AIX 6 System Administration II: Problem Determination

(Course code AU16)

Student Exercises

ERC 14.0

IBM certified course material

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM® is a registered trademark of International Business Machines Corporation.

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX®	AIX 5L™	DB2®
DS4000™	eServer™	FlashCopy®
General Parallel File System™	GPFS™	Micro-Partitioning™
Notes®	POWER™	POWER4™
POWER5™	POWER6™	POWER Gt1™
POWER Gt3™	pSeries®	Redbooks®
RS/6000®	SP™	System p™
Tivoli®	TotalStorage®	xSeries®

Alerts® is a registered trademark of Alphablox Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX® is a registered trademark of The Open Group in the United States and other countries.

Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

December 2007 edition

Contents

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM® is a registered trademark of International Business Machines Corporation.

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX®	AIX 5L™	DB2®
DS4000™	eServer™	FlashCopy®
General Parallel File System™	GPFS™	Micro-Partitioning™
Notes®	POWER™	POWER4™
POWER5™	POWER6™	POWER Gt1™
POWER Gt3™	pSeries®	Redbooks®
RS/6000®	SP™	System p™
Tivoli®	TotalStorage®	xSeries®

Alerts® is a registered trademark of Alphablox Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX® is a registered trademark of The Open Group in the United States and other countries.

Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Exercise Description

Each exercise in this course is divided into sections as described below. Select the section that best fits your method of performing exercises. You may use a combination of these sections as appropriate.

Exercise Instructions

This section tells you what to accomplish. There are no definitive details on how to perform the tasks. You are given the opportunity to work through the exercise given what you learned in the unit presentation, utilizing the Student Notebook, your past experience, and maybe a little intuition.

Exercise Instructions with Hints

This section is also an exact duplicate of the **Exercise Instructions** and contains solutions and additional tips for the students. If very inexperienced students take part in this course, they should work with this section.

Students can use this part to compare their work with the solutions.

When showing the SMIT method to accomplish a task, each line in bold represents a submenu or selector screen. You will need to press the Enter key after selecting each item as listed. When you reach the dialog screen, the field descriptions will be in regular text and the items you need to fill in will be in bold. Only the items that need to be changed will be shown, not the entire screen. Once you have reached the dialog screen portion of SMIT, press Enter ONLY after all indicated entries have been made.

The SMIT steps will be shown for the ASCII version of SMIT. Under most circumstances these steps match the steps taken if using the graphics version of SMIT. The exceptions relate to the use of the function keys. When instructed to press the **F3** key back to a particular menu, when in graphics SMIT, you will instead click the **Cancel** box at the bottom of the screen. When instructed to press the **F9** key to shell out, in graphics mode, simply open another window.

Note: The Web-based System Manager interfaces are currently not shown.

Optional Exercise Parts

Some labs provide additional practice on a particular topic. Specific details and hints are provided to help step you through the **Optional Exercises**, if needed. Not all exercises include **Optional Exercises**.

According to the group, the instructor can decide to do them or not. If there is time, the optional part should be executed by the students.

Text highlighting

The following text highlighting conventions are used throughout this book:

Bold	Identifies file names, file paths, directories, user names, principals, menu paths, and menu selections. Also identifies graphical objects such as buttons, labels, and icons that the user selects.
Italics	Identifies links to Web sites, publication titles, is used where the word or phrase is meant to stand out from the surrounding text, and identifies parameters whose actual names or values are to be supplied by the user.
`Monospace`	Identifies attributes, variables, file listings, SMIT menus, code examples, and command output that you would see displayed on a terminal, and messages from the system.
`Monospace bold`	Identifies commands, subroutines, daemons, and text the user would type.

Exercise 1. Problem Determination Introduction

What This Exercise Is About

This exercise will acquaint you with the system that you will be using throughout this course. You will recall some basic administration commands. This exercise also provides you with an opportunity to work with the Service Update Management Assistant (SUMA).

What You Should Be Able to Do

At the end of the lab, you should be able to:

- List volume groups, physical, and logical volumes on your system
- Identify real memory and paging space on your system
- Identify the hardware platform and processor type of your system
- Use the Service Update Management Assistant (SUMA)

Introduction

In this exercise, you will obtain and record information about your system using some basic administration commands with which you are probably already familiar. You will also use the Service Update Management Assistant (SUMA).

You will require **root** authority to complete this exercise.

Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Recording system information

__ 1. *Using commands rather than SMIT*, collect and record the following information regarding your system:

__ a. The volume groups on your system:

_____

__ b. The physical volumes for your system:

_____

__ c. The logical volumes in rootvg on your system:

_____

_____

_____

__ d. All paging space areas for your system:

_____

__ e. Real memory on your system:

_____

__ f. Hardware platform (architecture) of your system:

_____

__ g. Processor type of your system:

_____

__ 2. Identify the logical volumes that reside on your **hdisk0**.

Write down the command you used:

_____

From the fact that the number of LPs is equal to the number of PPs, what can you conclude?

_____

Using the Service Update Management Assistant (SUMA)

__ 3. Display the **man** page for the **suma** command.

Locate the examples illustrating use of this command. How many such examples are there?

_____

__ 4. As illustrated in the suma man page examples (example 7), enter a **suma** command that will create and schedule a repeating (**-a Repeats=y**) task that will check for the latest level of the **bos.rte.install** fileset on the 15th of each month at 3:30 AM.

__ 5. Enter a **suma** command that will list information regarding the SUMA task you just created.

End of exercise

Exercise 2. The Object Data Manager (ODM)

What This Exercise Is About

This exercise will review some of the most important ODM files and how they are used in device configuration. You will use the ODM command line interface.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Describe some of the most important ODM files
- Use the ODM command line interface
- Explain how ODM classes are used by device configuration commands

Introduction

This exercise has three parts:

1. Review of device configuration ODM classes (**PdDv**, **PdAt**, **CuDv**, **CuAt**, **CuDep**, **CuDvDr**).
2. Role of ODM during device configuration.
3. Optional Part: Creating self-defined ODM classes.

All instructions in this exercise require **root** authority.

Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Review of device configuration ODM classes

__ 1. Execute the **lsdev** command and identify all devices that are supported on your system. Tell the **lsdev** command to provide column headers in the output.

What is the command you used?

_____

Which ODM class is used by the **lsdev** command to generate this output?

_____

__ 2. Execute the **lsdev** command and identify all disk devices that are currently attached to your system. Tell the **lsdev** command to provide column headers in the output.

What is the command you used?

_____

Which ODM class is used by the **lsdev** command to generate this output?

_____

__ 3. Request the same listing as above, except customize the reported fields needed to complete the following list for disk **hdisk0**:

Name: _____

Status: _____

Location: _____

Physical location: _____

Description: _____

__ 4. Use the ODM command line interface and list the ODM object that describes this disk device. Also, use the ODM command line interface to list the ODM object that give the physical location code as part of the Vital Product Data information.

What command(s) did you used?

_____

From the output complete the following list for disk **hdisk0**:

Status: _____

Chgstatus: _____

Parent: _____

Location: _____

Connwhere: _____

PdDvLn: _____

Physical Location: _____

__ 5. From this output please answer the following questions.

What is the meaning of the descriptor `chgstatus`?

_____

The **`lsdev`** command provides a description field, which is not part of ODM class **CuDv**. Where does the description come from?

_____

_____

_____

_____

__ 6. Execute the **`lsattr`** command and identify the *physical volume identifier* for your **hdisk0**.

What is the command you used?

_____

Write down the physical volume ID of the disk:

pvid: _____

__ 7. Use the ODM command line interface, and list the ODM object that stores the *physical volume identifier*:

What is the command you used?

_____

__ 8. The /**dev** directory contains the special files to access the devices. Write down the major and minor number of the special file for **hdisk0**.

Major number: _____

Minor Number: _____

Which ODM class is used to identify the major number and minor number for the device driver?

_____

__ 9. List all your logical volumes that are part of the **rootvg**.

What is the command you used?

_____

Query the ODM class **CuDep** and identify all logical volumes that belong to **rootvg**.

What is the command you used?

_____

Role of ODM during device configuration

__ 10. During the following steps we will simulate the configuration of an SCSI disk without using **cfgmgr**.

Important: *This is just a simulation. You do not attach a real disk in this exercise. The ability to simulate a non-existent disk depends on having a physical SCSI adapter which is in an available state.*

For lab systems which only have virtual SCSI, this exercise section ends after this step (you may wish to go to the start of the following optional part).

The ODM contains predefined objects to support many types of different disks.

Use the **lsdev** command to list all predefined devices of class `disk`.

Write down the command you used.

_____

Locate the predefine device object that support the disks your system has configured. Use the PdDvLn value you previously recorded in step 4 to make the match up.

For example, if you have physical disks you might have a disk type of `osdisk` and a subclass of `scsi` (`Other SCSI Disk Drive`). Or, if you have virtual disks, you would have a disk type of `vdisk` and a subclass of `vscsi`.

Use **odmget** to identify the object in **PdDv** that describes your disk type and subclass.

Write down the command you used.

_____

From the output complete the following:

type: _____

class: _____

subclass: _____

prefix: _____

Device Driver: _____

Configuration Method: _____

__ 11. We will use the device **scsi0** (or an alternative device such as **scsi1** or **scsi2** if specified by your instructor) as the *parent device* for the disk we are configuring. This is where the new disk will be attached to the system.

__ 12. Before configuring the device, a free SCSI ID must be identified. List all ODM objects in **CuDv** where **scsi0** (or the alternative device specified by your instructor) is stored as the parent device.

Write down the command you used.

_____

From the output, write down the SCSI IDs which are in use:

SCSI IDs in use: _____

Choose a free SCSI ID and write it down in the table. You need to specify this ID later.

Free SCSI ID: _____

__ 13. Get the disk into the *defined* state using the **`mkdev -d`** command. You need to pass the following information to **`mkdev`**:

- Device class
- Device subclass
- Device type
- Parent device
- SCSI address

Write down the command you used to define the disk.

_____

What device name has been assigned to the disk?

Device name: _____

__ 14. Using this newly assigned name, list the object that stores information about your disk in the customized database.

Write down the command you used:

_____

__ 15. Try to configure the disk using **`mkdev -l`**. What happens?

_____

_____

__ 16. Finally, remove the disk from the system using **rmdev**.

Write down the command you used:

_____

Examine your **CuDv** object class. Did you find the removed disk in this object class?

_____

Optional Part: Creating self-defined ODM classes

__ 17. Before creating an ODM class you need to specify the descriptors that are contained in the class. Create the directory /**tmp**/**odm** to hold the specification file and **cd** to that directory.

Using an editor, create a file **parts.cre** (in your new working directory) with the following class structure:

```
class parts {
long        part_number;
char        part_description[128];
char        warehouse[4];
long        contained_in;
}
```

__ 18. Create the ODM class using this class structure and check the structure of this class. Write down the commands you used:

_____

_____

Identify in your working directory, which files have been created during this step.

_____

What do you think is the purpose of these files?

_____

_____

_____

Where does the ODM class **parts** reside?

_____

__ 19. Create some objects in ODM class **parts**, using the following data:

Part Number	Description	Warehouse	Contained In
10001	Wheel	a12	50001
10003	Frame	a19	50001
10005	Saddle	a01	50001

Part Number	Description	Warehouse	Contained In
10006	Front wheel brake	a03	50001
10007	Rear wheel brake	a03	50001
50001	City Bike Easy Rider	x99	

__ 20. List all objects that are contained in part `50001` (the `City Bike Easy Rider`). Write down the command you used:

_____

__ 21. Change the `warehouse` location for part `Wheel` to `b10`.

__ 22. Finally, remove ODM class parts from the system. Write down the command you used.

_____

End of exercise

Exercise 3. System Initialization Part 1

What This Exercise Is About

This exercise will review the hardware boot process of an AIX system.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Boot a machine in maintenance mode
- Repair a corrupted boot logical volume
- Alter bootlists

Introduction

This exercise has three parts:

1. Identify the bootlists of your system
2. Identify LVM information of your system
3. Repair a corrupted boot logical volume

All instructions in this exercise require **root** authority.

Requirements

- The program **`/home/workshop/ex3pro1`**
- Bootable media that matches the version and release of your system or a NIM server setup that can be used to execute a remote boot).

Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Part 1 - Working with bootlists and identifying information on your system

__ 1. What is the boot sequence of your system for a **normal** boot?

- Boot device:_____
- What is the command you used, to determine the bootlist? _____

__ 2. Does your model support a customized **service** bootlist? _____

- What command did you use? _____

- If your model supports a service bootlist for maintenance mode, write down the boot sequence for this boot mode:
  1) Boot device:_____
  2) Boot device:_____
  3) Boot device:_____
  4) Boot device:_____

__ 3. Identify which disks are contained within the **rootvg**:

_____

- What command did you use? _____

- Which disk is the bootable disk? (That means the disk that contains the boot logical volume **hd5)**: _____

  What command did you use? _____

- What is the **logical volume type** of **hd5**? _____

  What command did you use? _____

Part 2 - Identify LVM information from your system

__ 4. Alter the boot sequence of your system using the **bootlist** command. Set the **normal** bootlist so it contains only the bootable hard disk.

__ 5. The Logical Volume Manager uses names *and* IDs when storing information. Complete the following table that maps names to IDs:

rootvg VGID	
First disk PVID	
Second disk PVID	

- What command did you use to determine the **rootvg** VGID? _____
- What command did you use to determine the physical volume IDs? _____
- Using **odmget**, identify the attribute `pvid` of one of your disks from ODM class **CuAt**.
  - What command did you use?_____
- What difference do you see with the ID value?

Part 3 - Booting to maintenance mode

__ 6. Before creating any boot problems, verify that you can boot into maintenance mode. This will be crucial to fixing the problem.

For **local machines**, where your system console is a physical display, the procedure is fairly straight forward:

i. Place the provided installation media in the CD/DVD drive.

ii. Shut down your machine by using the shutdown -F command from the root level prompt.

iii. Power on your machine.

iv. Soon after the LED reads E1F1, the keyboard is discovered and a distinctive beeping will sound. Between that time and before the system begins to search for the boot image, press either the **F6** or numeric **6** key (depending on type of machine) to invoke the service mode bootlist.

v. The console should eventually display the **Installation and Maintenance** menu.

vi. Power off your machine and power it back on (which allows it to boot normally to multi-user mode).

For **remote machines** the procedure, at a high level, is as follows:

i. Shut down your AIX operating system by running the **shutdown -F** command from the AIX root level command prompt.

ii. Access the HMC and locate the icon for your LPAR.

iii. Activate the LPAR into maintenance mode, using the service bootlist.

iv. Shut down the LPAR from the current maintenance mode.

v. Start the LPAR back up into multi-user mode.

Except for the shutdown of a running AIX operating system, details of this will depend on the level of HMC with which you are working. The details for the different HMC versions are on the following pages:

If using **WebSM** with a version of **HMC prior to version 7**:

i. Access the HMC and locate your LPAR:

1) Start the Web-based System Manager client (icon on your lab workstation desktop). Note that the lab workstation may be a portal machine at the remote server location.
2) Enter the IP address of your assigned HMC (provided by instructor) and click **OK**.
3) Enter the provided userid and password to login.
4) In the navigation area, click on the address of the HMC.
5) In the content area on the right, double-click the **Server and Partition** icon.
6) Double-click the **Server Management** icon.
7) Click the **+** next to the name of the assigned managed system (provided by instructor).
8) Click the **+** next to **Partitions**.
9) Look for the name of your assigned partition (provided by instructor).

ii. Activate your LPAR into maintenance mode (with customized service bootlist):

1) When the partition state is **Not Activated**, proceed to activate the partition.
2) Select the partition and right click to get the context menu.
3) On the menu, click **Close Terminal Connection** and confirm when prompted.
4) Select the partition and right-click to get the context menu.
5) On the menu, click **Activate**.
6) When the **Activate Logical Partition** panel appears, click the box next to **open terminal window**, and click **OK** (the intent is to have you manually signal which bootlist to use later in step 9).
[Alternatively, you may choose to click the “Advanced” button and specify “Diagnostic with Stored Bootlist” as the boot mode. In that case you would not need to follow step 9.]
7) A terminal console window should appear.
8) The virtual console will display the discovered devices (soon after the HMC shows an operator panel value of E1F1).
9) When the keyboard is discovered, a distinctive beeping will sound and the “keyboard” key word will appear. Between that time and before the system begins to search for the boot image, press the numeric **6** key to invoke the service mode bootlist. This happens very quickly for LPARs, and soon after

staring the partition you may want to simply press the **6** key periodically right after activation.

10) The console should eventually display the **Installation and Maintenance** menu.

iii. Shut down the partition from maintenance mode (if accessing the root level command prompt, simply run **`shutdown -F`**):

1) In the HMC **Server and Partition** panel, right-click your partition icon.
2) On the resulting menu, click **Close Terminal Connection** and confirm when prompted.
3) On the **Server and Partition** panel, right-click your partition icon, and select **Shutdown Partition**.
4) On the **Shutdown Partitions** panel, select **immediate** and click **OK**.

iv. Start your partition in multiuser mode (normal bootlist):

1) When the partition state is **Not Activated** proceed to activate the partition.
2) Select the partition and right-click to get the context menu.
3) On the menu, click **Activate**.
4) When the **Activate Logical Partition** panel appears, click the box next to **open terminal window**, and click **OK**.
5) A terminal console window should appear.
6) The **Server and Partition** operator panel will display the LED values during boot up.
7) When the **`init`** process starts, the virtual console will display the remaining boot progress.
8) You should eventually receive a login prompt.

v. Opening a **virtual console**

1) Access the HMC and locate your LPAR (as described in earlier).
2) Select the partition and right-click to get the context menu.
3) Select **Close Terminal Connection** and confirm when prompted.
4) Select the partition and right-click to get the context menu.
5) Select **Open Terminal Window**.

If using a **Web browser** with **HMC version 7 or later**:

i. Access the HMC and locate your LPAR:

1) Start a browser on your lab workstation (note that the workstation may be a portal machine at a remote location).

2) Enter a URL of: **https://<IP address of your HMC>**.

This will take you to an HMC status window which has 3 status indicators and a link with the text:

"Log on and Launch the Hardware Management Console web application"

3) Click the log-on link to launch the HMC logon panel.

4) Enter the userid of **hscroot** and the password (either abc123 or abc1234) and click the logon button. This should launch the HMC Web interface.

5) In the left navigation area click **Systems Management**. The Systems Management item should expand to show "Servers" and "Custom Groups."

6) Click the **Servers** item. The "Servers" item should expand to show the managed systems.

7) Click the managed system which is assigned to your team. In the Content Area on the right, you should see a list of logical partitions defined for your assigned system.

8) Select your assigned logical partition by clicking the box under "Select" for your LPAR. After a short delay you should see a small menu icon appear to the right of your LPAR name, and the Tasks Area on the bottom half of the panel should update to reflect operations which are appropriate for the selected target.

9) If you left-click the new menu icon (to right of the LPAR name), then you should see a menu which is similar to what you see in the Tasks Area.

ii. Activate your LPAR into maintenance mode (with service bootlist):

1) When the partition state is **Not Activated**, proceed to activate the partition.

2) Select the partition (if not already selected).

3) When the small menu icon appears, click it to show the menu and click the **Operations** task.

4) When the subtasks appear, click the **Activate** subtask.

5) In the pop-up window labeled **Activate Logical Partition: <your lpar name>**, click the small box next to **Open a terminal window or console session** and also click the **Advanced** button. This should result in a new pop-up window labeled "Activate Logical Partition - Advanced".

6) In the new pop-up window, click the menu icon to the right of “Boot Mode” and select **Diagnostic with stored bootlist**.

   You may alternately boot in Normal mode, but you would then have to press numeric 6 at the appropriate time (right after keyboard discovery) to cause a service boot. Click **OK** to exit this pop-up.

7) On the **Activate Logical Partition: <your lpar name>**, click **OK**. A virtual terminal window should appear and you should see information about the progress of the BOOTP request, followed by a “Welcome to AIX” display, and finally followed by a prompt to “Define the System Console”.

8) Respond to the various boot prompts until you see the “Maintenance” menu.

iii. Shut down the partition from maintenance mode (if accessing the root level command prompt, simply run **`shutdown -F`**):

1) On the HMC Content Area, make sure your LPAR (and only your LPAR) is currently selected.

2) Click the menu icon, click the **Operations** task and then click the **Shutdown** subtask. This should result in a pop-up window.

3) In the shutdown window, select **Immediate** and then click **OK**.

4) The partition shutdown is complete when the “Status” field for your LPAR changes from “Running” to “Not Active”.

iv. Start your partition in multiuser mode (normal bootlist):

1) When the partition state is **Not Activated**, proceed to activate the partition.

2) Select the partition (if not already selected).

3) When the small menu icon appears, click it to show the menu and click the **Operations** task.

4) When the subtasks appear, click the **Activate** subtask.

5) In the pop-up window labeled “Activate Logical Partition: <your lpar name>”, click the small box next to “Open a terminal window or console session” (unless you already have a virtual console window open) and also click **OK**.

6) You should eventually see a login prompt appear in the virtual console window.

v. Opening a **virtual console**

1) Locate and select your LPAR, as described earlier.

2) Left-click the menu to the right of your LPAR name.

3) Left-click the “Console Window” item.
4) Left-click the “open terminal connection” item.

Part 4 - Repair a corrupted boot logical volume

__ 7. Execute the program **/home/workshop/ex3pro1**. When the prompt is returned, shut down and reboot the system.

__ 8. What happens on your system during the reboot?

__ 9. Boot to maintenance mode to do the repair:

__ 10. Repair the boot logical volume.

The procedure for using the maintenance menu to repair the boot logical volume is the same for all environments:

- Access the **rootvg** with all mounted file systems.

__ 11. In the maintenance shell, check that hdisk0 is in the normal bootlist and that the rootvg actually it has a boot logical volume on it. Correct if needed.

__ 12. In the maintenance shell, rebuild the boot image on the boot logical volume. Write down the command you used.

»

__ 13. If the command executes successfully, reboot your system in normal mode.

End of exercise

Exercise 4. System Initialization Part 2

What This Exercise Is About

This exercise will review the software boot process of an AIX system.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Boot a machine in maintenance mode
- Repair a corrupted log logical volume
- Analyze and fix an unknown boot problem

Introduction

This exercise has two parts:

1. Repair a corrupted log logical volume
2. Analyze and fix a boot failure

All instructions in this exercise require **root** authority.

Required Material

- Program **`/home/workshop/ex4pro1`**
- Bootable media that matches the version and release of your system or a NIM server setup that can be used to execute a remote boot)

Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Part 1 - Repair a Corrupted Log Logical Volume

Before starting the lab, read the following paragraph carefully.

If you have any questions, ask the instructor **before** starting the exercise steps.

Files or directories which are created or updated are stored with their i-nodes and the superblock of the file system in memory first. Most write requests are handled in memory first to improve system performance. Every minute or 16 KB of changes the **`syncd`** daemon writes the changes from memory to disk.

All changes in the JFS file systems (superblock, i-nodes, list of free data blocks, and so forth) are recorded in a log logical volume. The **rootvg** uses as default the log logical volume /**dev**/**hd8**. When the changes are written to the disk, the JFS transactions are removed from the log logical volume. This guarantees the integrity of a file system. Until the file system changes are written to disk, the changes are recorded and held in the log logical volume.

In this part of the lab, we corrupt the jfslog to stress a boot failure.

__ 1. Check to see if your **rootvg** file systems are JFS or JFS2. You will need this information later in this exercise.

__ 2. Execute the program **`/home/workshop/ex4pro1`**. This program may take as long as 30 seconds to run. It will shut down your machine.

__ 3. Power on your system.

__ 4. What happens during the reboot? Examine your Student Guide to find an explanation for the boot failure.

_____

_____

__ 5. Boot your machine in maintenance mode.

__ 6. From the maintenance menu access the **rootvg** before mounting the file systems. You need to do this, because mounting the file systems in **rootvg** will fail due to the corrupted log logical volume.

__ 7. Reformat the journal log logical volume. Be sure to do a file system check for all file systems that use /**dev**/**hd8**. If you like, use **`set -o emacs`** or **`set -o vi`**.

__ 8. Shut down your system and reboot your system in normal mode.

Part 2 - Analyze and fix a boot failure

__ 9. What happens during the reboot of the system? Write down the last LED code that is shown. What type of problem is this indicative of?

_____

_____

_____

__ 10. Reboot the system to maintenance mode.

__ 11. Examine your system and find the corrupted file that leads to the boot failure.

Be sure to set the `TERM` variable to `lft`, if you are working on a graphical display. Otherwise **`vi`** or SMIT will not work correctly in the maintenance shell.

__ 12. Repair the corrupted file. You will find an example in your student notebook. If you are not able to fix the boot failure, contact your instructor.

End of exercise

Exercise 5. LVM Tasks and Problems

What This Exercise Is About

This exercise has two parts. In the first part, you will be asked to complete some common LVM tasks. In the second part, you will analyze and fix LVM-related ODM problems. Two different lab sessions will be devoted to this exercise. So, you should stop after completing the first part of the exercise and not continue with the second part of the exercise.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Complete common LVM tasks
- Analyze an LVM-related ODM problem
- Fix an LVM-related ODM problem associated with the **rootvg**

Introduction

This exercise has two parts:

1. In the first part, you will complete some common LVM tasks. This may be a review for some members of the class.

   You should do this part during the first lab session allotted to this exercise.

2. In the second part, which has two sections, you will be asked to analyze and fix LVM-related ODM problems:

   a. Analyze and fix an LVM ODM failure manually.

   b. Analyze and fix an LVM ODM failure by using **`rvgrecover`**.

   You should do this part during the second lab session allotted to this exercise.

You will need **root** authority to complete this exercise.

Requirements

- **/home/workshop/ex5_corrupt_pvid**
- **/home/workshop/ex5_corrupt_odm**
- **/home/workshop/rvgrecover**

Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Part 1: Basic LVM Tasks

__ 1. If you have a disk which is not part of the rootvg, extend the rootvg to include that disk. You may get an error if the additional disk appears to already belong to a volume group. This can happen if the disk was not properly removed from a previous volume group (reducevg); in that case, use the force option (-f).

__ 2. Using **smit mklv**, create a *mirrored logical volume* with the name **mirrorlv**. Make it two logical partitions in size.

Use **lslv -m** to identify the physical partitions that have been assigned to your logical partitions. Use the output produced to complete the table below:

LP	PP1	PV1	PP2	PV2
0001				
0002				

Finally, remove the logical volume **mirrorlv**.

__ 3. Use **lspv -p** to determine a region (`outer edge`, `outer middle`, `center`, `inner middle`, `inner edge`) on **hdisk0** (or some other disk if so directed by your instructor) that has space available.

Record the region or regions with free partitions in the space provided below:

_____

__ 4. Use **smit mklv** to created an unmirrored logical volume **lvtmp1** on hdisk0 with a size of one logical partition. Choose an intra-physical policy that will choose physical partitions in a region where free partitions exist.

Use **lspv -p** to check where the partitions of **lvtmp1** reside.

__ 5. Using **smit chlv** or the **chlv** command, change the intra-physical policy to another disk region. Have the partitions been moved to another region?

_____

If not, use the **reorgvg** command. Use the **man** pages to identify how to reorganize a logical volume.

*Note: Do not reorganize the complete **rootvg**, because this takes too much time!*

Write down the command you used:

_____

Using **`lspv -p`** check where the partitions of **lvtmp1** reside now.

__ 6. Find two free partitions on a disk. Write down the partition numbers:

_____

Create a logical volume **lvtmp2** that uses an *allocation map*. The logical volume should have a size of two partitions and should use the two partitions you identified before. Here is an example for an allocation map:

```
hdisk0:82-83
```

After creating the logical volume, check where the partitions reside.

__ 7. What would be the maximum number of disks allowed if we created a volume group using the following command:

```
# mkvg -B -t 4 -y homevg hdisk11 hdisk99
```

End of Part 1

(If you are doing "Part 1" of this exercise, stop here. Do not go on to "Part 2.")

Part 2 - Fixing LVM-related ODM Problems

If you are doing "Part 2" of this exercise, start here.

Section 1: Analyze and Fix an LVM-related ODM Problem

__ 7. Execute **lspv** without any options to list all physical volumes in your system.

Complete the following table.

Disk name	PVID	Volume group

__ 8. Execute **lsvg -p** to list all physical volumes that are part of your **rootvg**.

Complete the following table:

PV_NAME	PV STATE	TOTAL PPs

__ 9. Execute **odmget -q** to see the disk information stored in ODM. Write down the command you used:

_____

PV_Name	PVID

Also, write down the structure of the stanza (that is, information labels) output by the above command. You will need this information in a later lab step.

_____

_____

_____

_____

_____

_____

_____

__ 10. Execute the program **/home/workshop/ex5_corrupt_pvid**.

__ 11. Repeat the **lspv** command to list your physical volumes. Complete the table and compare with the table from step 1.

Disk name	PVID	Volume group

__ 12. Repeat the **lsvg -p** command you used earlier to list the physical volumes in **rootvg**.

What is the output from the command?

_____

_____

_____

_____

__ 13. You learned that LVM stores information about volume groups, physical volumes and logical volumes in the ODM. Consider the output of **lspv** and **lsvg -p**. What data is missing? Where is the problem?

- Volume group objects?
- Physical volume objects?
- Logical volume objects?

Write down what you suspect:

_____

__ 14. Depending on your suspicion, identify the ODM entries which are shown in your student notes in Unit 5.

Find out which objects in which ODM class are missing by reviewing the material from your *student notes*.

_____

__ 15. Before you fix the problem, please consult one VGDA for each of the VGs on your system and compare the missing information with the data in the VGDA. Be sure that the information you wrote down in the tables above is correct, otherwise you will not be able to fix the problem.

What command allows you to query a VGDA?

__ 16. Fix the ODM problem by adding the missing objects into the ODM. Please work very carefully in this step!

Recover the missing entries for all disks that show a problem in the **lspv** listing, not just the ones that are part of a particular volume group.

Use your *student notes* to find out the layout of the corresponding ODM class. Write down the steps you executed to fix the problem.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

__ 17. Repeat the commands **lspv** and **lsvg -p** to check whether your fix works.

If you still have problems, the stanza file you created contains a typo. Find the typo, delete the objects you just created, and add the fixed file. Did you remember to include the 16 trailing zeros on your `pvid` valve?

Section 2: Analyze and Fix an LVM-related ODM Problem Using rvgrecover

__ 18. Execute the program **/home/workshop/ex5_corrupt_odm**.

__ 19. Verify the following information:

__ a. Check whether your volume groups are ok. Use **lsvg**.

__ b. Check whether your physical volumes are ok. Use **lspv**.

__ c. Check whether your logical volumes are ok. List all logical volumes that are part of your **rootvg**. Use **lsvg -l rootvg**.

What happens?

_____

_____

_____

__ 20. Display information for logical volume **hd2**. Use **`lslv hd2`**.

What happens?

_____

_____

_____

__ 21. Analyze the ODM problem by reviewing your student notebook. Compare the ODM entries for logical volumes from “Unit 5” with the ODM objects from your system.

What causes the ODM problems?

_____

_____

_____

__ 22. Examine the **`/home/workshop/rvgrecover`** script and modify it if necessary to match your situation (the specified disk must be one in your **rootvg**).

After making any required changes to the script, fix the ODM problem by executing **`/home/workshop/rvgrecover`**. Ignore the messages. This may take up to 1 minute, depending upon the speed of your lab system.

Check that your ODM problems have been fixed. Repeat **`lsvg -l rootvg`** and **`lslv hd2`**. They should work now without problems.

__ 23. Look into /**home**/**workshop**/**rvgrecover**. What two main steps fix your ODM problem?

__ 24. Remove the logical volumes **lvtmp1** and **lvtmp2** that were created in the first part of this exercise.

End of Part 2

End of exercise

Exercise 6. Mirroring rootvg

What This Exercise Is About

This exercise covers the process required to mirror the **rootvg**.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Mirror the **rootvg**
- Describe physical volume states
- Unmirror the **rootvg**

Introduction

In this exercise, you will mirror and unmirror **rootvg**.

Completion of this exercise requires **root** authority.

Requirements

- **/home/workshop/ex6_diskfailure**

Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Exercise: Mirror and Unmirror the Complete rootvg

__ 1. Write down on which disks your **rootvg** resides.

_____

_____

You might have a mixed installation, where the **rootvg** logical volumes are spread over two disks. Knowing which logical volumes reside on which disk is important because later (if you use **`mklvcopy`**), you will need to specify the target disk for the new mirror for each logical volume.

Which command displays the logical volumes that are contained on a disk?

_____

Execute this command for each disk in the **rootvg** on your system.

__ 2. Now mirror each logical volume as described in your student notebook. If you have a mixed **rootvg** installation, you must be careful when specifying the target disk name.

Do not synchronize the logical volumes in this step.

Write down the commands you executed in this step:

_____

_____

_____

_____

_____

_____

_____

_____

_____

__ 3. Display information about your **rootvg** using the command **`lsvg rootvg`**.

Use the output of the **`lsvg rootvg`** command to record the following information:

- Stale physical volumes: _____

- Stale physical partitions: _____

__ 4. Now, synchronize your **rootvg**. Depending on your system, this step may take 5 to 10 minutes to complete, depending on the speed of your system. This may be a good point to take a break.

Write down the command you executed:

_____

__ 5. Check by using **`lsvg rootvg`** that all partitions have been updated.

__ 6. Enter the following command to see which disks on the system are bootable:

```
# ipl_varyon -i
```

Examine the BOOT DEVICE column in the output produced by this command. Does the value YES appear in this column for both of the physical volumes in **rootvg**?

__ 7. Update your boot logical volumes and your bootlist (increase the size of the /tmp filesystem, if needed).

Write down the commands you executed:

_____

_____

__ 8. Use the **`ipl_varyon`** command again to see which disks on the system are bootable now.

Examine the `BOOT DEVICE` column in the output produced by this command. Does the value `YES` now appear in this column for both of the physical volumes in **rootvg**?

__ 9. Use the **`-m`** flag of the **`lslv`** command to confirm that there is a copy of the logical partition for the boot logical volume on the second disk in your **rootvg**.

__ 10. Now, reboot your system. When the system comes back up, check to see what device the system booted from.

Write down the commands you executed.

_____

_____

__ 11. The procedure **`/home/workshop/ex6_diskfailure`** simulates a disk failure. This procedure requires that your **rootvg** is mirrored completely.

Clear the error log and then execute the program: **`/home/workshop/ex6_diskfailure`**. Create some files in the /**tmp** file system after running the program.

__ 12. Analyze your AIX error log. Detailed information about working with the AIX error log is not provided until later in this course, so use SMIT to display the information in the error log. You may first want to look at the summary report:.

__ 13. The detailed report can get very long if you do not filter it, so you might only request hardware errors. In our case, we saw both hardware and software errors and the summary report was not very long; thus request the detailed report without any filtering.

Browse through your error report and identify the error log entries that have been created by the LVM. Note that the only information identifying the disks involved are the major and minor numbers, which are in hex and need converting to decimal before comparing to an **`lsdev`** listing. Here is some room for you to make notes about the error log entries:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

__ 14. Use the command **`lspv hdiskx`** to display information about each of the physical volumes in **rootvg**. Analyze the `PV STATE` field to check the state of each disk. Also check the `STALE PARTITIONS` entry for each of the disks.

Which disk is causing problems? Which `PV STATE` has been allocated to the failing disk?

_____

_____

_____

__ 15. Review the page in your *Student Notebook* that describes “Physical Volume States.” Find the physical volume state of the failing disk on the visual.

__ 16. Execute a **`varyonvg rootvg`** and check whether this fixes the disk problem.

_____

What happens? Check the `PV STATE` of the failing disk.

_____

_____

__ 17. By reviewing your *Student Notebook*, determine which command will bring the disk back into the `active` state.

_____

Execute this command and check the `PV STATE` of the failing disk.

_____

__ 18. Check whether your **rootvg** still contains *stale* partitions.

_____

_____

__ 19. Check your *Student Notebook* again. What command is the best to fix the stale partitions?

_____

Execute the command and check if the stale partitions are fixed.

__ 20. Unmirror the **rootvg** of your system.

Important: Unmirror your **rootvg** in a way so that one disk is *completely empty*. We need an empty disk in our next exercise.

Decide which of your disks you want to remove all mirrors.

Write down the command you executed to unmirror your **rootvg**:

_____

What recommendation regarding the **chpv** command do you get when you execute the **unmirrorvg** command?

_____

Follow this recommendation and clear the boot record.

_____

__ 21. Check that all logical volumes have been removed from the disk and remove the “empty disk” from **rootvg**.

_____

_____

__ 22. Finally update your boot logical volume and your bootlist.

Then, reboot your system. When the system comes back up, check to see what device the system booted from.

Write down the commands you executed:

_____

_____

_____

End of exercise

Exercise 7. Exporting and Importing Volume Groups

What This Exercise Is About

This exercise describes the steps to export and import volume groups.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Export a volume group
- Import a volume group

Introduction

As you have learned in this course, export and import can be used to move data from one system to another. Sometimes it is the only way to correct ODM failures with non-**rootvg** volume groups.

This exercise has two parts:

- Export and import a volume group
- Analyze import messages (Optional)

This exercise requires one disk to be completely empty. This disk will be used to create a new volume group. This volume group will be exported and imported.

All instructions in this exercise require **root** authority.

Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Part 1 - Export and import a volume group

__ 1. Create a new volume group named **datavg** on a disk that is empty. Check that this disk does not belong to another volume group. Set the physical partition size to 16 MB.

Write down the command you executed to create the new volume group:

_____

__ 2. Check if the new volume group has been varied on automatically. Write down the command you used.

_____

__ 3. Use the fastpath **`smit mklv`** to create a logical volume in **datavg** with the following characteristics:

- Logical volume name: **lv_raw**
- Number of logical partitions: 1

__ 4. Use the fastpath **`smit jfs`** to create two standard journaled file systems in **datavg** with the following characteristics:

- Size of file systems: 16 MB (65536 512-byte blocks)
- Mount points:
  - File system 1: /**home**/**jupiter**
  - File system 2: /**home**/**mars**

Note: Do NOT build these file systems on the **lv_raw** logical volume created in the previous step.

__ 5. Verify the new logical volumes are in **datavg** with the `lsvg` command.
Fill in the following table with the logical volume information in **datavg**:

LV NAME	TYPE	MOUNT POINT

__ 6. Mount the file systems and create some files in both file systems.

__ 7. Export the **datavg** volume group from your system.
Write down all the steps you executed to export the volume group.

_____
_____
_____

__ 8. Analyze your system to see if it contains any reference to the exported volume group. For example, check whether the file systems you have created exist. (Check /**etc**/**filesystems**.)

__ 9. Import the volume group into your system. Specify volume group name **datavg**, otherwise the system will generate a new volume group name.

Write down the command you executed:

_____

__ 10. Check whether the imported volume group, **datavg**, is varied on.

Check to see if the file system information is back.

__ 11. Mount the /**home**/**jupiter** and /**home**/**mars** file systems.
Check that no files have been lost.

Part 2: Analyze import messages (Optional)

In *Part 1: Export and import a volume group*, the export and import worked without problems, as the logical volumes and file systems did not exist during the import of the volume group.

This part will show what will happen when a volume group that is being imported has the same logical volume names of those that already exist on the system.

__ 12. Export the **datavg** volume group again. Repeat the steps from the last export.

__ 13. Use the fastpath `smit mklv` to create a logical volume in **rootvg** with the following characteristics:

- Logical volume name: **lv_raw**
- Number of logical partitions: 1

__ 14. Use the fastpath `smit jfs` to create two standard journaled file systems in **rootvg** with the following characteristics (these are the same as in Part 1).

- Size of file systems: 16 MB (65536 512-byte blocks)
- Mount points:
  - File system 1: /**home**/**jupiter**
  - File system 2: /**home**/**mars**

__ 15. What are the corresponding logical volume names that have been created for the file system?

Logical volume for /**home**/**jupiter**: _____

Logical volume for /**home**/**mars**: _____

__ 16. Mount the /**home**/**jupiter** and /**home**/**mars** file systems, and add a few files to each.

__ 17. At this stage, the following problems will come up when you import the **datavg** volume group:

- The **lv_raw, lv00** and **lv01** logical volumes already exist in **rootvg**
- The /**home**/**jupiter** file system already exists in **rootvg**
- The /**home**/**mars** file system already exists in **rootvg**

Let's see how **importvg** will react to this situation.

Import the **datavg** volume group into the system.

__ 18. Write down the new logical volume names that are created for **datavg** during the import.

_____

_____

_____

__ 19. Another problem that you should see at this stage, is that the /**home**/**jupiter** and /**home**/**mars** file systems already exist in **rootvg**.

To fix this problem, first unmount the /**home**/**jupiter** and /**home**/**mars** file systems from **rootvg**.

__ 20. Mount the file systems from **datavg** over the corresponding mount points. Use the new logical volume names that have been created. You have to specify the log device that is part of **datavg**.

Write down the commands you executed.

_____

_____

__ 21. Check the files you have created in /**home**/**jupiter** and /**home**/**mars**. They should exist in these directories.

__ 22. At the end of this exercise, all four file systems should be mounted at the same time.

Start with unmounting /**home**/**jupiter** and /**home**/**mars**.

__ 23. Create two new directories, /**datavg**/**jupiter** and /**datavg**/**mars**. These will be the new mount points for the file systems from **datavg**.

__ 24. Create two new stanzas in /**etc**/**filesystems** that describe the file systems from **datavg**. You must use the new logical volume names that have been created during the import of **datavg**.

__ 25. Mount the /**datavg**/**jupiter** and /**datavg**/**mars** file systems.

__ 26. Verify you can access all the files.

__ 27. Unmount the /**datavg**/**jupiter** and /**datavg**/**mars** file systems.

__ 28. Varyoff the **datavg** volume group.

__ 29. Export the **datavg** volume group.

__ 30. Remove the /**home**/**jupiter** and /**home**/**mars** file systems from the **rootvg** volume group.

End of exercise

Exercise 8. Saving and Restoring a User Volume Group

What This Exercise Is About

This exercise provides an opportunity to back up a non-**rootvg** volume group and then restore the data to simulate a failure of a complete volume group.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Use the **savevg** command
- Change volume group characteristics
- Use the **restvg** command

Introduction

All instructions in this exercise require **root** authority. There must be enough free space on the other disk to back up the user volume group. This disk will probably be the **rootvg** disk. If there are two students using the system, they must work on this exercise together.

NOTE: It is imperative that *Exercise 7: Exporting and Importing Volume Groups* was completed. This exercise assumes a user volume group (**datavg**) was created.

Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

__ 1. Check that your user volume group, **datavg,** is defined and varied on. Also, check that the /**home**/**jupiter** and /**home**/**mars** file systems are mounted. If not, import **datavg** and mount the file systems. (If you did the optional exercise in *Exercise 7: Exporting and Importing Volume Groups, Part 2 - Analyze import messages (Optional)* **datavg** should have been exported in the last exercise.)

__ 2. What is the physical partition size of **datavg**? _____

__ 3. How many partitions are allocated for /**home**/**jupiter**? _____

__ 4. Execute the **`mkvgdata`** command to create a control file for the **`savevg`** command. Write down the command you used.

__ 5. Before saving the volume group **datavg**, change the control file that is used during the restore process. Edit the file and change the number of logical partitions that are allocated for /**home**/**jupiter** to 4.

__ 6. Back up your user volume group, **datavg,** to a file image. Do not use SMIT to save the volume group. Write down the command you used.

__ 7. Unmount all file systems from **datavg**. Write down the commands you used.

__ 8. Varyoff the volume group **datavg**. Write down the command you used.

__ 9. Export the volume group from the system. Write down the command you used.

__ 10. Execute the **`restvg`** command and restore the volume group from your backup image. Write down the command you used.

__ 11. Write down the number of partitions that are allocated for /**home/jupiter**:

__ 12. Using SMIT, save the volume group **datavg** again. Specify the same backup file image as before. Write down the command that SMIT executes.

__ 13. Execute the same steps as before (**`umount`**, **`varyoffvg`**, **`exportvg`**) to remove the complete volume group **datavg** from the system.

__ 14. Using SMIT, restore the volume group, **datavg**, from the file image. In SMIT, specify a bigger physical partition size, for example 64 MB. Write down the command that SMIT executes.

__ 15. After restoring the volume group, check the physical partition size of **datavg**.

__ 16. How many partitions are allocated for /**home/jupiter**?

Do you still have four partitions for /**datavg/jupiter**?

End of exercise

Exercise 9. Error Log and `syslogd`

What This Exercise Is About

This exercise has two parts. In the first part, you will work with the AIX error logging facility. In the second part, you will work with the **`syslogd`** daemon and the ODM error notification class **errnotify**.

Two different lab sessions will be devoted to this exercise. So, you should stop after completing the first part of the exercise and not continue with the second part of the exercise.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Determine what errors are logged on your machine
- Generate different error reports
- Start concurrent error notification
- Identify errors and warnings sent by the **`syslogd`** daemon
- Create and maintain the /**etc**/**syslog.conf** file
- Automate error logging with **errnotify**
- Redirect **`syslogd`** messages to the error log

Introduction

In “Part 1” of this exercise, you will work with the AIX error logging facility. You should do this part of the exercise during the first lab session allotted to this exercise.

In “Part 2” of this exercise, you will work with the **`syslogd`** daemon and the ODM error notification class **errnotify**. You should do this part of the exercise during the second lab session allotted to this exercise.

You will need **root** authority to complete this exercise.

Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Part 1 - Working with the error log

__ 1. Generate a *summary report* of your system's error log. Write down the command that you (or SMIT) used:

_____

__ 2. Generate a *detailed report* of your system's error log. Write down the command that you (or SMIT) used:

_____

__ 3. Using SMIT, generate the following reports:

- A *summary report* of all errors that occurred during the past 24 hours. Write down the command that SMIT executes:

  _____

- A *detailed report* of all hardware errors. Write down the command that SMIT executes:

  _____

__ 4. If you are using local lab machines, this instruction requires that either: (a) a graphical desktop, such as CDE, is active or (b) that you have a windowing workstation where you can have a telnets from multiple windows. In this environment, start two windows.

If you are using a remote server, open an additional network connection (using a tool such as **`telnet`**) and log in as **root**, in order to have two windows to work with. In one window startup *concurrent* error logging, using the **`errpt`** command. Write down the command that you used:

_____

In the other window, execute the **errlogger** command to generate an error entry. Write down the command you used:

_____

Is the complete error text shown in the error report?

_____

Stop concurrent error logging.

__ 5. Write down the characteristics of your error log:

LOGFILE: _____

Maximum LOGSIZE: _____

Memory BUFFER SIZE: _____

What command have you used to show these characteristics?

_____

__ 6. List the entries that have an error class of operator.

__ 7. Clean up all error entries that have an error class of operator. Write down the command, you (or SMIT) used:

_____

__ 8. Verify that the operator entries are now gone.

End of Part 1

(If you are doing "Part 1" of this exercise, stop here. Do not go on to "Part 2.")

Part 2

If you are doing "Part 2" of this exercise, start here.

Section 1: Working with `syslogd`

__ 7. Edit the /**etc**/**syslog.conf** file and configure the **`syslogd`** daemon to log all daemon messages to a file with the name /**tmp**/**syslog.debug**.

Write down the line that you added to /**etc**/**syslog.conf**:

_____

__ 8. Execute the **`touch`** command and create the file /**tmp**/**syslog.debug**.

_____

__ 9. Refresh the **`syslogd`** daemon so it will pick up the changes. Write down the command that you used:

_____

__ 10. Stop the **`inetd`** daemon and restart it in debug mode. Use the appropriate *System Resource Controller* command to start the **`inetd`** daemon in debug mode (**`-d`** flag). Write down the commands that you used:

_____

_____

__ 11. Use the **`telnet`** command to **`telnet`** back to your own system, log in, and then log back out of the **`telnet`** session. This step is performed to log several debug messages. Use your login name when you **`telnet`** to your system.

_____

__ 12. Stop the **`inetd`** daemon and restart it without debug mode. Use the appropriate *System Resource Controller* command to start the **`inetd`** daemon. Write down the commands that you used:

_____

_____

__ 13. Analyze the content of the file /**tmp**/**syslog.debug**. Many debug messages from the **`inetd`** daemon processes are shown.

_____

__ 14. Change your /**etc**/**syslog.conf**. All messages should be directed to the AIX error log. Write down what you have changed:

_____

__ 15. Refresh the **`syslogd`** subsystem. Write down the command that you used:

_____

__ 16. Generate a **syslogd** message, for example, use an invalid password during a login. Check that the message is posted to the error log.

_____

_____

»

Section 2: Error Notification with errnotify

This part of the exercise demonstrates how to automate working with the error log.

__ 17. Create an **errnotify** object that mails a message to **root**, whenever an *operator message* is posted to the **errlog**. Write down the stanza that you added:

_____

_____

_____

_____

_____

_____

_____

__ 18. Execute the **errlogger** command and create an entry in the **errlog**. Write down the command that you used:

_____

__ 19. After a short time, check the mail for the **root** user. The mail processing is batched and it could take more than a minute before the mail is delivered; using **sendmail -q** may help to expedite this.

_____

End of Part 2

End of exercise

Exercise 10.Diagnostics

What This Exercise Is About

This exercise describes how to use diagnostic routines in several different modes.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Execute hardware diagnostics in the following modes:
  - Concurrent
  - Maintenance
  - Service (standalone)

Introduction

Only one person per machine can execute these commands.

Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Specifically, it requires either a a local machine were access does not depend upon network access or a remote LPAR which is accessible using a virtual terminal (HMC) with a physical Ethernet adapter.

__ 1. Determine if your system has a physical Ethernet adapter and whether it is configured.

__ 2. If your physical Ethernet adapter is not configured, then configure it with an private address which will not conflict with any existing lab subnets. For example, you might assign it 192.168.252.<your team number>. Check with the instructor if you are unsure.

__ 3. Start up diagnostic routines in concurrent mode and test a communication adapter that is in use on your system. What happens?

__ 4. Return to the **FUNCTION SELECTION** menu. Then, select **Diagnostic Routines**. What is the difference between **System Verification** and **Problem Determination**?

__ 5. Return to the **FUNCTION SELECTION** screen.
Using **Task Selection**, query the vital product data of one of your physical Ethernet adapters.

__ 6. Return to the **TASKS SELECTION LIST** screen.

»

f. Who will be notified when a hardware error is posted to the error log?

g. If **root** was not in the notification list, return to the **AUTOMATIC ERROR LOG ANALYSIS AND NOTIFICATION SERVICE AID** screen and add **root** to the notification list.

__ 7. Start up diagnostic routines in single user mode using the following steps:

a. You will need to be at the system console to do this. If you are using a remote LPAR, then first open a virtual terminal to your system from the HMC.

b. Shutdown your system to single user mode. (Note: In this particular case, it would have been sufficient to detach the interface related to the network adapter, rather than having to shut down to single user mode. But, there will be other situations where one may need to run diagnostics from single user mode, maintenance mode, or even booting with a diagnostic routine provided on CD or over the network.)

c. At your system console, login to single user mode using **root's** password.

d. Start the diagnostics facility.

__ 8. Test the communication adapter again in maintenance mode. What happens now?

__ 9. Exit the diagnostic utility.

__ 10. Start up the diagnostic utility in service mode from the hard drive using the following steps:

a. Shut down AIX and power off your machine or logical partition:

b. Boot your system to diagnostics using service mode off the hard drive.

__ 11. Test the communication adapter again in maintenance mode. What happens now?

__ 12. Exit the diagnostic utility.

__ 13. Boot your system in normal (multi-user) mode.

__ 14. When AIX finishes booting, log in as **root**.
View the contents of the diagnostics log using both the summary format and the detailed format. Did you find any errors?

End of exercise

Exercise 11.System Dump

What This Exercise Is About

This exercise allows you to become familiar with the AIX dump facility. In addition, you'll use the **snap** command to collect system data that is needed to analyze the system dump. During this exercise, you will also use the **kdb** command, but only at a very introductory level.

What You Should Be Able to Do

After completing this exercise, you should be able to:

- Initiate a dump
- Use the **snap** command

Introduction

In this exercise you will create a dump and use the **kdb** command to look at that dump.

You will need **root** authority to complete this exercise.

Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Note: All users must perform this exercise together if there is more than one user on your system.

Working with the AIX Dump Facility

__ 1. Record the following dump-related settings for your system:

primary dump device _____

secondary dump device _____

copy directory _____

dump compression (`ON` or `OFF`) _____

__ 2. Execute the command to display the *estimated size of a dump* and record the estimate you obtain:

_____

__ 3. Verify that the dump `copy directory` is large enough to hold the dump size reported on the previous command.

If there is not enough space, you must increase the size of the corresponding file system. (If necessary, use the **`chfs`** command to increase the size of the appropriate file system, typically /**var**.) After increasing the size, reverify that the filesystem is large enough.

__ 4. Set the value of the `autorestart` attribute for `sys0` to `true`. (If `autorestart` is set to `true`, the system will reboot after a crash.)

__ 5. Use the command **`sysdumpstart -p`** to start a dump to the primary dump device.

What LED code appears for several minutes after this command is entered? This is referred to as an `Operator Panel Value` (pre-HMCv7) or as the `Reference Code` (HMCv7) in the HMC display across from your LPAR name.

_____

__ 6. After the system reboots, determine and write down the size, uncompressed size, and filename for your system dump:

__ 7. Uncompress the dump file (for example, /**var**/**adm**/**ras**/**vmcore.0.BZ**). When doing the dump-uncompress, keep the original compressed file. Note, based on the

reported *Uncompressed Size* just reported, that you may need to further increase the size of /**var** to accommodate the size of the uncompressed dump (in addition to the already created compressed dump).

Then, execute the **kdb** command on the uncompressed dump that was created. Write down the commands you used:

_____

_____

__ 8. Use the **kdb** **stat** and **status** subcommands to show the system name and time of the dump, and the processes/threads running when the dump occurred. Leave the **kdb** command afterwards.

__ 9. Remove the uncompressed dump, but keep the original compressed dump. (This will ensure proper processing of the system dump by the **snap** command, which you will use in a subsequent lab step.)

__ 10. Check to see how much free space is currently available in /**tmp**.

If necessary, increase your /**tmp** file system so that there is at least *32 MB* of free space. We need this space in the next lab step.

Write down the commands you used:

_____

_____

__ 11. Run the command **snap -a**. (Note that this command will take approximately 10 minutes to run.)

Review the output of this command. This output will include a list of various directories (in /**tmp**/**ibmsupt**) to which the **snap** command writes its output.

In these directories, you will find files with names that end in **.snap**, which are ASCII files. Review the content of a few of these files.

End of exercise

Exercise 12.Basic Performance Commands

What This Exercise Is About

The purpose of this exercise is to provide basic performance commands.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Use **ps** to identify CPU and memory-intensive programs
- Execute a basic performance analysis
- Implement a Korn shell job queue
- Work with **nice** and **renice** to change the priorities of processes

Introduction

All instructions in this exercise should be executed with **root** authority.

Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Part 1 - Working with `ps`, `nice`, and `renice`

__ 1. Implement an alias **`top`** that shows a sorted output from **`ps aux`** according to the CPU usage. Write down the alias definition.

__ 2. Execute **`top`** and identify the process that consumes the most CPU.

__ 3. Start the program, **`/home/workshop/ex12_prog1`**, in background. Use the **`ps`** command to identify the assigned priority and nice value.

Priority: _____

Nice value: _____

__ 4. Stop **`ex12_prog1`** and restart it in background with a very low priority. Write down the command that you used.

Priority: _____

Nice Value: _____

__ 5. Without restarting **`ex12_prog1`**, increase the priority of the process. Write down the command you used.

Check that the priority has been increased.

Priority: _____

Nice Value: _____

__ 6. Stop the program **`ex12_prog1`**.

Part 2 - Basic Performance Analysis

__ 7. Start the program **/home/workshop/ex12_cpu** in background. Execute the **sar** command to analyze CPU usage on your system. Set it up to collect the data at two second intervals for five times. Write down the command that you used to monitor CPU usage.

From the output, what can you conclude?

__ 8. Use the **ps** command to check that the priority that has been assigned to the process. Is the priority high or low?

__ 9. Stop the program **ex12_cpu**.

__ 10. Create two JFS file systems on one of your hard disks then mount them. Use the following commands (these commands assume **rootvg** is on **hdisk0**, your disk name may be different):

```
# mklv -y lv1 -t jfs -u 1 rootvg 1 hdisk0
# crfs -v jfs -d lv1 -m /fs1
# mklv -y lv2 -t jfs -u 1 rootvg 1 hdisk0
# crfs -v jfs -d lv2 -m /fs2
# mount /fs1; mount /fs2
```

__ 11. Start the program **/home/workshop/ex12_io -c -w -t 120** in the background. (The value used for the **-t** option specifies how long to run this program in seconds.) Execute the **iostat** command to analyze your disk I/O while **ex12_io** is running. Look at **iostat** disk information for two second intervals five times. Write down the command that you used to monitor disk I/O.

From the output, what can you conclude?

__ 12. The **ex12_io** program should stop after 2 minutes (120 seconds). If it has not ended, stop the program.

__ 13. Start the memory intensive process **/home/workshop/ex12_memory** in the background. Execute the **vmstat** command to analyze your memory utilization. Run **vmstat** at five second intervals. Write down the command that you used to measure memory.

From the output, what can you conclude?

Part 3 - Working with a Korn Shell Job Queue

__ 14. Create a Korn shell job queue as shown in your student notes. Write down the definitions for the queue and the queue device:

__ 15. Bring down the queue. Write the command you used.

__ 16. Put the job **`/home/workshop/ex12_job`** into the **ksh** queue. Write down the command you used.

__ 17. Verify that the job is queued. Write down the command you used.

__ 18. Bring up the queue. Write down the command you used.

What happens?

End of exercise

Exercise 13.Performance Diagnostic Tool

What This Exercise Is About

The purpose of this exercise is to give students an opportunity to use the Performance Diagnostic Tool.

What You Should Be Able to Do

After completing this exercise, students should be able to use the Performance Diagnostic Tool (PDT) for ongoing data capture and analysis of critical system resources.

Introduction

This exercise deals with the Performance Diagnostic Tool (PDT) for on-going data capture and analysis of system resources.

Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

__ 1. Verify that PDT is loaded on your exercise system. Start PDT to enable default data collection and reporting.

__ 2. The **adm** user is needed to run this procedure. **`su`** to the **adm** user and change the **crontab** entry so PDT will collect data within 10 minutes from now and run the reports five minutes later. Make sure you check the system date first so you will know what hour and minutes to put in the **crontab** entries.

After you modify the **crontab** entry, continue onto the next step. The next step needs to run while PDT is collecting information.

__ 3. Run the script **`ex13_perf`** located in /**home**/**workshop**. You need to run this as **root**. This script will create some items that should be reported when PDT runs in 10 minutes.

__ 4. After the time frame is over in which the report should have been created (based on your entries in the **crontab** file), view the report.

__ 5. To change the severity level to severity level 2 and the user to whom the report is mailed, execute the **`pdt_config`** program. Once you are finished making the changes, exit the program.

__ 6. Run PDT again, this time from the command line. Do it twice. Once to see a severity level 2 report. The other time to do a severity level 3 report.

End of exercise

Exercise 14.Authentication and ACLs

What This Exercise Is About

This exercise will familiarize you with three security features: the **login.cfg** file, authentication, and access control lists (ACLs).

What You Should Be Able to Do

After completing this exercise, students should be able to:

- Customize the **login.cfg** file
- Add an additional primary authentication method for a user
- Implement access control lists (ACLs)

Introduction

This exercise consists of three parts:

1. Customizing the **login.cfg** file
2. Adding a primary authentication method
3. Access control lists (ACLs)

Requirements

- Program **`/home/workshop/ex14_login`**

Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Setting a new login herald

__ 1. Log in as **root** and edit /**etc**/**security**/**login.cfg.** Change the herald message to read:

```
*Restricted Access*
Authorized Users Only
Login:
```

__ 2. Use the **login** command to log in over yourself. You must be at the command line login to see your changes. If you are using the CDE graphical login, click the **options** button and select **Command line login**.

Does it look correct? If not, try step one again.

__ 3. Log in as **root**.

__ 4. Review the failed login attempts made on your machine.

__ 5. Review the **su** activity on your machine.

__ 6. Review all the logins on your system.

__ 7. Review all root logins on your system.

Adding a primary authentication method

In /**home**/**workshop** you find a procedure with the name **ex14_login**, which implements an additional primary authentication method. This method restricts a user to *one login session* on a system.

__ 8. With **root** authority, change to /**home**/**workshop** and analyze the procedure **ex14_login**. Which statement indicates a valid or invalid login?

_____

_____

Check that **ex14_login** is executable.

__ 9. Install the procedure **ex14_login** as an additional authentication method on your system. You will need to add a stanza in /**etc**/**security**/**login.cfg.** Write down the stanza you add and the name of the file in which you place the stanza:

_____

_____

_____

__ 10. Check to be sure that team01 is a defined user, with a password set and no ADMCHG flag to force password reset. The password should be the same as the user name.

__ 11. Install the additional authentication method for user **team01** in **/etc/security/user**. Write down the stanza definition for **team01**:

_____

_____

__ 12. Working in a graphical environment, open two windows and execute the **login** command in both of them. Login as **team01**. The second login should fail.

__ 13. Remove the additional authentication method from **team01**.

_____

Access control lists

__ 14. Use the **login** command to log in over yourself as **team01** and switch user to **root**. Create two new users named **michael** and **sarah**. Assign each new account a password that is the same as the login name.

_____

_____

_____

_____

__ 15. Return to your **team01** user ID.

__ 16. Create a shell script named **sample** in your home directory with the following content:

```
tput clear
banner We love AIX
print End of Program
```

Set the base permissions for **sample** to 700 and verify that the script works.

_____

_____

__ 17. Use the **login** command to log-in over yourself as **michael**. Change directory to **/home/team01**.

__ 18. Try to display the **sample** script, using the **cat** command. Try to execute **sample**. You should not be able to do either.

_____

_____

__ 19. Use the **login** command to log in over yourself **team01**. Set and export the `EDITOR` variable to `/usr/bin/vi` in your **.profile**.

Use the **login** command to log in over yourself as **team01**. The alternative is to execute the .profile you just created (without launching a subshell).

_____

_____

__ 20. Use the **acledit** command to change the extended permissions of the **sample** script so that **michael** has `rwx` access to the script, and **sarah** has only `r-x` access to the script. (Use the AIXC ACL type for this exercise.) Apply the modified ACL.

_____

_____

_____

_____

__ 21. Execute **ls -e** and check that **extended permissions** are set for **sample**.

_____

_____

__ 22. Use the **login** command to log in over yourself as **michael**. Change to the /**home**/**team01** directory and test the extended permissions by trying to add the **date** command to the end of the script. Execute the **sample** script afterwards.

_____

_____

__ 23. Use the **login** command to log in over yourself as **sarah**. Change to /**home**/**team01** and try to change the **sample** script by removing the **date** command. Does it work?

_____

__ 24. Use the **login** command to log in over yourself as **team01**. Change the *base* and *extended* permissions to the **sample** script so that members of the **staff** group can *read* and *execute* the script, *except* for **michael**.

_____

_____

_____

_____

__ 25. Use the **login** command to log in over yourself as **michael**. Issue the **groups** command to ensure you are part of the **staff** group. Change directory to /**home/team01**. Can you execute the **sample** script?

_____

_____

__ 26. Use the **login** command to log in over yourself as **team01**. Create a new file named **sample2**. Type a couple of lines in the file. Use **aclget** to see that no extended permissions are set on **sample2**.

_____

_____

__ 27. Using **aclget** and **aclput**, copy the access control information of the file **sample** to the new file **sample2**. Verify that the ACLs were copied over to **sample2**.

_____

_____

__ 28. Execute a **chmod 700** on **sample**. Execute **acledit** on **sample**. What is different?

_____

_____

End of exercise

Appendix A. Auditing

What This Exercise Is About

This exercise is an introduction to the use of the AIX auditing subsystem to trace and record security-relevant information.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Audit objects and application events
- Create audit classes
- Audit users
- Set up auditing in bin and stream mode

Introduction

Completion of this exercise requires **root** authority.

Requirements

- **/home/workshop/exappa_job**

Exercise Instructions

Bin mode auditing

__ 1. Answer the following question first: Where do you specify file system objects that should be audited?

_____

__ 2. Set up auditing of the program **/usr/bin/passwd**. When you are finished, whenever a user calls **passwd**, you should get an audit record. Add this object to the corresponding audit configuration file.

_____

_____

_____

Write down the event name you have created:

_____

__ 3. In which file do you have to specify the format definitions for your new event?

_____

__ 4. Add the format definition for your new event to the corresponding audit configuration file.

_____

_____

__ 5. In which file do you specify the *start mode* for the auditing subsystem?

_____

__ 6. Create a directory /**var**/**myaudit**. We want to use this directory to collect all audit-related files.

__ 7. Change the corresponding configuration file to start up the auditing subsystem in *bin mode*. Specify the following bin files:

```
bin1 = /var/myaudit/bin1
bin2 = /var/myaudit/bin2
trail = /var/myaudit/trail
```

_____

_____

__ 8. Start the auditing subsystem. Write down the command you used.

_____

__ 9. Log in as **team01**. Use the password **team01.** When prompted to change password, set it back to **team01**. If you use a graphical environment, execute the **login** command in a separate window.

__ 10. Execute the **passwd** command and change the password for **team01**.

__ 11. With **root** authority, stop the auditing subsystem. Write down the command you used.

_____

__ 12. Change to /**var**/**myaudit** and display the audit records that have been recorded. Write down the command you used.

_____

_____

Stream mode auditing

__ 13. In which file do you configure *audit classes* and *audit users*?

_____

__ 14. Change this configuration file in the following way:

- The auditing subsystem starts up in *stream mode*.
- Create an audit class `kill`, that contains an audit event whenever a process gets killed.
- Remove the **root** user in the `users` stanza.
- The user **team01** should be audited for the audit classes `kill` and `tcpip`.

_____

_____

_____

_____

__ 15. In which file do you configure the **auditstream** daemon?

_____

__ 16. Before starting the auditing subsystem in stream mode, change the configuration file for the **auditstream** daemon. All audit records shall be written to file /**var**/**myaudit**/**stream.out**. Be sure to terminate the command with a & sign.

_____

_____

__ 17. Start your auditing system.

_____

__ 18. Use the **touch** command and create an empty file /**var**/**myaudit**/**stream.out**. Use the **tail** command in a separate window to display the audit records real-time.

_____

_____

__ 19. Log in as **team01** and trigger the events that you are auditing for this user:

- Execute the **ftp** command. Use your local host as destination host. You should see the corresponding audit records in file /**var**/**myaudit**/**stream.out**.
- Start the program **/home/workshop/exappa_job** in *background*. Kill the started program afterwards. You should see audit records for the `kill` audit class you have created.

_____

_____

__ 20. Stop the auditing subsystem.

_____

End of exercise

Exercise Instructions With Hints

Bin mode auditing

__ 1. Answer the following question first: Where do you specify file system objects that should be audited?

_____

Hint: /etc/security/audit/o...

__ 2. Set up auditing of the program **/usr/bin/passwd**. When you are finished, whenever a user calls **passwd**, you should get an audit record. Add this object to the corresponding audit configuration file.

_____

_____

_____

Hint: Add an `x-event` for the program **/usr/bin/passwd**.

Write down the event name you have created:

_____

__ 3. In which file do you have to specify the format definitions for your new event?

_____

Hint: /etc/security/audit/e...

__ 4. Add the format definition for your new event to the corresponding audit configuration file.

_____

_____

Hint: Specify the event name and add a **printf** definition.

__ 5. In which file do you specify the *start mode* for the auditing subsystem?

_____

Hint: /etc/security/audit/c...

__ 6. Create a directory /**var**/**myaudit**. We want to use this directory to collect all audit-related files.

__ 7. Change the corresponding configuration file to startup the auditing subsystem in *bin mode*. Specify the following bin files:

```
bin1 = /var/myaudit/bin1
bin2 = /var/myaudit/bin2
trail = /var/myaudit/trail
```

_____

_____

Hint: You must change the `start` and `bin` stanzas.

__ 8. Start the auditing subsystem. Write down the command you used.

_____

Hint: Execute **`audit...`**

__ 9. Log in as **team01**. Use password **team01**. When prompted to change password, set it back to **team01**. If you use a graphical environment, execute the **`login`** command in a separate window.

__ 10. Execute the **`passwd`** command and change the password for **team01**.

__ 11. With **root** authority, stop the auditing subsystem. Write down the command you used.

_____

Hint: Execute **`audit...`**

__ 12. Change to /**var**/**myaudit** and display the audit records that have been recorded. Write down the command you used.

_____

_____

Use the **`auditpr`** command and query the *trail*.

Stream mode auditing

__ 13. In which file do you configure *audit classes* and *audit users*?

_____

Hint: /etc/security/audit/c...

__ 14. Change this configuration file in the following way:

- The auditing subsystem starts up in *stream mode*.
- Create an audit class `kill`, that contains an audit event whenever a process gets killed.
- Remove the **root** user in the `users` stanza.
- The user **team01** should be audited for the audit classes `kill` and `tcpip`.

Hint: You must change the `start`, `classes` and `users` stanzas.

_____

_____

_____

_____

__ 15. In which file do you configure the **auditstream** daemon?

_____

Hint: /etc/security/audit/str...

__ 16. Before starting the auditing subsystem in stream mode, change the configuration file for the **auditstream** daemon. All audit records shall be written to file **/var/myaudit/stream.out**. Be sure to terminate the command with a & sign.

_____

_____

Hint: Change the **auditpr** command.

__ 17. Start your auditing system.

_____

__ 18. Use the **touch** command and create an empty file **/var/myaudit/stream.out**. Use the **tail** command in a separate window to display the audit records real-time.

Hint: # **tail -f /var/myaudit/stream.out**

_____

_____

__ 19. Log in as **team01** and trigger the events that you are auditing for this user:

- Execute the **ftp** command. Use your local host as destination host. You should see the corresponding audit records in file **/var/myaudit/stream.out**.
- Start the program **/home/workshop/exappa_job** in *background*. Kill the started program afterwards. You should see audit records for the `kill` audit class you have created.

_____

_____

Hint: Execute **ftp localhost**. Use the **kill** command to kill the command that runs in background.

__ 20. Stop the auditing subsystem.

End of exercise

Exercise Instructions With Solutions

Bin mode auditing

__ 1. Answer the following question first: Where do you specify file system objects that should be audited?

/etc/security/audit/objects

__ 2. Set up auditing of the program **/usr/bin/passwd**. When you are finished, whenever a user calls **passwd** you should get an audit record. Add this object to the corresponding audit configuration file.

```
# vi /etc/security/audit/objects

/usr/bin/passwd:
x = "X_EVENT"
```

Write down the event name you have created:

X_EVENT

__ 3. In which file do you have to specify the format definitions for your new event?

/etc/security/audit/events

__ 4. Add the format definition for your new event to the corresponding audit configuration file.

```
# vi /etc/security/audit/events

X_EVENT = printf "%s"
```

__ 5. In which file do you specify the *start mode* for the auditing subsystem?

/etc/security/audit/config

__ 6. Create a directory /**var**/**myaudit**. We want to use this directory to collect all audit-related files.

```
# mkdir /var/myaudit
```

__ 7. Change the corresponding configuration file to start up the auditing subsystem in *bin mode*. Specify the following bin files:

```
bin1 = /var/myaudit/bin1
bin2 = /var/myaudit/bin2
trail = /var/myaudit/trail
```

```
# vi /etc/security/audit/config

bin:
trail = /var/myaudit/trail
bin1 = /var/myaudit/bin1
```

```
bin2 = /var/myaudit/bin2
binsize = 10240
cmds = /etc/security/audit/bincmds
```

__ 8. Start the auditing subsystem. Write down the command you used.

```
# audit start
```

__ 9. Log in as **team01**. Use password **team01**. When prompted to change password, set it back to **team01**. If you use a graphical environment, execute the **login** command in a separate window.

```
# login
```

__ 10. Execute the **passwd** command and change the password for **team01**.

```
$ passwd
```

__ 11. With **root** authority, stop the auditing subsystem. Write down the command you used.

```
# su
# audit shutdown
```

__ 12. Change to /**var**/**myaudit** and display the audit records that have been recorded. Write down the command you used.

```
# cd /var/myaudit
# auditpr -v < trail
```

You should see the audit event `X_EVENT` that has been created, triggered by the execution of **passwd**.

Stream mode auditing

__ 13. In which file do you configure *audit classes* and *audit users*?

/etc/security/audit/config

__ 14. Change this configuration file in the following way:

- The auditing subsystem starts up in *stream mode*.
- Create an audit class `kill`, that contains an audit event whenever a process gets killed.
- Remove the **root** user in the `users` stanza.
- The user **team01** should be audited for the audit classes `kill` and `tcpip`.

```
# vi /etc/security/audit/config

start:
binmode = off
streammode = on

classes:
```

```
kill = PROC_Kill

users:
team01 = kill,tcpip
```

__ 15. In which file do you configure the **auditstream** daemon?

/etc/security/audit/streamcmds

__ 16. Before starting the auditing subsystem in stream mode, change the configuration file for the **auditstream** daemon. All audit records shall be written to file **/var/myaudit/stream.out**. Be sure to terminate the command with a & sign.

```
# vi /etc/security/audit/streamcmds

/usr/sbin/auditstream | auditpr -v > /var/myaudit/stream.out &
```

__ 17. Start your auditing system.

```
# audit start
```

__ 18. Use the **touch** command and create an empty file **/var/myaudit/stream.out**. Use the **tail** command in a separate window to display the audit records real-time. (Depending on your environment, you may want to start AIXwindows for the next step (# **xinit**).)

```
# touch /var/myaudit/stream.out

# tail -f /var/myaudit/stream.out
```

__ 19. In a separate window log in as **team01** and trigger the events that you are auditing for this user:

```
# login
```

- Enter login name as **team01**. Provide **team01**'s password.
- Execute the **ftp** command. Use your local host as destination host. You should see the corresponding audit records in file **/var/myaudit/stream.out**.

```
$ ftp localhost
Name: team01
Password: team01

ftp>bye
```

- Start the program **/home/workshop/exappa_job** in *background*. Kill the started program afterwards. You should see audit records for the `kill` audit class you have created.

```
$ /home/workshop/exappa_job &
$ kill %1
```

__ 20. Stop the auditing subsystem.

```
# audit shutdown
```

End of exercise