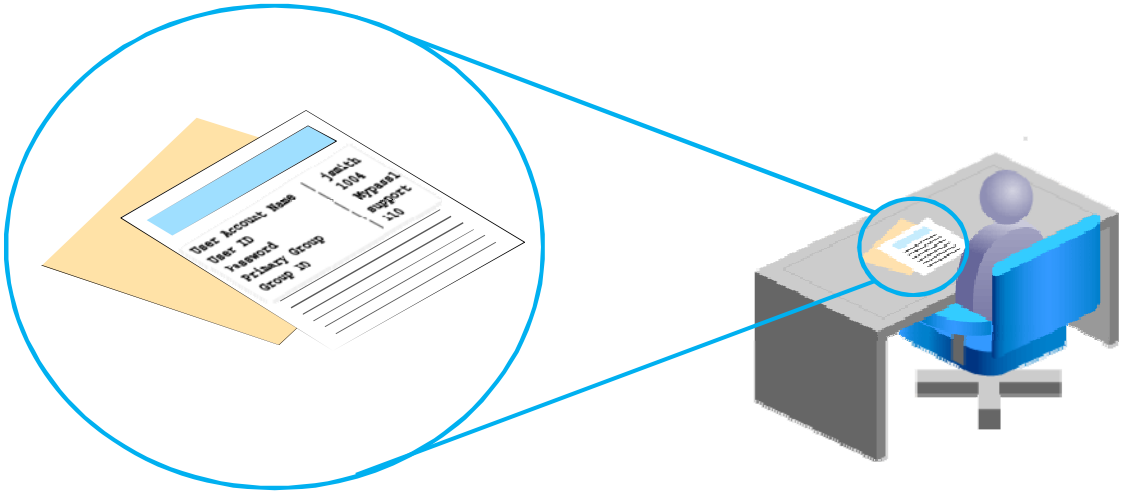
# Setting Up User Accounts

- 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






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, refer to http://docs.oracle.com/cd/E36784_01/html/E36818/usersetup-21417.html#scrolltoc.

# Creating the User Accounts Default File

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

```
# ls /usr/sadm/defadduser
/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 in 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.

When the `-D` option is used with the `–g`, `-b`, `-f`, `-e`, `-A`, `-p`, `-P`, `-R`, or `–K` option, it sets the default values for the specified fields. 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 a 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:
```



As you learned in the section on getting started with 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 in the slide. 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 made him a member of the `support` group that you created earlier. This is `jsmith`'s primary group. In addition, you also made him a member of a secondary group called `itgroup`. Next, you defined the full path name for the user's home directory, which you created earlier. You used the comment option to specify the user's full name (Joe Smith), and 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/passwd` 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:/usr/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 in the slide, you check 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 in the slide, you check whether an entry for Joe Smith is 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 is 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, contain at least two alphabetic characters, and contain 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 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 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 is 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 the example in the slide, 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 whether the group exists by using grep *groupname* /etc/group, and then use id loginname.

```
# grep support /etc/group
support::110:
# id jsmith
uid=1003(jsmith) gid=110(support)
```



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 whether 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 in the slide, you check whether the support group is in the /etc/group file. As you see, it is and the group ID number is 110. You might recall from an earlier discussion about /etc/group 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 the following:

```
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 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 the username. 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 the 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 a 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 the password 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.

# Quiz

`/var/sadm/defadduser` is the file that you use to add new users.

a. True
b. False



**Answer: b**

# Quiz

When you create a new user, which of the following files receives user-related information?

a. /etc/skell
b. /etc/shaddow
c. /etc/group
d. /etc/password



**Answer: c**

# Lesson Agenda

- Getting Started with the User Administration
- Setting Up User Accounts
- **Maintaining User Accounts**
- Managing User Initialization Files
- Configuring User Disk Quotas
- Using Shell Metacharacters

# 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 was 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
# zfs list
..
rpool/export/home/jsmith          35K  4.32G    35K  /export/home/jjones
...
# zfs rename rpool/export/home/jsmith rpool/export/home/jjones
# zfs list
...
rpool/export/home/jjones          35K  4.32G    35K  /export/home/jjones
...
# grep jjones /etc/passwd
jjones:x:1003:110:joe jones:/export/home/jjones:/usr/bin/bash
```



To modify a user's account, use the `usermod` command followed by the appropriate user attribute. In the example in the slide, you change `jsmith`'s login name and home directory to `jjones`. When you use the `zfs list` command and view the ZFS file systems, observe that the `usermod` command changed the home directory of the user but not the ZFS file system name. The ZFS file system name for the user continues to display `jsmith`. To change the ZFS file system name to reflect `jjones`, use the `zfs rename` command and view the resulting change by using the `zfs list` command. Next, you verify that the change has been made in the `/etc/passwd` file. In the `/etc/passwd` file, you see that the login name, the user's name, and the home directory have been changed. The user ID, group ID, and shell remain the same.

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 a user
- **`-g` *`gid`***: Specifies an existing group's integer ID or a character-string name. It redefines a user's primary group membership.
- **`-G` *`gid`***: Defines a new user's supplementary group membership

- **-d *dir*:** Specifies the new home directory of a 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 a 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.

# Deleting a User Account

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

```
# userdel -r ckent
```



The employee `ckent` has left the company, so you should now 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.

**Notes**

- 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.
- 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 itadmin support
# grep itadmin /etc/group
itadmin::110::
# grep itadmin /etc/group
itadmin::110::
# id jjones
uid=1003(jjones) gid=110(itadmin)
```



The `groupmod` command is used to modify group entries. The `groupmod` utility modifies the group definitions in the `/etc/group` file. It 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. The name argument is a string of not more than eight bytes consisting of characters from the set of lowercase alphabetic characters and numeric characters. A warning message is displayed if the restrictions are not met.

To modify a group entry, use the `groupmod` command followed by the group attribute. In the example in the slide, you change the name of the `support` group to `itadmin` 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 `itadmin` 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
# 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
# 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 in the slide, 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, 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 in the slide, 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.

**Note:** You can also set up and manage users by using the Oracle Solaris User Manager GUI. You can use the User Manager GUI to perform most of the tasks that can be performed by using the equivalent CLI (`useradd`, `usermod`, `userdel`, and so on). For more information about the User Manager GUI and its use, refer to http://docs.oracle.com/cd/E36784_01/html/E36818/usersetupgui-1.html#scrolltoc.

# Practice 10-1 and Practice 10-2 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

ORACLE



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

- **Practice 10-1:** Setting up user accounts
- **Practice 10-2:** Maintaining user accounts
- **Practice 10-3:** Managing user initialization files
- **Practice 10-4:** Exploring shell metacharacters and user quotas

You find Practices 10-1 and 10-2 in your *Activity Guide*. It should take about 30 minutes to complete Practice 10-1 and 30 minutes to complete Practice 10-2.

# Lesson Agenda

- Getting Started With the User Administration
- Setting Up User Accounts
- Maintaining User Accounts
- **Managing User Initialization Files**
- Configuring User Disk Quotas
- Using Shell Metacharacters

# 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 table in the slide describes the shell options that are supported in this release. In this Oracle Solaris release, the following shell features and behaviors are supported:

- The user account that is created when you install the Oracle Solaris 11 release is assigned the GNU Bourne-Again Shell (bash) by default.
- The standard system shell (bin/sh) is now the Korn Shell 93 (ksh93).
- The default interactive shell is the Bourne-again (bash) shell (/usr/bin/bash).
- Both the bash and ksh93 shells feature command-line editing, which means you can edit commands before executing them.
- There are a few ways in which you can display default shell and path information:
  - Use the echo $SHELL and which commands:

```
$ grep root /etc/passwd
root:x:0:0:Super-User:/root:/usr/bin/bash
$ echo $SHELL
/usr/bin/bash
$ which ksh93
/usr/bin/ksh93
```

- Use the `pargs` command:

  ```
  $ pargs -l $$
  /usr/bin/bash
  ```

- The `ksh93` shell also has a built-in variable called `.sh.version`, which can be displayed as follows:

  ```
  $ echo ${.sh.version}
  Version JM 93u 2011-02-08
  ```

- To change to a different shell, type the path of the shell that you want to use.
- To exit a shell, type `exit`.

**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. For example, if you are in a `bash` shell, you can display the complete history of the commands that you have run by using the `history` command. To display a specific number of commands executed previously, include an integer in the history command, for example, `history 2`.

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:

- **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 alternative `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:

- **Site 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

# Site Initialization Files

- You are responsible for maintaining the site initialization files.
- Site 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 site initialization files in accordance with the needs and requirement of the users in the system. These files provide an environment for the entire community of users who log in to the system. The site initialization files reside in the `/etc` directory. The `/etc/profile` file and the `/etc/.login` file are the two main site 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 site 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.

# Bash Shell Initialization Files

For the Bash shell, initialization files are run in the following sequence:

1. Commands in `/etc/profile` are executed if present.
2. Commands from the `$HOME/.bash_profile`, `$HOME/.bash_login`, and `$HOME/.profile` file are executed.
3. When an interactive shell that is not a login shell is started, `bash` reads and executes commands from the `$HOME/.bashrc` file if it is present.
4. When startup processing is complete, the `bash` shell begins reading commands from the default input device, the terminal.
5. Upon exiting the shell, `bash` reads and executes `$HOME/.bash_logout`.

ORACLE



Bash shell initialization files run in a particular sequence after the user logs in to the system:

1. Commands in `/etc/profile` are executed if present. `/etc/profile` is the global initialization file for the Bash shell.
2. Commands from the `$HOME/.bash_profile`, `$HOME/.bash_login`, and `$HOME/.profile` files (located in the user's home directory) are executed. The system reads and executes commands from the first file that exists and is readable. Typically either `.bash_profile` or `.profile` is used. The file contains commands to specify the terminal type and environment.
3. When an interactive shell that is not a login shell is started (for example, a new terminal window is opened), `bash` reads and executes commands from the `$HOME/.bashrc` file if it is present. Typically, you may see shell aliases placed in this file.
4. When startup processing is complete, the `bash` shell begins reading commands from the default input device, the terminal.
5. Upon exiting the shell, `bash` reads and executes `$HOME/.bash_logout` (if it exists).

For more information about the `bash` shell, refer to the `bash`(1) man page.

# Managing User Initialization Files

- Setting up site-wide initialization files
- Setting up the user initialization files
- Customizing the user's work environment



When you set up the home directory of a user, you need to set up the shell initialization files for that user's login shell (also called "user initialization files"). A shell initialization file is a shell script that runs automatically each time the user logs in. The initialization file sets up the work environment, and customizes the shell environment for the user. The primary job of the shell initialization file is to define the user's shell environment, such as the search path, environment variables, and windowing environment. Each UNIX shell has its own shell initialization file (or files), which is located in the user's home directory.

# 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 in the slide. 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
```

# Modifying the Site Initialization Files

To edit a site 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 the 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><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 |

ORACLE



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

The 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 |

ORACLE



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 a work environment that is common to all users. You can also modify these files to provide a 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/sbin
export PATH
#
```



When a user logs in to the system, the user's login shell is invoked. The shell program looks for its initialization files in the correct order for the shell. The shell program then executes the commands contained in each file and, when it is finished, displays the shell prompt on the user's screen.

Default user initialization files (such as `.cshrc`, `.profile`, and `.login`) are created automatically in the user's home directory when a new user account is added. You can predefine the contents of these files, or you can choose to use the system default files. Oracle Solaris provides default user initialization files for each shell in the `/etc/skel` directory on each system. You can use these initialization files as a starting point and modify them to create a standard set of files that provide a work environment that is common to all users. You can also modify them to provide a working environment for different types of users.

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. 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 in the slide, 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).

# Quiz

Which of the following is an enhanced C shell?

a. /usr/bin/csh
b. /usr/bin/tcsh
c. /usr/bin/ksh
d. /usr/bin/bash



**Answer: d**

# Practice 10-3 Overview: Managing User Initialization Files

This practice covers the following topics:

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



This practice should take about 45 minutes to complete.

# Lesson Agenda

- Getting Started With the User Administration
- Setting Up User Accounts
- Maintaining User Accounts
- Managing User Initialization Files
- **Configuring User Disk Quotas**
- Using Shell Metacharacters



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

# 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



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 several 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. 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, refer to http://docs.oracle.com/cd/E36784_01//html/E36835/gazud.html#ZFSADMINgitfx.

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
rpool/export/home/jjones    quota     10g       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 in the slide, 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@<name>=` 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 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    jjones      7K      10g
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:** 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 by 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.

# Lesson Agenda

- Getting Started With the User Administration
- Setting Up User Accounts
- Maintaining User Accounts
- Managing User Initialization Files
- Configuring User Disk Quotas
- **Using Shell Metacharacters**

# 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



As you saw in the preceding section, the shell is the user's window to the system. Knowing how to use shell metacharacters enables you to move within the system more easily and to locate files and directories more quickly.

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 now look at each of these characters, 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
$
```

ORACLE



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:
$
```

ORACLE



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
$
```

# Quiz

If you want to change to your home directory, which of the following characters helps you do that?

a. Tilde (~) character
b. Dot (.) character
c. Asterisk (*) character
d. Dash (-) character



**Answer: a**

# Practice 10-4 Overview: Exploring Shell Metacharacters and User Quotas

This practice covers the following topics:

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



This practice should take about 30 minutes to complete.

# Summary

In this lesson, you should have learned how to:

- Get started with user administration
- Set up user accounts
- Manage user accounts
- Manage user initialization files
- Configure user disk quotas
- Use shell metacharacters

11

# Managing System Processes and Scheduling System Tasks

# Objectives

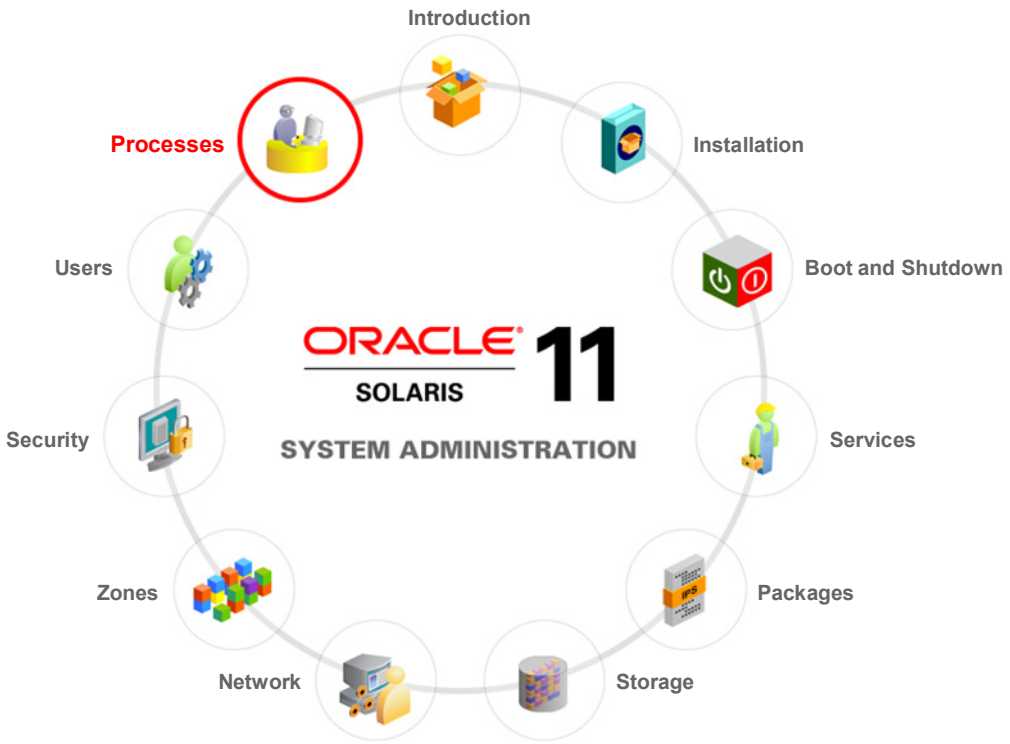
After completing this lesson, you should be able to:

- Explain 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. You also learn how to schedule system tasks by using the `at` command and `crontab` file, and how to administer `crontab` files.

# Workflow Orientation






Before you start the lesson, orient yourself as to where you are in the job workflow. 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

- **Managing System Processes**
- Scheduling System Administration Tasks

# Importance of System Processes Management

System processes management ensures that you can:

- Determine what processes are running in 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


ORACLE



Your company's Oracle Solaris 11 implementation should include system processes management. System processes management is more about ensuring that you are ready to manage processes on an Oracle Solaris 11 system and to perform basic system process management tasks, such as being able to determine the processes that are running on the system, the state the processes are in, and the system resources that the processes are utilizing. It is also important to understand how to control processes and terminate unwanted processes. Because your time is valuable, process management allows you 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, how you can interact with them, and how to administer `crontab` files.

# System Processes: Overview

A process is:

- Any program that is running in the system
- Assigned a unique process identification (PID) number that is:
  - 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.

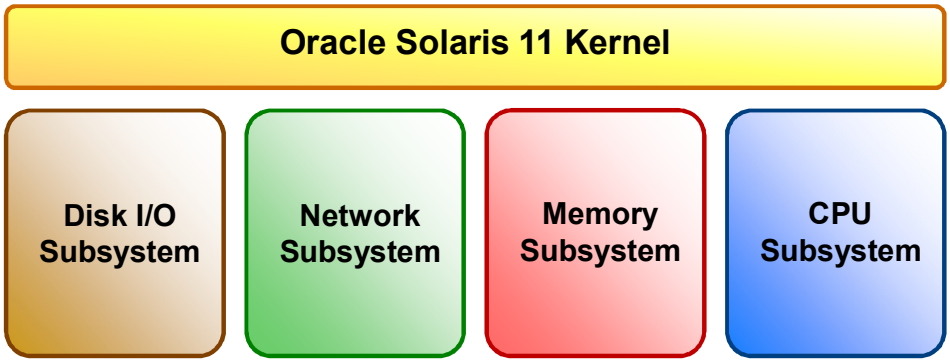
The system starts processes called daemons. Daemons are processes that run in the background and provide services. For instance, the desktop login daemon (`dtlogin`) provides a graphical prompt that you use to log in to the operating system.

Every process has a unique process identification (PID) number that 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 in 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 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 parent process ID (PPID) number
  - The new process is called the child process
- The parent and child processes interact as follows:
  - While the child process runs, the parent process waits.
  - When the child process 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.

# 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 in 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 for 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

A process can be in one of the following states:

| State | Description |
|---|---|
| `run` | The process is in the run queue and running on a CPU. |
| `sleep` | The process is waiting for work. |
| `zombie` | The parent process has terminated. |
| `stop` | The process is stopped. |

# Commands for Managing Processes

| Command | Description |
|---|---|
| `ptree` | Displays the process trees for the specified process ID |
| `ps` | Displays detailed information about the active processes in the system |
| `pgrep` | Displays information about a process based on specific criteria |
| `prstat` | Displays statistics for the active processes in 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. 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 in 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.
- Users with the `root` role 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 may be times when you must terminate an unwanted process. The process may be in an endless loop or you may have started a large job that you want to stop before it is completed. You can kill (stop) any process that you own. A user with the `root` role 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 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 signals presented in the table in the slide can be defined 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 re-read its configuration file.
- **2, SIGINT:** Interrupt signal from your keyboard, usually from a Ctrl + C key combination
- **9, SIGKILL:** 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:** 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.

# Managing System Processes

- Viewing the parent/child process relationship
- 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.

# 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  1345  1280   0   Jul 31 ?           0:01 gnome-panel
---
---

# ptree 1345
1032   /usr/sbin/gdm-binary
  1046   /usr/lib/gdm-simple-slave --display-id /org/gnome/DisplayManager/Displa
    1258   /usr/lib/gdm-session-worker
      1280   gnome-session
        1345   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 process' tree, use the `ptree` command followed by the process ID. The output from the command contains the specified PIDs or users, with the child processes indented from their respective parent processes. In the example in the slide, you are looking at the `gnome-panel` process (PID `1345`). 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 `1345`. As you can see, this process is the child of the `gnome-session` process (PID `1280`), which is the child of the `gdm-session-worker`, and so on up the tree.

# Listing System Processes

To list the active processes in a system, use `ps`.

```
# ps
  PID TTY          TIME CMD
 4605 pts/4       0:00 bash
 4604 pts/4       0:00 su
 5880 pts/4       0:00 ps
```



To manage the processes in the system, you must know what processes are running. To list the processes that are currently running in 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 third 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 process that is currently running, 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    6     0   0 06:50:40 ?           0:02 kmem_task
  root    1     0   0 06:50:43 ?           0:00 usr/sbin/init
...
...
...
```



To generate a more detailed list of every process that is currently running in 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`, `netcfg`, `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 in the slide, which presents a partial output for the command, shows several processes that belong to root and that are associated with the following commands: `sched`, `zpool-rpool`, `kmem_task`, and `usr/sbin/init`.

# Displaying Information About Processes

To display the PID of a particular process, use `pgrep` *process*.

```
# pgrep sched
0
9179
29414
```

```
# pgrep -l manager
  4238 updatemanagerno
  4283 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 in the system: the Update Manager process (PID `4238`) and the NWAM Manager process (PID 4283).

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
------------------------------------------------------------------------
 26264 root       38M  372K run      10    0 183:40:15  95% sysconfig/1
  4297 oracle     99M   75M run      49    0   2:33:14 0.8% java/20
   739 root       39M 9552K sleep    59    0   2:14:44 0.6% pkg.depotd/64
  4668 oracle    131M   20M sleep    59    0   0:01:40 0.6% gnome-terminal/2
   832 oracle     73M   46M sleep    59    0   0:04:55 0.5% Xorg/3
  4327 oracle     13M 2320K sleep    59    0   1:26:15 0.4% VBoxClient/3
  5890 root       11M 3244K cpu0     49    0   0:00:00 0.3% prstat/1
   516 root       11M  916K sleep    59    0   0:08:21 0.1% VBoxService/7
   519 root       19M 6212K sleep    59    0   0:09:17 0.1% named/4
  4185 oracle    128M   16M sleep    59    0   0:00:07 0.0% metacity/1
  4605 root       10M 2672K run      39    0   0:00:00 0.0% bash/1
  4289 oracle    134M   19M sleep    59    0   0:04:33 0.0% isapython2.6/1
  7605 root       14M 4408K sleep    59    0   0:02:40 0.0% nscd/120
    15 root       20M   16M sleep    59    0   0:04:55 0.0% svc.configd/27
  4238 oracle     62M   27M sleep    12   19   0:04:43 0.0% updatemanagerno/1
Total: 198 processes, 1075 lwps, load averages: 1.42, 1.39, 1.43
```

ORACLE



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 in 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
  - **cpuN:** The process is running on the CPU.
  - **sleep:** The process is waiting for an event to complete.
  - **run:** The process is in the run queue.
  - **zombie:** The process is terminated and the parent is not waiting.
  - **stop:** The process is stopped.
- **PRI:** The 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 you 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 `4668)` process is owned by the `oracle` user. It has a total virtual memory size of `131M` and a resident set size of `20M`. 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 1 minute 40 seconds and was using `0.6%` of the CPU's capacity.

The `updatemanagerno/1` process (PID `4238`) is also owned by the `oracle` user. It has a total virtual memory size of `62M` and a resident set size of `27M`. 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 4 minutes 43 seconds and is not utilizing any CPU capacity.

# Displaying Active Process Statistics

```
# prstat -s cpu 20 3
   PID USERNAME  SIZE   RSS STATE    PRI NICE      TIME  CPU PROCESS/NLWP
------------------------------------------------------------------------------
 26264 root       38M  372K run      30    0 186:38:44  96% sysconfig/1
  4297 oracle     99M   75M sleep    49    0   2:34:36 0.8% java/20
   739 root       39M 9552K run      59    0   2:15:45 0.6% pkg.depotd/64
  4327 oracle     13M 2320K run      59    0   1:27:00 0.5% VBoxClient/3
  4668 oracle    131M   20M sleep    59    0   0:01:41 0.2% gnome-terminal/2
  5987 root       11M 3620K cpu0     59    0   0:00:00 0.2% prstat/1
<output omitted>
Total: 199 processes, 1078 lwps, load averages: 1.45, 1.40, 1.38
```

```
# prstat -s rss 20 3
   PID USERNAME  SIZE   RSS STATE    PRI NICE      TIME  CPU PROCESS/NLWP
------------------------------------------------------------------------------
  4297 oracle     99M   75M run      39    0   2:34:38 0.8% java/20
   528 root       61M   58M sleep    59    0   0:00:52 0.0% hald-addon-acpi/1
   832 oracle     74M   47M sleep    59    0   0:05:00 0.3% Xorg/3
 26129 oracle    142M   43M sleep    49    0   0:01:13 0.0% nautilus/3
  4210 oracle    147M   31M sleep    49    0   0:00:04 0.0% nautilus/1
  1354 root      141M   29M sleep    59    0   0:00:03 0.0% gedit/1
  4238 oracle     62M   27M run      12   19   0:04:46 0.0% updatemanagerno/1
  5894 oracle    138M   25M sleep    49    0   0:00:01 0.0% gedit/1
<output omitted>
Total: 199 processes, 1077 lwps, load averages: 1.38, 1.38, 1.37
```



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 in the slide, the `java/20` process is using the most CPU at `0.8%`.

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, 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 in the slide, you have a process called `rptpgm` (for the report program) that is taking 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 steps for this are as follows:

1. Identify the PID for the process by using the `pgrep` command. It is `3366`.
2. Temporarily stop the process by using the `pstop` command.
3. 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
$
```

ORACLE



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 for 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 in a system |
| `pgrep` | Displays information about a process based on specific criteria |
| `prstat` | Displays statistics for the active processes in a system |
| `kill, pkill` | Terminates a process |



The table in the slide summarizes the commands that you can use to list, control, and kill processes, as well as to display process information.

# 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**

# Practice 11-1 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 11-1:** Managing system processes
- **Practice 11-2:** Scheduling system tasks

You will find Practice 11-1 in your *Activity Guide*. It should take about 30 minutes to complete.

# Lesson Agenda

- Managing System Processes
- **Scheduling System Administration Tasks**

# Scheduling a Single Job Using the `at` Command

- You can schedule a job for execution at a later time by using the `at` command.
- The job can consist of a single command or a script.
- The `at` command allows you to schedule the automatic execution of routine tasks.
- `at` files execute their tasks once after which they are removed from their directory.
- The `at` command is most useful for running simple commands or scripts that direct output into separate files for later examination.



The `at` command enables you to schedule a job for execution at a later time. The job can consist of a single command or a script. Similar to `crontab`, the `at` command allows you to schedule the automatic execution of routine tasks. However, unlike `crontab` files, `at` files execute their tasks once. They are then removed from their directory. Therefore, the `at` command is most useful for running simple commands or scripts that direct output into separate files for later examination.

Submitting an `at` job involves typing a command followed by the `at` command syntax to specify options to schedule the time your job will be executed. The `at` command stores the command or script you ran, along with a copy of your current environment variable, in the `/var/spool/cron/atjobs` directory.

The file name for an `at` job consists of a long number that specifies its location in the `at` queue followed by the `.a` extension, for example, `793962000.a`. The `cron` daemon checks for `at` jobs at startup, and listens for new jobs that are submitted. After the `cron` daemon executes an `at` job, the at job's file is removed from the `atjobs` directory. For more information, see the `at`(1) man page.

# Creating an `at` Job

1. Start the `at` utility, specifying the time you want your job to be executed.
2. At the `at` prompt, type the commands or scripts that you want to execute, one per line.
3. Press Control-D to exit the `at` utility and save the `at` job.

```
$ at -m 1930
at> rm /home/jones/*.backup
at> <Press Control-D>
job 897355800.a at Thu Jul 12 19:30:00 2004
```



The `at` utility has the following syntax:

```
at [-m] time [date]
```

where,

- `-m` specifies to send you an email after the job is completed.
- `time` specifies the hour that you want to schedule the job. Add AM or PM if you do not specify the hours according to the 24-hour clock. Acceptable keywords are midnight, noon, and now. Minutes are optional.
- `date` specifies the first three or more letters of a month, a day of the week, or the keywords today or tomorrow.

Your `at` job is assigned a queue number, which is also the job's file name. This number is displayed when you exit the `at` utility.

# `at` Commands

| Command | Description |
|---|---|
| `atq` | Displays status information about the `at` jobs that you have created<br>**Note:** You can also use this command to verify that you have created an `at` job. |
| `at -l` *[job-id]* | Displays information about the execution times of your `at` jobs |
| `at -r` *[job-id]* | Removes the `at` job from the queue before the job is executed |



### Example for Verifying an `at` Job

```
$ atq
Rank Execution    Date  Owner Job          Queue Job Name
1st  Jul 12, 2004 19:30 jones 897355800.a      a  stdin
2nd  Jul 14, 2004 23:45 jones 897543900.a      a  stdin
3rd  Jul 17, 2004 04:00 jones 897732000.a      a  stdin
```

### Example for Displaying an `at` Job

```
$ at -l
897543900.a Sat Jul 14 23:45:00 2004
897355800.a Thu Jul 12 19:30:00 2004
897732000.a Tue Jul 17 04:00:00 2004
$ at -l 897732000.a
897732000.a Tue Jul 17 04:00:00 2004
```

### Example for Removing an `at` Job

```
$ at -l
897543900.a Sat Jul 14 23:45:00 2003
897355800.a Thu Jul 12 19:30:00 2003
897732000.a Tue Jul 17 04:00:00 2003
$ at -r 897732000.a
$ at -l 897732000.a
at: 858142000.a: No such file or directory
```

# Denying Access to the `at` Command

1. Assume the `root` role.
2. Edit the `/etc/cron.d/at.deny` file by using the `pfedit` command.
3. Add the names of users, one username per line, that you want to prevent from using the `at` commands.

```
$ pfedit /etc/cron.d/at.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
...
```



You can set up a file to control access to the `at` command, permitting only specified users to create, remove, or display queue information about their `at` jobs. The file that controls access to the `at` command, `/etc/cron.d/at.deny`, consists of a list of usernames, one username per line. The users who are listed in this file cannot access the `at` commands.

The `at.deny` file, which is created during Oracle Solaris software installation, contains the following usernames:

- `daemon`
- `bin`
- `smtp`
- `nuucp`
- `listen`
- `nobody`
- `noaccess`

With `root` role privileges, you can edit the `at.deny` file to add other usernames whose `at` command access you want to restrict.

### Verifying That the `at` Command Access Is Denied

To verify that a username was added correctly to the `/etc/cron.d/at.deny` file, use the `at -l` command while logged in as the user. For example, if the logged-in user `smith` cannot access the `at` command, the following message is displayed:

```
# su smith
Password:
# at -l
at: you are not authorized to use at. Sorry.
```

Likewise, if the user tries to submit an `at` job, the following message is displayed:

```
# at 2:30pm
at: you are not authorized to use at. Sorry.
```

This message confirms that the user is listed in the `at.deny` file.

If `at` command access is allowed, the `at -l` command returns nothing.

# Scheduling Repetitive System 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

ORACLE



Most administrators have many system tasks that occur on a regular 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 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* | This is the full path name to the command to be run. |



An example of a `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 following 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 after 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.

# 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 user with the root role 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, which 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 the 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.

# Default `cron.deny` File

```
# cat /etc/cron.d/cron.deny
daemon
bin
nuucp
```



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.

# 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, refer to http://docs.oracle.com/cd/E36784_01/html/E36819/index.html.

# 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`.

ORACLE



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
30 17 * * 5 /usr/bin/banner "Time to go!" > /dev/console
# ls -l /var/spool/cron/crontabs
-rw-r--r--   1 root     sys          190 Sep 19 16:23 adm
-rw-------   1 root     staff        225 Nov  5 09:19 jjones
-rw-r--r--   1 root     root        1063 Nov  5 16:23 lp
-rw-r--r--   1 root     sys          441 Sep 19 16:25 root
-rw-------   1 root     staff         60 Nov  5 09:15 smith
-rw-r--r--   1 root     sys          308 Sep 19 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 Sep 19  16:23 adm
-rw-r--r--   1 root     root        1063 Nov  5  16:23 lp
-rw-r--r--   1 root     sys          441 Sep 19  16:25 root
-rw-------   1 root     staff         60 Nov  5  09:15 smith
-rw-r--r--   1 root     sys          308 Nov 19  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
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 name or users' names 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 the `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 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.

# 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**

# Practice 11-2 Overview: 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 how to:

- Explain system processes management
- Manage system processes
- Schedule system administration 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.