



Linux

LPIC Level 2 Certification

LPIC-2 Courseware

Version 3.0

www.firebrandtraining.com



LPIC-2 200 Capacity Planning

Copyright © Property of Firebrand Training Ltd

200.1 Measure & Troubleshoot Resource Usage

Weight 6

Description:

Candidates should be able to measure hardware resource and network bandwidth, identify and troubleshoot resource problems.

Key Knowledge Areas:

- › Measure CPU usage.
- › Measure memory usage.
- › Measure disk I/O.
- › Measure network I/O.
- › Measure firewalling and routing throughput.
- › Map client bandwidth usage.
- › Match / correlate system symptoms with likely problems.
- › Estimate throughput and identify bottlenecks in a system including networking.



Measuring and Troubleshooting Resource Usage

GNU/Linux makes available a great amount of information about its own inner workings. Detailed information is available for every level of the system, and administrators control a variety of tunable parameters. Performance tuning is hard work and requires patience, a methodical approach, and careful analysis:

- Establish a baseline performance
- Collect and review historical information about your system
- Tune your system in a way that allows comparison with baseline
- Have a rollback plan

To a first approximation, only the following have much effect on performance:

- CPU time
- Memory
- Hard disk I/O
- Network I/O

Remember though that most serious performance issues often lie within applications and have little to do with the underlying operating system.

06/23/14

3



Measuring CPU usage

You will probably want to gather the following CPU data:

- Load average:
 - **#top** and **#uptime** commands show load averages over 1,10,15minutes intervals. This is an indication of the aging and/or duration of a system state. This is an indication of all demand for CPU time
- Overall Utilization
 - **#vmstat** shows averages within the previous sample period (5sec default). Takes two arguments, no. of seconds and no. of reports. First line of output reports averages since system boot. Under cpu heading:
 - us(user)** :higher numbers here generally indicate computation
 - sys(system,kernel)**:higher numbers processes making a lot of system calls
 - id**:idle time
 - wa**:time waiting for I/O
 - Under system heading:
 - cs**:context switches per interval
 - in** : interrupts per interval.
 - Extremely high **cs** or **in** indicate a misbehaving or mis-configured hardware

device

06/23/14

4

.....cont



🌀 Per process CPU consumption

- A CPU is a discrete state machine. Therefore percentage values given by utilities indicate a time interval that the system's processes were found to be active on the CPU. e.g. 45% of the samples taken by the utility found your process active on the CPU.
- The following commands display per process CPU utilization

```
#top & ps aux
```

 - Unlike the `ps` command, `top` presents “live” view of a running system. Option `-d` to `top` specifies the delay between screen updates in ss.tt(e.g. `top -d 1.2`)
 - Deferring the execution of CPU hogs or reducing their priority makes the CPU available to other processes
 - The `w` command displays information about current system users including command line of their current process, idle time, time used by current user processes
 - `sar -q` will report CPU run queue, task list, load averages for the last 1,5, & 15min as well as blocked tasks waiting for I/O to complete

06/23/14



Memory Usage

- 🌀 Linux manages memory in units called pages currently 4KiB. The kernel allocates virtual memory to processes as they request memory.
- 🌀 Most of these units can be swapped(or “paged”) in or out of RAM(some pages are locked and can't be swapped)
- 🌀 When a running process requires more RAM one or more pages of RAM are “swapped out” to make RAM available.
- 🌀 If “swapped out” pages are required by a process they can be “swapped back”
- 🌀 `vmstat` command reports virtual memory statistics

```
#vmstat 5 10
```

(will run ten updates, five seconds apart.)
- 🌀 Columns `free`, `si` and `so` are for free memory, page-ins, and page-outs respectively. High values of `so` can be indicative of not enough RAM

06/23/14



Disk I/O

✿ A common performance bottleneck on Linux systems is disk bandwidth because hard disks are mechanical systems. It takes many milliseconds to locate a disk block and fetch its contents.

✿ Swap devices are normally disk drives so this also affects performance

✿ **iotstat** command can be used to monitor disk performance. It accepts optional arguments to specify an interval in seconds and a repetition count.

- Its first line of output is a summary since boot
- It gathers its information from `/proc`
- Columns **tps**, **Blk_read/s**, **Blk_wrtn/s**, **Blk_read** and **Blk_wrtn** are for transfers/sec, blocks read/sec, blocks written/sec, total blocks read and written respectively
- Cost of seeking is the most important factor affecting disk drive performance. It's best to split for example swap area among several disks.

06/23/14

7



Network I/O

✿ A good understanding of networking protocols e.g. TCP/IP is helpful while analyzing network issues

✿ For network interfaces, you should monitor total number of packets received, sent, and dropped as well as rate (i.e. bits/second)

✿ Linux provides quite a few tools to aid one in analyzing network performance

- **iptraf -l eth0** displays interface eth0 statistics (bytes in/out etc)
- **lsof -i** will list all open Internet network files
- **netstat --statistics, -s** displays statistics for each protocol
- **ethtool -S eth0** dumps NIC & driver specific statistics
- **sar -n DEV 1** will display NIC statistics updating every second
- **ss** will dump socket statistics. e.g. **-m** option displays socket memory usage

06/23/14

8



Map client bandwidth usage.

Linux provides a number of utilities to help one determine how processes are using available network bandwidth

- **nethogs** command groups bandwidth utilization by process. The output has a column for user running the program

06/23/14

9



200.2 Predict Future Resource Needs

Weight: 2

Description: Candidates should be able to monitor resource usage to predict future resource needs.

Key Knowledge Areas:

- Use **collectd** to monitor IT infrastructure usage.
- Predict capacity break point of a configuration.
- Observe growth rate of capacity usage.
- Graph the trend of capacity usage.
- Awareness of monitoring solutions such as Nagios, MRTG and Cacti

06/23/14

10



Collectd - System Statistics Collection daemon

- ✿ **collectd** receives system information from plugins that it then makes available in a number of ways
- ✿ Its behavior is controlled by a configuration file (collectd.conf by default) but an alternate one can be used with option -C
- ✿ Check /usr/share/doc/collectd for a README file that contains a list of plugins including a short description of what the plugins do. Broadly speaking there are input and output plugins.
- ✿ Example input plugin:cpu plugin;output plugin:for getting dispatched values from daemon e.g. and then writing to RRD-files for use with RRDtool(a data logging and graphing tool)

....cont

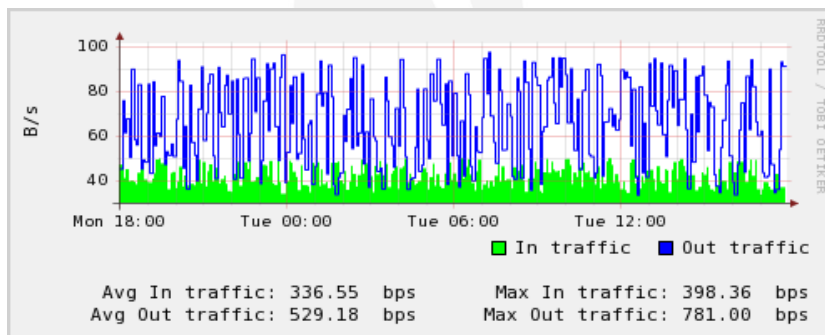
06/23/14

11



..cont collectd

- ✿ Example network traffic graph generated by RRDtool



06/23/14

12



Monitoring System Behavior Using collectd

- ✿ **collectd** supports monitoring(>version 4.3.0). So user can define thresholds which if breached collectd should notify the user.
- ✿ Data collected can be used with tools like RRD-tool to generate graphs over a period of time which can easily show usage trend
- ✿ Can help in predicting capacity break point for a given system configuration.
- ✿ Some interesting data that can be collected by **collectd** daemon
 - Hard disk temperature, NIC traffic-including packets and errors, motherboard sensors:temp, fanspeed etc, MySQL statistics, Apache web server and Nginx statistics, network latency etc(see the collectd README for more information)

06/23/14

13



Other Monitoring Utilities

- ✿ There is quite a few available monitoring utilities for Linux including
 - **Nagios**: a very popular industry standard monitoring tool which also supports SNMP. Can also monitor network switches, applications etc. Its capabilities are extensible via plugins.
 - Alerts can be delivered via email, SMS etc. It provides a web based dashboard. Provides performance graphs too.
 - **Cacti**: A frontend to RRDTool. Uses MySQL database for storage of system performance data that it has gathered. Has SNMP support too. Once one or more data sources are defined, an RRDTool graph can be created using the data.
 - **MRTG** (Multi Router Traffic Grapher) was originally meant to allow user to see traffic load on a network in graphical form. However now it can create graphs and statistics for almost anything. It is written in Perl so it is cross-platform

06/23/14

14



201.1 Kernel Components

Weight 2

Description

Candidates should be able to utilize kernel components that are necessary for specific hardware, hardware drivers, system resources and requirements. This objective includes implementing different types of kernel images, identifying stable and development kernels and patches, as well as using kernel modules.

Key Knowledge Areas

- ✿ Kernel 2.6.x documentation
- ✿ Kernel 3.x documentation
- ✿ The following is a partial list of the used files, terms and utilities:
- ✿ /usr/src/linux
- ✿ /usr/src/linux/Documentation
- ✿ zImage
- ✿ bzImage

06/24/14

1



Microkernel and Monolithic Kernels

- ✿ The Linux kernel is the core component of the system. It is essentially monolithic.
 - ✿ A monolithic kernel - all components are built into the kernel. This means any changes will need a recompile of the kernel
 - ✿ A microkernel kernel - minimum functionality in kernel mode and implement the rest in user space.
 - ✿ However to reduce the size of the Linux kernel, some functionality can be provided via loadable kernel modules e.g. file system support, NIC driver.
 - ✿ The version of the kernel can be shown with the uname command
- #uname -r**
- ✿ The kernel numbering system is as follows. If your kernel is 2.6.25
 - 2. is the kernel number and only changes when there is a main concept change in the kernel
 - 6. is the major number. Odd numbers are development kernels
 - 25 is the minor or patch level changes when security patches etc are applied

06/24/14



Version 3 kernel

- ✿ Linus Torvalds announced the version 3.0 kernel in April 2011
- ✿ Several reasons behind this - one of them being Linux's 20th birthday and 15 years after the launch of version 2.0
- ✿ Additional graphics driver support
- ✿ Support for newer chipsets such as Fusion and Sandy Bridge
- ✿ Support for Microsoft Kinect
- ✿ Latest version at time of printing - 3.15.1, note use of odd numbers

06/24/14

3



/boot directory

- ✿ The compiled kernels are stored in the /boot directory and are generally gzipped or bziped
 - zImage or bzImage can be used for this
- ✿ These files could be linked or may be replaced by the real kernel name
 - /boot/vmlinuz -----linked--> /boot/vmlinuz-2.6.XXXX
- ✿ Other files that may exist in the /boot directory are
 - config-2.6.XXXX (This is the kernel configuration file used to compile this version of kernel)
 - initrd-2.6.XXXX (The Initialisation Ram Disk containing modules for boot devices)
 - System.map-2.6.XXXX (System.map is a "phone directory" list of function in a particular build of a kernel)
- ✿ The choice of kernel that loads is defined in the LILO boot loader or the GRUB boot loader configuration files.
 - /etc/lilo.conf
 - /boot/grub/grub.conf

06/24/14

4



The kernel source code

- On earlier kernels (2.4 and prior), Kernel source code is stored in
 - /usr/src/linux/2.4.18-3
- On newer 2.6 versions of the kernel they can be found
 - /usr/src/kernels/2.6.23.1-42.fc8-i386
- Version 3 kernels also use /usr/src/kernels
- Under these directories you will find the kernel source code, the Makefile and documentation.
- Documentation can also be found in
 - /usr/share/doc/

06/24/14

5



Getting the kernel source

- The kernel source can be found in many locations. As a rule of thumb download the kernel source relative to your distribution, i.e. if you have Redhat, download a Redhat kernel source or patch.
- If you have a vanilla kernel then you can download from <http://www.kernel.org>
- You can also use the repository tools like yum and apt-get
 - `#yum install kernel-source-`uname -r``

06/24/14

6



201.2 Compiling a kernel

Weight 3

Description

Candidates should be able to properly configure a kernel to include or disable specific features of the Linux kernel as necessary. This objective includes compiling and recompiling the Linux kernel as needed, updating and noting changes in a new kernel, creating an initrd image and installing new kernels.

Key Knowledge Areas

- /usr/src/linux/
- GRUB configuration files
- Kernel 2.6.x make targets
- Kernel 3.x make targets

Terms and Utilities:

- mkinitrd
- mkinitramfs
- make
- make options (config, xconfig, menuconfig, oldconfig, cloneconfig, prepare-all, mrproper, zlmage, bzlmage, modules, modules_install)

06/24/14

7



The Kernel config file

- In the root of the source tree you will find all the files needed to compile a kernel. All work is done from this directory.
- In addition to this you may want/need the current kernel build configuration file which is located
/boot/config-2.6.XXXX
- Copy the above file into the root of the source tree. You can rename it **.config**
- If you take a .config file from an existing kernel version (from the /boot directory), and you have the source code for a newer kernel version, then you need to update the .config file to the newer version of source code. To do this type

#make oldconfig

06/24/14

8



Editing the config file

- ✿ To choose what is compiled into the kernel you need to edit the `.config` file. This is usually done with a tool
- ✿ There are various versions of the kernel editing tool, and are usually invoked using the `make` command
 - `#make config` (A text based configuration tool, 300 odd questions)
 - `#make menuconfig` (A text menu based system using ncurses)
 - `#make xconfig` (A GUI based tool using qt libraries, requires X)
 - `#make gconfig` (Another GUI based tool)
- ✿ All the above do the same, i.e. edit the `.config` file. This will allow you to choose things like what CPU, whether you are building a modular or monolithic kernel etc.
- ✿ When you exit the configuration tool, you will save the changes to the `.config` file.

06/24/14

9



The kernel Makefile

- ✿ The top level **Makefile** is also located in the source code tree root directory. The **Makefile** contains such important information as the kernel version number, the location of the source code, the location of compilers etc.
- ✿ It contains the configuration and cleaning targets for the `make` command, and these can be seen by typing
 - `#make help`

06/24/14

10



Cleaning targets

✿ Two cleaning targets are available, these commands will remove partial compiled kernels, and configuration files

✿ The **clean** target removes most generated files, but leaves behind the **.config** files

#make clean

✿ The **mrproper** target removes all generated files and **.config** files as well

#make mrproper

06/24/14

11



Compiling the kernel

✿ It is now time to compile the kernel.

✿ The following commands can be used to compile the kernel

#make bzImage (zImage can be used to make a kernel load into low mem)

or

#make

✿ After compilation the kernel image can be found in the **/usr/src/kernels/kernel-ver/arch/x86/boot** directory called **bzImage**.

06/24/14

12



Modular Kernel steps

🌀 If you have defined a modular kernel then you now need to compile the kernel modules. To do this

#make modules

🌀 The modules then need to be installed in the correct directory under **/lib/modules/kernel-ver**

#make modules_install

🌀 Note, the above steps can be ignored if you have a monolithic kernel



Install the kernel into the /boot

🌀 The next step is to install three files into **/boot** directory as well as modification to your kernel grub configuration file.

🌀 To do this use

#make install

🌀 The following are moved to boot from the compilation directory

- **System.map-2.6.XX**
- **config-2.6.XX**
- **vmlinuz-2.6.XX**



The init Ram Disk

✿ If you are using SCSI disks then you need to build an initrd file

✿ Change to the /boot directory and run the following command

```
#mkinitrd -o initrd.img-2.6.XX 2.6.XX
```

Or

```
#mkinitramfs
```

RedHat uses a program called dracut to manage the ram disk.



Modify your grub.conf or lilo.conf

✿ You must now edit your bootloader configuration file to point to the new kernel.

✿ For grub

```
/boot/grub/grub.conf
```

✿ Next generation of grub is grub2 slight change in the way it loads the first sectors

✿ Grub2 uses grub.cfg

✿ For lilo

```
/etc/lilo.conf
```

Note - Remember to run lilo after modifying the lilo.conf



Patching the Kernel

- ✿ Instead of downloading a full kernel source you can patch the current kernel source code with a patch file
- ✿ A patch file is a difference file between one version of a source code tree and a newer version of the source code tree.
- ✿ If you have to go from 2.6.10 to 2.6.20 then you would have to apply all patches from 11-20 which could be time consuming.
- ✿ Patches are generally used if you have made custom modifications to the original source code, which you don't want to lose.

06/24/14

17



The patch command

- ✿ When you download a patch you need to extract it into the current kernel source.
- ✿ It is recommended that you copy the original source code tree into a new directory with the name of the new kernel version as the directory name. Then use the following commands to patch the existing tree
 - #zcat 2.6.18-patch.gz | patch -p0**
 - This patches the kernel source with the files contents
 - #cat 2.6.18-patch | patch -p0**
 - For when the file is uncompressed.
- ✿ The **-p0** tells the patch command not to strip any filepath parts in the patch file.
- ✿ The **-R** option can be used for revoking a patch, if it was applied by mistake.

06/24/14

18



Finishing the patching

Once the directories have been patched, then follow the same process to build the kernel as in the previous slides.

- **make clean**
- **make oldconfig**
- **make bzImage**
- **make modules**
- **make module_install**
- **etc..**

06/24/14

19



201.3 Kernel Runtime Management and Troubleshooting

Weight 4

Description

Candidates should be able to manage and/or query a 2.6.x or 3.x kernel and its loadable modules.

Candidates should be able to identify and correct common boot and run time issues.

Candidates should understand device detection and management using udev. This objective includes troubleshooting udev

rules

Key Knowledge Areas

Use command-line utilities to get information about the currently running kernel and kernel modules.

Manually load and unload kernel modules.

Determine when modules can be unloaded.

Determine what parameters a module accepts.

Configure the system to load modules by names other than their file name.

/proc filesystem

Content of /, /boot, and /lib/modules

Tools and utilities to analyze information about the available hardware

.....cont.

06/24/14

20



201.3 Kernel runtime management and troubleshooting

Terms and Utilities

- ✿ /lib/modules/kernel-version/modules.dep
- ✿ module configuration files in /etc
- ✿ /proc/sys/kernel/
- ✿ /sbin/depmod
- ✿ /sbin/rmmod
- ✿ /sbin/modinfo
- ✿ /bin/dmesg
- ✿ /sbin/lspci
- ✿ /usr/bin/lsdev
- ✿ /sbin/lsmmod
- ✿ /sbin/modprobe
- ✿ /sbin/insmod
- ✿ /bin/uname
- ✿ /usr/bin/lsusb
- ✿ /etc/sysctl.conf, /etc/sysctl.d/
- ✿ /sbin/sysctl
- ✿ Udevmonitor
- ✿ udevadm monitor
- ✿ /etc/udev

06/24/14

21



Modules

- ✿ Kernel modules have a .o or a .ko extension depends on which kernel version 2.6 is .ko and previous versions are .o
- ✿ Modules are kept in /lib/modules/kernel-version
 - **modules.dep** (the list of module dependencies)
 - block - all of the modules for block devices
 - cdrom - all modules for CD-ROMs
 - ipv4 - IP networking modules
 - net - network interface driver modules
 - scsi - all SCSI device drivers
 - video - drivers for video capture devices
 - misc - sound, joysticks, serial, parallel, USB
- ✿ Modules have dependencies, relationships
 - Each module contains a list of dependencies needed
 - Mapping contained in /lib/modules/kver/modules.dep
 - Built with "**depmod -a**", called by rc.sysinit script
 - Must be run after compiling or changing modules

06/24/14

22



Loading and unloading modules

- ✿ Manually loading modules

```
#insmod fat
```

- ✿ Manually removing modules

```
#rmmod fat
```

- ✿ Listing loaded modules

```
# lsmod
```

Module	Size	Used by
vfat	10736	0 (autoclean)
fat	31264	0 (autoclean) [vfat]
mousedev	3968	1
input	3424	0 [mousedev hid]
usb-ohci	16464	0 (unused)

06/24/14

23



The modprobe command

- ✿ The **modprobe** command can be used to load and unload groups of modules and is used by the kernel when managing modules.

- ✿ modprobe gets its configuration information from one of the following files

- /etc/modules.conf
- /etc/conf.modules
- /etc/modprobe.d (directory)

- ✿ To insert a module or module stack with modprobe

```
#modprobe -a vfat
```

- ✿ To remove a module and all dependencies

```
#modprobe -r vfat
```

- ✿ To remove all autoclean modules i.e. inactive for 60+ seconds


```
#modprobe -r
```

06/24/14


24



Other modprobe options

 To list available modules of a specific type. This is relative to the `/lib/modules/kernel-ver/kernel/drivers` directory

#modprobe -l -t net

 To make modprobe send errors to the syslog rather than standard output

#modprobe -s badmodule

 To display all module configuration including `/etc/modules.conf` or `/etc/conf.modules`

#modprobe -c module.ko



The modinfo command


 To display a description and author information for a module as well to find out possible module options

#modinfo bonding



Configuring Modules

 Configuration data for modules is stored in `/etc/modules.conf`

 Some of the commonly used entries are as follows

- **keep** - found before a path directive, causes paths to be retained
- **depfile=file** - over-rides the default `modules.dep` file
- **path=path** - specifies additional directories to search
- **options modulename module-specific-options** - name and value pairs
- **alias aliasname result** - Aliases can be used to associate a generic name with a specific module

`alias /dev/ppp ppp_generic`


- **install module command** - allows a specific shell command to override the default module-insertion command

06/24/14


27



The module dependency file

 The module dependency file contains a relationship between modules that are required when loading the module.

`/lib/modules/kernel-version/modules.dep`

 A typical entry is shown below for the `ac97_codec`

`/lib/modules/2.4.9/kernel/drivers/sound/ac97_codec.o:`

`/lib/modules/2.4.9/kernel/drivers/sound/maestro3.o: \
/lib/modules/2.4.9/kernel/drivers/sound/ac97_codec.o`

 `depmod` rebuilds `modules.dep`

06/24/14

28



Modifying Kernel Parameters

- ✿ The directories `/proc` and `/sys` (virtual file systems) contain parameters that can be modified on the fly and thus modify kernel behaviour
- ✿ `/proc/net/ipv4` contains “tweakable” network parameters e.g. tcp congestion control algorithm. You can do this using `sysctl` command `/sbin/sysctl -w kernel.domainname="mine.com"`
- ✿ To make changes permanent edit `/etc/sysctl.conf`.
- ✿ `lspci` - lists all PCI devices. Option `-k` shows kernel drivers(modules) handling each device
- ✿ `lsusb` - lists USB devices
- ✿ `lsdev` - gathers information about installed hardware like IRQ,I/O ports

06/24/14

29



Udev

- ✿ Udev is the Linux dynamic device management system. The `udev` daemon receives device uevents from the kernel whenever a device is added or removed e.g. USB memory stick
- ✿ Udev then examines a file with configured rules and if it finds a match for the device it applies the rules which could include creating dynamically a device entry in `/dev` directory
- ✿ `udev` expects its main configuration file at `/etc/udev/udev.conf`
- ✿ Rule files are normally here `/lib/udev/rules.d/`
- ✿ Custom rules can also be found here `/etc/udev/rules.d/`
- ✿ `udevadm` is a utility that can query and modify the behavior of the `udev` daemon
`udevadm monitor` will monitor kernel uevent and display what the event `udev` sends out

06/24/14

30





202.1 Customizing system startup and boot processes

Weight 3

Description

Candidates should be able to query and modify the behaviour of system services at various run levels. A thorough understanding of the init structure and boot process is required. This objective includes interacting with run levels.

Key Knowledge Areas

-  Linux Standard Base Specification (LSB)
-  SysV init environment

Terms and Utilities:

-  /etc/inittab
-  /etc/init.d/
-  /etc/rc.d/
-  chkconfig
-  update-rc.d
-  init and telinit



The Linux Standard Base LSB

✿ *The goal of the LSB is to develop and promote a set of open standards that will increase compatibility among Linux distributions and enable software applications to run on any compliant system even in binary form. In addition, the LSB will help coordinate efforts to recruit software vendors to port and write products for Linux Operating System.*

✿ <http://www.linuxfoundation.org/collaborate/workgroups/lsb>

09/14/12



The Boot process

✿ The **/sbin/init** process is started with Process ID of 1.

✿ The **/sbin/init** process reads the **/etc/inittab** configuration file.

✿ The **/etc/inittab** is very important to the system administrator. If you can find and read this file, it identifies the whole boot-up order, and everything else can be found from here.

09/14/12



/etc/inittab

✿ The `/etc/inittab` describes the processes started at boot time.

✿ The lines are in the format of

id:runlevels:action:process

- ***id*** is a unique sequence of 1-4 characters which identifies an entry in `inittab`
- ***runlevels*** lists the runlevels for which the specified action should be taken.
- ***action*** describes which action should be taken.
- ***process*** specifies the process to be executed. If the process field starts with a '+' character, `init` will not do utmp and wtmp accounting for that process. This is needed for gettys that insist on doing their own utmp/wtmp housekeeping.

09/14/12



/etc/inittab continued

✿ The following line at the start of the `inittab` defines the default run level to enter on boot

id:5:initdefault:

✿ This entry defines the default system initialisation script to run, it may be named differently on different distributions


si::sysinit:/etc/rc.d/rc.sysinit **Fedora/RH**

si::sysinit:/etc/rcS **Debian derivatives**


09/14/12




/etc/inittab continued


 The next section defines how to run the defined run levels.

```
l0:0:wait:/etc/rc 0
l1:1:wait:/etc/rc 1
l2:2:wait:/etc/rc 2
l3:3:wait:/etc/rc 3
l4:4:wait:/etc/rc 4
l5:5:wait:/etc/rc 5
l6:6:wait:/etc/rc 6
```

 In the above the run control script will execute passing an argument of the run level to it.

 The rc script then executes the K* and S* script links in the corresponding run level directory. The example below is the network S Script which would link to the daemon script

```
/etc/rc.d/rc3.d/S50Network -----linked to -----> /etc/rc.d/init.d/network
```


 This will either start or stop the daemons at that run level. The daemon scripts are stored in the following directory

```
/etc/rc.d/init.d/
```




09/14/12


/etc/inittab continued

 The next line defines how to handle CTRL +ALT +DEL. In this case a shutdown is issued.

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

 The next and final entries will start the default terminals. The respawn option stops them from being terminated.


```
1:23:respawn:/sbin/mingetty tty1
2:23:respawn:/sbin/mingetty tty2
3:23:respawn:/sbin/mingetty tty3
4:23:respawn:/sbin/mingetty tty4
x:5:respawn:/etc/X11/prefdm -nodaemon
```

 The user is then prompted with a login either in a console or in an XDM login manager defined in the `/etc/X11/prefdm` script.



09/14/12

Other Actions in the /etc/inittab

 Valid actions for the *action* field are:

 **respawn**

- The process will be restarted whenever it terminates (e.g. `getty`).

 **wait**

- The process will be started once when the specified runlevel is entered and `init` will wait for its termination.

 **once**

- The process will be executed once when the specified runlevel is entered.

 **boot**

- The process will be executed during system boot. The *runlevels* field is ignored.

 **bootwait**

- The process will be executed during system boot, while `init` waits for its termination (e.g. `/etc/rc`). The *runlevels* field is ignored.

 **off**

- This does nothing.

09/14/12




Other Actions in the /etc/inittab

 **ondemand**


- A process marked with an **ondemand** runlevel will be executed whenever the specified **ondemand** runlevel is called. However, no runlevel change will occur (**ondemand** runlevels are 'a', 'b', and 'c').

 **initdefault**

- An **initdefault** entry specifies the runlevel which should be entered after system boot. If none exists, `init` will ask for a runlevel on the console. The *process* field is ignored.

 **sysinit**

- The process will be executed during system boot. It will be executed before any **boot** or **bootwait** entries. The *runlevels* field is ignored.

 **powerwait**

- The process will be executed when the power goes down. `init` is usually informed about this by a process talking to a UPS connected to the computer. `init` will wait for the process to finish before continuing.

09/14/12



Other Actions in the `/etc/inittab`

`powerfail`

- As for `powerwait`, except that `init` does not wait for the process completion.

`powerokwait`

- This process will be executed as soon as `init` is informed that the power has been restored.

`powerfailnow`

- This process will be executed when `init` is told that the battery of the external UPS is almost empty and the power is failing (provided that the external UPS and the monitoring process are able to detect this condition).

`ctrlaltdel`

- The process will be executed when `init` receives the SIGINT signal. This means that someone on the system console has pressed the **CTRL-ALT-DEL** key combination. Typically one wants to execute some sort of **shutdown** either to get into single-user level or to reboot the machine


09/14/12



What daemon runs at what runlevel


 The runlevels are defined by the links in the `/etc/rc.d/rcX.d` where X is the runlevel.

 These links can be added/viewed and removed using various tools. Two of these command line tools are **update-rc.d** (Debian) and **chkconfig** (Fedora/RH and now Debian)

 Inserting and removing these links **does not** disable the daemon immediately. You must stop the daemon manually or bounce the run level down and back up.

09/14/12



 To view the current daemons configured

On RedHat/Fedora systems

```
#chkconfig --list
```

```
#chkconfig --list httpd
```

```
#chkconfig --levels 345 httpd on
```

```
#chkconfig --levels 345 httpd off
```

```
#chkconfig --add mydaemon
```

```
#chkconfig --del mydaemon
```

On Debian Systems


```
#update-rc.d apache2 defaults
```

```
#update-rc.d -f apache2 remove
```


09/14/12




Changing runlevels

 To view your current and previous runlevel.


```
#runlevel
```

 Changing the current runlevel

```
#init 3 or telinit 3
```

 To force a re-read of the /etc/inittab after a change has been made

```
#init q or telinit q
```

 Shutting down the system to a halted state

```
#shutdown -h now
```

Note: When init is executed by a user process, it will actually run /sbin/telinit

09/14/12



Upstart and Systemd

- ✿ Upstart and systemd are not part of the exam, but they are now used on many distributions.
- ✿ Systemd is becoming the dominant replacement for traditional **init** daemon. It is comparable with SysV and LSB init scripts but it provides aggressive parallelization capabilities. (<https://fedoraproject.org/wiki/Systemd>)
- ✿ Upstart is a daemon which replaces the **/etc/inittab** with a content with a set of files located in **/etc/event.d**
- ✿ The **/etc/inittab** now only contains the default runlevel
- ✿ In the directory **/etc/event.d** there are separate script files for each of the following
 - **rcS**
 - **rc[0-6]**
 - **control-alt-delete**
 - **Serial**

06/14/12



202.2 System Recovery

Weight: 4

Description:

Candidates should be able to properly manipulate a Linux system during both the boot process and during recovery mode. This objective includes using both the **init** utility and **init**-related kernel options.

Candidates should be able to determine the cause of errors in loading and usage of bootloaders. GRUB version 2 and GRUB Legacy are the bootloaders of interest.

Key Knowledge Areas

GRUB version 2 and Legacy

grub shell

boot loader start and hand off to kernel

kernel loading

hardware initialization and setup

daemon/service initialization and setup

Know the different boot loader install locations on a hard disk or removable device

Overwriting standard boot loader options and using boot loader shells

Awareness of UEFI

06/23/14



GRUB version 2 & Legacy

- ✿ GRUB (GRand Unified Boot Loader) is now the default boot loader for most Linux distributions. Its job is to choose a kernel from a previously assembled list and to load it with options specified by the administrator
- ✿ There are two branches of the GRUB lineage: the original GRUB, now called GRUB Legacy and the newer GRUB 2. They differ in config file syntax
- ✿ GRUB 2 releases actually have version numbers between 1 and 2
- ✿ GRUB supports a powerful command-line interface as well as facilities for editing file entries on the fly.

06/23/14

17



GRUB Legacy

- ✿ By default, GRUB reads its boot configuration from `/boot/grub/menu.lst` or `/boot/grub/grub.conf` at startup
- ✿ The `menu.lst` and `grub.conf` files are slightly different but have a similar syntax. Red Hat systems use `grub.conf`, SUSE, and Ubuntu still use `menu.lst`. Sample `grub.conf` file:

```
default=0
timeout=10
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
title Red Hat Enterprise Linux Server (2.6.18-92.1.10.el5)
root (hd0,0)
kernel /vmlinuz-2.6.18-92.1.10.el5 ro root=LABEL=/
```
- ✿ Only one OS configured which is booted automatically (`default=0`) if it doesn't receive input within 10 second (`timeout=10`). Root file system of OS (`hd0,0`) first partition on first hard disk. The `/vmlinuz-2.6.18-92.1.10.el5` is loaded and displays a splash screen from file `/boot/grub/splash.xpm.gz`

06/23/14

18



GRUB Command-line Interface

✿ To enter command-line mode, type `c` from the GRUB boot screen. From here you can boot OSes that are not listed in `grub.conf`, display system information, and perform basic file system testing

✿ Command completion and cursor movement are also supported. Press `<Tab>` key to obtain a quick list of available commands

- `reboot` (reboots system)
- `find` (finds files on all mountable partitions)
- `root` (specifies the root device)
- `kernel` (loads a kernel from the root device)
- `help` (interactive help for a command)
- `boot` (boots the system from the specified kernel image)

06/23/14

19



Kernel Boot-time Options

✿ GRUB lets you pass command-line options to the kernel which typically modify values. e.g.

- `acpi=off` (Disables Advanced Configuration and Power Interface components)
- `init=/bin/bash` (Starts only the bash shell; useful for emergency recovery)
- `root=/dev/foo` (Use `/dev/foo` as the root device)
- `single` (Boots to single-user mode)

✿ When edited at boot time, kernel options are not persistent. Edit the appropriate kernel line in `grub.conf` or `menu.lst` to make the change persist across reboots

06/23/14

20



Installing GRUB using grub-install

✿ The command **grub-install** will install GRUB where you specify as command argument

- **#grub-install /dev/sda** will install into the MBR of /dev/sda
- To specify a different directory to put images (default /boot) use **--boot-directory** argument
- grub-install copies GRUB images into /boot/grub, and uses grub-setup to install grub into the boot sector

06/23/14

21



MBR, BIOS, GPT, AND UEFI

✿ Master Boot Record (MBR) is the area at the beginning of a hard disk (512bytes) which contains boot code (GRUB stage 1) and a partition table (DOS) that allows up to 4 primary partitions and additional logical partitions. This is used on PC BIOS platforms.

✿ The GRUB stage 1.5 is normally installed between the MBR and the first partition (this area is usually at least 31KiB in size). This means you must ensure your first file system begins at 63 sectors from start of the disk

✿ GRUB stage 2 which is responsible for loading the OS and can either be called directly by stage 1 or through stage 1.5 (contains file system drivers etc so it can locate stage 2 anywhere on the filesystem)

✿ Due to limitations of the MBR system, newer systems use the GUID partition table (GPT) format. This was specified as part of the Extensible Firmware Interface (EFI) which is to replace BIOS eventually. With GPT, it is possible to reserve a whole partition for GRUB, called the BIOS Boot Partition (minimum size 31KiB but 1MiB better to allow room for growth).

✿ Use GNU parted utility to create a GPT format partition.

06/23/14

22



GRUB 2

- ✿ Legacy GRUB is no longer under development
- ✿ GRUB 2 is complete rewrite of GRUB. It stores its configuration files differently. New features include ability to boot live CD ISOs from the hard drive, support for other architectures like PowerPC
- ✿ Uses `/boot/grub/grub.cfg` configuration file which contains scripts. Do not edit this file directly, instead edit `/etc/default/grub` or create custom files for menus by editing `/etc/grub.d/40_custom`
- ✿ Remember to run `grub-mkconfig` after making any changes. See man page for `grub-mkconfig` for more information.
- ✿ Also note that in GRUB 2 the first partition is 1 not 0 but the first device is still `hd0`.

06/23/14

23



202.3 Alternate Bootloaders

Weight: 2

Description: Candidates should be aware of other bootloaders and their major features.

Key Knowledge Areas

LILO

SYSLINUX, ISOLINUX, PXELINUX

Understanding of PXE

Terms and Utilities

`lilo`, `/etc/lilo.conf`

`Syslinux`

`Extlinux`

`pxelinux.cfg/*`

`isolinux.bin`

`isolinux.cfg`

`Pxlinux.0`

06/23/14

24



LILO

- ✿ LILO(Linux Loader) is a previously popular Linux kernel loader. It is hardly used these days.
- ✿ Uses `/etc/lilo.conf` for its configuration but requires that you run `/sbin/lilo` command if the configuration is modified
- ✿ Can be located in MBR or the boot sector of a partition. In the latter case the MBR code must be able to load LILO.

06/23/14

25



PXE (Pre-Execution Environment)

- ✿ PXE makes possible to boot computers using the network. This is usually the case for diskless computers.
- ✿ A working PXE set up requires DHCP server and a TFTP server to provide the network information(e.g. Assigning IP address and specifying IP address of TFTP server) and location of boot image
- ✿ PXE initiates a bootstrap session by using DHCP to obtain an IP address, a list of PXE Boot servers (IP addresses).Once contacted, a boot server can respond with a complete path to download a network bootstrap image via TFTP.
- ✿ PXELINUX(pxelinux.0) is an example of a network bootstrap image which is part of the SYSLINUX suite of lightweight bootloaders which includes ISOLINUX (for booting ISO images), EXTLINUX (to boot from ext2/3/4 filesystems)
- ✿ dnsmasq dhcp server is ideal for setting up a PXE system as it requires very little modification to make PXE ready
- ✿ EXTLINUX is a Syslinux variant which boots from a Linux filesystem

06/23/14

26






203.1 Operating the Linux filesystem

Weight 4








Description

Candidates should be able to properly configure and navigate the standard Linux filesystem.
This objective includes configuring and mounting various filesystem types.

Key Knowledge Areas

-  The concept of the fstab configuration
-  Tools and utilities for handling SWAP partitions and files
-  Use of UUIDs

Terms and Utilities:

-  /etc/fstab
-  /etc/mtab
-  /proc/mounts
-  mount and umount
-  sync
-  swapon
-  swapoff



The /etc/fstab

- ✿ The /etc/fstab holds information about what device files are mounted where.
- ✿ The format of the fstab is as follows and can use device names, volume names or UUIDs

Device	Mount Pt	Filesystem	Options	Dump	FSCK
/dev/sda1	/boot	ext3	defaults	1	1
UUID=bd3a8d84-9846 /		ext3	defaults	1	1
LABEL=swap	swap	swap	defaults	0	0

- ✿ To mount an entry that is in the fstab, you can mount by device name or mount point

#mount /dev/sda1 or **#mount /boot**

08/23/14

3



The /etc/mtab

- ✿ The /etc/mtab holds the currently mounted filesystems

#more /etc/mtab

- ✿ You can also look in the /proc/mounts


#more /proc/mounts

08/23/14

4



Volume names and UUID


 UUID = Universally Unique Identifier

 Volume names can be viewed in the `/dev/disk/by-label`

```
#ls -lah /dev/disk/by-label/
```

You can also use `dumpe2fs`

```
#dumpe2fs /dev/sda1 | grep -i volume
```

 UUIDs can be viewed in the `/dev/disk/by-uuid`


```
#ls -lah /dev/disk/by-uuid/
```

or using `dumpe2fs`

```
#dumpe2fs /dev/sda1 | grep -i uuid
```



The sync command

 The `sync` command forces the disk buffers to be flushed and committed to the disk blocks

```
#sync
```



Swap Space

Linux can use swap files or swap partitions.

To show what your system is currently using, use the following command

```
#swapon -s
```

This extracts its information from the `/proc/swaps` file

08/23/14

7



Swap partition

First the partition must exist which is created with `frisk`, but set the partition type to **82**.

Once the disk has been partitioned, use the `mishap` command to format it

```
#mishap /dev/sda3
```

It must then be enabled

```
#swapon /dev/sda3
```

For a permanent change edit the `/etc/fstab`

```
/dev/sda3    swap    swap    defaults    0 0
```

08/23/14

8



Swap file

✿ To create a swap file you must first make an empty raw file of the required size. The block size should be equal to the paging size. The count will then equal the size of the file, i.e. $1024 \times 1024 = 1\text{MB}$

```
#dd if=/dev/zero of=/extra swap bs=1024 count=1024
```

✿ Then use the **mishap** command to initialise it

```
#mishap /extra swap
```

✿ To enable it

```
#swapon /extra swap
```

✿ Check that it is enabled

```
#swapon -f
```

✿ For permanent swap edit the **/etc/fstab** and add the following line

```
/extra swap swap swap defaults 0 0
```

08/23/14

9



203.2 Maintaining a Linux filesystem

Weight 3

Description

Candidates should be able to properly maintain a Linux filesystem using system utilities. This objective includes manipulating standard filesystems and monitoring SMART devices.

Key Knowledge Areas

✿ Tools and utilities to manipulate and ext2, ext3 and ext4

✿ Tools and utilities to manipulate xfs

✿ Awareness of Btrfs

Terms and Utilities:

✿ fsck (fsck.*)

✿ mkfs (mkfs.*)

✿ dumpe2fs, xfsdump, xfsrestore

✿ debugfs

✿ tune2fs

✿ mishap

✿ Xfs_info, xfs_check and xfs_repair

✿ Smartd, smartctl

08/23/14

10



Tools for tuning ext2 and ext3

Ext2 and ext3 filesystems can be tuned using the tune2fs

Some examples are shown below

```
#tune2fs -j /dev/sda1      add a journal to an ext2
#tune2fs -c 10 /dev/sda3  set max mount count
#tune2fs -g users /dev/sda1    set group for reserved block
#tune2fs -L boot /dev/sda1    set a volume label
```

08/23/14

11



File system check with fsck

The file system check is done with fsck on an ext2/3/4 file system. It is important to unmount a filesystem before running an fsck as this can break the filesystem

```
#fsck -v -f /dev/sda1
#fsck.ext2 -v -f /dev/sda1
#fsck.ext3 /dev/sda1
#fsck.ext4 /dev/sda4
#fsck -c /dev/sda2      (runs badblocks to report bad blocks)
```

You can use the debugfs tool for low level tuning of the ext2/3 filesystems

```
#debugfs /dev/sda1
```

08/23/14

12



Ext4 filesystem

- Journaling file system successor to ext3
- Introduced in kernel 2.6.28
- Supports file up to 16 terabytes and volumes up to 1exabyte
- The file system uses extents, contiguous blocks of physical space
- Subdirectories increased from 32,000 in ext3 to 64,000
- Backwards compatible with commands such as e2fsck

08/23/14

13



SMART

- Self-Monitoring, Analysis and Reporting Technology (SMART) is a monitoring system for many ATA and SCSI hard disks
- The monitoring is hoped will make possible to predict an impending disk failure perhaps due to increased level of errors
- The specification also includes drive self-tests that can be executed to check state of the disk
- On Linux, /usr/sbin/smartd daemon is used to perform the SMART monitoring, logging errors and changes
- It polls the disk every 30 minutes by default
- smartd configuration file is /etc/smartd.conf
- smartctl can be used to query directly a SMART compliant hard drive from the command-line

08/23/14

14



Xfs file system

- Journaling file system created by Silicon Graphics
- Supports a maximum file system of 8 Exabytes
- Also uses extents
- To check an xfs file system you can use:
 - #fsck.xfs /dev/sda1
- #xfs_info provides information about a filesystem
- #xfs_check to check filesystem consistency
- #xfs_repair is similar to xfs_check
- These have to be invoked manually unlike fsck

08/23/14

15



Btrfs

- B-tree file system(Btrfs) is an advanced file system for Linux. It is described as a copy-on-write system

- Features include:

- Self-healing
- Online defragmentation
- RAID levels
- Snapshots
- Extent based file storage
- 2^{64} byte == 16 EiB maximum file size

It appears its goal is to be comparable to Oracle ZFS in terms of feature set.

08/23/14

16



203.3 Creating and configuring filesystem options

Weight 2

Description

Candidates should be able to configure automount filesystems using AutoFS. This objective includes configuring automount for network and device filesystems. Also included is creating filesystems for devices such as CD-ROMs and a basic feature knowledge of encrypted filesystems.

Key Knowledge Areas

- autofs configuration files
- UDF and ISO9660 tools and utilities
- Awareness of CD-ROM filesystems (UDF, ISO9660, HSF)
- Awareness of CD-ROM filesystem extensions (Joliet, Rock Ridge, El Torito)
- Basic feature knowledge of encrypted file systems

Terms and Utilities:

- /etc/auto.master
- /etc/auto.[dir]
- mkisofs

08/23/14

17



What is AutoFS

- AutoFS or automount automatically mounts a filesystem when the user attempts to access the directory where the filesystem would be mounted

08/23/14

18



The files related to AutoFS

- ✿ The `/etc/auto.master` looks like this
`/auto/etc/auto.misc --timeout=60`
- ✿ The 1st column sets where all mounts are to be mounted
- ✿ The 2nd column defines the mount file.
- ✿ The 3rd column sets the timeout value i.e. how long after use before autofs unmounts the FS

08/23/14

19



The `/etc/auto.misc`


- ✿ The `/etc/auto.misc` looks like
`kernel -ro,soft,intr ftp.kernel.org:/pub/linux`
`cd -fstype=iso9660,ro :/dev/cdrom`
`zip -fstype=auto :/dev/hdd4`
`Floppy -fstype=vfat :/dev/fd0`
`host_share -fstype=smbfs ://172.16.0.5/share`
- ✿ The 1st column defines where to mount, based upon `auto.master` previous, so `cdrom` would be mounted `/auto/cd`
- ✿ The 2nd column defines the option to pass at mounting
- ✿ The 3rd column defines the device file, `:` indicates a local device file, `kernel` would be an `nfs` share

08/23/14


20



Creating iso file systems

 The mkisofs command is used to create iso filesystems

```
#mkisofs -J -T -r -o backup.iso /home/luke
```

 The options are as follows


- -J creates a Joliet directory records that are compatible with windows based systems
- -T creates the TRANS.TBL mapping file in each directory for non Rock Ridge capable devices
- -r sets ownership to 0 and all bits to readable
- -o sets the output file

08/23/14

21




Using cdrecord

 If you have a CD-writer and a separate CD-ROM drive.

```
#cdrecord -v dev=0,6,0 speed=2 -isosize /dev/scd0
```

this reads the data stream from the CD-ROM drive attached as /dev/scd0 and writes it directly to the CD-writer.

 To create an image file

```
#dd if=/dev/scd0 of=cdimage
```

08/23/14

22



CD File Systems

- HSF was the early logical file system for CD-ROMs
- ISO-9660 is now the standard CD-ROM file system
- UDF (Universal Disc Format) is the next generation for DVD discs



CD File System Extensions

- There are range of extensions supported buy Linux to allow for differences in naming conventions.
- Rock Ridge extensions allow for support of Unix style long filenames, permissions and symbolic links.
- Joliet extensions were created by Microsoft for use by Windows. Linux supports Joliet as part of the iso9660 driver
- El Torito is an extension to the ISO-9660 standard to allow for the creation of bootable CD-ROMs



Encrypted File Systems

- Linux offers a number of encrypted file system options with differences in the way encryption is implemented.
- Loop-AES - filesystem and swap encryption package. Loop devices do not store any data directly but redirect to an underlying block device, encrypting on the way.
- DM-crypt - transparent encryption of block devices. Creates a new device in /dev
- | - a software system for on-the-fly-volume encryption (probably the best known). Data is encrypted/decrypted before it is loaded or saved, needs a password to access the data





204 Advanced Storage Device Administration

Copyright © Property of Firebrand Training Ltd


204.1 Configuring RAID

Weight 3


Description

Candidates should be able to configure and implement software RAID. This objective includes using and configuring RAID 0, 1 and 5.

Key Knowledge Areas


 Software raid configuration files and utilities

Terms and Utilities:

 mdadm.conf

 mdadm

 /proc/mdstat

 Partition type 0xFD



RAID

- ✿ RAID stands for Redundant Array of Inexpensive Disks. It can be used for redundancy and also for performance increase
- ✿ The main RAID versions are 0,1 and 5, but other levels are available
- ✿ **RAID 0** is called striping without parity, allowing increased throughput by reading in parallel. **Storage capacity of the array is equal to the sum of the capacity of the member disks.**
- ✿ **RAID 1** is a true redundancy by mirroring over to another drive. Minimum drives required is 2. **Array capacity is equal to the capacity of the smallest member disk.**
- ✿ **RAID 5** is striping with parity, which allows redundancy and an increase in speed. Minimum drives required is 3. **Array capacity is equal to the capacity of member disks, minus capacity of one member disk.**

06/23/14

3



Linux and RAID

- ✿ For the LPI we will be looking at software based RAID done through the operating system rather than a hardware device.

06/23/14

4



Setting up RAID 0

✿ First use `fdisk` to create new partition on the drives you are going to use. This will be one partition that fills the drive. You must make these of partition type **FD**.

✿ In this scenario we will use two drives

- `/dev/sdb1` and `/dev/sdc1` and of type **FD**

✿ Next using the `mdadm` tool create the RAID

```
#mdadm --create --verbose /dev/md0 --level=0 --raid-devices=2  
/dev/sdb1 /dev/sdc1
```

✿ Once created you must then format the device file with the desired filesystem and mount it

```
#mkfs /dev/md0  
#mount /dev/md0 /mnt/raid0
```

06/23/14

5



Setting up RAID 1

✿ First use `fdisk` to create new partition on the drives you are going to use. This will be one partition that fills the drive. You must make these of partition type **FD**.

✿ In this scenario we will use two drives

- `/dev/sdd1` and `/dev/sde1` and of type **FD**

✿ Next using the `mdadm` tool create the RAID

```
#mdadm --create --verbose /dev/md1 --level=1 --raid-devices=2  
/dev/sdd1 /dev/sde1
```

✿ Once created you must then format the device file with the desired filesystem and mount it

```
#mkfs /dev/md1  
#mount /dev/md1 /mnt/raid1
```

06/23/14

6



Setting up RAID 5

✿ First use `fdisk` to create new partition on the drives you are going to use. This will be one partition that fills the drive. You must make these of partition type **FD**.

✿ In this scenario we will use two drives

- `/dev/sdf1` , `/dev/sdg1` and `/dev/sdh1` of type FD

✿ Next using the `mdadm` tool create the RAID

```
#mdadm --create --verbose /dev/md2 --level=5 --raid-devices=3  
/dev/sdf1 /dev/sdg1 /dev/sdh1
```

✿ Once created you must then format the device file with the desired filesystem and mount it

```
#mkfs /dev/md2  
#mount /dev/md2 /mnt/raid5
```

06/23/14

7



The `/etc/mdadm.conf`

✿ The `/etc/mdadm.conf` holds information of the configuration for the raid sets.

✿ To build the `mdadm.conf` you can run the `mdadm` tool after the previous stage of building the raid

```
#mdadm --detail --scan  
  
ARRAY /dev/md5 level=raid5 num-devices=3  
UUID=1ba7e95a:24d106f6:7756543e:82c7b110  
devices=/dev/sdf1,/dev/sdg1,/dev/sdh1
```

✿ Add the output without line breaks to the `/etc/mdadm.conf`

06/23/14

8



Querying the RAID

✿ The `/proc/mdstat` holds information of the raids configured

```
#more /proc/mdstat
Personalities : [raid1]
read_ahead 1024 sectors
md5 : active raid1 sdb5[1] sda5[0]
    4200896 blocks [2/2] [UU]
md6 : active raid1 sdb6[1] sda6[0]
    2104384 blocks [2/2] [UU]
md7 : active raid1 sdb7[1] sda7[0]
    2104384 blocks [2/2] [UU]
md2 : active raid1 sdc7[1] sdd8[2] sde5[0]
    1052160 blocks [2/2] [UU]
unused devices: none
```

06/23/14

9



Querying the RAID

✿ Finally, remember that you can always use `raidtools` or `mdadm` to check the arrays out.

✿ Using `mdadm`

```
#mdadm --detail /dev/md0
```

✿ Using `lsraid`

```
#lsraid -a /dev/md0
```

06/23/14

10



204.2 Adjusting Storage Device Access

Weight 2

Description

Candidates should be able to configure kernel options to support various drives. This objective includes software tools to view & modify hard disk settings.

Key Knowledge Areas

- ✿ Tools and utilities to configure DMA for IDE devices including ATAPI and SATA
- ✿ Tools and utilities to manipulate or analyze system resources (e.g. interrupts)

Terms and Utilities:

- ✿ `hdparm, sdparm`
- ✿ `tune2fs`
- ✿ `sysctl`
- ✿ `/dev/hd* & /dev/sd*`
- ✿ `iscsiadm, scsi_id, iscsid` and `iscsid.conf`
- ✿ WWID, WWN, LUN numbers

06/23/14

11



Using hdparm

- ✿ `hdparm` allows you to tune the hard drive parameter, this is usually for IDE or EIDE, but basic functionality can be done on SATA as well.

```
#hdparm /dev/hda
/dev/hda:
multcount = 0 (off)
I/O support = 0 (default 16-bit)
unmaskirq = 0 (off)
using_dma = 0 (off)
keepsettings = 0 (off)
nowerr = 0 (off)
readonly = 0 (off)
readahead = 8 (on)
geometry = 1870/255/63, sectors = 30043440, start = 0
```

06/23/14

12



Turning on various settings

```
#hdparm -c3 -m16 /dev/hda
```

/dev/hda:

setting 32-bit I/O support flag to 3

setting multcount to 16

multcount = 16 (on)

I/O support = 3 (32-bit w/sync)

- **multcount:** This controls how many sectors are fetched from the disk in a single I/O interrupt. Almost all modern IDE drives support this. Can increase IO from 30% to 50%
- **I/O support:** This flag controls how data is passed from the PCI bus to the controller. Almost all modern controller chipsets support mode 3, or 32-bit mode w/sync.

06/23/14

13



Setting DMA using hdparm

✿ To set the correct DMA and transfer mode use the following

✿ For Multiword DMA mode 2 drives

```
#hdparm -d1 -X mdma2 /dev/hda
```

✿ For Simple DMA Mode 1 drives

```
#hdparm -d1 -X sdma1 /dev/hda
```

✿ For Ultra DMA Mode 2 drives


```
#hdparm -d1 -X udma2 /dev/hda
```

06/23/14

14



Viewing disk access timings


 To view information about read and write access speeds

```
#hdparm -Tt /dev/hda
```

```
/dev/hda:
```


```
Timing buffer-cache reads: 128 MB in 1.34 seconds =95.52 MB/sec
```


```
Timing buffered disk reads: 64 MB in 17.86 seconds = 3.58 MB/sec
```

 Use `sdparm` for scsi and sata disks




Using sysctl

 The `/sbin/sysctl` allows the system administrator to configure the kernel tuneable parameters

 To view the current settings

```
#sysctl -a
```

 Parameters can be set permanently in the configuration file

```
/etc/sysctl.conf
```



Using sdparm

- ✿ **sdparm** is used to display and change scsi devices mode pages. It can also send commands to the SCSI device e.g. starting and stopping device
- ✿ Mode pages hold meta data about a SCSI device which can, in some cases, be changed by the user.
- ✿ To see a list of generic mode page names that sdparm has some information about use: '**sdparm -e**'
- ✿ **sdparm -e -t sas** will display a list transport specific mode page names
- ✿ **sdparm -a /dev/sda** lists all known fields for given device
- ✿ **sdparm -c capacity /dev/sda** displays the number of blocks , length and capacity in MiB

06/23/14

17



Working with iSCSI Devices

- ✿ In iSCSI, commands for SCSI devices are carried inside an IP packet and delivered to the device via the network.
- ✿ In iSCSI, the storage node is called *target* and the source of the commands *initiator*. A LUN(logical unit number) represents an individual SCSI device. iSCSI Qualified Names (IQN) are used to refer to either initiator or target e.g.

iqn.2012-1.com.example:storage:diskarrays-sn-a12345

- ✿ Use **iscsiadm** command for administration of iSCSI. e.g.

iscsiadm --mode discovery --type sendtargets --portal 192.168.100.100

Above command will discover targets at the stated address

06/23/14

18




204.3 Logical Volume Manager


Weight 3


Description


Candidates should be able to create and remove logical volumes, volume groups, and physical volumes. This objective includes snapshots and resizing logical volumes.

Key Knowledge Areas


 Tools in the LVM suite


 Resizing, renaming, creating, and removing logical volumes, volume groups, and physical volumes


 Creating and maintaining snapshots

 Activating volume groups

Terms and Utilities:

 /sbin/pv*

 /sbin/lv*

 /sbin/vg*

 mount


 /dev/mapper/

06/23/14


19



What is an LVM

 Logical Volume Manager gives the System Administrator the ability to change the sizes of partitions without disrupting services.

 It can also be used as a backup service by using snapshots

 On smaller systems you don't have to be concerned with the size of partitions as you can resize it at a later date.

06/23/14

20



Overview of a Volume group

```
-----[ Volume Group ]-----
| filesystem      | filesystem      | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| logical volume  | logical volume  |
|+++++|+++++|+++++|+++++|+++++|+++++|+++++|+++++|+++++|+++++|
| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| physical volume | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|-----|
| partition      | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|-----|
```

mkfs

lvcreate

vgcreate

pvcreeate

fdisk (0x8e)

06/23/14

21



Creating an LVM

- ✿ An LVM can be created with a minimum of one partition, but to see its full use, then more than one partition is required.
- ✿ If you create an LVM on a single partition, you can, at a later date add another physical volume to the original LVM to extend the original capacity.

06/23/14

22



Create the physical volumes

✿ First create the physical volume. In order to do this use `fdisk` and create a partition of partition type **8e**. Consider the following partitions

- `/dev/sdb1` and `/dev/sdc1` and `/dev/sdd1` of type **8e**

✿ After rebooting, convert these partitions into physical volumes

```
#pvcreate /dev/sdb1 /dev/sdc1 /dev/sdd1
```

06/23/14

23



Create the Volume Group

✿ Next you need to create the Volume Group. In this example we will call it `VolumeA`

```
#vgcreate VolumeA /dev/sdb1 /dev/sdc1 /dev/sdd1
```

✿ This will create a device file in `/dev` called `VolumeA`

```
#ls -l /dev/VolumeA
```

✿ The size will be relative to the three partitions sizes that you added

06/23/14

24



Create the Logical Volume

✿ Within the Volume Group you now need to create the Logical Volumes.

To create a logical volume called lv0 of size 1GB

```
#lvcreate -L 1024 -n lv0 VolumeA
```

✿ Now make a filesystem in the logical volume lv0

```
#mkfs /dev/VolumeA/lv0
```

06/23/14

25



Extending and reducing the logical volume

✿ To extend the logical volume

```
#lvextend -L +2048 /dev/VolumeA/lv0
```

✿ You now need to make the FS the same size

```
#resize2fs /dev/VolumeA/lv0 2048M
```

✿ Similarly to reduce a logical volume

```
#resize2fs /dev/VolumeA/lv0 1024M
```

```
#lvreduce -L1G /dev/VolumeA/lv0
```


Note, reduce the filesystem size before reducing the volume

06/23/14

26




Other commands

 To activate and de activate the volume group

`#vgchange -ay VolumeA` Activate

`#vgchange -an VolumeA` Deactivate


 To add another physical volume to the Volume group create a new partition /dev/sde1 of type 8e

`#pvcreate /dev/sde1`


`#vgextend VolumeA /dev/sde1`




Removing the Volume

 To remove the logical volume

`#lvremove /dev/VolumeA/lv0`

 To completely remove a volume group


`#vgremove VolumeA`

 To remove the physical volume

`#pvremove /dev/sde1`




Displaying information about the Logical Volumes and Physical Volumes

 To display information about the Volume Group


#vgscan **Minimum information**

#vgdisplay **Most information**

 To display information about the Physical Volumes

#pvscan **Minimum information**

#pvdisplay **Most information**


 To display information about the Logical volumes


#lvscan **Minimum information**


#lvdisplay **Most information**



Volume snapshot


 An LVM snapshot is an exact copy of an LVM partition that has all the data from the LVM volume from the time the snapshot was created.

 The advantage of LVM snapshots is that they can be used to greatly reduce the amount of time that your services/databases are down during backups because a snapshot is usually created in fractions of a second.


 After the snapshot has been created, you can back up the snapshot while your services and databases are in normal operation.



Creating the snapshot

 To create a volume snapshot of lv0, first ensure that the Volume Group VolumeA has sufficient space. Then create a snapshot volume and link it to the lv0

```
#lvcreate -L1G -s -n lv0snapshot /dev/VolumeA/lv0
```

 Mount the snapshot volume to ensure it is correct

```
#mount /dev/VolumeA/lv0snapshot /mnt/snapshot
```





205 Networking Configuration



205.1 Basic networking configuration

Weight 3

Description

Candidates should be able to configure a network device to be able to connect to a local, wired or wireless, and a wide-area network. This objective includes being able to communicate between various subnets within a single network including both IPv4 and IPv6 networks.

Key Knowledge Areas

-  Utilities to configure and manipulate ethernet network interfaces
-  Configuring wireless networks

Terms and Utilities:

-  /sbin/route
-  /sbin/ipconfig
-  /sbin/imp
-  /user/sbin/arp
-  /sbin/iwconfig
-  /sbin/iwlist



205.2 Advanced Network Configuration and Troubleshooting

Weight 4

Description

Candidates should be able to configure a network device to implement various network authentication schemes. This objective includes configuring a multi-homed network device and resolving communication problems.

Key Knowledge Areas

- 🌸 Utilities to manipulate routing tables
- 🌸 Utilities to configure and manipulate ethernet network interfaces
- 🌸 Utilities to analyze the status of the network devices
- 🌸 Utilities to monitor and analyze the TCP/IP traffic

Terms and Utilities:

- 🌸 /sbin/route

09/14/12



205.2 Advanced Network Configuration and Troubleshooting

- 🌸 /sbin/ifconfig
- 🌸 /bin/netstat
- 🌸 /bin/ping
- 🌸 /usr/sbin/arp
- 🌸 /usr/sbin/tcpdump
- 🌸 /usr/sbin/lsof
- 🌸 /usr/bin/nc
- 🌸 /sbin/ip
- 🌸 nmap
- 🌸 wireshark

09/14/12



ifconfig

✿ The `ifconfig` command is used to set ip addresses on interfaces.

✿ To set the loopback address

```
#ifconfig lo 127.0.0.1
```

- There should be a corresponding entry in `/etc/hosts`

```
127.0.0.1 localhost
```

✿ To set the interface ip and netmask

- `#ifconfig eth0 172.16.0.1 netmask 255.255.255.0`

09/14/12



Wireless cards

✿ Wireless cards are configured using the `iwconfig` command and various `iw*` commands

✿ The main tool is `iwconfig` and can be used to set various wireless settings


```
#iwconfig wlan0 mode managed
```

```
#iwconfig wlan0 essid homersimpson
```

09/14/12



Typical Fedora setup

 The configuration file can contain the settings for the wireless card i.e. on a Fedora/RH system

- `/etc/sysconfig/network-scripts/ifcfg-wlan0`

`DEVICE=wlan0`

`ONBOOT=yes`

`BOOTPROTO=dhcp`

`TYPE=wireless`

`ESSID=homersimpson`

`CHANNEL=8`


`MODE=master`

`RATE=auto`

09/14/12



Other wireless tools

 To create a wpa configuration file use the `wpa_supplicant` tool. It will prompt you for the passphrase and create a `wpa_supplicant.conf` file

```
#wpa_supplicant -Dwext -i wlan0 -c /etc/wpa_supplicant.conf
```


 To see the available wireless networks

```
#iwlist wlan0 scan
```

09/14/12



The ip command

 The `/sbin/ip` command can be used to show and manipulate routing, devices and tunnels.

`#ip link show`

`#ip link set eth0 promisc on`

`#ip addr show`

`#ip monitor all`


`#ip route show`

`#ip route add 10.10.10.0/8 via 192.168.1.1 dev eth0`


09/14/12



The route command

 To view the current routing table issue the route command

`#route`

 You can also use the netstat command

`#netstat -rn`


09/14/12



Adding and removing a default route

 To add a default gateway


```
#route add default gw 172.16.0.1
```

 To delete the default route

```
#route del default gw 172.16.0.1
```

 To add a route to a network

```
#route add -net 10.0.0.0 netmask 255.0.0.0 dev eth1
```


 To delete a route from the route table


```
#route del -net 10.0.0.0 netmask 255.0.0.0 dev eth1
```

09/14/12




Arp entries


 The address resolution protocol or ARP is used to map IP address to hardware address.

 To view the current entries

```
#arp -a
```

 To add a static entry

```
#arp -s 172.16.0.1 00:23:45:23:52:25
```

 To delete an entry

```
#arp -d 172.16.0.1
```

09/14/12



The hosts file

✿ The `/etc/hosts` is a static mapping between the IP address and the host names.

✿ The contents look like

Sample `/etc/hosts` entry for localhost

127.0.0.1 localhost

172.16.0.100 mailserver

✿ The host resolution order is governed by `/etc/nsswitch.conf` particularly the line as follows

hosts: dns files

09/14/12



The `nsswitch.conf` fine control

✿ The `nsswitch.conf` file entries can give finer control through various parameters between the definitions

Hosts: dns [!UNAVAIL=return] files

✿ These parameters can be set to


[' (!? STATUS '=' ACTION)+ ']

- STATUS => success | notfound | unavail | tryagain
- ACTION => return | continue

09/14/12



More nsswitch.conf


 The STATUS values are the results of a call to a lookup function of a specific service.

- **success** No error occurred and the required entry is returned. The default action for this is 'return'.
- **notfound** The lookup process works ok but the required value was not found. The default action is 'continue'.
- **unavail** The service is permanently unavailable. This can either mean the needed file is not available, or, for DNS, the server is not available or does not allow queries. The default action is 'continue'.
- **tryagain** The service is temporarily unavailable. This could mean a file is locked or a server currently cannot accept more connections. The default action is 'continue'.

09/14/12



The lsof command


 The lsof command is used to list open files held open by processes.

```
#lsof -t `which apache`    the process ID using the apache process
#lsof /dev/sda1           what files are opened on sda1
#lsof /etc/passwd        who is using the /etc/passwd
#lsof -c init            shows what files are opened by init process
#lsof -u apache,luke     processes opened by user apache and user luke
#lsof +p 30297           what files are using the process whose PID is 30297
#lsof -i :80             list opened internet sockets on port 80
```

09/14/12



The netstat command

 Netstat prints information about the Linux networking subsystem. By default, **netstat** displays a list of open sockets. If you don't specify any address families, then the active sockets of all configured address families will be printed.

#netstat -r show routing table

#netstat -s show summary statistics

#netstat -p show PID and name of each program

#netstat -l show listening ports only

09/14/12




205.3 Troubleshooting network issues

Weight 4


Description

Candidates should be able to identify and correct common network setup issues, to include knowledge of locations for basic configuration files and commands.


Key Knowledge Areas


 Location and content of access restriction files


 Utilities to configure and manipulate ethernet network interfaces

 Utilities to manage routing tables

 Utilities to list network states.

 Utilities to gain information about the network configuration

 Methods of information about the recognized and used hardware devices

 System initialization files and their contents (SysV init process)

Terms and Utilities:

 /sbin/ifconfig

09/14/12



205.3 Troubleshooting network issues

- ✿ /sbin/route
- ✿ /bin/netstat
- ✿ /etc/network || /etc/sysconfig/network-scripts/
- ✿ System log files such as /var/log/syslog & /var/log/messages
- ✿ /bin/ping
- ✿ /etc/resolv.conf
- ✿ /etc/hosts
- ✿ /etc/hosts.allow & /etc/hosts.deny
- ✿ /etc/hostname | /etc/HOSTNAME
- ✿ /bin/hostname
- ✿ /usr/sbin/traceroute
- ✿ /usr/bin/dig
- ✿ /bin/dmesg
- ✿ /usr/bin/host

09/14/12



Redhat/Fedora Config files

- ✿ The configuration file for a RH/Fedora distribution is
`/etc/sysconfig/network-scripts/ifcfg-eth0`
- ✿ With a static configuration you would see output similar to:
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.1.73
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
- ✿ If the interface is configured for DHCP, you would see output similar to:
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp

09/14/12



Debian style systems

✿ The configuration file for a debian system can be found at **/etc/network/interfaces**

✿ This would produce output similar to: auto lo eth0

```
iface lo inet loopback
iface eth0 inet static
    address 192.168.15.5
    netmask 255.255.255.0
    network 192.168.15.0
    broadcast 192.168.15.255
    gateway 192.168.15.2
```

09/14/12



Troubleshooting with ifconfig

✿ The ifconfig command will show general information about the card including ip address and netmask.


```
#ifconfig eth0
```

```
eth0 Link encap:Ethernet HWaddr 00:10:60:58:05:36
inet addr:192.168.2.3 Bcast:192.168.2.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:1398185 errors:0 dropped:0 overruns:0 frame:0
TX packets:1411351 errors:0 dropped:0 overruns:0 carrier:0
collisions:829 txqueuelen:100
RX bytes:903174205 (861.3 Mb) TX bytes:201042106 (191.7 Mb)
Interrupt:11 Base address:0xa000
```


09/14/12



Connectivity and route

 You can use the ping command to send an ICMP echo request to the host

```
#ping -c4 172.16.0.1
```


 The traceroute command sends a packet that has an incrementing TTL value to track a packet over the network


```
#traceroute www.bbc.co.uk
```

09/14/12




Troubleshooting the route

 Make sure your routing is correct by issuing the route command or netstat command.

 Check that the default gateway is set and a route to your default network

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.11.0	0.0.0.0	255.255.255.0	U 0	0	0	0	eth0
145.66.8.0	0.0.0.0	255.255.252.0	U 0	0	0	0	eth1
0.0.0.0	145.66.11.254	0.0.0.0	UG	0	0	0	eth1

 Flags can be as follows

- U - The route is UP
- G - The route is a GATEWAY
- H - The route is to a HOST

09/14/12



Packet analysis

🌀 There are various tools for packet analysis, these include Wireshark, tcpdump.

🌀 These use the libpcap packet capture library.

🌀 Wireshark is the GUI version of tcpdump

#wireshark

#tcpdump

09/14/12



Dig and nslookup

🌀 To check names resolution, use either dig or nslookup.

🌀 Names resolution depends on three files

- **/etc/nsswitch.conf**
- **/etc/hosts**
- **/etc/resolv.conf**

🌀 Use dig to check for names resolution

#dig www.google.com

#dig ocf.co.uk MX

#dig ocf.co.uk NS

#dig www.ocf.co.uk @172.16.0.6

Similarly

#nslookup www.ocf.co.uk

09/14/12



Netcat

Netcat is used for many uses. It can create a listener on a port, can be used to telnet connect to a port, port scan and tunnel traffic.

To create a listener on port 9999 that will execute a bash shell when connected to, issue the following command.

```
#nc -l -d -e /bin/bash -p 9999
```

09/14/12



IP version 6 (IPv6)

IPv6 was first supported by Linux in 1996 but it wasn't until the kernel 2.6 where implementation was standard.

A 128 bit address represented as 8 blocks of hexadecimal numbers

A full address would be: fe80:0000:0000:0000:0a2e:5fff:fe10:5f53

Leading zeros can be omitted and successive blocks of zeros can be concatenated to :: so the address becomes: fe80::a2e:5fff:fe10:5f53

The IPv6 loopback address is ::1 (all zeros with a 1 at the end)

09/14/12



IP version 6

- ✿ If both protocol stacks are loaded the ifconfig command will show an inet (IPv4) and an inet6 (IPv6) address
- ✿ The ping command for IPv6 is ping6
e.g. ping6 ::1 to ping the loopback
- ✿ A host can have multiple IPv6 addresses simultaneously, the type of address being identified by the prefix:
 - fe80: is a link local address, the local LAN
 - fec0: is a site local address, the local private network
 - 2xxx: is a public global unicast address

09/14/12



IP version 6

- ✿ No more broadcast addresses. There are three types of IPv6 address:
 - Unicast
 - Multicast
 - Anycast

09/14/12



IP version 6

- ✿ To check if IPv6 is running do an `ifconfig` or
- ✿ Check for an entry in `/etc/net/if_net6`
- ✿ To trace an IPv6 address use `traceroute6` or `tracert6`
- ✿ Tools like `tcpdump` can parse IPv6 addresses
- ✿ In DNS a host record becomes an AAAA record
- ✿ The `ip` command now becomes `ip -6`
e.g. `ip -6 addr show dev <interface>`





206 System Maintenance

206.1 Make and install programs from source

Weight 2

Description

Candidates should be able to build and install an executable program from source. This objective includes being able to unpack a file of sources.

Key Knowledge Areas

- Unpack source code using common compression and archive utilities.
- Understand basics of invoking make to compile programs.
- Apply parameters to a configure script.
- Know where sources are stored by default.

Terms and Utilities:

- /usr/src/
- gunzip
- gzip
- bzip2
- tar
- configure
- Make
- Uname
- Install
- patch

06/23/14



Installing programs from source

- ✿ Source compiling advantages
 - Allows inspecting code for flaws, security
 - Available earlier than binary packages
 - Compiling optimizes for current system
 - Feeds that “bleeding edge” need
- ✿ Source compiling disadvantages
 - More complex installation
 - Needed dependencies may not exist
 - Often poorly documented, readme.txt only
 - Difficult to uninstall source packages
- ✿ Programs can be downloaded as source code. It is generally delivered in a tar.gz or tarball format.

06/23/14



The configure script and Makefile

- ✿ The **configure** script checks the system
 - Ensures the proper compilers are present
 - Updates the Makefile
- ✿ **Makefile** are made up of:
 - **Platform** - The platform of the system
 - **Debug** - How to handle errors
 - **Optimize** - Items that are customized by .configure
 - **Source** - Where the source files are found
- ✿ Paths and Variables in the **Makefile**
 - **install-prefix** = .
 - **bin_dir** = \$(install-prefix)/bin/\$(SYS_TYPE)
 - **uparm_dir** = \$(install-prefix)/lib/uparm
 - **include_dir** = \$(install-prefix)/include

06/23/14



Changing the values in the Makefile

✿ The code in the Makefile is not normally edited, but various parameters are passed to the **configure** script to change the values required.

✿ Some parameters could be

- **--prefix** sets the directory where to install the software, by default `/usr/local/bin`
- **--bindir** defines where the binary will be located
- **--sysconfdir** sets where the machine specific data is stored i.e. `/etc`

✿ An example would be when the source code was not in the current directory

```
#!/configure --srcdir=/usr/src
```

✿ You can also pass variables to the configure script

```
#!/configure CC=gcc
```

06/23/14



Installing apps from source

✿ First download the source and extract it. Source code should generally be extracted to `/usr/src/progname` subdirectory

```
#tar -zxvf mysrcapp.tar.gz
```

✿ Now change to the directory that contains the route of the source

```
#cd mysrcapp
```

✿ In this directory there could be **README** or **INSTALL** files which you can view with the **more** command.

✿ Normally the configure script can be run first, which will check all dependencies are installed and build the **Makefile**

✿ Now run the **make** command to compile the source code.

✿ If the build was successful, then run the **make install** which should install the files in the correct location. Note, this can only be run by root if the program installs into system directories. This generally installs onto `/usr/local/bin`

06/23/14









206.2 Backup operations

Weight 3

Description

Candidates should be able to use system tools to back up important system data.

Key Knowledge Areas







-  Knowledge about directories that have to be included in backups
-  Awareness of network backup solutions such as Amanda, Bacula and BackupPC?
-  Knowledge of the benefits and drawbacks of tapes, CDR, disk or other backup media
-  Perform partial and manual backups.
-  Verify the integrity of backup files.
-  Partially or fully restore backups.

06/23/14



206.2 Backup operations

Terms and Utilities:

-  /bin/sh
-  dd
-  tar
-  /dev/st* and /dev/nst*
-  mt
-  rsync

06/23/14



Backups

- ✿ Backups are important to prevent loss of valuable data and to keep uptime to a maximum.
- ✿ Data loss can be caused by many issues, including hardware failure, software errors, natural disasters and user error.

06/23/14



What to backup

- ✿ What to backup is important on a Linux machine. Some directories like /tmp /proc and swap partitions would not required backup.
- ✿ Consideration should be made on files in /var (spool files) and maybe /bin and /sbin. Some of these files may never change and only require backing up infrequently.
- ✿ Other issues to consider are time frame to create the backup, time required to restore the data. You may only have a limited backup window.

06/23/14



Backup types

✿ The types of backup are as follows

- **Copy** = copies files, no archive bit reset
 - File ghosting, slower
- **Full** = all files on system, archive bit reset
 - Safe, slower, easy restore
- **Incremental** = all since last inc or full, bit reset
 - Fast, only changed files, takes longer to restore
- **Differential** = all since last inc or full, no bit reset
 - Best combo of all, use towards end of week

06/23/14



Backup software

✿ There are many commercial and open source products to schedule your backups, these include


- AMANDA <http://www.amanda.org/>
- Bacula <http://www.bacula.org/en/>
- BackupPC <http://backuppc.sourceforge.net/>


✿ But don't forget the installed software like **tar**, **dump** and **cpio** and also the compression software like **bzip**, **compress** and **zip**.

06/23/14



Using tar


 The tar command can be used to roll up many files into one single file. It can then be compressed with bzip or gzip to create a tar ball.

 To tar the contents of a directory

```
#tar -cf tarfile.tar *
```

```
#tar -czf tarfile.tar.gz
```


Note tar is unusual as it does not require the - in front of the switches

 To extract the files from the tape archive use

```
#tar -xf tarfile.tar
```

```
#tar -zxf tarfile.tar.gz
```

If the file is bzipipped then use -jxf


 To view the contents of a tar file


```
#tar -tf tarfile.tar
```

06/23/14




Using dd

 dd can be used for a variety of functions, one of its main ones being cloning drives or partitions from one place to another


 To create an image of a disk or partition

```
#dd if=/dev/sda of=/root/diskimage.dd
```

```
#dd if=/dev/sda1 of=/root/part1.dd
```

 To create a clone of one disk to another

```
#dd if=/dev/sda of=/dev/sdb
```

 To wipe a hard drive

```
#dd if=/dev/zero of=/dev/sda
```

06/23/14



Using rsync

- ✿ Rsync is a tool that allows you to copy only the changes made to a filesystem. It is like doing a copy based upon a **diff** over a network.
- ✿ Only the actual diffs are copied and they are compressed to increase performance.
- ✿ This allows backup and mirroring simply
- ✿ Rsync requires one machine to be a server and run the rsync daemon.
- ✿ Then changes can be copied from and to the servers from one or more systems.

06/23/14



A typical /etc/rsyncd.conf on server side


- ✿ The following file would be saved on the rsync server

```
motd file = /etc/rsyncd.motd
log file = /var/log/rsyncd.log pid
file = /var/run/rsyncd.pid
lock file = /var/run/rsync.lock
[modulename]
  path = /rsync_files_here
  comment = My Very Own Rsync Server
  uid = nobody
  gid = nobody
  read only = no
  list = yes
  auth users = username
  secrets file = /etc/rsyncd.scrt
```


06/23/14



Starting the rsync server

 The following is a breakdown of the previous configuration file

- path - this is the actual filesystem path to where the files are rsync'ed from and/or to.
- comment - a short, descriptive explanation of what and where the path points to for listings.
- auth users - you really should put this in to restrict access to only a pre-defined user that you specify in the following secrets file - does not have to be a valid system user.
- secrets file - the file containing plaintext key/value pairs of usernames and passwords.

 Once set up, then run the rsync in daemon mode to start the server

```
#rsync --daemon
```

06/23/14



Using rsync client

 To download changes from the rsync server use the rsync command

```
#rsync -avzrpog luke@server.lpiclass.lab::modulename /path/to/put-data
```

The switches are as follows

archive(a), verbose(v), recursive(r), preserve permissions(p), preserve owner(o), preserve group(g), compress (z)

 You can view what is available on the rsync server

```
#rsync server.lpiclass.lab
```

06/23/14



Simple rsync commands

✿ Some simple rsync techniques

✿ To sync a directory to another

```
#rsync -av /path/to/source /home/luke/rsync/daily
```

✿ To sync a directory to another server

```
#rsync -av /path/to/source luke@picclass.lab:/home/luke/rsync/daily
```

✿ The above only adds, to delete

```
#rsync -av -delete /path/to/source  
luke@picclass.lab:/home/luke/rsync/daily
```

✿ To only sync isos

```
#rsync -zrv -include="*.iso" host:/home/luke /home/
```

06/23/14




Tape device control using mt

✿ The tape device can be controlled using the mt command. It can be used to rewind, eject, fast forward, list status of the tape and erase the tape.


06/23/14




Using the mt and tar command to backup

 Rewind tape drive:


- # mt -f /dev/st0 rewind

 Backup directory /www and /home with tar command (z - compressed):

```
# tar -czf /dev/st0 /www /home
```

 Find out what block you are at with mt command:

```
# mt -f /dev/st0 tell
```

 Display list of files on tape drive:

```
# tar -tzf /dev/st0
```


06/23/14




Using the mt and tar command to backup

 Restore /www directory:


```
# cd /  
# mt -f /dev/st0 rewind  
# tar -xzf /dev/st0 www
```

 Unload the tape:

```
# mt -f /dev/st0 offline
```

 Display status information about the tape unit:

```
# mt -f /dev/st0 status
```

 Erase the tape:

```
# mt -f /dev/st0 erase
```

06/23/14



206.3 Notify users on system-related issues

Weight 1

Description

Candidates should be able to notify the users about current issues related to the system.

Key Knowledge Areas

🌀 Automate communication with users through logon messages.

🌀 Inform active users of system maintenance

Terms and Utilities:

🌀 `/etc/issue`

🌀 `/etc/issue.net`

🌀 `/etc/motd`

🌀 `wall`

🌀 `/sbin/shutdown`

06/23/14



The message of the day

🌀 The Message of the Day (Not Match of the Day) is a text file which holds a message displayed to users after login

`/etc/motd`

🌀 The `/etc/issue` and `/etc/issue.net` is displayed to users before they have logged in.


`/etc/issue`

`/etc/issue.net` (Only users coming in on telnet)


06/23/14



Write and wall

 If you wish to notify an individual user on the system, use the write command

```
#write lcrowe /dev/pts/2
```


 To notify all users on the system with wall

```
#wall "Message to all users, system going down in 5 Mins"
```

06/23/14



Notifying users during shutdown

 The shutdown command has the ability to notify all users when shutdown has been initiated

```
#shutdown -h 15:00 "System shutting down at 15:00"
```

06/23/14






207.1 Basic DNS server configuration

Weight 3

Description

Candidates should be able to configure BIND to function as a caching-only DNS server. This objective includes the ability to convert older BIND configuration files to newer format, managing a running server and configuring logging.

Key Knowledge Areas

-  BIND 9.x configuration files, terms and utilities
-  Defining the location of the BIND zone files in BIND configuration files
-  Reloading modified configuration and zone files

Terms and Utilities:

-  /etc/named.conf
-  /var/named/*
-  /usr/sbin/rndc
-  Kill
-  dig



207.2 Create and maintain DNS zones

Weight 3

Description

Candidates should be able to create a zone file for a forward or reverse zone or root level servers. This objective includes setting appropriate values for records, adding hosts in zones and adding zones to the DNS. A candidate should also be able to delegate zones to another DNS server.

Key Knowledge Areas

- BIND 9 configuration files, terms and utilities
- Utilities to request information from the DNS server
- Layout, content and file location of the BIND zone files
- Various methods to add a new host in the zone files, including reverse zones

Terms and Utilities:

- /var/named/*
- zone file syntax
- resource record formats
- dig
- nslookup
- host

06/23/14



207.3 Securing a DNS server

Weight 2

Description

Candidates should be able to configure a DNS server to run as a non-root user and run in a chroot jail. This objective includes secure exchange of data between DNS servers.

Key Knowledge Areas

- BIND 9 configuration files
- Configuring BIND to run in a chroot jail
- Split configuration of BIND using the forwarders statement

Terms and Utilities:

- /etc/named.conf
- /etc/passwd
- DNSSEC
- Dnssec-keygen
- dnsec-signzone

06/23/14



What is BIND

- ✿ BIND is the Berkley Internet Name Daemon or commonly known as Named or DNS. Its purpose is to provide names resolution to IP addresses.
- ✿ The current version is version 9 and can be installed from the distribution repository using apt or yum.
- ✿ Version 9 supports many enhancements including IPv6 and DNSSec.

06/23/14



How it works

- ✿ DNS is a hierarchical tree structure with the top level being a '.' known as root. Under the root there are the TLDs or Top Level Domains (.com .edu .org .net)
- ✿ When a query arrives at your name server it first looks in the cache. If it is in the cache then it will answer from the cache.
- ✿ If no entry is in the cache then the name server will see if it can match a smaller part i.e. www.bbc.co.uk, then bbc.co.uk, then co.uk, then uk. Then finally it will check '.'

06/23/14



Types of DNS Implementation

Master/Primary Name server

- A master or primary name server is a DNS server that is Authoritative for one or more name spaces or zones. The domain information is directly updated on these machines

Slave/Secondary Name Server

- The slave or secondary name server holds the IP addresses of machines in its domain, but transfers them from the Master/Primary names server. This is known as a zone transfer. No direct domain modification takes place on these servers

06/23/14



Types of DNS Implementation

Caching name server

- Only caches name server information and does not hold any files

Forwarding name server

- A forwarding name server is one which simply forwards all requests to another DNS and caches the results. This can be used for split configuration when hosting internal and external DNS servers. This can also be known as a Stealth or DMZ DNS configuration.

06/23/14



Query types

- ✿ **Recursive** - When performed for a client, the DNS server stays with the query until it is resolved. The answer is returned or an error is returned.
- ✿ **Iterative** - The server when it does not have the answer will refer the client to another name server that may have the answer. The best answer the name server has is returned even if it is partial. Usually used between name servers to obtain partial name resolutions.
- ✿ **Reverse** - The client provides the IP address and asks for the name. In other queries the name is provided, and the IP address is returned to the client. Reverse lookup entries for a network 192.168.100.0 is "100.168.192.in-addr.arpa".

06/23/14



The configuration files

- ✿ The location of the files can depend on distribution and whether you are running named in a chrooted environment
- ✿ The main configuration file is named.conf and can be found in one of the following locations
 - `/etc/named.conf`
 - `/var/named/chroot/etc/named.conf`
- ✿ The other files are the **zone files** which hold the address mappings for each zone stated in the named.conf. They can be found in following directories
 - `/var/named/`
 - `/var/named/chroot/var/named/`

06/23/14



The named.conf file

- ✿ This file defines how the DNS server will operate, what zones it is authoritative for and where the zone files are located
- ✿ It can also hold information for the DNS security.

06/23/14



Global options in the named.conf

- ✿ Here is a typical global section of the named.conf shown with comments //

```
options {
//directory states location of zone files relative to root or chroot
    directory "/var/named";
// version statement - inhibited for security
    version "get lost";
// optional - disables all transfers
// slaves allowed in zone clauses
    allow-transfer {"none"};
// Closed DNS - permits only local IPs to issue recursive queries
// remove if an Open DNS required to support all users
// or add additional ranges
    allow-recursion {192.168.3.0/24};
};
```

06/23/14



Required zone information in the named.conf

There are 3 entries in the /etc/named.conf that are required for bind to work. They are as follows

```
root-servers
zone "." in{
    type hint;
    file "named.ca";
};
localhost
zone "localhost" in{
    type master;
    file "localhost";
};
reverse-map
zone "0.0.127.in-addr.arpa" in{
    type master;
    file "127.0.0";
};
```

06/23/14




The authoritative zone configuration in the named.conf

- The following set of slides shows various configurations depending on whether the server is a master, slave, caching or forwarder, reverse lookup.
- Only put the entries relative to the type of server you are implementing.

06/23/14



A sample master configuration in the named.conf


 This is the entry in the named.conf file for a master server for the domain lpi.test

```
// lpi.test fragment from named.conf
// defines this server as a zone master
zone "lpi.test" in{
    type master;
    file "pri.lpi.test";
};
```

06/23/14



A sample master-reverse lookup configuration in the named.conf


 This is an example of a master reverse lookup entry in the named.conf file for a ip address 192.168.23.0 network.

```
// the reverse map zone declaration would look
// something like this
zone "23.168.192.in-addr.arpa" in{
    type master;
    file "192.168.23.rev";
};
```

06/23/14



A sample slave configuration in the named.conf

 This is the entry in the named.conf file for a slave server for the domain lpi.test

```
// lpi.test fragment from named.conf
// defines this server as a zone slave
zone "lpi.test" in{
    type slave;
    file "sec.lpi.test";
    masters {172.16.0.5};
};
```

06/23/14



A sample caching configuration in the named.conf


 This is the entry in the named.conf file for a caching name server

```
// options section fragment of named.conf
// recursion yes is the default and may be omitted
options {
    directory "/var/named";
    version "not currently available";
    recursion yes;
};
// zone section ....
// the DOT indicates the root domain = all domains
zone "." IN {
    type hint; file "named.ca";
};
```

06/23/14



A sample global forwarding configuration in the named.conf


 This is the entry in the named.conf file for a forwarding name server which will forward all requests to 10.0.0.1 or 10.0.0.2

```
// options section fragment of named.conf
// forwarders can have multiple choices
options {
    directory "/var/named";
    version "not currently available";
    forwarders {10.0.0.1; 10.0.0.2;};
    forward only;
};
// zone file sections ....
```

06/23/14



A sample per zone forwarding configuration in named.conf

 This configuration only forwards requests for example.com to 10.0.0.1 and 10.0.0.2

```
// zone section fragment of named.conf
zone "example.com" IN {
    type forward;
    forwarders {10.0.0.1; 10.0.0.2;};
};
```

06/23/14



The zone files

Zone files are stored in `/var/named/` or `/var/named/chroot/var/named/` directory.

As with the `named.conf` there are three zone files that are required to run

`/var/named/named.ca`

`/var/named/localhost`

`/var/named/127.0.0`

06/23/14



A typical zone file

```
$TTL 3D
@      IN SOA      ns.lpi.test. hostmaster.lpi.test. (
        199802151 ; serial, todays date + todays serial #
        8H      ; refresh, seconds
        2H      ; retry, seconds
        4W      ; expire, seconds
        1D )    ; minimum, seconds
;

NS     ns          ; Inet Address of name server
MX     10 mail.lpi.test. ; Primary Mail Exchanger
MX     20 mail.sec_lpi.test. ; Secondary Mail Exchanger
;
localhost      A      127.0.0.1
gw              A      192.168.196.1
ns              A      192.168.196.2
mail           A      192.168.196.4
```

06/23/14



A typical reverse lookup record

```
$TTL 3D
@ IN SOA ns.lpi.test. hostmaster.lpi.test. (
    199802151      ; Serial, todays date + todays
    serial 8H      ; Refresh
    2H            ; Retry
    4W            ; Expire
    1D)           ; Minimum TTL
NS ns.lpi.test.

1 PTR gw.lpi.test.
2 PTR ns.lpi.test.
4 PTR mail.lpi.test.
```

06/23/14



Reloading the zone records

✿ Whenever a change is made to a zone file, the serial number must be incremented.

✿ Next run the `rndc` to reload the zone file

```
#rndc reload
```

06/23/14



Checking the name server is working

✿ The easiest way to test the DNS server is to use the **dig** command.

✿ To query for a specific host

```
#dig mail.lpi.test
```

✿ To query the domain for a list of mail exchangers

```
#dig lpi.test MX
```

✿ To query for the name servers for lpi.test

```
#dig lpi.test NS
```

✿ To query for a host using a specific DNS server

```
#dig www.bbc.co.uk @172.16.0.5
```

✿ You can also use nslookup or host

```
#host www.bbc.co.uk or #nslookup www.bbc.co.uk
```

06/23/14



DNS Security

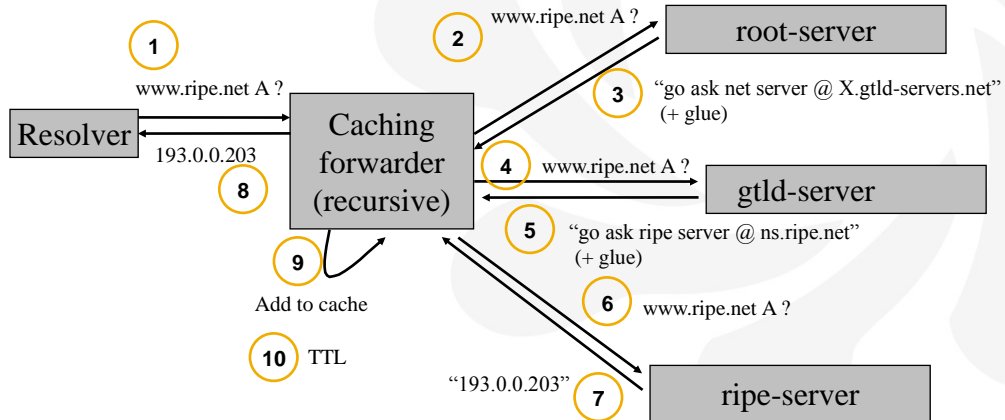
✿ **DNSSEC** was designed to protect the Internet from certain attacks, such as DNS cache poisoning. It is a set of extensions to DNS.

06/23/14



Reminder: DNS Resolving

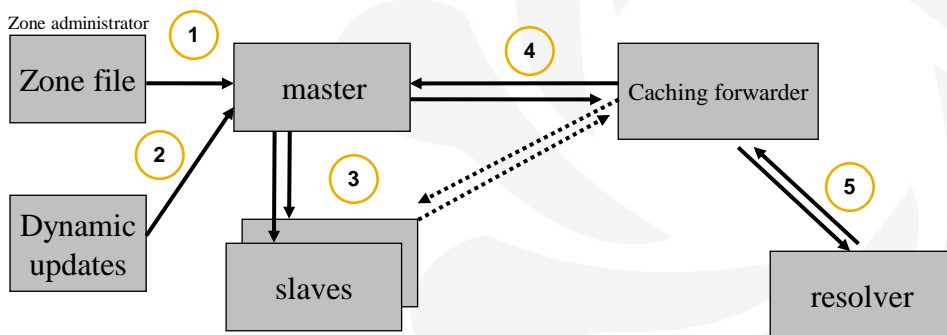
www.ripe.net A



06/23/14



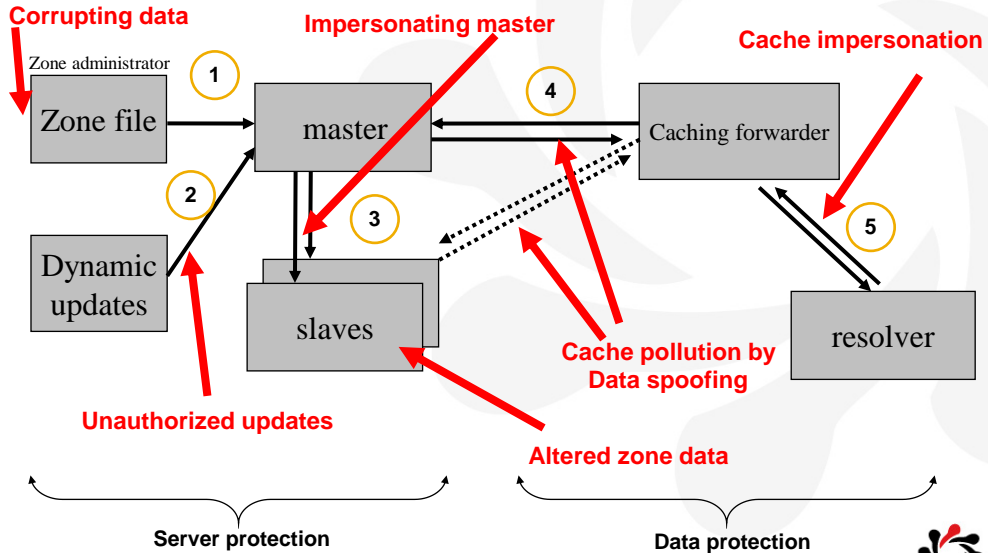
DNS: Data Flow



06/23/14



DNS Vulnerabilities



DNS Protocol Vulnerability

- DNS data can be spoofed and corrupted between master server and resolver or forwarder
- The DNS protocol does not allow you to check the validity of DNS data
 - Exploited by bugs in resolver implementation (predictable transaction ID)
 - Polluted caching forwarders can cause harm for quite some time (TTL)
 - Corrupted DNS data might end up in caches and stay there for a long time
- How does a slave (secondary) know it is talking to the proper master (primary)?

What does DNSSEC protect

DNSSEC protects against data spoofing and corruption

- **TSIG/SIG0**: provides mechanisms to authenticate communication between servers
- **DNSKEY/RRSIG/NSEC**: provides mechanisms to establish authenticity and integrity of data
- **DS**: provides a mechanism to delegate trust to public keys of third parties
- A secure DNS will be used as an infrastructure with public keys
 - However it is **NOT** a PKI

06/23/14



The Bind DNSSEC Tools

• **dnssec-keygen**

- Is used to generate keys of various types

• **dnssec-signzone**

- Is used to sign a zone that you wish to secure

• **dig**

- Is used to troubleshoot problems with DNS and DNSSEC

#**dig +dnssec @...**


• **named-checkzone** & **named-checkconf**

- These tools check the syntax for the zonefiles and the /etc/named.conf


06/23/14



Turning on DNSSEC in the named.conf

 To turn on DNSSEC, set in “options” statement

- **dnssec-enable yes;**

 It is also advisable to turn on logging for troubleshooting

- Several categories
- Categories are processed in one or more channels
- Channels specify where the output goes
- The following slide shows typical settings

06/23/14




Logging Configuration in the named.conf

```
logging {
    channel query_channel {
        file "log/querylog" versions 3;
        print-time yes;
    };
    channel dnssec_log {
        file "log/dnssec" versions 2;
        print-time yes; // add timestamp the entries
        print-category yes; // add category name
        print-severity yes; // add severity level
        severity debug 3; // print debug messages
    };
    channel everything_else {
        file "log/runlog" versions 3;
        print-time yes;
        print-severity yes;
        print-category yes;
    };
    category dnssec { dnssec_log; };
    category security { dnssec_log; everything_else; };
    category queries { query_channel; };
    category default { everything_else; };
};
```


06/23/14



Creating keys using dnssec-keygen

 To generate the key pair

```
#dnssec-keygen -a RSASHA1 -b 1024 -n zone example.net.
```


 The above command will create 2 files.


- **Kexample.net.+005+20704.key**
 - This will contains the public key and will be included in the zone file
- **Kexample.net.+005+20704.private**
 - This contains the private key and **should be kept secret**

06/23/14




Signing the files

 You only need to sign authoritative records

 The NS records for the zone itself are signed

 NS records for delegations are not signed

- DS RRs are signed!

 Glue is not signed

06/23/14



Preparing the zonefile

- ✿ The public keys generated must be set in the zonefile
 - #cat **Kexample.net.+005+20704.key** >> **example.net**
- ✿ Next, check the zonefile is correct using the **named-checkzone**
- ✿ Remember to increase the SOA serial number

06/23/14



Sign the zone

- ✿ The next thing to do is to sign the zone
 - #**dnssec-signzone** [options] zonefile [ZSK's]
- ✿ The signed zonefile will be called
 - zonefilename.signed**
- ✿ Keyset is created as a bonus...
 - ready to go to parent
- ✿ To create DS records from keyset files:
 - use **-g** option


06/23/14



Publishing the signed zone

 Edit named.conf:

```
zone "example.net" {  
    type master;  
    file "zones/example.net.signed";  
    allow-transfer { 10.1.2.3 ;  
                    key mstr-slave.example.net.; };  
    notify yes;  
};
```


 To check the named.conf file, use `named-checkconf`

 To enable the signed zone, Reload zone


06/23/14



Setting up a verifying resolving name server

 To verify the content of a zone:

- Get the public (key signing) key and check that this key belongs to the zone owner

 Configure the keys you trust as secure entry points in named.conf

```
trusted-keys {  
    "example.net." 256 3 1 "AQ...QQ==";  
};
```

06/23/14



Testing a verifying forwarder

🌀 To test the forwarder

```
#dig +dnssec @172.16.0.5 record [TYPE]
```

🌀 Answer Flags are relevant

🌀 The following example is a query to an authoritative nameserver

```
; <<> DiG 9.1.1 <<> +dnssec @193.0.0.202 www.example.net
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1947
;; flags: qr aa rd; QUERY: 1, ANSWER: 4, AUTHORITY: 3, ADDITIONAL: 4
```



06/23/14



Testing a verifying forwarder dig: an example

```
; <<> DiG 9.3.0s20020122 <<> +dnssec @127.0.0.1 example.net NS
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31630
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, udp= 4096
;; QUESTION SECTION:
;example.net. IN NS

;; ANSWER SECTION:
example.net. 600 IN NS ns1.example.net.
example.net. 600 IN NS ns2.example.net.
example.net. 600 IN SIG NS 1 2 600 20020314134313
                20020212134313 47783 example.net.
DVC/ACejHtZylifpS6VSSqLa15xPH6p33HHnr3hC7eE6/QodM6fBi5z3
fsLhbQuuJ3pCEdi2bu+A0duuQ1QMiHPvrkYia4bKmoyyvWHwB3jcyFhW
lV4YOzX/fgkLUmu8ysGoID9C0CkSvNSE6rBCA0a3hfkkSht4FBsuA1oQ
yoc=
```

“use DNSSEC, if you can”

Authenticated Data

06/23/14





208 Web Services






208.1 Implementing a web server

Weight 4

Description

Candidates should be able to install and configure a web server. This objective includes monitoring the server's load and performance, restricting client user access, configuring support for scripting languages as modules and setting up client user authentication. Also included is configuring server options to restrict usage of resources. Candidates should be able to configure a web server to use virtual hosts and customize file access.

Key Knowledge Areas

-  Apache 2.x configuration files, terms and utilities
-  Apache log files configuration and content
-  Access restriction methods and files
-  mod_perl and PHP configuration
-  Client user authentication files and utilities



208.1 Implementing a web server

- ✿ Configuration of maximum requests, minimum and maximum servers and clients
- ✿ Apache 2.x virtual host implementation (with and without dedicated IP addresses)
- ✿ Using redirect statements in Apache's configuration files to customize file access

Terms and Utilities:

- ✿ access logs and error logs
- ✿ .htaccess
- ✿ httpd.conf
- ✿ mod_auth
- ✿ htpasswd
- ✿ AuthUserFile, AuthGroupFile
- ✿ apache2ctl
- ✿ httpd

06/24/14

3



Apache Installation

- ✿ If you haven't installed Apache when you installed the server, then you can install from a repository or download the sources from <http://httpd.apache.org/download.cgi>
- ✿ For Fedora/Redhat
 - #yum groupinstall "Web Server"**
- ✿ For Debian based systems
 - #apt-get install apache2**

06/24/14

4



The configuration files

✿ When installed the main configuration files are

For Redhat/Fedora

`/etc/httpd/conf/httpd.conf`

For Debian

`/etc/apache2/apache2.conf`

✿ There is also a sub directory called

`/etc/httpd/conf.d` or `/etc/apache2/conf.d`

✿ The `conf.d` directory generally hold small files that contain individual Apache directives like `mod_ssl` configuration. If you install extra web server functions you might find the changes in here.



What is in httpd.conf

✿ The `httpd.conf/apache2.conf` file contains configuration data including

- How many server children to start and maintain for performance
- Where your web page files are stored
- What page to displayed by default
- What modules to enable
- Where error messages are stored
- Where the web server logs to
- Virtual hosting information



The contents of the httpd.conf

- ✿ The configuration file is split into three sections, the **Global Environment**, **Main Server Configuration** and **Virtual Hosts**
- ✿ The Global environment defines control of the Apache Server Process, how many requests it can handle, where the configuration files are etc.
- ✿ The Main Server Configuration defines how the server responds to requests.
- ✿ The Virtual Hosts defines how the server handles multiple websites.

06/24/14

7



The Global Section

✿ The global section of the `/etc/httpd/conf/httpd.conf`


```
ServerType standalone
ServerRoot "/etc/httpd"
PidFile run/httpd.pid
StartServers 16
MinSpareServers 16
MaxSpareServers 64
MaxClients 512
User www
Group www
Listen 80
LoadModule auth_basic_module modules/mod_auth_basic.so
DirectoryIndex index.htm index.htm
```

06/24/14


8



The Main Section

 In the main section it defines the document root, where the html documents are stored

DocumentRoot “/var/www/html”

 Underneath this entry is a directive which defines the access to the document root. By default this is very strict.

<Directory />

Options FollowSymLinks

AllowOverride None


</Directory>

06/24/14

9



The directory directive

 Once the root directory directive has been assigned with high security, you then need to allow features within each directory

<Directory /var/www/html>


Options Indexes FollowSymLinks

AllowOverride None

Order allow,deny

Allow from all

</Directory>

 The above sets directory, directive sets the default

- **Options** - are features you would like to enable
- **AllowOverride** - enables the use of .htaccess files (in the above case htaccess files are ignored)
- **Order** - Tells apache in which order to apply the allow and deny directives
- **Allow** - sets that anybody can access as there is no deny directive

06/24/14

10



Controlling access

✿ If you wish to control access to a directory then you can use authentication, authorization and access control

- Authentication - validating who someone is, e.g. Username and password
- Authorization - Once a person is authenticated, then see if they are authorised to use the resource
- Access control - Access control can be granted or denied based upon many factors such as network address, time of day, etc.

06/24/14

11



Basic authentication

✿ First you must create a password file. The `htpasswd` command will do this and prompt you for a password

```
#htpasswd -c /usr/local/apache/passwd/passwords luke (new password file)
```

```
#htpasswd /usr/local/apache/passwd/passwords john (password file exists)
```

✿ Create the `.htaccess` in the document directory to protect the resource. The contents of the `.htaccess` file will look like

```
AuthType Basic
```

```
AuthName "By invitation"
```

```
AuthUserFile /usr/local/apache/passwd/passwords
```

```
Require user luke,john
```

✿ Note - This could be `Require valid-user` for any user in the passwords file

06/24/14

12



Applying access control in the httpd.conf

- ✿ Suppose the directory you wish to protect is `/var/www/html/protected`
- ✿ In the above directory, put the `.htaccess` file from the previous slide
- ✿ In the `httpd.conf` add a directory directive as follows

```
<Directory /var/www/html/protected>  
    AllowOverride AuthConfig  
</Directory>
```

06/24/14

13



Starting/Stopping the Apache Server Daemon

- ✿ The server can be controlled in many ways
- ✿ On Debian system

```
#!/etc/init.d/apache start  
#!/etc/init.d/apache stop
```
- ✿ On Fedora systems

```
#!/service httpd start  
#!/service httpd stop
```
- ✿ On both systems the `apachectl` or `apache2ctl` command can be used

```
#!/apache2ctl stop  
#!/apache2ctl start
```

06/24/14

14



The Apache binary

✿ The binary that executes when the daemon starts is either **httpd** on Fedora/RH or **apache2** on Debian.

✿ The binary can be used to control/show/test various aspects of the server. To display the help page

On Redhat/Fedora

```
#!/usr/sbin/httpd -h
```

On Debian

```
#!/usr/sbin/apache2 -h
```

06/24/14

15



Logfiles

✿ Apache generally logs to the following directories

```
/var/log/apache2
```

```
/var/log/httpd
```

✿ In this directory the logfiles will be **access_log** for client access logging and **error_log** for web server error messages

06/24/14

16



The logfile directives in the httpd.conf

✿ The **ErrorLog** directive tells apache where to send apache error messages

ErrorLog logs/error_log

✿ The **LogLevel** directive tells apache what level to log error messages at (Same as syslog)

LogLevel warn

✿ The **LogFormat** directive defines a name for a defined format (**combined** in the case below)

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" **combined**

✿ The **CustomLog** directive defines where the client access logs are stored and references the **LogFormat** name (**combined**)

CustomLog logs/access_log **combined**

06/24/14

17



Virtual Hosting

✿ A web server can host more than one site. Apache allows three types of hosting

✿ Domain name based virtual hosting

✿ IP address based virtual hosting

✿ Port based virtual hosting

06/24/14

18



Port based virtual hosting



This is what the httpd.conf file looks like for port based virtual hosting

```
Listen 80
Listen 8080

NameVirtualHost 172.20.30.40:80
NameVirtualHost 172.20.30.40:8080

<VirtualHost 172.20.30.40:80>
ServerName www.example1.com
DocumentRoot /www/domain-80
</VirtualHost>

<VirtualHost 172.20.30.40:8080>
ServerName www.example1.com
DocumentRoot /www/domain-8080
</VirtualHost>

<VirtualHost 172.20.30.40:80>
ServerName www.example2.org
DocumentRoot /www/otherdomain-80
</VirtualHost>

<VirtualHost 172.20.30.40:8080>
ServerName www.example2.org
DocumentRoot /www/otherdomain-8080
</VirtualHost>
```

06/24/14

19



Domain name based virtual hosting



This is what the httpd.conf file looks like for domain based virtual hosting

```
NameVirtualHost *:80

<VirtualHost *:80>
DocumentRoot /www/example1
ServerName www.example1.com
</VirtualHost>

<VirtualHost *:80>
DocumentRoot /www/example2
ServerName www.example2.org
</VirtualHost>
```

06/24/14

20



IP addressed virtual hosting

✿ This is what the httpd.conf file would look like for IP based virtual hosting

Listen 80

```
<VirtualHost 172.20.30.40>  
DocumentRoot /www/example1  
ServerName www.example1.com  
</VirtualHost>
```

```
<VirtualHost 172.20.30.50>  
DocumentRoot /www/example2  
ServerName www.example2.org  
</VirtualHost>
```

06/24/14

21



208.2 Apache configuration for HTTPS

Weight 3

Description

Candidates should be able to configure a web server to provide HTTPS.

Key Knowledge Areas

✿ SSL configuration files, tools and utilities

✿ SSL certificate handling

Terms and Utilities:

✿ SSL configuration files, tools and utilities

✿ Ability to generate a server private key and CSR for a commercial CA

✿ Ability to generate a self-signed Certificate from private CA

✿ Ability to install the key and Certificate

✿ Awareness of the issues with Virtual Hosting and use of SSL

✿ Security issues in SSL use

06/24/14

22



Loading Modules

Apache can extend its functionality by loading modules in. Not all versions of apache support this, so to identify if yours does execute the following command

```
#httpd -l
```

If the output includes `mod_so.c` then it will support modules.

Modules can include cluster support, sql authentication support, cookies and many others. Take a look at <http://modules.apache.org>

06/24/14

23



Enabling a module

In the `/etc/httpd/conf/httpd.conf` file, you can add a line as follows

This module allows basic authentication

```
LoadModule auth_basic_module modules/mod_auth_basic.so
```

This module allows HTTPS

```
LoadModule ssl_module modules/mod_ssl.so
```

To see if a module is available use

```
#httpd -M
```

06/24/14

24



OpenSSL and the Secure Socket Layer

- ✿ Apache web server is able to implement SSL through the OpenSSL system.
- ✿ This provides secure connections through Transport Layer Security (TLS) and Public/Private Key encryption

06/24/14

25



How SSL works

- ✿ The client browser connects to the Apache web server via a HTTPS request.
- ✿ The server returns the site's certificate which also includes the server public key. (mydomain.com_cert.pem)
- ✿ The certificate is checked by the browser for validity and tells the user if there is a problem (e.g. was it issued by a recognized, trusted certificate authority?).
- ✿ The browser creates a session key, which is encrypted with the server's public key (typically 1024, but could be 2048bits), which is then sent to the server.
- ✿ The server then decrypts this information using its private key (mydomain.com_key.pem) and extracts the session key sent by the browser.
- ✿ The session key is then used to create the session. This is a symmetric key used to encrypt and decrypt data exchanged by the browser and server. Browsers and servers usually negotiate the strongest mutually supported session.

06/24/14

26



Enabling SSL

✿ Make sure `mod_ssl` is installed and added into your configuration file. It should look like

```
LoadModule ssl_modules modules/mod_ssl.so
```

Listen 443

✿ Paths will vary between different distributions.

✿ Fedora 8 uses `/etc/httpd/conf.d/ssl.conf` which is referenced from `/etc/httpd/conf/httpd.conf` with the line **Include conf.d/*.conf**

✿ Debian uses a directory called `/etc/apache2/mods-enabled` this directory contains two files, `ssl.load` and `ssl.conf` if `ssl` is enabled.



The certificates and keys

✿ The next thing to do is to generate the certificate and keys.

✿ You must choose between a self signed certificate for internal use, or a CA signed certificate for external use.



Generating the Keys

✿ First you need to generate the public/private keys

```
#openssl genrsa -des3 -out mydomain.com.key 1024
```

✿ Now create the self signed certificate

```
#openssl req -new -key mydomain.com.key -x509 -out  
mydomain.com.crt
```

✿ If you wish to have a signed certificate from a CA then you need to generate a Cert Signing Request and get the CA to sign it.

```
#openssl req -new -key mydomain.com.key -out mydomain.com.csr
```



Where the file should be put

✿ The certificate files must be copied to the correct location as stated in the configuration files talked about earlier. The lines in the configuration file look like the following

✿ Fedora for example

```
SSLCertificateFile /etc/pki/tls/certs/mydomain.crt
```

```
SSLCertificateKeyFile /etc/pki/tls/private/mydomain.key
```

✿ For Debian systems

```
SSLCertificateFile /etc/ssl/certs/mydomain.crt
```

```
SSLCertificateKeyFile /etc/ssl/private/mydomain.key
```



208.3 Implementing a proxy server

Weight 2

Description

Candidates should be able to install and configure a proxy server, including access policies, authentication and resource usage.

Key Knowledge Areas

- 🌸 Squid 3.x configuration files, terms and utilities
- 🌸 Access restriction methods
- 🌸 Client user authentication methods
- 🌸 Layout and content of ACL in the Squid configuration files

Terms and Utilities:

- 🌸 squid.conf
- 🌸 acl
- 🌸 http_access

06/24/14

31



Squid Proxy

- 🌸 Squid proxy is the standard proxy server supplied with most Linux distributions. It supports HTTP, HTTPS and FTP. It allows the administrator to cache content and control access to the web.
- 🌸 It can be installed through the repo system or downloaded from the site
www.squid-cache.org

06/24/14

32



The config file

✿ The main configuration file is `/etc/squid/squid.conf`

✿ This defines such things as the port that squid listens on

`http_port 3128`

✿ The access log and where to log to

`access_log /var/log/squid/access.log`

✿ Where to store the cached pages

`cache_dir /var/spool/squid 100 16 256`

Where 100 is the MiB to store under this dir

16 is the number of first level sub dirs to create (L1 cache)

256 is the number of second level sub dirs to create (L2 cache)

06/24/14

33



http_access

✿ First the acl must be defined. Here we define an acl name of `localnet` and what that local net is.

`acl localnet src 192.168.1.0/255.255.255.0`

✿ We then define who can access http with the `http_access` directive selecting the acl name as defined above

`http_access allow localnet`

`icp_access allow localnet`

06/24/14

34



Proxy Authentication

Proxy server authentication uses the same protocols and techniques as web server authentication, but sends a challenge with the **proxy-authenticate** field rather than the **www-authenticate** field.

It is configured in the `squid.conf` file and requires three entries to be set

- The realm
- The access control list
- The authentication module

06/24/14

35



Authentication realm and access control list

The Realm defines what the user sees to authenticate against the proxy

```
# realm default
proxy_auth_realm Squid proxy-caching web server
```

The Access Control List section


```
# set up the acl name for the local network
acl localnetwork proxy_auth foo.bar.baz/xy.zz.y
# set up the acl name for user authentication
acl localusers proxy_auth REQUIRED
# set up all the denies for those not in the local network
http_access deny !localnetwork
# set up the user authentication
http_access allow localusers
# set up the allows for the local network
http_access allow localnetwork
# deny anything that passes beyond this point
http_access deny all
```

06/24/14

36



The authentication module

 Squid supports many auth types


- **LDAP**
Authenticates against LDAP databases. This needs open LDAP libraries from Openldap.org.
- **MSNT**
Microsoft NT domain authentication. This needs configuration changes made to the source.
- **NCSA**
Authenticates against the same type of password file as many NCSA-compliant web servers. No visible documentation, but the code is readable.
- **PAM**
Pluggable Authentication Module. Ideal for PAM-enabled systems like Debian Linux. PAM is configurable to use a variety of authentication systems.
- **SMB**
Authenticates against an SMB server such as Windows NT or Samba.
- **getpwnam**
Authenticates off the Unix password or shadow password file, or similar files which can be read by the C `getpwnam()` library function. There is no visible documentation or readable code. `man getpwnam` discusses the function. To use the shadow password file, the authenticator would need to be `setuid root`.

06/24/14

37



Setting the Authentication modules

 The authentication module is configured with the option

```
#authenticate_program authentication module authentication file  
authenticate_program example authenticate_program  
/squid/bin/ncsa_auth /squid/etc/passwd
```

06/24/14

38



208.4 Implementing Nginx as a web server and a reverse proxy

Weight: 2


Description: Candidates should be able to install and configure a reverse proxy server, Nginx. Basic configuration of Nginx as a HTTP server is included.

Key Knowledge Areas

 Nginx

 Reverse Proxy

 Basic Web Server

 Terms and Utilities

 /etc/nginx/


 nginx


06/24/14


39



Installing Nginx

 Nginx (pronounced engine x) is new web server that is fast and easy to configure. It can also function as a reverse proxy server as well as a mail proxy server

 For Linux, installation is best done using packages specific to a Linux distribution. These are available at:
http://nginx.org/en/linux_packages.html

 It has a modular architecture with one master and several worker processes; worker processes run under an unprivileged user

06/24/14

40



Nginx Configuration

- ✿ The default configuration file is `nginx.conf` and can be located here
`/usr/local/nginx/conf` or `/etc/nginx` or `/usr/local/etc/nginx`
- ✿ Start nginx by running the nginx binary. Thereafter it can be controlled by using `-s` option to the nginx binary

e.g. `#nginx -s stop`, `#nginx -s reload`
- ✿ The pid of the master process is written to
`/usr/local/nginx/logs/nginx.pid` or `/var/run/nginx.pid`
- ✿ The nginx configuration file consists of directives that control the nginx modules. These directives are divided into simple block directives

06/24/14

41



Sample Nginx Configuration File

- ✿ Create a directory where content to be served by the web server will be located e.g `/var/nginx/www`
- ✿ Open the configuration file (`nginx.conf`) and uncomment the following server block and add the following:

```
http {  
    server {  
        location / {  
            root /var/nginx/www;  
        }  
    }  
}
```

- ✿ With this configuration the server will serve content on port 80. Test <http://localhost/> after applying the configuration: `nginx -s reload`

06/24/14

42



Ngix as a Simple Proxy Server

 In this configuration, Nginx receives requests and passes them to the 192.168.4.5

```
http {  
    server {  
        listen 8080;  
        location / {  
            proxy_pass http://192.168.4.5;  
        }  
    }  
}
```



209.1 SAMBA Server Configuration






Weight 5

Description

Candidates should be able to set up a SAMBA server for various clients.

This objective includes setting up Samba for login clients and setting up the workgroup in which a server participates and defining shared directories and printers. Also covered is configuring a Linux client to use a Samba server. Troubleshooting installations is also tested.

Key Knowledge Areas


-  Samba 3 documentation
-  Samba configuration files
-  Samba tools and utilities
-  Mounting Samba shares on Linux
-  Samba daemons

06/23/14







1



209.1 SAMBA Server Configuration

-  Mapping Windows usernames to Linux usernames

Terms and Utilities:

-  smb, nmbd
-  smbstatus, testparm, smbpasswd, nmblookup
-  smbclient
-  net
-  /etc/smb/*
-  /var/log/samba/

06/23/14

2



Samba

- ✿ Samba allows you to share portions of your filesystem and printers as if they were a windows device.
- ✿ It can also be used to connect a Linux machine to connect to a windows share.
- ✿ There are two components to samba
 - SMBD - Which shares the portions of the filesystem
 - NMBD - Which provides Netbios Naming service
- ✿ When you start the SMB service both daemons are started.

06/23/14

3



The SMB Configuration file

- ✿ The main configuration files is
`/etc/samba/smb.conf` or `/usr/local/samba/lib/smb.conf`
- ✿ This file can be edited using vi or modified using an interface like SWAT.
- ✿ Each section of the smb.conf file that specifies a share, or a meta-service, is called a stanza. The global stanza specifies settings that affect all the other stanzas in the smb.conf file.
- ✿ Stanza are generally enclosed within [] and can include
 - [global]*** Holds information about the overall server, netbios name etc
 - [homes]*** Stanza sharing the home directories of the users
 - [printers]*** Stanza sharing the printer off the linux system (cups/lpd)

06/23/14

4



A typical smb.conf

```
[global]
workgroup = METRAN
encrypt passwords = yes
wins support = yes
log level = 1
max log size = 1000
read only = no

[homes]
browsable = no
map archive = yes

[printers]
path = /var/tmp
printable = yes
min print space = 2000


[test]
browsable = yes
read only = yes
path = /usr/local/samba/tmp
```

06/23/14

5



Typical entries in [global]

 The **[global]** section defines such items as browsing/server identification, network bindings, security level, password support, debugging/logging settings, password syncing.

workgroup = myworkgroup makes the samba server a member of the **myworkgroup** workgroup.

server string = %h server this becomes the Netbios server name for the samba server.

wins support = no turns off wins support in NMBD.


interfaces = 172.16.0.0/24 eth0 Forces Samba to listen for a specific network and interface.

06/23/14

6



System logging

 This section defines how to log and where to log messages and client connections.

log file = /var/log/samba/log.%m defines the name and location of the logfile for each machine that connects.

max log size = 1000 defines the maximum size of the logfile in kB.

syslog only = no set to yes if you only want to log through syslog server


syslog = 0 Sets the amount of information to log through syslog server, 0 indicates don't send to syslog. Used in combination with **syslog only** setting

06/23/14

7



User password control

 This section defines whether a users Samba password (changed with **smbpasswd** command) will change the Linux password in **/etc/shadow** at the same time.

unix password sync = yes

passwd program = /usr/bin/passwd %u

passwd chat = *Enter\snew\spassword:* %n\n

Retype\snew\s*\spassword:* %n\n *password\supdated\s*successfully

.

pam password change = yes

06/23/14

8



Authentication of Clients

- Each user who attempts to connect to a share not allowing guest access, must provide a password to make a successful connection. What Samba does with that password is the arena of the **security** configuration option. Samba currently supports four security levels on its network: *share*, *user*, *server*, and *domain*.
- The next slide shows the options that can be set under the **security** option.

06/23/14

9



Levels of security


- **Share-level security**
 - Each share in the workgroup has one or more passwords associated with it. Anyone who knows a valid password for the share can access it.
- **User-level security (this is the default)**
 - Each share in the workgroup is configured to allow access from certain users. With each initial tree connection, the Samba server verifies users and their passwords to allow them access to the share.
- **Server-level security**
 - This is the same as user-level security, except that the Samba server uses another server to validate users and their passwords before granting access to the share.
- **Domain-level security**
 - Samba becomes a member of a Windows NT domain and uses one of the domain's domain controllers—either the PDC or a BDC—to perform authentication. Once authenticated, the user is given a special token that allows them access to any share with appropriate access rights. With this token, the domain controller will not have to revalidate the user's password each time they attempt to access another share within the domain. The domain controller can be a Windows NT/2000 PDC or BDC, or Samba acting as a Windows NT PDC.

06/23/14

10



Share Level security

 With share-level security, each share has one or more passwords associated with it, with the client being authenticated when first connecting to the share. This differs from the other modes of security in that there are no restrictions as to whom can access a share, as long as that individual knows the correct password. Shares often have multiple passwords.

[global]

security = share

[accounts]

path = /home/samba/accounts

guest ok = no

writable = yes


username = luke, polly, andrew

06/23/14

11



User Level security

 In this example the **default mode of security** with Samba is *user-level security*. With this method, each share is assigned specific users that can access it. When a user requests a connection to a share, Samba authenticates by validating the given username and password with the authorised users in the configuration file and the passwords in the password database of the Samba server.

[global]

security = user

[accounts]

writable = yes

valid users = luke, john, peter

06/23/14

12



Server Level Security

- ✿ In this example the Samba is to use a separate password server under server-level security with the use of the password server global configuration option, as follows:

[global]

security = server

password server = account_svr1 account_svr2

06/23/14

13



Share definitions in smb.conf

- ✿ After the global section, the printer and share definitions are defined.

- ✿ A typical directory share may look like

[public]

comment = Public Stuff

path = /home/public

public = yes writable = yes

printable = no

write list = @staff The group that can write to this share from /etc/groups

[homes]

comment = Home Directories

browseable = no

read only = no

create mode = 0750

06/23/14

14



Sharing printers

✿ If an entry named **printers** is listed in the **smb.conf** file, then samba will share out the default printing system to samba. i.e. cups or lpd

```
[printers]
comment = All Printers
security = server
path = /var/spool/lpd/lp
browseable = no
printable = yes
public = yes
writable = no
create mode = 0700
```

✿ Otherwise you can define individual printers as follows

```
[deskjet]
printable = yes
path = /var/spool/samba/print
valid users = luke peter jack heather ed
```

06/23/14

15



Checking your configuration file

✿ The configuration file syntax can be checked using the **testparm** command

```
#testparm /etc/samba/smb.conf
```

✿ You can check the shares that are available on the server using the **smbclient** tool

```
#smbclient -L localhost
```

✿ Don't forget you can check the logfiles for issues with the daemon

```
#more /var/log/samba
```

06/23/14

16



Starting and stopping Samba

🌀 On Redhat/Fedora servers, the daemons can be configured to run using **chkconfig** command and the **service** command.

```
#chkconfig --levels 345 smb on
```

```
#service smb start
```

🌀 On Debian servers, the service command does not exist, so must be started using the daemon script directly

```
#!/etc/init.d/samba start
```

06/23/14

17



User administration

🌀 In Samba V3 the username and passwords are now stored in the tdbsam

🌀 To add a user to the database

```
#smbpasswd -a luke
```

🌀 To change your smb password

```
#smbpasswd
```

🌀 The tdbsam can be queried and edited using the **pdbedit** command. To list the settings on an account called luke


```
#pdbedit -Lv luke
```

06/23/14


18




smbstatus, smbtree and nmblookup tools

 To view the current build status of the `smbd`


```
#smbd -b | more
```

 To view the current status of the smb share use the `smbstatus` tool

```
#smbstatus
```

 To view all available shares on the network you can use the `smbtree`.
When you press enter you will be prompted for a password. Press return

```
#smbtree
```

 The `nmblookup` tool can be used to query for netbios names on remote servers

```
#nmblookup -A 172.16.0.5
```

06/23/14


19




Connecting to a share

 If you are using windows, then you need to map a network share and connect to the server using the UNC

```
c:\net use e: \\172.16.0.5\lpi
```

 Using `smbclient` command line to get an ftp like prompt

```
#smbclient //172.16.0.5/lpi -U lpi
```

 Using Linux command line to mount it as part of the filesystem

```
#mount -t smbfs -o username=lpi,password=password,rw  
//172.16.0.5/lpi /mnt/share
```

```
#smbmount //server/shared /mnt/share -o  
username=user,password=pass,rw
```

06/23/14

20



209.2 NFS Server Configuration

Weight 3

Description

Candidates should be able to export filesystems using NFS. This objective includes access restrictions, mounting an NFS filesystem on a client and securing NFS.

Key Knowledge Areas

- 🌀 NFS version 3 configuration files
- 🌀 NFS tools and utilities
- 🌀 Access restrictions to certain hosts and/or subnets
- 🌀 Mount options on server and client
- 🌀 TCP wrappers
- 🌀 Awareness of NFSv4

06/23/14

21



209.2 NFS Server Configuration

Terms and Utilities:

- 🌀 /etc/exports
- 🌀 exportfs
- 🌀 showmount
- 🌀 nfsstat
- 🌀 /proc/mounts
- 🌀 /etc/fstab
- 🌀 rpcinfo
- 🌀 mountd
- 🌀 portmapper

06/23/14

22



The /etc/exports

✿ The **/etc/exports** is the main configuration file for NFS. It contains a list of what directories you want to export and to whom you want to allow it.

✿ The format is as follows

```
/usr/local 192.168.0.1(ro) 192.168.0.2(ro)
```

```
/home 192.168.0.1(rw) 192.168.0.2(rw)
```

```
/temp 192.168.0.1/255.255.255.0(ro)
```

Note - You can use wildcards such as **.foo.com* or *192.168.* instead of hostnames

06/23/14

23



Options in the /etc/exports

✿ The directory entered includes sub directories within the directory

✿ The Machine name can be IP or hostname

✿ The options you can set are as follows

- **ro** - The directory is exported as read only (default)
- **rw** - The directory is exported as read write
- **no_root_squash** - Any file request made by user root on the client machine is treated as if it is made by user nobody on the server. If **no_root_squash** is selected, then root on the client machine will have the same level of access to the files on the system as root on the server.

06/23/14

24



An /etc/exports WARNING

✿ These are not the same in /etc/exports

`/export/dir hostname(rw,no_root_squash)`

`/export/dir hostname (rw,no_root_squash)`

✿ The first will grant **hostname** **rw** access to /export/dir without squashing root privileges.

✿ The second will grant **hostname** **rw** privileges with **root squash** and it will grant *everyone* else read/write access, without squashing root privileges.

06/23/14

25



The NFS daemon

✿ The NFS daemon depends on various RPC binaries and will be started when NFS is started. The following order of starting

- `rpc.portmap`
- `rpc.mountd`, `rpc.nfsd`
- `rpc.statd`, `rpc.lockd` (if necessary), and `rpc.rquotad`

✿ To start the nfs daemon

`#!/etc/init.d/nfs start`

`#service nfs start`

✿ On starting the nfs daemon the contents of the /etc/exports will be available to the clients.

06/23/14

26



Checking the daemon is running

✿ The `rpcinfo` tool is used to query the nfs status. The portmapper (`rpcbind`) binds to port 111 tcp.

#rpcinfo -p

program vers proto port

100000 2 tcp 111 portmapper

100011 1 udp 749 rquotad

100005 1 udp 759 mountd

100005 1 tcp 761 mountd

100005 3 udp 769 mountd

100003 2 udp 2049 nfs

100003 3 udp 2049 nfs

100024 1 udp 944 status

100024 1 tcp 946 status

100021 1 udp 1042 nlockmgr

100021 3 udp 1042 nlockmgr

06/23/14

27



Exporting the filesystem

✿ Once changes have been made to the `/etc/exports`, you must then export the filesystem.

✿ Make sure the nfs daemon is running and if any changes have been made to the `/etc/exports` re-export the FS

#exportfs -a

✿ This will export all entries in the `/etc/exports`

✿ You can also export a directory directly from the command line

- Export `/usr/bin` and `/var/adm` read-only to the world:

#exportfs -i -o ro /usr/bin /var/adm

- Export `/usr/bin` read-write only to systems, jupiter and saturn, when using DNS as the name service:

#exportfs -i -o rw=saturn:jupiter /usr/bin

06/23/14

28



NFS security with TCPd wrappers

- Security can be controlled directly through the `/etc/exports` but also by using tcpd wrappers
- The entries for the daemons must be put in the `/etc/hosts.allow` and `/etc/hosts.deny` respectively

```
/etc/hosts.deny  
lockd:ALL  
mountd:ALL  
rquotad:ALL  
statd:ALL
```


```
/etc/hosts.allow  
lockd: 192.168.0.1 , 192.168.0.2  
rquotad: 192.168.0.1 , 192.168.0.2  
mountd: 192.168.0.1 , 192.168.0.2  
statd: 192.168.0.1 , 192.168.0.2
```

06/23/14


29




Client Side NFS

 To mount the NFS filesystem on the client, the mount command can be used.

```
#mount saturn:/home /mnt/homes
```

 And similarly to unmount the FS

```
#umount /mnt/homes
```

 Set up permanent mounts in the `/etc/fstab`

```
saturn:/home /mnt nfs rw 0 0
```

06/23/14

30



Soft Mounting

- ✖ If a file request fails, the NFS client will report an error to the process on the client machine requesting the file access. Some programs can handle this with composure, most won't.
- ✖ This setting is not recommended; it is a recipe for corrupted files and lost data. You should especially not use this for mail disks --- if you value your mail, that is.

06/23/14

31



Hard Mounting

- ✖ The program accessing a file on a NFS mounted file system will hang when the server crashes. The process cannot be interrupted or killed (except by a "sure kill") unless you also specify **intr**. When the NFS server is back online the program will continue undisturbed from where it was. It is best to use **hard,intr** on all NFS mounted file systems.

06/23/14

32



Checking the NFS status

✿ The **showmount** tool can be used to check the current status of the NFS exported directories on a server

```
#showmount -e 172.16.0.5
```

```
172.16.0.5:/home
```

✿ The statistics of the server can be displayed with the **nfsstat** tool

```
#nfsstat
```

06/23/14

33



NFS version 4

✿ NFS version 4 is now shipping as standard on many Linux distributions

✿ Some enhancements featured in NFSv4 include

- Works better with firewalls and NAT
- Lock and mount protocols are now part of core NFS protocol
- Support for ACLs, Unicode filenames
- Support for replication and migration
- Strong, integrated security
- Requires support of RPC over streaming network transport protocols such as TCP
- An NFSv4-only client cannot communicate with versions 2 and 3

06/23/14

34



210.1 DHCP configuration

Weight 2

Description

Candidates should be able to configure a DHCP server. This objective includes setting default and per client options, adding static hosts and BOOTP hosts. Also included is configuring a DHCP relay agent and maintaining the DHCP server.

Key Knowledge Areas

- DHCP configuration files, terms and utilities
- Subnet and dynamically-allocated range setup

Terms and Utilities:

- dhcpd.conf
- dhcpd.leases
- /var/log/daemon.log and /var/log/messages
- arp
- dhcpd

06/23/14

1



What is DHCP Daemon


- The DHCP daemon or dhcpd as it is known on Linux is used to dynamically assign an IP address to client hosts when connected to the network.
- The client side is called **dhcpcd**, but can also be known as **dhclient**
- Both server and client are from ISC (Internet Systems Consortium)
- The relay daemon from the same organization is called **dhcrelay**

06/23/14

2



The /etc/dhcpd.conf

 Below is a typical /etc/dhcpd.conf file


```
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-name "lpiclass.org";
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.10 192.168.1.100;
    range 192.168.1.150 192.168.1.200;
}
```

06/23/14

3



Reserved IP addresses in the /etc/dhcpd.conf

 To reserve an IP address for a specific host with MAC address

```
host laptop10 {
    hardware ethernet 08:00:2b:4c:59:23;
    fixed-address 192.168.1.222;
}
```

06/23/14

4



Starting the DHCP daemon

✿ The leases file may have to be created if it doesn't exist. This file holds the currently used IP address leases.

```
#touch /var/state/dhcp/dhcpd.leases
```

✿ To start the DHCP daemon you can use

```
#service dhcpd start
```

```
#!/etc/init.d/dhcpd start
```

06/23/14

5



BOOTP

✿ The dhcpd server provides BOOTP support. Unlike DHCP, bootp does not provide a way of recovering an assigned address once it is no longer needed, therefore by default addresses are granted to clients in perpetuity.

✿ BOOTP was used in the past by diskless workstations but these now support DHCP

A simple bootp declaration:

```
host haagen {
    hardware ethernet 08:00:23:4a:5f:aa;
    fixed-address 239.252.197.9;
    filename "/tftpboot/pc23.boot";
}
```

06/23/14

6



DHCP Relay Agent (dhcrelay)

- ✿ The dhcrelay agent provides a means for relaying DHCP and BOOTP requests from a subnet with no DHCP server to with a DHCP server
- ✿ It supports both IPv4 and IPv6
- ✿ You invoke the command with a list of one or more server addresses to which queries should be relayed. Example:
 - /usr/sbin/dhcrelay -q -4 192.168.9.10(q:quiet,4 :IPv4)

06/23/14

7



Address Resolution protocol ARP


- ✿ The arp cache holds the relationship between a MAC address and an IP address.
- ✿ When trying to find where to send a packet, the client broadcasts for the IP address **Who has 192.168.254.1 tell 192.168.254.5** on the Layer 2 broadcast address **FF:FF:FF:FF:FF:FF**
- ✿ The response that comes back from 192.168.254.1 is an arp reply with **192.168.254.1 is at 00:11:2F:2E:4C:8D**
- ✿ This entry is held in the clients arp cache

06/23/14


8




Displaying the ARP cache

 To display the arp cache


#arp -a

 To set a static entry

#arp -s 172.16.0.1 12:df:23:3e:23:ef

 To delete a static entry

#arp -d 172.16.0.1

 To make permanent setting create a file `/etc/ethers` and put the entries of MAC and IP addresses

00:08:20:61:CA 172.16.0.5

06/23/14

9



210.2 PAM authentication


Weight 3

Description

The candidate should be able to configure PAM to support authentication using various available methods.


Key Knowledge Areas

 PAM configuration files, terms and utilities


 passwd and shadow passwords

Terms and Utilities:

 /etc/pam.d

 pam.conf

 nsswitch.conf

 pam_unix, pam_cracklib, pam_limits, pam_listfile

06/23/14

10



What is PAM

✿ PAM stands for Pluggable Authentication Modules, and allows Linux to authenticate with many different subsystems (/etc/shadow, LDAP, Samba, RADIUS etc..), also allowing many programs (ftp, login, etc..) to decouple themselves from the authentication system.

06/23/14

11



What are PAM aware applications

✿ To identify if an authentication application is pam aware you need to look at the libraries that are linked to the application

```
#ldd /bin/login | grep libpam
```

```
libpam.so.0 => /lib/libpam.so.0 (0xb7f47000)
```

```
linpam_misc.so.0 => /lib/libpam_misc.so.0 (0xb7f47000)
```

✿ The stack of modules that each PAM aware application uses to perform each of the four authentication related activities are in the /etc/pam.d folder with a file named after the program

```
#more /etc/pam.d/sshd
```

06/23/14

12



The PAM configuration files

- ✿ The main configuration files are either `/etc/pam.conf` or a subdirectory `/etc/pam.d`
- ✿ If `/etc/pam.d` exists then `/etc/pam.conf` would not exist.
- ✿ The `pam.conf` file will contain multiple lines for all services it handles. On the other hand `pam.d` directory will hold one file per service it handles.
- ✿ These make up what are commonly known as a module stack. When the PAM aware application requests authentication, then each module in the stack is queried and the results passed back with a pass or fail to the PAM library.
- ✿ The PAM library combines these pass/fail results into a pass/fail result for the stack as a whole. This result is then returned to the application.

06/23/14

13



`/etc/pam.conf` example

✿ Below is an example of `/etc/pam.conf`

```
login auth requisite pam_authtok_get.so.1
login auth required pam_dhkeys.so.1
login auth required pam_dial_auth.so.1
login auth binding pam_unix_auth.so.1 server_policy
login auth required pam_ldap.so.1
#
sshd auth requisite pam_authtok_get.so.1
sshd auth required pam_dhkeys.so.1
sshd auth sufficient pam_unix_auth.so.1
sshd auth required pam_ldap.so.1 try_first_pass
sshd account required pam_unix_account.so.1
```

06/23/14

14



/etc/pam.d typical configuration file

In the /etc/pam.d there are files for each service, so as per the previous example, sshd and login would have a file each.

/etc/pam.d/sshd

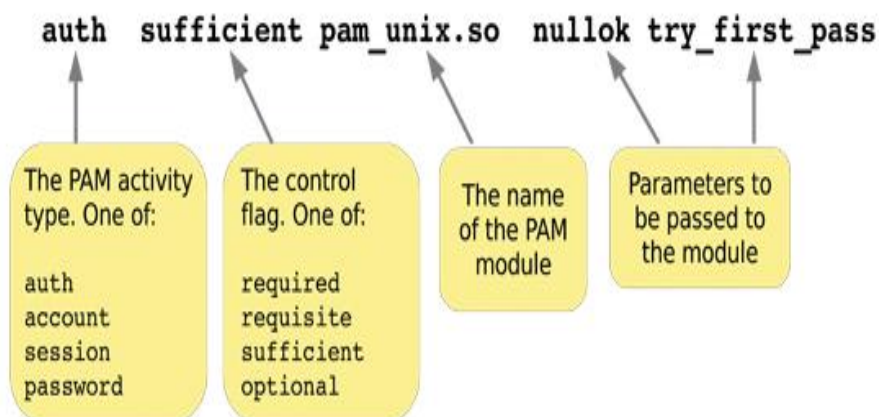
```
auth    include    system-auth
account required  pam_nologin.so
account include   system-auth
password include  system-auth
session optional  pam_keyinit.so force revoke
session include  system-auth
session required  pam_loginuid.so
```

06/23/14

15



What does it mean



06/23/14

16



The PAM activity

- ✿ **auth** - this deals with proving who you are by supplying valid credentials. Usually user name and password, but could be biometric etc.
- ✿ **account** - decides after authentication if you can log in. These modules can include time of day restrictions.
- ✿ **session** - allocates resources that a user might require during the login session.
- ✿ **password** - deals with users updating their credentials like password

06/23/14

17



The control flag

- ✿ **requisite**: If this module fails, PAM immediately returns a failure result to the application; no further modules in the stack are called.
- ✿ **required**: If this module fails, PAM returns a failure result to the application but it will continue to call the next module in the stack. (If you find you have trouble remembering the distinction between requisite and required, then join the club! So do we...)
- ✿ **sufficient**: If this module succeeds, PAM returns a 'pass' result to the application and no further modules in the stack are called. (This assumes, of course, that a required module hasn't failed higher up the stack.
- ✿ **optional**: The pass/fail result of this module is ignored, which generally means that the module is being called to perform some operation, rather than participating in the pass/fail decision for the stack. For example, the pam_keyinit module is used as an 'optional' module by sshd to create a new 'session keyring' for the new login.

06/23/14

18



The PAM module and parameters passed

✿ This field holds the name of the PAM module to be used at this layer of the module stack. The values after it are the parameters to pass to the module.

✿ Modules are usually stored in

`/lib/security`

✿ Typical modules are shown on the next page, but information can be found in the

`/usr/share/doc/pam-0.99.8.1/txts/*`

This is relative to my version of PAM



The PAM module and parameters passed

✿ This field holds the name of the PAM module to be used at this layer of the module stack. The values after it are the parameters to pass to the module.

✿ Modules are usually stored in

`/lib/security`

✿ Typical modules are shown on the next page, but information can be found in the

`/usr/share/doc/pam-0.99.8.1/txts/*`

This is relative to my version of PAM



Typical modules

Module	Activities	Description
pam_unix	auth session password	Performs traditional Unix-style authentication against hashed passwords stored in /etc/shadow. You'll find it included in the pam config files of many applications, either directly or via an include directive.
pam_rootok	auth	Succeeds if you're root and fails if you're not. It's as simple as that. It is usually used in combination with some other authentication module to establish a policy of "if you're root you can go ahead and do it; otherwise you need to authenticate."
pam_cracklib	password	Performs password strength checking, testing the password against a system dictionary and a set of rules for identifying poor choices. It's usually used on a password stack to verify password strength before handing the password on to the next module in the stack (typically pam_unix) to actually update the password.
pam_passwdqc	password	An alternative module for password strength checking; it also provides support for pass phrases and can provide randomly generated ones. Like pam_cracklib it would typically be used on a password stack to verify password strength before updating it.
pam_permit	auth account session password	This module just says "yes" to everything. It always returns success.
pam_deny	auth account session password	This module just says "no" to everything. It always returns failure. Typically it would only be used right at the bottom of a PAM stack to guarantee failure.

06/23/14

21



Module	Activities	Description
pam_warn	auth account session password	Simply logs a message (including the service name, the terminal, the user name and the remote host) to the message logging service syslogd. It might be used, for example, near the bottom of a PAM stack to log a failed authentication attempt, just prior to denying it with pam_deny.
pam_nologin	auth account	Prevents users (other than root) from logging in to the system when the file /etc/nologin exists. In some distros, this file is created by the shutdown program to prevent users logging in when the system is about to be stopped.
pam_wheel	auth account	Used by programs like su, this module allows root access only if the requesting user is a member of the group called wheel.
pam_limits	Session	Assigns user session limits. By default limits are taken from the /etc/security/limits.conf config file. Then individual *.conf files from the /etc/security/limits.d/ directory are read. The files are parsed one after another in the order of "C" locale.
pam_listfile	auth account session password	This module provides a way to deny or allow services based on an arbitrary file # deny ftp-access to users listed in the /etc/ftpusers file # auth required pam_listfile.so onerr=succeed item=user \ sense=deny file=/etc/ftpusers

06/23/14

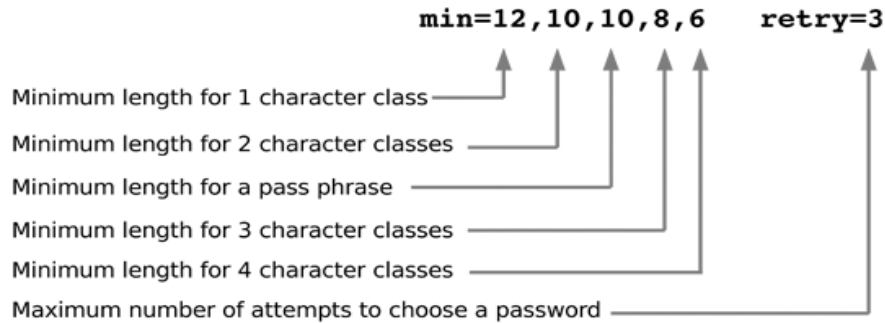
22



A typical module entry

🌀 Lets look at password length enforcement information and how it is checked

password requisite pam_passwdqc.so min=12,10,10,8,6 retry=3



06/23/14

23



210.3 LDAP client usage

Weight 2

Description

Candidates should be able to perform queries and updates to an LDAP server. Also included is importing and adding items, as well as adding and managing users.

Key Knowledge Areas

- 🌀 LDAP utilities for data management and queries
- 🌀 Change user passwords
- 🌀 Querying the LDAP directory

Terms and Utilities:

- 🌀 ldapsearch
- 🌀 ldappasswd
- 🌀 ldapadd
- 🌀 ldapdelete

06/23/14

24



What is LDAP

- ✿ LDAP stands for Lightweight Directory Access Protocol
- ✿ LDAP, is an Internet protocol that email and other programs use to look up information from a server.
- ✿ It is based on the ITU-T X.500 DAP

06/23/14

25



Adding an entry to the ldap

- ✿ Use the command-line tool `ldapadd` to add entries to the directory. `ldapadd` opens a connection to the directory and authenticates the user.
#`ldapadd -h myhost -p 389 -D "cn=orcladmin" -w welcome -f jhay.ldif`
- ✿ Using this command, user `orcladmin` authenticates to the directory `myhost`, located at port `389`. The command then opens the file `jhay.ldif` and adds its contents to the directory. The file might, for example, add the entry `uid=jhay,cn=Human Resources,cn=acme,dc=com` and its object classes and attributes.

06/23/14

26



ldapdelete

Use the command-line tool **ldapdelete** to remove leaf entries from a directory. **ldapdelete** opens a connection to a directory server and authenticates the user. Then it deletes specified entries.

```
#ldapdelete -h myhost -p 389 -D "cn=orcladmin" -w welcome \  
"uid=hricard,ou=sales,ou=people,dc=acme,dc=com"
```

This command authenticates user **orcladmin** to the directory **myhost**, using the password **welcome**. Then it deletes the entry **uid=hricard,ou=sales,ou=people,dc=acme,dc=com**.



ldapsearch


Use the command-line tool **ldapsearch** to search for specific entries in a directory. **ldapsearch** opens a connection to a directory, authenticates the user performing the operation, searches for the specified entry, and prints the result in a format that the user specifies.

```
#ldapsearch -h myhost -p 389 -s base -b "ou=people,dc=acme,dc=com" \  
"objectclass=*"
```


This command searches the directory server **myhost**, located at port **389**. The scope of the search (**-s**) is **base**, and the part of the directory searched is the base DN (**-b**) designated. The search filter **"objectclass=*"** means that values for all of the entry's object classes are returned. No attributes are returned because they have not been requested. The example assumes anonymous authentication because authentication options are not specified.



ldappasswd

 The following command lets Luke Crowe change his own user password, connecting over simple authentication:

```
#ldappasswd -h host -D uid=lcrowe,ou=people,dc=example,dc=com -j  
old.pwd -T new.pwd -t old.pwd  
uid=lcrowe,ou=people,dc=example,dc=com
```

 The following command lets Luke Crowe change Barbara Jensen's password, connecting over simple authentication:

```
#ldappasswd -h host -D uid=lcrowe,ou=people,dc=example,dc=com -w  
- -A -S uid=bjensen,ou=people,dc=example,dc=com
```

06/23/14

29











210.4 Configuring an OpenLDAP server

Weight: 4

Description: Candidates should be able to configure a basic OpenLDAP server including knowledge of LDIF format and essential access controls. An understanding of the role of SSSD in authentication and identity management is included.

Key Knowledge Areas

-  OpenLDAP
-  Access Control
-  Distinguished Names
-  Chantype Operations
-  Schemas and Whitepages
-  Directories
-  Object IDs, Attributes and Classes
-  Awareness of System Security Services Daemon (SSSD)

06/23/14

30



210.4 Configuring an OpenLDAP server

Terms and Utilities

- 🌀 slapd
- 🌀 slapd.conf
- 🌀 LDIF
- 🌀 slapadd
- 🌀 slapcat
- 🌀 slapindex
- 🌀 /var/lib/ldap/*
- 🌀 loglevel

06/23/14

31



Configuring A Basic LDAP Server

- 🌀 Installation of OpenLDAP on Red Hat/CentOS:

```
#yum install openldap openldap-clients openldap-servers openldap-servers-overlays
```

- 🌀 You then need to edit the main configuration file slapd.conf
- 🌀 The accompanying lab gives you the opportunity to do so.

06/23/14

32



211.1 Using e-mail servers

Weight 4

Description

Candidates should be able to manage an e-mail server, including the configuration of e-mail aliases, e-mail quotas and virtual e-mail domains. This objective includes configuring internal e-mail relays and monitoring e-mail servers.

Key Knowledge Areas

- Configuration files for postfix
- Basic knowledge of the SMTP protocol
- Awareness of sendmail and exim

Terms and Utilities:

- Configuration files and commands for postfix `/etc/postfix/*`
- `/var/spool/postfix`
- sendmail emulation layer commands
- `/etc/aliases`
- mail-related logs in `/var/log/`

06/23/14

1



SMTP (Simple Mail Transfer Protocol)

- The original RFC for SMTP is 821. It has now been obsolete by RFC2821 the introduction of which says

“The objective of the Simple Mail Transfer Protocol (SMTP) is to transfer mail reliably and efficiently.....SMTP's strength comes primarily from its simplicity. Experience with many protocols has shown that protocols with few options tend towards ubiquity, whereas protocols with many options tend towards obscurity.”

- Typical SMTP commands:

- EHLO (to start conversation with server)
- MAIL FROM
- RCPT TO

- The server responses are equally simple

06/23/14

2



Postfix

- ✿ Postfix is an easy to configure MTA (Mail Transfer Agent)
- ✿ The majority of Postfix configuration is handles by 2 files:
 - main.cf and master.cf
- ✿ master.cf controls how clients connect to the server while main.cf is where we customize the server
- ✿ Inside main.cf each option is structured as follows:
 - *action = value e.g. command_directory = /usr/sbin*

06/23/14

3



Postfix configuration files

This file contains typical entries

```
mydomain = virtual.domain (virtual interface)
myorigin = $mydomain (probably desirable: "user@$mydomain")
mydestination = $myhostname localhost.$mydomain localhost (domains
to receive from)
mynetworks = 127.0.0.0/8 168.100.189.2/32 (authorize client to relay
from)
relay_domains = $mydomain (domains to relay to)
relayhost = $mydomain (deliver via local mailhub)
alias_maps = hash:/etc/aliases
```

06/23/14

4



Reload postfix


 Any change to the `/etc/postfix/main.cf` requires a reload of postfix
`#postfix reload`

06/23/14

5



The `/etc/aliases`

 A typical `/etc/aliases`

`# Our Own Aliases`

`www: root`

`admin: root`

`sysadmin: root`

`webmaster: root`

`support: helpdesk`

`# Person who should get root's mail`

`root: john <-- John will receive all system/security
email alerts meant for root.`

`# People who have left our organisation - Mail redirection...`

`sarah: sarah@otherdomain.org`

`tom: tom@differentorganisation.org`

06/23/14

6



Mailing list through aliases

✿ To create a mailing list create a file `/etc/mail/mailling-list`

```
alice@wonderland.com  
peter@nevernever.org  
harry@potterworld.net
```

✿ In the `/etc/aliases` add a line as follows

```
mailling-list: :include:/etc/mail/mailling-list
```

✿ Any changes to the `/etc/aliases` you must run the following command

```
#newaliases
```

06/23/14

7



Spool and logging directories

✿ Mail that is destined for outbound is generally stored in

```
/var/spool/postfix
```

- To view what messages are waiting for delivery use the following command

```
#mailq
```

Mail waiting for delivery to local users

```
/var/spool/mail/username
```

✿ Postfix has a Sendmail compatibility interface to allow one to use Sendmail commands on Postfix (“old habits die hard”)

– e.g. `sendmail -bp`, `mailq`, `sendmail -l`, `newaliases`

✿ The logging file is stored in the following

```
/var/log/maillog
```

06/23/14

8



211.2 Managing Local E-Mail Delivery

Weight 2

Description

Candidates should be able to implement client e-mail management software to filter, sort and monitor incoming user e-mail.

Key Knowledge Areas

- ✿ procmail configuration files, tools and utilities
- ✿ Usage of procmail on both server and client side

Terms and utilities:

- ✿ `-.procmail`
- ✿ `/etc/procmailrc`
- ✿ `procmail`
- ✿ `mbx` and `Maildir` formats

06/23/14

9



What is procmail

- ✿ Procmail is an MDA or Mail Delivery Agent, and is the default MDA for Linux.
- ✿ When mail is delivered to the system or user, the procmail system will split the mail header from the message body.
- ✿ Globally the recipe file is in `/etc/procmailrc` and on a per user basis it is in the users home directory and is `$HOME/.procmailrc`
- ✿ The `procmailrc` which contains recipes on how to sort the mail and where to store it.

06/23/14

10



Procmail recipes

✿ The beginning of a recipe is `:0` It has the following format:

`:0 [flags] [: [locallockfile]]`

<zero or more conditions (one per line)>

<exactly one action line>

✿ Conditions start with a leading `*`, everything after that character is passed on to the internal `egrep` literally, except for leading and trailing whitespace.

✿ Conditions are combined together with an `and`

✿ If there are no conditions the result will be true by default.

✿ See <http://www.partmaps.org/era/procmail/quickref.html>

06/23/14

11



What does it mean

✿ The recipe consists of the following parts:

Notation	Meaning
----------	---------

<code>:0</code>	Begin a recipe
-----------------	----------------

<code>:</code>	Use a lock file
----------------	-----------------

<code>*</code>	Begin a condition
----------------	-------------------

`^Subject:.*test` Match lines in the message.


<code>testing</code>	Store message in the given mailbox in case matching.
----------------------	------------------------------------------------------

06/23/14

12




Sample recipes

 To store mailing list messages in a separate folder, use a recipe like this:

```
:0:  
^TO_procmail@informatik\.rwth-aachen\.de
```


procmail

 To make procmail case sensitive use the D flag

```
:0D:
```

```
* ^Subject:.*ADV
```

Potential.SPAM

 To make a copy of all mail into a backup

```
:0c:
```

```
/var/mail/backup
```

 To dump all spam to the bit bucket

```
:0
```

```
* ^From:.*@cyberspam\.com
```

```
/dev/null
```

06/23/14

13



A simple .procmailrc file

 A simple \$HOME/.procmailrc

```
# Next two are needed if you invoke programs, such as
```

```
# formail, sendmail, or egrep, from your procmailrc
```

```
# SHELL=/bin/sh
```

```
# PATH=/usr/bin:/bin
```

```
# Put # before LOGFILE if you want no logging (not recommended)
```

```
LOGFILE=.procmail.log
```

```
# To insert a blank line between each message's log entry,
```

```
# uncomment next two lines (this is helpful for debugging)
```

```
# LOG=" # "
```

```
# NOTE: Upon reading the next line, Procmail does a chdir to $MAILDIR
```

```
MAILDIR=$HOME/Mail
```

```
# Make sure this directory exists!
```

```
:0:
```

```
* ^Subject:.*test
```

```
testing
```

06/23/14

14



211.3 Managing Remote E-Mail Delivery

Weight 2

Description

Candidates should be able to install and configure POP and IMAP daemons.

Key Knowledge Areas

- ✿ Courier IMAP and Courier POP configuration

- ✿ Dovecot configuration

- ✿ **The following is a partial list of the used files, terms and utilities:**

- ✿ /etc/courier/*

- ✿ dovecot.conf

06/23/14

15



dovecot

- ✿ Dovecot is an IPMAP4 server that also contains a POP3 server

- ✿ Supports mail in both maildir and mbox formats

- ✿ Configuration file is stored in /etc/dovecot/dovecot.conf

- ✿ Service can be enabled using `chkconfig dovecot on` and then started using `service dovecot start`

06/23/14

16



dovecot

- ✿ Common Dovecot configuration settings:
- ✿ Protocols - i.e. imap, pop3
- ✿ Listen - the IP address (and optionally the port) to listen on
- ✿ Login_process_per_connection - does each process launch it's own process, default=yes
- ✿ Mail_location - location of mbox files or maildir directories

06/23/14

17



The Courier mail server

- ✿ Courier can provide SMTP, IMAP and POP3 services.
- ✿ The main configuration files are stored in /etc/courier directory and are described over the next few slides

06/23/14

18



Courier SMTP settings

✿ The following two files define what domains the SMTP server will accept mail for

- `/etc/courier/locals`
- `/etc/courier/esmtpacceptmailfor.dir/domains`

✿ These two files contain a line by line entry for each domain

`localhost`

`lukecrowe.co.uk`

`mydomain.com`

✿ Next run the `makeacceptmailfor` which takes the above files and creates the following file

- `/etc/courier/esmtpacceptmailfor.dat`

06/23/14

19



Courier SMTP aliasing

✿ Aliasing is done through the following file

- `/etc/courier/aliases/system`

✿ This file is very similar to the Sendmail alias file

`postmaster: fred`

✿ Once set, run the script `makealiases` to build the following

- `/etc/courier/aliases.dat`

✿ You can test your configuration of aliases by the following command


`#echo "To: fred" | sendmail`

06/23/14


20




Accepting mail from other domains (Relaying)

 In order for the smtp server to route to other domains, you need to edit the following file

- `/etc/courier/smtpaccess/default`

 This file contains an entry by default of 127.0.0.1, which only allows localhost to send mail

 Next run the `makesmtpaccess` script to build the database

- `/etc/courier/smtpaccess.dat`



212.1 Configuring a router

Weight 3

Description

Candidates should be able to configure a system to perform network address translation (NAT, IP masquerading) and state its significance in protecting a network. This objective includes configuring port redirection, managing filter rules and averting attacks.

Key Knowledge Areas

- iptables configuration files, tools and utilities
- Tools, commands and utilities to manage routing tables.
- Private address ranges
- Port redirection and IP forwarding
- List and write filtering rules that accept or block datagrams based on source or destination protocol, port and address
- Save and reload filtering configurations
- Awareness of ip6tables and filtering

Terms and Utilities:

- /proc/sys/net/ipv4
- /etc/services
- iptables

06/23/14

1



Standard configuration files

- The `/etc/services` file hold a list of TCP and UDP ports and their respective services. Applications look it up to translate between port name and number
- For routing to be enabled in the kernel the following file in `/proc` must be set to 1
 - `/proc/sys/net/ipv4/ip_forward`
- This can also be set using `/etc/sysctl.conf`, put an entry as follows
`net.ipv4.ip_forward=1`

06/23/14

2



Private address ranges

✿ The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private internets:

- 10.0.0.0 - 10.255.255.255 (10/8 prefix)
- 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

06/23/14

3



Iptables

✿ Netfilter is a Linux kernel based framework that provides options for packet filtering, network address translation(NAT) and port translation(PAT)

✿ Iptables (IPv4) is the userland program for setting up, maintaining, and inspecting the tables of IPv4 packet filter rules in the Linux kernel

✿ The IPv6 equivalent is ip6tables

✿ To see if netfilter is enabled in your kernel check in your config file in the /boot directory

```
# cat /boot/config-`uname -r` | grep -i "CONFIG_IP_NF"
```

06/23/14

4



The Iptables main files

✿ Rules created using iptables are ephemeral. To survive a reboot they need to be saved. **iptables-save** will generate the current rules in the kernel in a format that is easy for iptables to parse. The start up scripts can then read the file during boot

– /sbin/iptables-save > /etc/sysconfig/iptables

✿ On Red Hat **/etc/sysconfig/iptables** contains the said rule sets.

✿ The iptables binary is **/sbin/iptables**

06/23/14

5



Viewing and flushing the current rules

✿ To view the current rules use the iptables tool

#iptables -L

✿ To view a specific rule type

#iptables -L -t nat

#iptables -L -t filter

#iptables -L -t mangle

✿ To flush the current filter rules

#iptables -F

#iptables -F -t nat

06/23/14

6



Rules

- ✿ The lowest level objects are the rules that are performing the packet filtering or manipulation. A **rule** is made of several parts.
 1. The **Table** to which this rule should be added. If no table is defined the rule will be added to the **filter** table.
 2. The **Chain** to which this rule should be added i.e. **INPUT** or **FORWARD**
 3. The filtering or manipulation instructions.
 4. The **Target** of the rule. This target decides what should be done with the packet if it matches the rule. The most important targets are **DROP**, which drops the packet without any further action, **ACCEPT**, which will let the packet pass the firewall and the data is sent to the receiver and **LOG** that simply writes some information (src. IP, ports etc.) about the packet that matches to the Syslog.

06/23/14

7



The tables

- ✿ There are three tables available filter, nat and mangle.
 - **filter** - This table is used for packet filtering as with typical firewalls
 - **nat** - This table is made for all kinds of Network Address Translation which is a technology used to change the source and destination attributes of the packet. IP Masquerading is the most common form of nat
 - **mangle** - This table is designed to hold chains and rules that change other attributes of the packets or sending them into the user space to be processed by any other application.

06/23/14

8



The Chains and Targets

- There are three chains that hold rules
 - **INPUT** - Any traffic coming into the server (The most common)
 - **OUTPUT** - Any traffic coming out of your server
 - **FORWARD** - Any traffic going across your server (Routing)
- There are a further two rules for Natting
 - **PREROUTING** - NATs packets when the destination address of the packet needs to be changed.
 - **POSTROUTING** - NATs packets when the source address of the packet needs to be changed.
- When a rule is matched within a chain then it is assigned to a Target
 - **ACCEPT** - Packets are accepted
 - **DROP** - The packet is dropped without notification to the sender
 - **REJECT** - The sender is notified with an ICMP code that his packet has been blocked

06/23/14

9



The State

- As iptables can be used as a stateful filter, then the states can be defined
 - **NEW** - The 1st server sends the 2nd server a SYN packet that it wants to create a new connection.
 - **RELATED** - The 2nd server receives the SYN packet and sends the 1st server a SYN-ACK packet which tells that everything is alright.
 - **ESTABLISHED** - The 1st server receives the SYN-ACK packet and sends the 2nd server an ACK packet which is the final acknowledgment, the connection finishes establishing and the traffic start between the two servers.

06/23/14

10



A Simple rule

✿ Lets have a look at a simple rule

```
# iptables -A INPUT -s 192.168.1.10 -d 10.1.15.1 -p tcp --dport 22 -j ACCEPT
```

-A Tells iptables to append this rule to the INPUT Chain

-s Source Address. This rule only pertains to traffic coming FROM this IP. Substitute with the IP address you are SSHing from.

-d Destination Address. This rule only pertains to traffic going TO this IP. Substitute with the IP of this server.

-p Protocol. Specifying traffic which is TCP.

--dport Destination Port. Specifying traffic which is for TCP Port 22 (SSH)

-j Jump. If everything in this rule matches then 'jump' to ACCEPT

06/23/14

11



Whitelist or blacklist

✿ The firewall can be set up to either

A. Allow all traffic and only drop packets you don't want
(**Blacklist**)

B. Block all traffic and explicitly allow certain traffic through
(**Whitelist**)

✿ Which you choose depends on speed and length of the chains rule sets.

06/23/14

12



Setting up a whitelist

🌀 First assign a default policy on the chain to block all traffic

```
#iptables -P INPUT DROP
```

🌀 Now define the traffic you wish to allow

Unlimited access to loop back

```
#iptables -A INPUT -i lo -j ACCEPT
```

```
#iptables -A OUTPUT -o lo -j ACCEPT
```

Now allow the white traffic

```
#iptables -A INPUT -s 172.16.0.0/24 -j ACCEPT
```

```
#iptables -A INPUT -p tcp --dport 21:25:80:443 -j ACCEPT
```

```
#iptables -A INPUT -p udp --dport 53 -d 172.16.0.5 -j ACCEPT
```

06/23/14

13



Setting up a blacklist

🌀 First assign a default policy on the chain to accept all traffic

```
#iptables -P INPUT ACCEPT
```

🌀 Now define the traffic allowed and disallowed

Lets make sure we can access the box regardless if we are remoting in

```
# iptables -A INPUT -s 172.16.0.100 -p tcp --dport 22 -j ACCEPT
```

This will stop us locking ourselves out by setting a rule that blocks SSH

Now lets use the state tracking function, these must be defined to enable communication

```
# iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
# iptables -A FORWARD -i eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
# iptables -A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
```

Rejects first

BAD GUYS (Block Source IP Address):

```
# iptables -A INPUT -s 172.34.5.8 -j DROP
```

06/23/14

14



Iptables blacklist continued




```
NO SPAMMERS (notice the use of FQDN):
# iptables -A INPUT -s mail.spammer.org -d 172.16.0.2 -p tcp --dport 25 -j REJECT
Now open up the rest
MYSQL (Allow Remote Access To Particular IP):
# iptables -A INPUT -s 172.50.3.45 -d 172.16.0.2 -p tcp --dport 3306 -j ACCEPT
SSH:
# iptables -A INPUT -d 172.16.0.2 -p tcp --dport 22 -j ACCEPT
Sendmail/Postfix:
# iptables -A INPUT -d 172.16.0.2 -p tcp --dport 25 -j ACCEPT
FTP: (Notice how you can specify a range of ports 20-21)
# iptables -A INPUT -d 172.16.0.2 -p tcp --dport 20:21 -j ACCEPT
Allow Ping
# iptables -A INPUT -d 172.16.0.0/16 -p icmp -j ACCEPT
Now reject everything else
# iptables -A INPUT -j REJECT
# iptables -A FORWARD -j REJECT
```

06/23/14

15



Some examples

```
 Redirect any traffic going to IP 10.1.1.1 (port 26) and send it to IP
10.1.1.1 (port 25):
#iptables -A PREROUTING -p tcp -m tcp -d 10.1.1.1 --dport 26 -j DNAT --
to 10.1.1.1:25
 Log Ping packets
#iptables -A INPUT --protocol icmp --icmp-type echo-request -j LOG
 Natting anything coming from 10.10.10.0 network
#iptables -t nat -A POSTROUTING -s 10.10.10.0/24 -d ! 10.10.10.0/24 -j
MASQUERADE
#sysctl -w net.ipv4.ip_forward=1
```

06/23/14

16



Inserting Rules and saving rules

To insert a new rule into the input chain into line 3 of the INPUT chain

```
#iptables -I INPUT 3 -p tcp --dport 80 -j ACCEPT
```

To save the rules

```
#iptables-save > /etc/sysconfig/iptables
```

or

```
#!/etc/init.d/iptables save
```

06/23/14

17



An example

(1) Traffic initiated by the Firewall that is destined for the Internet

We are running a DNS on the Firewall that needs to be able to consult other DNSes on the Internet (which use the UDP protocol and listen to PORT 53). We want to be able to use **ssh** (which uses the TCP protocol and port 22) to connect to other systems on the Internet. We are participating in a distributed.net project RC564 cracking and are running a proxy server on the Firewall (which uses the TCP protocol and PORT 2064 to communicate with the keyserver). We want to be able to **ping** hosts on the Internet (**ping** uses the ICMP protocol and message type "ping"). The Firewall communicates with the Internet through interface eth1.

```
iptables -t filter -A OUTPUT -o eth1 -p udp --destination-port dns -m state --state NEW -j ACCEPT
iptables -t filter -A OUTPUT -o eth1 -p tcp --destination-port ssh -m state --state NEW -j ACCEPT
iptables -t filter -A OUTPUT -o eth1 -p tcp --destination-port 2064 -m state --state NEW -j ACCEPT
iptables -t filter -A OUTPUT -o eth1 -p icmp --icmp-type ping -m state --state NEW -j ACCEPT
```

(2) Traffic initiated by the Internet that is destined for the Firewall

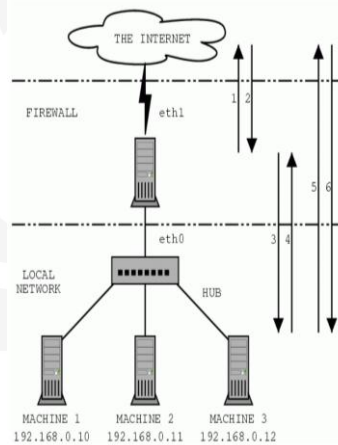
We want to be able to use **ssh**, which uses the TCP protocol and port 22, to connect to our Firewall from other systems on the Internet.

```
iptables -t filter -A INPUT -i eth1 -p tcp --destination-port ssh -m state --state NEW -j ACCEPT
```

(3) Traffic initiated by the Firewall that is destined for the internal network

We want to be able to use **ssh**, which uses the TCP protocol and port 22, to connect to one of our internal machines from our Firewall.

```
iptables -t filter -A OUTPUT -o eth0 -p tcp --destination-port ssh -m state --state NEW -j ACCEPT
```



06/23/14

18



An example

(4) Traffic initiated by the internal network that is destined for the firewall

The machines on the internal network, using the dns of the firewall, must be able to connect to the firewall using ssh, are processing RC564 keys, must be able to talk to the proxy on the firewall using port 2064 and must be able to ping the Firewall for system administrative purposes.

```
iptables -t filter -A INPUT -i eth0 -p udp --destination-port dns -m state --state NEW -j ACCEPT
iptables -t filter -A INPUT -i eth0 -p tcp --destination-port ssh -m state --state NEW -j ACCEPT
iptables -t filter -A INPUT -i eth0 -p tcp --destination-port 2064 -m state --state NEW -j ACCEPT
iptables -t filter -A INPUT -i eth0 -p icmp --icmp-type ping -m state --state NEW -j ACCEPT
```

(5) Traffic initiated by the Internal Network that is destined for the Internet

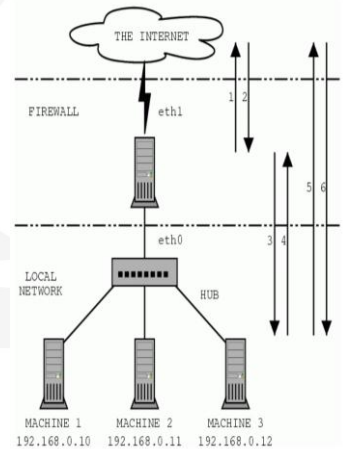
Every connection from a machine on the internal network to a machine on the Internet is allowed.

```
iptables -t filter -A FORWARD -i eth0 -o eth1 -m state --state NEW -j ACCEPT
```

(6) Traffic initiated by the Internet that is destined for the Internal Network

The local network uses private IP addresses that can't be used on the Internet. Our local machines aren't visible from the Internet. Therefore need to port forward from external network. To make a machines available on the Internet we need to connect to a certain port on the firewall and use NAT to redirect them to a port on one of the machines on the Internal Network. Suppose Machine 2 has a web-server running which listens to port 2345 and people from the outside must be able to connect to that program. The solution here is to tell Machine 4 that all data from the outside that is aimed at port 80 should be routed to port 2345 on Machine 2.

```
iptables -t nat -A PREROUTING -i eth1 -p tcp --destination-port 80 -j DNAT --to-destination 192.168.0.11:2345
iptables -t filter -A FORWARD -i eth1 -p tcp --destination-port 2345 -m state --state NEW -j ACCEPT
```



06/23/14

19



An example

(7) Traffic as a result of initiated traffic

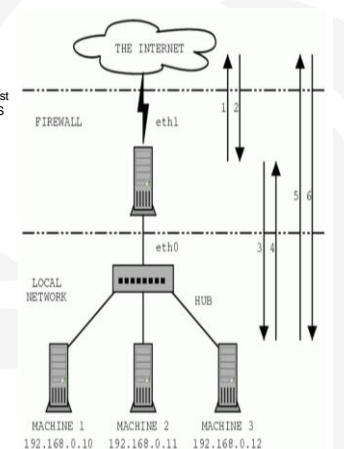
We have only specified that connection initiation traffic is allowed, but that is not enough. We also must allow ESTABLISHED and RELATED traffic.

Let's tell the firewall that all ESTABLISHED and RELATED traffic, regardless of type, interface etc. is allowed. We must also allow the initiation of traffic on the firewalls lo interface because otherwise some services, such as a caching DNS server, will not work.

```
iptables -t filter -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A INPUT -m state --state NEW -i lo -j ACCEPT
```

Finally we need to masquerade all packets from our internal network destined for the external networking using Source Network address translation

```
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o eth1 -j MASQUERADE
```



06/23/14

20



212.2 Securing FTP servers

Weight 2

Description

Candidates should be able to configure an FTP server for anonymous downloads and uploads. This objective includes precautions to be taken if anonymous uploads are permitted and configuring user access.

Key Knowledge Areas

- Configuration files, tools and utilities for Pure-FTPd and vsftpd
- Awareness of ProFTPd?
- Understanding of passive vs. active FTP connections

Terms and Utilities:

- vsftpd.conf
- important Pure-FTPd command line options

06/23/14

21



vsftpd

- The vsftpd is an ftp daemon, which stands for very secure ftp daemon. Can be launched by xinetd, inetd or stand alone.

- The main configuration files are

- /etc/vsftpd/vsftpd.conf
- /etc/vsftpd/ftpusers
- /etc/vsftpd/user_list

06/23/14

22



The vsftpd.conf file

✿ This file governs the overall running of the ftp server. It sets options in the format **option=value**.

✿ Some options are shown below

- `anonymous_enable=yes` allows anonymous access
- `anon_upload_enable=yes` allow anonymous to upload
- `write_enable=yes` allows users to write to ftp server
- `chroot_local_user=yes` users who log in will be placed in own dir
- `listen=yes` sets vsftpd in standalone mode
- `local_enable=yes` turns on local users access
- `tcp_wrappers=yes` enable TCPd support

06/23/14

23



The /etc/vsftpd/user_list

✿ If `userlist_enable=yes` in the vsftpd.conf file. The vsftpd server will check the `user_list` file before checking their password.

✿ If the username is in the `user_list` file, then they will be denied access

✿ If `userlist_deny=yes` then users in the `user_list` will be granted access.

06/23/14

24



The /etc/vsftpd/ftpusers file

- ✿ By default this file contains a list of users that are not allowed to access the ftp server

06/23/14

25



Log files for vsftpd


- ✿ The logfiles for transfers are in
 - /var/log/xferlog
- ✿ The daemon logfile is stored in
 - /var/log/vsftpd.log


06/23/14

26



Pure ftp daemon


 Pure ftp daemon is a small simple to use ftp server. It can implement many modern extensions like ldap authentication and SSL/TLS using OpenSSL. It can also support authentication through PAM

 The main configuration file is

- /etc/pure-ftpd/pure-ftpd.conf



ProFTP Daemon

 The ProFTP server is a powerful ftp service. Its configuration file is very similar to an apache httpd.conf using directives.

- /etc/proftpd.conf



212.3 Secure shell (SSH)

Weight 4

Description

Candidates should be able to configure and secure an SSH daemon. This objective includes managing keys and configuring SSH for users. Candidates should also be able to forward an application protocol over SSH and manage the SSH login.

Key Knowledge Areas

- OpenSSH configuration files, tools and utilities
- Login restrictions for the superuser and the normal users
- Managing and using server and client keys to login with and without password
- Usage of XWindow and other application protocols through SSH tunnels
- Configuration of ssh-agent
- Usage of multiple connections from multiple hosts to guard against loss of connection to remote host following configuration changes

06/23/14

29



212.3 Secure shell (SSH)

Terms and Utilities:

- ssh
- sshd
- /etc/ssh/sshd_config
- Private and public key files
- ~/.ssh/authorized_keys
- PermitRootLogin, PubKeyAuthentication, AllowUsers, PasswordAuthentication, Protocol

06/23/14

30



The secure shell - ssh

SSH is a tool for secure remote login over insecure networks. It provides an encrypted terminal session with strong authentication of both the server and client, using public-key cryptography

Some examples of using the basic ssh command

```
#ssh -v username@172.16.0.100
```

```
#ssh 172.16.0.100
```

```
#ssh -l luke 172.16.0.100
```

```
#ssh 172.16.0.100 uptime
```

06/23/14

31



The standard configuration files

The global configuration files for ssh are stored in /etc/ssh

- /etc/ssh/ssh_config
- /etc/ssh/sshd_config

V1 Protocol Keys

- /etc/ssh/ssh_host_key (Private Key)
- /etc/ssh/ssh_host_key.pub (Public Key)

V2 Protocol Keys

- /etc/ssh/ssh_host_rsa_key (Private RSA Key)
- /etc/ssh/ssh_host_rsa_key.pub (Public RSA Key)
- /etc/ssh/ssh_host_dsa_key (Private DSA Key)
- /etc/ssh/ssh_host_dsa_key.pub (Public DSA Key)

06/23/14

32



Generating the keys

✿ To generate the keys use the `ssh-keygen` command. The following command generates a V2 protocol public and private keys

```
#ssh-keygen -t dsa
```

```
#ssh-keygen -t rsa
```

✿ The location of these keys will depend on who you are creating them for.

✿ For the system, they will be stored in `/etc/ssh` directory.

✿ For users who wish to use their own keys, then these will be stored in their home directory under

```
~/.ssh
```

06/23/14

33



Personal ssh keys

✿ You can use `ssh` with your own keys to increase encryption of the `ssh` system.

✿ Issue the following command to create the keys for your personal use

```
#ssh-keygen -t dsa
```

✿ This will create two files in your `~/.ssh`

```
~/.ssh/id_dsa    (Private key)
```

```
~/.ssh/id_dsa.pub    (Public Key)
```

06/23/14

34



Personal ssh keys

✿ Now you need to copy the public key to the remote server and place it in your home directory on there.

```
#scp ~/.ssh/id_dsa.pub luke@172.16.0.100:~/.ssh/id_dsa.pub
```

✿ Now ssh to that machine

```
#ssh luke@172.16.0.100
```

✿ Now create the authorized_keys with the contents of the id_dsa.pub file

```
#mkdir .ssh
```

```
#cd .ssh
```

```
#touch authorized_keys
```

```
#chmod 600 authorized_keys
```

```
#cat ../id_dsa.pub >> authorized_keys
```

06/23/14

35



Personal ssh keys

✿ Everything is now in place to use these keys.

✿ Log back into the system using the new keys

```
#ssh -2 -v luke@172.16.0.100
```

06/23/14

36



The ssh-agent

✿ The ssh-agent is used to store passphrases that are used to protect your personal keys.

✿ The ssh-agent is applied to the shell

```
#ssh-agent bash    or #ssh-agent $SHELL
```

✿ Next you need to add the passphrases

```
#ssh-add
```

✿ After this, the ssh-add program will ask you for your passphrase. After you entered your password the key is loaded in the key manager ssh-agent

✿ To list the currently loaded keys

```
#ssh-add -l
```

06/23/14

37



SSH and X

✿ SSH can tunnel all X traffic through an ssh tunnel. To do this you can run

```
#ssh -Y luke@172.16.0.100
```

✿ Once logged in, you can run any X application and it will be displayed to your X Server on the client machine.

```
#firefox &
```

06/23/14

38



212.4 Security Tasks

Weight 3

Description

Candidates should be able to receive security alerts from various sources, install, configure and run intrusion detection systems and apply security patches and bugfixes.

Key Knowledge Areas

- Tools and utilities to scan and test ports on a server
- Locations and organizations that report security alerts as Bugtraq, CERT, CIAC or other sources
- Tools and utilities to implement an intrusion detection system (IDS)
- Awareness of OpenVAS and Snort
- Terms and Utilities
 - telnet
 - nmap
 - fail2ban
 - nc
 - iptables

06/23/14

39



212.4 Security tasks

Weight 3

Description

Candidates should be able to receive security alerts from various sources, install, configure and run intrusion detection systems and apply security patches and bugfixes.

Key Knowledge Areas

- Tools and utilities to scan and test ports on a server
- Locations and organisations that report security alerts as Bugtraq, CERT, CIAC or other sources
- Tools and utilities to implement an intrusion detection system (IDS)
- Awareness of OpenVAS?

Terms and Utilities:

- telnet
- nmap
- snort
- fail2ban
- nc
- iptables

06/23/14

40



Nmap

🌀 Nmap is a port scanner.

🌀 Running nmap is easy

🌀 To perform a three way handshake against port 80

```
#nmap -sT 172.16.0.100 -p 80
```

🌀 Half open scan or syn scan against a range

```
#nmap -sS 172.16.0.1-254 -p 25
```



Netcat

🌀 Netcat is known as the Swiss army knife of the internet. Netcat is a featured networking utility which reads and writes data across network connections, using the TCP/IP protocol. It is designed to be a reliable "back-end" tool that can be used directly or easily driven by other programs and scripts. At the same time, it is a feature-rich network debugging and exploration tool, since it can create almost any kind of connection you would need and has several interesting built-in capabilities.



Security Sites

- ✿ The web has many web sites that have security and vulnerability information
- ✿ **Bugtraq** is an electronic mailing list dedicated to issues about computer security. On-topic issues are new discussions about vulnerabilities, vendor security-related announcements, methods of exploitation, and how to fix them. It is a high-volume mailing list, and almost all new vulnerabilities are discussed there. Bugtraq is now hosted on SecurityFocus.com

06/23/14

43



Security Sites

- ✿ **CERT** or Computer Emergency Response Team is based out of Carnegie Mellon University. It hosts the CERT Co-ordination Centre, which holds information vulnerabilities in software. See www.cert.org
- ✿ **Computer Incident Advisory Capability (CIAC)** was the original computer security incident response team at the Department of Energy. It is now called DOE-CIRC and located at <http://www.doecirc.energy.gov>

06/23/14

44



Fail2ban

✦ **Fail2ban** scans log files like `/var/log/pwdfail` or `/var/log/apache/error_log` and bans IP that makes too many password failures. It updates firewall rules to reject the IP address.

✦ See <http://www.fail2ban.org>



OpenVAS

✦ **OpenVAS** stands for Open Vulnerability Assessment System and is a network security scanner with associated tools like a graphical user front-end. The core component is a server with a set of network vulnerability tests (NVTs) to detect security problems in remote systems and applications.



Snort

- Snort is an open source network intrusion prevention and detection system (IDS/IPS)
- It uses signature, protocol and anomaly-based inspection.

06/23/14

47



212.5 OpenVPN

Weight: 2

Description: Candidates should be able to configure a VPN (Virtual Private Network) and create secure point-to-point or site-to-site connections.

Key Knowledge Areas

OpenVPN

Terms and Utilities

• /etc/openvpn/*

• openvpn

06/23/14

48



Configuring OpenVPN Server for Point-to-Point VPN

openvpn - secure IP tunnel daemon

🌀 Install OpenVPN from the distribution repository

🌀 When you start the daemon (e.g. `service openvpn start`), it will search `/etc/openvpn` directory for files ending with `.conf` and load any VPNs defined in there.

Example configuration file `myvpn.conf`

```
#Network configuration
Dev tun
Port 1194
Proto udp
Server 10.10.0.0 255.255.255.0
Keepalive 10 20
.....cont.
```

06/23/14

49



Configuring OpenVPN Server for Point-to-Point VPN

#logging configuration

log-append /var/log/openvpn.log

status /var/log/openvpn-status.log

verb 4

mute 20

security configuration

user nobody

group nobody

persist-key

persist-tun

#compression

comp-lzo

06/23/14

50



Configuring OpenVPN Authentication

🌀 OpenVPN allows a variety of authentication mechanisms including preshared keys, two-factor authentication and TLS/SSL certificates

🌀 To generate a static key

- `Sudo openvpn --genkey -secret /etc/openvpn/secret.key`

🌀 You would then securely copy this key to the VPN client

