###Features of Red Hat Enterprise v6###

Features:

1. Current Release: 6
2. Based on 2.6.x kernel
3. Supports graphical and text-based installations
 a. Graphical installer is more feature-rich than text-based installer
4. Downloadable via: HTTP from redhat.com
 a. Available as trial and/or subscription
5. Installable from:
 a. Local media: CDs, DVDs (1-DVD ISO image), USB
 b. Network: HTTP, NFS, FTP
 c. PXE - Network Installation
6. Virtualization - KVM
7. EXT4 - Default FS for new installations
 a. Also works with: /boot due to GRUP support
8. Disk encryption, including root (/) file system
9. Platforms:
 a. x86(32-bit)
 b. x86_64 (AMD64 & Intel64)
 c. IBM Power
 d. System z

10. Compatibility cupport for older programs (compat* RPMS)

Note: These are libraries to ensure the operation of older programs

Note: This ensures that certified applications continue to run

11. Anaconda auto-formats disk with:
 a. '/boot'
 b. '/' - root
 c. '/home' (if >= 50GB of storage are available)
 d. 'swap'

12. Anaconda & running system use: NetworkManager to auto-configure networking
 a. Uses DHCP by default
 b. Requires minimum intervention

13. Anaconda (installer) still supports absolute control over variables
 a. i.e. network settings may be specified during installation

##Prep Installation (HTTP) Server###

Features:

1. Easy access to ISO image contents

Tasks:

1. Mount ISO image in web-accessible directory on: 192.168.75.101/LinuxCBT/EL-6/Misc/RHEL6/

Note: Boot ISO image may be downloaded from redhat.com or created from DVD ISO image:

2. Reboot server and supply the following boot string:
 a. Press 'Tab' to edit boot option
 b. 'linux repo=http://192.168.75.101/LinuxCBT/EL-6/Misc/RHEL6 resolution=800x600 ip=192.168.75.20 netmask=255.255.255.0 gateway=192.168.75.1 dns=192.168.75.101'

###Basic Linux Skills###

Features:

1. A number of key commands

Tasks:

1. 'tty' - reveals connected terminal
 a. '/dev/pts/0' - psuedo-terminal 1

Note: Terminals are either real (console) or fake (pseudo)

Note: SSH and GUI terminals are pseudo

Note: Physical console terminals are real (tty?)

2. 'whoami' - reveals currently-logged-in user
3. 'w & who'
 a. 'who -a' - reveals ALL users and their processes
 b. 'w' - reveals currently-logged-in user and processes, etc.
4. 'pwd' - prints working directory of full path from '/' root of file system
5. 'cd' - changes directory
 a. 'cd ..' - moves one level up in the directory tree
 b. 'cd .' - current directory
 c. 'cd /' - moves using absolute path
 d. 'cd ~' - changes to currently-logged-in user's $HOME
 e. 'cd with tab-completion' - shortens navigation time

Note: Use Tab-completion with BASH shell commands to shorten navigation time

6. 'ls' - lists directories/files
 a. 'ls' - lists current directory in short form
 b. 'ls -l' - lists "" in long form
 c. 'ls -l /' - lists '/' in long form
 d. 'ls -al' - lists current directories entries including hidden items
 e. 'ls -l .Xauthority .ssh/' - lists multiple items
 f. 'ls -ltr' - sorts with most recent at the bottom
7. 'touch' - creates zero-byte file or updates timestamp on pre-existing file
8. 'echo' - echoes information to a default of STDOUT
 a. 'echo "This is a test" '
9. 'cat' - concatenates (brings together) content

a. 'cat test.txt'

Note: '$?' var contains the exit status of the most recently executed command

b. 'cat test.txt test2.txt > test3.txt'

10. 'id' - returns: UID, GID, GROUPS, SELinux Context (if enabled)

11. 'mkdir' - creates new directories

a. 'mkdir temp'

12. 'rm' - removes file(s)/directory(ies) - removes recursively

a. 'rm -rf temp*'

b. 'rm -rf temp[34]' - removes a range of items using Regular Expression (RegEx) - Character-Class

13. 'which' - searches current $PATH for executable

a. 'which cat' && 'which ls'

14. 'echo $PATH' - reveals the current $PATH

15. Redirection:

a. '<' - INPUT - Usually defaults to a source file

b. '>' - OUTPUT - clobbers target file

c. '>>' - APPEND - appends to target file if it exists and creates it if it doesn't

Examples:

a. 'cat test.txt' - reads the file 'test.txt' as STDIN (Standard INPUT)

Note: However, most commands will wait for keyboard input if no input file is specified

i.e. b. 'cat ' - waits on STDIN for input

Note: Use: 'CTRL-D' to quit STDIN from keyboard

Note: 'cat -' does the same as: 'cat'

b. 'cat test.txt > helloworld.txt' - bypasses STDOUT (Standard OUTPUT)

c. 'cat test.txt >> helloworld.txt' - "" but APPENDS to target file

16. Linux | UNIX Pipes - connects output stream of command a to input stream of command b

a. 'cat /var/log/messages | less' - pipes output of 'cat...' into 'less'

b. 'cat /var/log/messages | grep kernel | less' - parses '/var/log/messages' for keyword 'kernel' then pipes the output to 'less' to display one pageful at a time

Note: When piping, STDIN becomes the content of the pipe

17. Command Chaining

a. 'cat /var/log/messages | grep kernel | wc -l'

b. 'rm -rf temp* ; ls -l' - runs both commands independently

Note: Command Chaining is not dependent upon the exit status of the most-recently executed command

18. Command Dependency: AND || OR

a. 'rm -rf temp* || ls -l' - run 'ls -l' if 'rm -rf temp*' fails

b. 'rm -rf temp* && ls -l' - run 'ls -l' if and only if 'rm -rf temp*' works

10. Command History - built-in command (BASH)

a. 'history'

Note: BASH maintains a number of variables per shell

a. 'OLDPWD' - updated as you navigate the directory tree

b. 'LOGNAME'

c. 'SHELL'

etc.

11. 'export' exports vars

a. 'export PATH=$PATH:/tmp' -appends '/tmp' to current shell's PATH

12. 'more' - similar to 'less'

13. 'cp' - copies data

a. 'mkdir temp && cp -v test.txt temp/ && ls -l temp/'

14. 'mv' - moves data

a. 'mv test.txt temp/ && ls -l . && ls -l temp/ && echo $?'

Note: In scripts, prefix exit status with meaningful text:

i.e. 'echo "EXIT STATUS: " $? '

Note: BASH Shell allows simple navigation using:

a. 'CTRL-a' - takes you to the beginning of the line

b. 'CTRL-e' - takes you to the end of the line

c. 'CTRL-b' - back one character

d. 'CTRL-f' - forward one character

15. 'dmesg' - Kernel Ring Buffer - Pre-Syslog

a. '/var/log/dmesg' - 'dmesg'

Note: Also contains how the most recent invocation of the kernel took place (command-line)

b. 'dmesg | grep -i 'command line' ' - returns current kernel command line

16. 'head & tail' - Returns header and footer of text documents

a. 'head /var/log/messages' - returns first 10 lines

b. 'tail /var/log/messages' - returns last 10 lines

17. 'file' - returns the type of data stored in a file

18. 'ps' - lists processes

a. 'ps' -lists processes for current user

19. 'top' - dynamic 'ps', 'free', 'uptime', 'vmstat'

20. 'free' - memory allocation - RAM & SWAP

21. 'uptime' - shows system uptime and load average

22. 'df' - shows disk allocation and mount point

a. 'df -h' - human-readable format

Note: '-h' often means human-readable for many commands

23. 'cat /proc/cpuinfo' - enumerates detected CPUs

Note: '/proc' is a virtual (in-RAM) FS which houses system statistics

Note: System utilities read from: /proc to display values: i.e. 'free', 'top', etc.

24. 'uname' - enumerates kernel version
25. 'seq' - generates a sequence of numbers - useful with looping in the $SHELL

###Compression Utilities###

Features:

1. de/Compression of content
2. 'gzip/gunzip'
3. 'bzip2/bunzip2'
4. 'zip/unzip'
5. 'tar'

Tasks:

1. 'gzip'

a. 'gzip -c 1million.txt ' - redirects compressed file to STDOUT

b. 'gzip -c 1million.txt > 1million.txt.gz' - redirects compressed output to file

c. 'gzip -l 1million.txt.gz' - returns compression statistics

d. 'zcat 1million.txt.gz' - dumps (catenates) the contents of 1million.txt.gz to STDOUT

e. 'gunzip 1million.txt.gz' - overwrites, with permission, the original file

f. 'gunzip -c 1million.txt.gz > 1million.txt2'

Note: Typical compressed file online resembles: 'filename.tar.gz'

2. 'bzip2'

a. 'bzip2 -c 1million.txt > 1million.txt.bz2'

b. 'bunzip2 -c 1million.txt.bz2 > 1million.txt3' - redirects source

c. 'bzcat 1million.txt.bz2' - dumps original content to STDOUT

3. 'zip & unzip'

a. 'zip 1million.txt.zip 1million.txt'

b. 'unzip 1million.txt.zip' - attempts to overwrite original file

c. 'zcat 1million.txt.zip'

4. 'tar' - creates archives

a. 'tar -cvf temp.tar temp/' - creates an archive without compression

b. 'tar -tvf temp.tar' - enumerates contents of tarball

c. 'tar -cvzf temp.tar.gz temp/' - creates Tar - Gzip image

d. 'tar -cvjf temp.tar.bz2 temp/ temp2/' - create Tar - Bzip2 image

e. 'tar -xvf temp.tar.gz' - extracts file, recreating hierarchy

Note: 'du' - shows disk utilization for directory hierarchy

a. 'du -ch' - returns storage of hiearchy from current directory, below

b. 'du -chs' -returns total storage sans individual items

###Checksums###

Features:

1. Integrity checks on content (files)
2. Included tools:

a. 'md5sum' - 128-bit

b. 'sha1sum' - 160-bit

c. 'sha256sum' - 256-bit

d. 'sha512sum' - 512-bit

Tasks:

1. 'md5sum'

a. 'md5sum 1million.txt' - returns string that is unique to its content

b. 'md5sum 1million.txt2' - returns the same string because the content are identical

c. Alter content in various files and compare MD5SUMs

Note: A single bit differential will cause the checksum to vary

2. 'sha1sum'

a. 'sha1sum 1million.txt'

Note: Backticks are used to support shell-based command-substitution

i.e. 'rpm -qf `which sha1sum`' OR 'rpm -qf $(which sha1sum)'

b. 'sha1sum 1million* > 1million.txt.sha1sums'

c. 'sha1sum -c 1million.txt.sha1sums' - confirm SHA1SUMs wholesale

3. 'sha256sum' - 256-bit

a. 'sha256sum 1million* > 1million.txt.sha256sums'

b. 'sha25sum --quiet -c 1million.txt.sha25sums' - quietly checks ALL sums

Note: Returns error if 1 or more fail

4. 'sha512sum' - 512-bit

a. 'sha512sum 1million* > 1million.txt.sha512sums'

Note: If file changes during checksum calculation, then its checksum will be incorrect, resulting in confirmation failures

###GREP###

Features:

1. Processes lines using regular expressions (normal and metacharacters)
2. Returns entire lines when keyword is matched
3. Searches are case-sensitive, by default (use: '-i' to enable case-insensitivity)
4. Shares regular expressions with: Awk & Sed

Tasks:

1. Create file with content
2. Peform queries

a. 'grep "Linux" grep.test.txt' - returns ALL matches for the case: 'Linux'

b. 'grep -i "linux" grep.test.txt' - returns ALL cases of the word: 'linux'

c. 'grep "2" grep.test.txt' - returns ALL lines containing the number 2

3. Metacharacters

a. 'grep "2011$" grep.test.txt' - returns lines that terminate with: '2011'

Note: '$' means to search for content @ the end of the line

b. 'grep "^Linux" grep.test.txt ' - returns lines beginning with: 'Linux' - case-sensitive

c. 'grep -i "^Linux" grep.test.txt ' - returns lines beginning with: 'Linux' - case-insensitive

Note: '^' & '$' are anchor tags

d. 'grep "L.*" grep.test.txt ' - searches for 'L' followed by any characters

e. 'grep '^L.*' grep.test.txt ' - searches for 'L' where begins the line, etc.

Note: '.*' - means 0 or more matches

f. 'grep -i '^L.*CBT$' grep.test.txt - searches where 'L' begins the line and 'CBT' ends the line

g. 'grep -i '^L.*CBT $' grep.test.txt ' - searches where 'L' begins the line and ' ' ends the line

h. 'grep -i '^L.*CBT.* $' grep.test.txt' - searches where 'L' begins the line and ' ' ends the line with variations between

i. grep -i '[Red|2011]' grep.test.txt' - uses character classes

4. Parse system log

a. 'grep -i '^Jan  9' /var/log/messages-20110109 '

b. 'grep -i '^Jan  7' /var/log/messages-20110109 | grep -i 'kernel' '

c. 'grep -i '^Jan  [89]' /var/log/messages-20110109 | grep -i 'kernel' ' - searches for both: 'Jan  8' and 'Jan  9'

###Awk###

Features:

1. Field Processor
2. Supports grep-style (POSIX) regular expressions
3. Default field-delimiter is whitespace
4. Stores fields (columns) into tokens, which then become accessible during processing
5. Loops over input one line at a time
6. Will accept input from: file or STDIN or pipe

Tasks:

1. awk '{ print $0 }' grep.test.txt - prints each line in its entirety
2. awk '{ print $1 }' grep.test.txt - prints column #1 from each line
3. awk '{ print $2 }' grep.test.txt - prints column #2 from each line
4. awk '{ print $2,$1 }' grep.test.txt - prints column #1 then #2
5. awk '/Red/ { print $0}' grep.test.txt - prints ALL columns where line includes 'Red'
6. awk '/Red/ { print $1,"-",$2,"-",$3}' grep.test.txt - prints ALL columns, with transformations, where line includes 'Red'
7. awk '{ if ($2 ~ /2011/) print $0 }' grep.test.txt - prints ALL columns of records containing '2011' in the second column
8. awk '/2011$/ { print $0 }' grep.test.txt - prints lines ending in: '2011'
9. awk '/2011$/ { print $0 }' - waits on STDIN for input
10. grep 2011 /var/log/messages | awk '/2011$/ { print $0 }' - accepts a pipe
11. awk '{ if ($2 ~ /9/) print $3,$4,$5,$6 }' /var/log/messages - prints columns $3-$6 where colum 2 = '9'

###Sed (Stream Editor)###

Features:

1. Stream Editing
2. Manipulate text at any point
3. Instructions may be specified on command line or via file
4. Supports POSIX Regular Expressions (Grep & Awk)

Tasks:

1. 'sed -n '1p' grep.test.txt ' - prints the first line of the file
2. 'sed -n '2p' grep.test.txt ' - prints the second line ...
3. 'sed -n '$p' grep.test.txt ' - prints the last line ...
4. 'sed -n 4,13p grep.test.txt ' - prints lines 4 - 13 ...
5. 'sed -n '1!p' grep.test.txt ' - prints ALL but line 1
6. 'sed -n '1,3!p' grep.test.txt ' - prints ALL but lines 1-3
7. 'sed -n -e '/2011/p' grep.test.txt ' - prints lines containing '2011'

8. 'sed -n -e '/2011$/p' grep.test.txt ' - prints lines ending with '2011'

9. 'sed -n -e '/^2011/p' grep.test.txt ' - prints lines beginning with '2011'

10. 'sed -n -e '/^2011$/p' grep.test.txt ' - prints lines starting & ending with '2011'

11. 'sed -n -e '/[0-9]/p' grep.test.txt - prints lines containing numbers

12. 'sed -n -e '/^[0-9][0-9][0-9][0-9]$/p' grep.test.txt' - prints lines containing 4 juxtaposed numbers

13. 'sed -n -e '/^[0-9]\{4\}$/p' grep.test.txt ' - returns lines containing 4 juxtaposed numbers that begin and end the line

14. 'sed -n -e '/^Red/,/Linux/p' grep.test.txt - extracts a range of lines from string: '^Red' to 'Linux'

15. 'sed -n -e '/^Red/,+2p' grep.test.txt' - extracts line with 'Red' and 2 others

16. 'sed -e '/^$/d' grep.test.txt' - deletes blank lines

17. 'sed -e '/^$/d' grep.test.txt > grep.test.txt2' - deletes blank lines and saves results

18. 'sed -i.bak -e '/^$/d' grep.test.txt' - deletes blank lines in-place and archives original(source) file

19. 'sed -n -e 's/2010/2011/p' grep.test.txt '

Note: '-n' suppresses non-matching lines

###Perl###

Features:

1. All-purpose scripting environment

Tasks:

1. Exploring Perl Environment

a. 'perl -e 'print "Hello World\n;"' - prints 'Hello World' to STDOUT

b. ' perl -e 'print "Hello World\n";' -e 'print "Learning about the magic of Perl\n"; ' -w '

c. ' perl -e '$fname = "Deano"; $lname = "Davis"; print "$fname $lname\n"; ' -w

2. Write simple script

Note: All shell scripts should include a shebang header: i.e. '#!/path/to/script_engine'

a. create simple script

b. check for errors - 'perl -c name_of_script'

c. flag script executable: 'chmod +x perl_script_1.pl'

###User & Group Management###

Features:

1. GUI

2. TUI - Text User Interface tools

Tasks:

1. 'system-config-users' - create additional users and evaluate

Note: If user's $SHELL is set to: '/sbin/nologin' the user will not be able to obtain a shell, nor will 'root' be able to 'su' as that user: i.e. 'adm', 'daemon', 'bin', etc.

Note: System accounts typically are present in the process listing sans TTY because they do not need a $SHELL

Note: Regular users who are defined with: '/sbin/nologin' as their $SHELL may not access the system via a $SHELL. i.e. via 'SSH' or 'Telnet', however, they may access the system via an appropriate daemon. i.e. 'FTPD'

Note: Defaults are assigned to new accounts, including, but not limited to:

1. $SHELL = /bin/bash

2. $HOME = /home/$USER

2. $SHELL Tools

a. 'groupadd linuxcbt4'

b. 'useradd -d /home/linuxcbt4 -s /bin/bash -g linuxcbt4 linuxcbt4'

c. 'passwd linuxcbt4'

Note: Account information, by default, is stored in:

a. '/etc/passwd' - general account data: username, uid, gid, $HOME, $SHELL, reference to shadow

b. '/etc/shadow' - password and policy data

Sample '/etc/shadow' entry:

linuxcbt:$CqvB.$o4lwrI5pS2Ovh6IgyA9w3FDwGi9wJjEXYcbot6o5NsjahpEQK5GzHz8ccj7pX3rnPq2ozE7fwQEchJmEZB8T8/:14981:0:99999:7:::

d. '/etc/shadow':

d1. login name

d2. encrypted password

d3. Days since Unix epoch, password was last changed

d4. Days before password may be changed

d5. Days after which password must be changed

d6. Days before password is to expire that user is warned

d7. Days after password expires that account is disabled

d8. Days since Unix epoch, that account is disabled

d9. Reserved

Note: 'usermod' - basic: /etc/passwd changes

Note: 'chage' - /etc/shadow policy changes

3. Use 'chage' to alter account policy for users

a. 'chage -M 10 linuxcbt4 && chage -l linuxcbt4'

b. 'chage -M 3 -m 1 linuxcbt3 && chage -l linuxcbt3'

4. Explore: '/etc/login.defs'

a. Contains account policy settings

b. Modify defaults to company policy

5. Test policy changes by creating new account

a. 'groupadd linuxcbt5 && useradd -g linuxcbt5 -d /home/linuxcbt5 -s /bin/bash linuxcbt5 && chage -l linuxcbt5'

6. 'userdel'

a. 'userdel -r linuxcbt5' - removes user, group, $HOME, $MAIL traces

###File Types - Permissions###

Features:

1. Classification of files

2. Permissions

Tasks:

1. Classification of files

a. Use: 'ls -l' to expose file properties

'-rw-rw-r--. 1 linuxcbt linuxcbt 6888896 Jan  7 16:46 1million.txt'

'-' -> standard file

'drwxr-xr-x. 2 linuxcbt linuxcbt    4096 Jan  7 11:14 Desktop'

'd' -> directory

Note: RHEL6 uses color templates for classifying files:

'black' -> standard file

'blue' -> directory

'red' -> compressed file

'green' -> executable

Note: The color pattern is subject to change, so don't always rely upon it

'crw-------. 1 root root      4,   1 Jan  7 11:31 tty1'

'c' -> character device

'lrwxrwxrwx. 1 root root         15 Jan  7 11:03 stdin -> /proc/self/fd/0'

'l' -> symbolic link

'brw-rw----. 1 root disk      8,   0 Jan  7 11:03 sda'

'b' -> block (storage) device - i.e. hard drive, USB stick, etc.

2. Permissions

a. Represented by 9-rightmost bits in 10-bit permissions block

'-rw-rw-r--. 1 linuxcbt linuxcbt 6888896 Jan  7 16:46 1million.txt'

'rw-' - owner bits - 2,3,4 = 4+2 = 6

'rw-' - group owner bits - 5,6,7 = 4+2 = 6

'r--' - other/everyone bits - 8,9,10 = 4+0 = 4

Permissions Values:

'r' = 4 = read

'w' = 2 = write

'x' = 1 = execute

b. Use 'chmod' to influence permissions on file objects - it changes the octal mode

c. Default permissions are inherited from the $UMASK var

d. 'chmod 666 /tmp/1million.txt'

e. 'chmod u-w /tmp/1million.txt' - removes owner's ability to write to the content

e. 'chmod o-w /tmp/1million.txt' - removes other/everyone's ability to write to the content

f. 'chmod g-w /tmp/1million.txt' - removes group's ability to write to the content

'drwxrwxr-x. 2 linuxcbt linuxcbt 4096 Jan  7 17:23 temp'

Default directory permissions is octal: 775

g. 'chown linuxcbt /tmp/1million.txt && ls -l /tmp/1million.txt'

h. 'chmod o-r /tmp/1million.txt && stat /tmp/1million.txt'

i. 'chmod 600 /tmp/1million.txt && stat /tmp/1million.txt'

j. 'chown linuxcbt:linuxcbt /tmp/1million.txt && stat /tmp/1million.txt'

k. 'chgrp linuxcbt /tmp/1million.txt && stat /tmp/1million.txt'

3. SETUID/SETGID/STICKY Bit

a. 'chmod 4755 perl_script_1.pl' - causes script to always run as user/owner

Note: permission will reflect: '4755' with 'rws'

Note: The 's' replaces the 'x' for the owner to indicate SETUID

b. 'mkdir /projectx && chmod 2755 /projectx' - causes files created in directory to inherit group permissions

b1. 'chmod g=s /projectx'

b2. 'groupadd projectx'

b3. 'chown linuxcbt:projectx /projectx'

b4. 'chmod 2775 /projectx && stat /projectx'

c. '/tmp' -> example of sticky bit - leading value of: '3'

c1. 'chmod 3777 /tmp' || 'chmod +t /tmp' - sets sticky bit on object

###Symbolic Links###

Features:

1. Two types

a. 'symlinks' - soft - facilitate intra/inter-file-system links

a1. based on file names in the file system, NOT inodes

b. 'hard links' - hard - facilitate intra-file-system links
b1. based on inodes, NOT file names

Tasks:
1. Symlinks - shortcuts
a. 'ln -s /tmp/1million.txt ./tmp.1million.txt'

```
lrwxrwxrwx. 1 linuxcbt linuxcbt   17 Jan 11 11:56
tmp.1million.txt -> /tmp/1million.txt
```

Creates a link to the actual file name
Note: Soft-links do NOT increment the link counter returned by 'ls -l' || 'stat'
Note: So long as the source file name and directory location remain unchanged, the soft-links will work

2. Hardlinks - shortcuts to inodes - may not span (go across) file systems
a. 'ln /tmp/1million.txt /projects/hard.1million.txt' - increments the link counter
b. 'ls -li filename' - reveals inode
Note: permissions apply to ALL linked (hard & soft) files

###Quota Implemenation###
Features:
1. Limit storage consumption per user/group
2. Based on: disk block usage or inode usage
3. Imposed in 2 stages (thresholds): soft & hard
a. Soft limit: may be execeeded for up to the grace period
b. Hard limit: may never be execeeded under any circumstance

Requires:
1. 'quota*' RPM
2. Must associates file system(s) with quota management: user and/or group

Steps:
1. Enable in: '/etc/fstab'
a. 'defaults,usrquota,grpquota' - impose on: '/home'
2. Remount the file system: '/home'
Note: Effect quota management during single-user / installation modes to avoid disconnects in service
a. 'mount -o remount /home' - remounts the file system
Note: Optional methods of remounting the file system include: umount/mount OR reboot the system
b. 'mount' - reflects whether or not: 'usrquota', 'groupquota' options have been enabled

3. Create quota database files and generate disk usage table - defines baseline
a. 'quotacheck -cug /home' - applies user and group quotas
Note: 'quotacheck' should be run in: Single-user mode OR when the system reboots to facilitate: read-only remount of target file system
b. Use: '-m' option to override

4. Check defined quota database:
a. 'quotacheck -amvug' - checks quotas - forces check

5. Assign quota policies per user and/or group:
a. 'edquota linuxcbt4' - uses default editor ($EDITOR)

6. Run 'quotacheck -avugm' to update stats
7. Run 'repquota /home' to show FS-wide usage report
8. Use: 'edquota -t' to modify grace period
a. 'edquota -T linuxcbt4'

9. Use: 'quotaon ...' - to enter production mode
a. 'quotaon -vug /home' - enters production mode
b. 'quotaon -p /home' - echoes current quota status

Note: Default grace period is 7-days
10. Attempt to write data beyond soft limit grace period

###Provision Partitions & File Systems###
Features:
1. Ability to provision additional storage

Tools:
1. 'fdisk'
2. 'parted'
3. 'mke2fs' - ext2,ext3,ext4 FSs

Storage Hierarchy:
Disk
-Partition(s)
-File System(s)

Tasks:
1. Enumerate available storage:
a. 'fdisk -l' - enumerates disks and partitions
b. 'parted -l' - ""

2. Provision additional storage:
a. Select disk: /dev/sdb
b. 'parted /dev/sdb'
c. 'mkpart primary 1 10GB'
d. 'mke2fs -t ext4 -j /dev/sdb1' - overlays EXT4 FS on: /dev/sdb1
e. 'mkdir /temp10G1'

f. 'mount /dev/sdb1 /temp10G1 && mount'

g. Create content in new repository

3. Repeat process on the same disk

4. Make partitions available across reboots:

a. '/etc/fstab'

5. Unmount both partitions and re-mount via: '/etc/fstab'

a. 'umount /temp10G1 && umount /temp10G2 && mount'

b. 'mount -a' - reads the contents of: '/etc/fstab'

Note: Paritioning is typically handled during installation and/or within runlevel 1

###Provision Swap Space###

Features:

1. Generates additional virtual memory
2. Temporary fix for RAM-shortage. Permanent fix is to add more RAM.
3. Requires no system downtime
4. Works with dedicated partitions
5. Works with existing file systems
6. Works across disks, consequently improving performance

Tasks:

1. Define swap partition and provision

a. 'fdisk /dev/sdb' - create partition and set to type '82' with 't' option

b. 'mkswap /dev/sdb3' - i.e. similar to: 'mke2fs'

Note: If necessary, reboot the system after using: 'fdisk' or 'parted' to provision new swap partition

c. 'swapon -s' displays current swap devices

d. 'swapon -v /dev/sdb3' - enables swapping on specific device

e. 'swapoff /dev/sdb3' - disables swapping on specific device: /dev/sdb3

2. Define swap storage on existing file system

a. 'dd if=/dev/zero of=/swapfile1G  bs=1024 count=1048576' - generates a file that we can overlay a swap file system on of size: 1G

b. 'mkswap /swapfile1G'

c. 'swapon -v /swapfile1G'

###Logical Volume Managment (LVM)###

Features:

1. Volume sets - aggreate storage from disparate sources
2. Resize storage on-the-fly
3. Provision storage as necessary

Tasks:

1. LVM Storage Hierarchy

Logical Volume - configure file system at this level

- Volume Groups - represents one or more physical volumes

- Physical Volumes: (i.e. /dev/sdb4, /dev/sdc3, etc.) - partition, using fdisk or parted: LVM type (8e)

2. Create LVM Storage Hierarchy - 6-Steps

a. Create LVM partitions on available disks

a1. 'parted /dev/sdb'

a2. 'mkpart primary start end'

a3. 'set partition_num lvm on'

a4. 'reboot'

b. 'pvcreate /dev/sdb4 /dev/sdc3' - create physical LVM volumes from partitions

b1. 'pvdisplay'

c. 'vgcreate volgroupvar /dev/sdb4 /dev/sdc3' - allocates both volumes to the volume group

d. 'lvcreate -L 5GB -n logvolvar volgroupvar'

e. 'mke2fs -t ext4 -j /dev/volgroupvar/logvolvar' - overlays EXT4 FS on LVM volume

f. 'mkdir /lvmvar1 && mount /dev/volgroupvar/logvolvar /lvmvar1'

g. Update: '/etc/fstab' for persistence

3. Resize LVMs

a. 'lvresize -L 6GB /dev/volgroupvar/logvolvar'

b. 'resize2fs /dev/volgroupvar/logvolvar 6G'

c. 'lvresize -L 4GB /dev/volgroupvar/logvolvar'

d. 'resize2fs /dev/volgroupvar/logvolvar 4G'

Note: Reductions will likely return errors resulting in re-provisioning of the FS

4. Rename Logical Volume

a. 'lvrename volgroupvar logvolvar logvolopt' - renames volume, NOT volume group

b. 'lvresize -L 6GB /dev/volgroupvar/logvolopt' - restores to 6GB

5. Rename Volume Group

a. 'vgrename volgroupvar volgroupopt' - renames the volume group

b. update: '/etc/fstab' - to reflect volume group name change

6. Assign more partitions(storage) to LVM

a. 'parted /dev/sdc'

b. 'mkpart primary 16.1GB 26.1GB'

c. 'set 4 lvm on'
d. 'pvcreate /dev/sdc4' - assigns LVM partition to LVM management
e. 'vgextend volgroupopt /dev/sdc4' - extends volume group: 'volgroupopt'
f. 'lvresize -L 15GB /dev/volgroupopt/logvolopt' - online resize
g. 'resize2fs /dev/volgroupopt/logvolopt 15G' - online resize

7. LVM GUI
a. 'system-config-lvm'
b. 'ssh -X root@192.168.75.20' - redirects X.org session back to local GUI
c. Extend storage of: '/dev/volgroupopt/logvolopt' to: 16GB
Note: GUI will send appropriate commands to system to:
a. Resize logical volume (logvolopt)
b. Resize EXT4 FS to appropriate size

8. Recreate LVM hierarchy
a. Unmount any partitions tied to: '/dev/sd[bc]'
b. 'parted /dev/sdb' - remove partitions & create new LVM partitions
c. 'init 6' - reboot
d. Use: 'system-config-lvm' to create volume group from: '/dev/sdb1' & '/dev/sdc1'
e. Create logical volume: 'logvolopt'
f. Mount at: '/opt'

###RAID###
Features:
1. Data spread across 2 or more disk/partitions
2. Redundancy - recover from catastrophy
3. Levels: 0,1,4,5,6,10

Tasks:
1. RAID0 - volume set creation i.e. LVM
a. Create multiple partitions: /dev/sd[bc][5-8] - of type '83' || 'linux'
b. 'init 6' - reboot
c. 'mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sdb5 /dev/sdc5'
d. 'mke2fs -t ext4 -j /dev/md0'
e. 'mkdir /raid0 && mount /dev/md0 /raid0'
f. 'nano /etc/fstab'

2. RAID1 - mirroring - halves the storage
a. 'mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sdb6 /dev/sdc6'
b. 'mke2fs -t ext4 -j /dev/md1'
c. 'mkdir /raid1 && mount /dev/md1 /raid1'

3. RAID5 - striping with parity - sacrifices the equivalent of 1-drive(partition)
a. 'mdadm --create /dev/md2 --level=5 --raid-devices=4 /dev/sdb7 /dev/sdb8 /dev/sdc7 /dev/sdc8'
b. 'mke2fs -t ext4 -j /dev/md2'
c. 'mkdir /raid5 && mount /dev/md2 /raid5 && seq 1000000 > /raid5/1million.txt && ls -l /raid5'
d. nano /etc/fstab
e. test auto-mount during system initialization

###RAID Management###
Features:
1. Create
2. Assemble: assembles pre-existing array(s)
3. Manage: Use to fail devices to take them offline
4. Monitor: E-mail, run processes, etc.
5. Misc: '--query', '--detail', '--examine'(individual RAID components'

Tasks:
1. 'cat /proc/mdstat' - enumerates currently-available RAID-arrays (sets)
2. 'mdadm --query /dev/md[0-2]' - returns information about the 3 arrays: 0-2
3. Publish RAID array as a read-only volume
a. 'umount /dev/md0' - unmounts the RAID array
b. 'mdadm -o /dev/md0' - flags, in the superblock, the array: /dev/md0 as Read-Only
c. 'mount /dev/md0 /raid0'
d. 'mount'
4. Publish RAID array as a read-write volume
a. 'umount /dev/md0' - unmounts the RAID array
b. 'mdadm -w /dev/md0' - flags, in the superblock, the array: /dev/md0 as Read-Write
c. 'mount /dev/md0 /raid0'
d. 'mount'
5. Stop RAID volume for management purposes
a. 'mdadm --manage --stop /dev/md0' - facilitates offline management
Note: Stopping/deactivating the array will remove its '/dev/md?' entry
Note: There are multiple ways to reassemble RAID arrays:
1. command-line: 'mdadm -A /dev/md0 /dev/sdb5 /dev/sdc5' - restarts (reassembles) '/dev/md0' from its component parts
2. '/etc/mdadm.conf' - associates DEVICES & ARRAYS and management/notification info.
a. 'DEVICE /dev/sdb[5678] /dev/sdc[5678]'

b. 'ARRAY /dev/md0 devices=/dev/sdb5,/dev/sdc5'

6. Other options:
a. 'mdadm -D /dev/md[0-2] - enumerates info. about ARRAYS
b. 'mdadm -E /dev/sd[bc][78] - enumerates info. about the 4 partions on the 2 drives: /dev/sd[bc]

###Package Management with RPM###
Features:
1. Compression of packages
2. SHA-256 hashes are used to sign packages
3. RPM DB: '/var/lib/rpm' - tracks installed packages, attributes of package files, etc.
4. 5-Modes of operations:
a. Install
b. Uninstall
c. Upgrade
d. Query
e. Verify
5. Caveat: Does NOT auto-resolve dependencies: Use 'yum'
6. Caveat: RPM does NOT track non-RPM programs/apps: i.e. '*.tar.gz' || '*.tar.bz2'

Tasks:
1. Query
a. 'rpm -qa' - dumps ALL installed packages (RPMs)
b. 'rpm -qa | grep grep' - 'grep-2.6.3-2.el6.i686'
'grep' - main name of package
'2.6.3-2' - package version
'el6.i686' - RedHat Version & Platform
c. 'rpm -qi grep' - returns metadata about 'grep' package
d. 'rpm -ql grep' - enumerates the contents of the package: 'grep'
e. 'rpm -qf /bin/grep' - enumerates the file's package membership
f. 'rpm -qd grep' - enumerates the included documentation
g. 'rpm -qc lftp' - enumerates a package's configuration file(s)
h. 'rpm -qpi http://192.168.75.101/LinuxCBT/EL-6/Misc/RHEL6/Server/Packages/unix2dos-2.2-35.el6.i686.rpm'

2. Verify - Verifies file system contents against installed package in RPM DB
Note: Returns: '.' per test performed if the test passed
Note: If test fails, one of the following will be returned: 5(MD5), S(file size), L(symlink), T(mod time), D(device), M(mode), ?(unreadable file), U(user), G(group)
a. 'rpm -Vvf /bin/grep' - compares: /bin/grep to 'grep' RPM
b. 'mv /bin/grep /bin/grep.original && touch /bin/grep'
SM5....T.    /bin/grep

3. Install - Works if package does NOT exist on the system
a. 'rpm -ivh http://192.168.75.101/LinuxCBT/EL-6/Misc/RHEL6/Server/Packages/unix2dos-2.2-35.el6.i686.rpm'

4. Upgrade - Installs and/or Upgrades
a. 'rpm -Uvh http://192.168.75.101/LinuxCBT/EL-6/Misc/RHEL6/Server/Packages/dos2unix-3.1-37.el6.i686.rpm'
b. 'rpm -Uvh --replacepkgs http://192.168.75.101/LinuxCBT/EL-6/Misc/RHEL6/Server/Packages/grep-2.6.3-2.el6.i686.rpm'
5. Uninstall
a. 'rpm -e grep' - checks dependencies and warns where appropriate

6. Import RedHat RPM GPG Key to confirm package signatures:
a. 'rpm --import

###YUM###
Features:
1. Package management
2. Auto-dependency resolution
3. Ability to specify multiple package sources

Tasks:
1. Mirror 'Packages' directory on local system
a. 'lftp http://192.168.75.101/LinuxCBT/EL-6/Misc/RHEL6/Packages/'
b. 'mirror -v'

2. Run 'createrepo' against: '/var/www/html/RHEL6' - creates sub-directory: 'repodata' and various DB files to serve packages to 'yum' clients
a. Confirm that 'createrepo' RPM is installed
b. 'createrepo /var/www/html/RHEL6' - queries ALL 2679 packages and generates a SQLlite DB and ancillary files beneath: 'repodata' dir

3. Setup first 'yum' client: localhost
a. '/etc/yum.repos.d/linuxcbtserv2.repo'
'[linuxcbtserv2]
name=linuxcbtserv2

```
   baseurl=http://192.168.75.21/RHEL6 '
```

4. Search & Install packages:
 a. 'rpm -e dos2unix unix2dos' - removes both packages
 b. 'yum search unix2dos' - searches for package
 c. 'yum info unix2dos' - returns/dumps/enumerates package metadata
 d. 'yum install unix2dos' - installs the package once
 e. 'yum reinstall unix2dos' - reinstalls package. i.e. '--replacepkgs' with 'rpm'
 f. 'yum -y reinstall unix2dos' - assumes yes when prompted
 g. 'yum history' - returns usage history. i.e. BASH Shell history
 h. 'yum -y erase unix2dos dos2unix' - assumes yes and removes both packages
 i. 'yum deplist lftp' - dependencies and their providers are returned
 j. 'yum localinstall dos2unix-3.1-37.el6.i686.rpm' - Note: The entire file name is indicated

5. Define: 'linuxcbtserv1' as a 'yum' client of 'linuxcbtserv2
6. Define: 'linuxcbtserv1' as a 'yum' server

 a. 'lftp http://192.168.75.101/LinuxCBT/EL-6/Misc/RHEL6/Packages/'
 b. 'mirror -v'
 c. Confirm that 'createrepo' RPM is installed
 d. 'createrepo /var/www/html/RHEL6' - queries ALL 2679 packages and generates a SQLlite DB and ancillary files beneath: 'repodata' dir

7. Define: 'linuxcbtserv2' as a 'yum' client of 'linuxcbtserv1

Note: This configuration will provide YUM server redundancy via: 2-repo files per 'yum' client

```
'[linuxcbtserv1]
   name=linuxcbtserv1
   baseurl=http://192.168.75.20/RHEL6 '
```

8. Test YUM redundancy by enabling/disabling HTTPD(Apache) on both systems and installing/uninstalling packages

###PackageKit###

Features:
1. GUI for package management
2. Front-end to YUM
3. Supports YUM plug-ins

Tasks:
1. Explore Interface

###Cron###

Features:
1. Job Scheduler
 a. minutely
 b. hourly
 c. daily
 d. monthly
 e. yearly

Note: Fields: a-e are specified as per the order above in appropriate config. file

2. Assumes computer is always on unlike: anacron
3. Maintains: global and per-user schedules
4. /var/spool/cron - stores crontabs for: /etc/passwd users or LDAP or otherwise
5. Checks ALL config files every minute, including: /etc/anacrontab
6. Supplies 'crontab' utility to manage jobs
7. Runs in ALL multi-user modes. Does NOT execute in: Single-User (1) mode

Tasks:
1. Analyze current cron setup
 a. 'ps -ef | grep cron'
 b. '/etc/crontab'

2. Define system-wide job
 a. '*/1 * * * * linuxcbt  /usr/bin/uptime >> /home/linuxcbt/uptime.stat'

3. Define per-user job
 a. 'crontab -e' - run as user principle: 'linuxcbt'

4. Manipulate 'linuxcbt's' job as 'root'
 a. 'crontab -e -u linuxcbt' - run as 'root' - edits user's job(s)
 b. 'crontab -l -u linuxcbt' - run as 'root' - lists user's job(s)

4. Restrict Cron-access
 a. '/etc/cron.allow' - add 'linuxcbt to list - User MUST be on the list in order to submit jobs to 'cron'
 b. '/etc/cron.deny' - add 'linuxcbt2' to list

###Anacron###

Features:
1. Runs jobs once per day during an allowed interval
2. Assumes computer is NOT always on, unlike: Cron
3. Facilitates delays in starting jobs - reduces resource contention
4. Maintains one schedule: '/etc/anacrontab'
5. Requires little-to-no intervention; handled by the system

Tasks:
1. Examine: '/etc/anacrontab'

###'at' and 'batch'###

Features:
1. One-off job schedulers
2. 'at' runs based on time schedule
3. 'batch' runs based on system-utilization stats: default < 0.8 for load average

Tasks:
1. Use 'at' to run jobs
 a. 'at 15:58'
 b. 'at 16:01'
 c. 'at -f at.job.1 16:02'
 d. 'at now + 1 day' - runs job 1-day from now (time submitted to job-queue)

2. Use 'batch' to run jobs
 a. 'batch' - supply instructions on STDIN

Note: 'batch' accepts no command-line options
Note: 'at' runs the jobs on behalf of 'batch'
Note: 'batch' is simply a special invocation of 'at'

###Syslog###

Features:
1. Logs daemon information
2. Logs remotely
3. Accepts, if configured, logs from remote hosts: i.e. routers, switches, firewalls, content switches, Linux hosts, etc.
4. Supports: Unix Domain Sockets (/dev/log)
5. Supports: Internet Sockets: (UDP:514) and/or (TCP:514)
6. Runs in ALL multi-user levels: 2-5

Tasks:
1. Exploration of environment
 a. '/etc/rsyslog.conf' - primary config file
 b. '/etc/sysconfig/rsyslog' - ancillary config file, containing startup options

2. '/etc/resyslog.conf' - exploration

```
Selector(s)
            Action(s)
*.info;mail.none;authpriv.none;cron.none
/var/log/messages

# The authpriv file has restricted access.
authpriv.*                                    /var/log/secure
```

3. Configure UDP:514 routing of messages from Cisco Router
 a. '/etc/rsyslog.conf' - uncomment UDP section
 b. Setup selector in: '/etc/rsyslog.conf'
  b1. 'local4                    /var/log/cisco/ciscorouter.log'
 c. Create: '/var/log/cisco' - 'mkdir /var/log/cisco'
 d. Configure router to log, via UDP, to our RHEL-6 Server

```
'Jan 18 17:09:49 192.168.75.1 12987: 012457: Jan 18 17:10:44.123 EST: %SYS-6-LOGGINGHOST_STARTSTOP: Logging to host 192.168.75.21 port 514 started - CLI initiated '
```

Note: Syslog ALWAYS includes a: timestamp & hostname/IP prefix & message

Note: Syslog supports a number of levels (0-7):
Debug(0), info, notice, warning, error, critical, alert, emerg(7)
Note: Syslog supports a variety of facilities:
 a. MAIL
 b. AUTH
 c. LOCAL0-7

4. Configure TCP:514 routing of messages from Cisco Router
 a. '/etc/rsyslog.conf' - uncomment TCP section
 b. Update router configuration

###Log Rotation###

Features:
1. Management of logs
2. Reduction/control of size of log files
3. Config files: '/etc/logrotate.d'
4. Primary config file: '/etc/logrotate.conf'
5. Auto-includes files in: '/etc/logrotate.d' into main config file: '/etc/logrotate.conf'

6. Rotates based on criteria: time || size-based

```
'/etc/logrotate.d' - entry
/var/log/httpd/*log {
   missingok
   notifempty
   sharedscripts
   delaycompress
   postrotate
      /sbin/service httpd reload > /dev/null 2>/dev/null || true
   endscript
}
```

Tasks:

1. Update 'logrotate' to handle: '/var/log/cisco/*log' - '/etc/logrotate.d/syslog'

2. Create separate file to handle: '/var/log/cisco/*log' - '/etc/logrotate.d/cisco'
3. Update directives to rotate based on size-based criteria

###Common Network Utilities###

Features:

1. Determine if remote host is up/available: 'ping'
2. Determine if local/remote service is available: 'telnet'
3. Determine network sockets stats/connections: 'netstat'
4. View L2 information: 'arp'
5. View path taken by packets to remote system: 'traceroute'
6. Hostname-to-IP and reverse resolution: 'nslookup', 'dig'
7. Learn more information about and IP and/or block: 'whois'

Tasks:

1. Explore Packet Internet Groper (PING)
 a. 'rpm -qf `/bin/ping`' - member of 'iputils' package
 b. 'ping -c 3 192.168.75.1 -s 32' - sends 32-bytes + 8-bytes (ICMP overhead)
 c. 'ping -c 3 -i 3 192.168.75.1' - sends 3-packets of 56-bytes, every 3-seconds to target

Note: PING may be blocked by L3 devices on your network and/or the Internet

2. Telnet - Don't use for TTY access to remote host. Use SSH. Use Telnet to test port-accessiblity.
 a. 'telnet 192.168.75.1 22' - Install if necessary using 'yum install telnet'

3. Netstat - reveals TCP:UDP:Unix Sockets - '/proc/net'
 a. 'netstat -a' - dumps ALL sockets with: service/port and hostname resolution
 b. 'netstat -an' - same as above, but suppresses name resolution
 c. 'netstat -ntl' - suppresses name resolution, shows ONLY TCP sockets, and listeners
 d. 'netstat -ntlp' - same as above, includes programs bound to ports

Note: 'Use '-p' option as root to reveal ALL programs'

Note: ':::514' - means that port is bound to ALL IPv6 addresses configured on the host

Note: '0.0.0.0:514' - means that port is bound to ALL IPv4 addresses configured on the host

 e. ' netstat -i'
 f. 'netstat -nulp' - returns ALL UDP listeners
 g. 'netstat -rn' - returns kernel routing table

4. ARP - Address Resolution Protocol
 a. 'arp -a || arp -e'

Note: ARP is usually self-managing.

5. Traceroute - follows path taken by packets across the network (Intra/Internet)
 a. 'traceroute 192.168.75.1'
 b. 'traceroute www.linuxcbt.com'

6. 'nslookup'
 a. 'nslookup www.linuxcbt.com'

DNS client tools use: '/etc/resolv.conf' to determine which DNS servers to query

7. 'dig'
 a. 'dig www.linuxcbt.com'
 b. 'dig -x 71.6.195.206' - performs a reverse lookup
 c. 'dig linuxcbt.com mx'

8. 'whois' - Finds IP/domain ownership information
 a. 'whois linuxcbt.com'

###IPv4 Configuration###

Features:

1. DHCP
2. Static
3. Virtual (Sub) Interfaces - supports single physical connected to multiple logical

i.e. 192.168.75.0/24 && 192.168.76.0/24 && 10.0.0.0/30

Tasks:

1. Explore key: Directories & Files

a. '/etc/sysconfig/network' - system-wide settings: i.e. hostname, gateway, enabled|disabled

b. '/etc/sysconfig/networking' - 'system-config-network' tool controls this directory. Don't edit manually.

c. '/etc/hosts' - local name DB - should contain a record for the localhost: i.e. 'localhost.localdomain'

```
192.168.75.21   linuxcbtserv2.linuxcbt.internal
          linuxcbtserv2     # Added by NetworkManager
127.0.0.1       localhost.localdomain    localhost
::1       linuxcbtserv2.linuxcbt.internal    linuxcbtserv2
          localhost6.localdomain6            localhost6
```

Note: Add hosts to: '/etc/hosts', for which you cannot or should not resolve via DNS

d. '/etc/sysconfig/network-scripts'

d1. Interface configuration files - describes up/down config of interfaces: i.e. eth0

d2. Control files - describes how interfaces are to be brought: up/down - scripts

d3. Network function files - contain key network information required for the stack

d4. 'ifup-eth' - brings up ethernet interfaces: i.e. 'eth0', 'eth1', etc.

d5. 'ifdown-eth' - brings down ethernet interfaces: i.e. 'eth0', 'eth1', etc.

e. 'ifconfig' - enumerates configuration of interfaces

Note: At minimum, a routeable, connected system has at least 2 interfaces:

1. 'lo' - loopback - 127.0.0.1
2. 'eth0' - Ethernet0 - Your Routeable IP/Net

e1. 'ifconfig'

```
eth0      Link encap:Ethernet  HWaddr 00:11:11:A2:A2:D0
      inet addr:192.168.75.21  Bcast:192.168.75.255
Mask:255.255.255.0
      inet6 addr: 2002:4687:db25:2:211:11ff:fea2:a2d0/64
Scope:Global
      inet6 addr: fe80::211:11ff:fea2:a2d0/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500
Metric:1
      RX packets:14048921 errors:0 dropped:0 overruns:0
frame:0
      TX packets:9107918 errors:0 dropped:0 overruns:0
carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:469081450 (447.3 MiB)  TX
bytes:4022814991 (3.7 GiB)

lo      Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:4698 errors:0 dropped:0 overruns:0
frame:0
      TX packets:4698 errors:0 dropped:0 overruns:0
carrier:0
      collisions:0 txqueuelen:0
      RX bytes:7374035 (7.0 MiB)  TX bytes:7374035 (7.0
MiB)
```

e2. 'ifconfig eth0:1 192.168.75.22 netmask 255.255.255.0'

e2.1. 'ping -c 3 -I 192.168.75.22 192.168.75.21' - sources traffic as: 192.168.75.22

e3. 'ifconfig eth0:2 192.168.75.23 netmask 255.255.255.0'

e4. Preserve changes across system restart/ 'NetworkManager' service restart

e4.1. 'cp -v /etc/sysconfig/network-scripts/ifcfg-eth0 ifcfg-eth0:1'

f. 'ifcfg eth0:3 add 192.168.75.24/24' - Does duplicate address detection & sends ARP to hosts on the same Net as the interface

f1. 'ifcfg eth0:1 delete 192.168.75.22/24' - removes the sub-interface

f2. 'ifconfig eth0:3 del 192.168.75.24' - removes the sub-interface

###IPv6 Configuration###

Features:

1. Self-configuring - Prefix (/64), is auto-derived from Router
2. Can be configured via: Neighbor discovery auto-config by router, DHCPv6, Statically(manually)

Tasks:

1. 'less /etc/sysconfig/network-scripts/ifup-ipv6' - peruse config
2. Peruse Router Config

2a. '2002:4687:DB25:2:21A:2FFF:FEE3:F240'

2002:4687:DB25:2 - left-most 64-bits describes the subnet: /64 prefix - globally unique

:21A:2FFF:FEE3:F240 - right-most 64-bits describes the host. Includes 48-bit unique MAC address

3. PING6 various devices

a. 'ping6 -c 3 -I eth0 2002:4687:DB25:2:21A:2FFF:FEE3:F240'

b. 'ping 2002:4687:db25:2:211:11ff:fea2:a2d0' - from the router, PING6 RHEL-6 box

4. Use browser to access Apache via: IPv6

a. 'http://[2002:4687:db25:2:211:11ff:fea2:a2d0]/' - escape IPv6 address with '[]' || use: '/etc/hosts' || DNS

Note: IPv6 is auto-configured, by default, so long as Router or DHCPv6 provides a usable prefix.

Note: Update host configuration: i.e. '/etc/hosts' and/or DNS to reflect name-to-IPv6 mappings

Note: Test with desired applications: i.e. 'ssh', 'http client', etc.

###Very Secure File Transfer Protocol Daemon (VSFTPD)###

Features:

1. Anonymous (Default) and user-based FTP sessions
2. SSL support (provided by SSH) no need for VSFTPD
3. Does not permit 'root' or 'service accounts' access, by default
4. Does not currently support IPv4 & IPv6 simultaneously with the same daemon.

Tasks:

1. Install using: 'yum'
2. Enable 'vsftpd' in multi-user runlevels

a. 'chkconfig vsftpd on'

3. Start 'vsftpd' and explore access

4. Disable Anonymous access
5. Test local user access and update SELinux configuration

a. 'getsebool -a | grep ftp' - dumps FTP-related SELinux booleans

b. 'setsebool -P ftp_home_dir=1'

Note: RHEL6 enables SELinux in 'enforcing' mode, requiring a slight change to the booleans to permit VSFTPD or any FTPD daemon to transition user into their: $HOME directory

6. Enable Dual-Logging

a. 'dual_log_enable=yes'

7. Enable server time for display of files/directories

a. 'use_localtime=yes'

Note: 'man vsftpd.conf' for useful directives that apply to your application

###LFTP###

Features:

1. Interactive (Shell-like) & Non-interactive modes
2. Scriptable
3. Servers supported: FTP, FTPS, SSH(SFTP), HTTP, etc.
4. Mirroring of content: forward (download) & reverse (upload)
5. Regular expressions
6. Job Engine

Tasks:

1. Use 'lftp' to connect to VSFTPD

a. 'lftp localhost' && 'open -u linuxcbt'

Note: LFTP batches authentication commands and submits when control-channel commands such as 'ls' are received

---- Connecting to localhost (127.0.0.1) port 21 - (no connection)

<--- 220 Welcome to linuxcbtserv2.linuxcbt.internal FTP service. - (traffic from server to client)

---> FEAT - (traffic from client to server)

2. Use 'lftp' to connect and mirror content

a. 'mirror temp*' - forward mirror - downloads content from server to client

b. 'mirror -Rv *' - reverse mirror - puts content on server from client

3. Run external commands with: '!command'

a. '!bash' - launches an instance of BASH SHELL from within 'lftp'

b. 'exit' - returns to 'lftp'

4. Test rate-limiting with 'vsftpd'

a. 'local_max_rate=10000' - B/s (Bytes per second)

5. Job Management - Backrounding

a. Use: 'CTRL-Z' to background jobs

b. Use: 'jobs' to view progress of jobs

c. Use: 'fg job_num' to foreground a specific job

6. Explore LFTP environment

a. '/etc/lftp.conf' - system-wide config file

7. Connect using 'lftp' to: SSH & HTTP servers

a. 'lftp http://192.168.75.101/LinuxCBT/EL-6/Misc/RHEL6'

b. 'lftp -u linuxcbt sftp://192.168.75.101'

###Curl###

Features:

1. Non-interactive file transfers with: HTTP|FTP|Telnet|etc
2. Default downloads to STDOUT
3. Like 'wget'

Tasks:

1. 'curl http://192.168.75.101/LinuxCBT/EL-6/Misc/RHEL6/EULA' - dumps content of target file to STDOUT

Note: This can be useful when used with pipes, etc.

2. Create multiple files on HTTP server and download, one-shot, with 'curl'
   a. 'for i in `seq 5`; do seq 1000000 > file$i.txt; done' - execute on target HTTP server
   b. 'curl -O http://192.168.75.101/LinuxCBT/EL-6/Misc/file[1-5].txt' - downloads file1..file5.txt to local system

3. Create files on multiple HTTP servers and aggregate with 'curl'
   a. 'curl -O http://192.168.75.{101,21}/LinuxCBT/file[1-5].txt'

4. Rate-Limit
   a. 'curl -O --limit-rate 1000k http://192.168.75.101/LinuxCBT/file[1-5].txt'

###Rsync###

Features:

1. Network Copies
2. Optionally, local copies
3. Ability to synchronize content quickly: i.e. staging -> production sites
4. Uses SSH as a conduit
5. Requires 'rsync' on client/server systems
6. Non-interactive client
7. Syntax is similar to: 'scp'

Tasks:

1. 'rsync -av SRC DST'
2. 'rsync -av --delete SRC DST' - removes superfluous content on DST (reverse mirror)

###TFTPD###

Features:

1. Fast, UDP-based file transfers
2. Unreliable, however, in a LAN-connected environment, it is rather reliable
3. Update devices that function as TFTP clients: Cisco devices (routers, switches, firewalls, etc.)
4. Managed via: 'XINETD'

Tasks:

1. Install 'tftp-server' RPM
   a. 'yum search tftp && yum -y install tftp-server'

'/var/lib/tftpboot' - directory where TFTPD-served content lives

'/etc/xinetd.d/tftp' - primary, XINETD-controlled, config file - enable/disable TFTPD here

'/usr/sbin/in.tftpd' - binary (daemon) - invoked by XINETD when necessary

2. Enable TFTP Server (TFTPD)
   a. '/etc/xinetd.d/tftp'
   b. 'service xinetd start'
   c. 'netstat -nul ' - ensure that: 'UDP:69' is listening and controlled by: 'xinetd'

3. Backup Cisco Router Configuration
   a. 'ssh linuxcbt@192.168.75.1'
   b. 'cp running-config tftp://192.168.75.21/linuxcbtrouter1.config'

Note: '/var/lib/tftpboot/' - root indicated in above URI, NOT the root (/) of the Linux FS

   c. 'touch /var/lib/tftpboot/linuxcbtrouter1.config && chmod 666 /var/lib/tftpboot/linuxcbtrouter1.config'
   d. Attempt to backup the configuration

4. Restore Cisco Router Configuration
   a. 'copy tftp://192.168.75.21/linuxcbtrouter1.config running-config'

Note: Sometimes, the restoration will generate errors. Check for accuracy

5. Use TFTP client to move data

Note: SFTP/SCP/FTPS are preferred, however, TFTP client may be convenient

Note: TFTP client is both: interactive & non-interactive

   a. 'tftp -v 192.168.75.21'

6. Overwrite TFTP Server data from rogue client:
   a. 'ssh 192.168.75.101 && tftp -v 192.168.75.21 -c put linuxcbtrouter1.config'

Note: Best practice suggests that you should run TFTPD only when/if necessary. Disable when not needed and, flag files in: '/var/lib/tftpboot' to restrictive permissions: i.e. 'chmod 644 /var/lib/tftpboot/*'

###TELNETD###

Features:

1. Clear-text means of accessing a TTY (PTY) across the wire
2. XINETD-controlled
3. Does NOT allow 'root' to access TTY via Telnet: '/etc/securetty'
4. Reads, as a banner, '/etc/issue.net | /etc/issue'
5. Reads, post-login, '/etc/motd' - publish useful info. here

Note: contents of: '/etc/motd' are also read by: SSHD

6. Assigns pseudo-terminals akin to: SSHD , however, they are flagged as unencrypted

Tasks:

1. Install
   a. 'yum -y install telnet-server'

2. Examine Configuration
   a. '/etc/xinetd.d/telnet'

3. Use Telnet Server
   a. 'telnet 192.168.75.21'
   b. 'su ' - switches context to 'root'

Note: Be very careful when using 'su' with 'telnet' due to clear-text exposure of passwords

Note: Loopback connections do NOT traverse the wire. It's ALL virtual (local). It's relatively safe.

c. 'telnet 192.168.75.21' - exposes session to switch-port (network)

4. Disable Telnet Server
   a. '/etc/xinetd.d/telnet' - set 'disable = yes'
   b. 'service xinetd restart'
   c. 'netstat -ntl | grep 23' - confirm whether TELNETD is still listening to: TCP:23
   d. 'netstat -ant | grep 23' - search for stale/existing sockets

Note: TELNETD does NOT facilitate SSH functions/features such as:

1. File Transfers: i.e. 'scp', 'sftp'
2. PKI: i.e. public key/private keypairs
3. Remote commands via command-line (one-off)
4. Pseudo-VPNs

###Network Time Protocol Daemon###

Features:

1. Time synchronization
2. Multiple sources
3. Supports symmetric keys for time sync with other, controlled(trusted), servers
4. Multiple strata are supported in a hierarchy:
   a. Strata range: 1(most accurate)-16(least accurate)

Note: Most accurate means that the stratum level 1 server has access to an external clock (GPS, radio, etc.)

5. NTP will NOT set your system's clock if it is skewed (off) by 1000 or more seconds
6. If '-g' invocation option is used, '1000s' skew is overridden
7. NTP is dynamic in its calculations; always adjusting the values surrounding target NTP servers

Tasks:

1. Explore configuration
   a. '/etc/ntpd.conf' - primary config file
2. Start service
   a. 'service ntpd start'
3. Query NTPD
   a. 'ntpq -np'

Note: Clocks labeled at: stratum 16 are considered unreliable

Note: NTP uses: UDP:123 for source and destination ports

4. Sync Cisco Router
   a. 'sh ntp ass'
   b. 'ntp server 192.168.75.21'

5. Sync Windows Server
   a. 'rdesktop 192.168.75.105'

6. Sync Debian Server with RedHat server & vice versa

Note: Configure NTP to sync with 3 or more clocks

###Add Network Interfaces to Hosts###

Features:

1. On-the-fly NIC provisioning

Tasks:

1. Explore NIC layout on: 'linuxcbtserv2'
   a. 'ifconfig -a' - enumerates detected NICs - named: 'ethn'
   b. 'ethtool eth1'
   c. Explore: '/etc/sysconfig/network-scripts/ifcfg*' - search for device scripts
   c. 'nm-applet' - configure 'eth1' with static address

Note: 'nm-applet' will create: '/etc/sysconfig/network-scripts/ifcfg-eth1' script

Note: This will ensure that the interface is resumed upon reboot/runlevel-switch

2. Explore NIC layout on: 'linuxcbtserv1'
   a. 'ifconfig -a'

Note: The presence of an IPv6 link-local address: 'fe80::' means that the link is connected to another device: i.e. switch, host, etc.

b. 'ethtool eth1' && 'ethtool eth2'
c. 'system-config-network'

d. Enumerate 'ifcfg-eth1' script from both locations:

```
ls -li /etc/sysconfig/{networking/devices,network-scripts}/ifcfg-eth1
```

```
1055028 -rw-r--r--. 3 root root 180 Jan 22 11:24 /etc/sysconfig/networking/devices/ifcfg-eth1
1055028 -rw-r--r--. 3 root root 180 Jan 22 11:24 /etc/sysconfig/network-scripts/ifcfg-eth1
```

Note: Now both: 'linuxcbtser1' and 'linuxcbtserv2' are both configured to allow DHCP configuration on their private subnet

'linuxcbtserv2' - DHCP Server

'linuxcbtserv1' - DHCP Client

Note: Ensure that interface script file contains: 'ONBOOT=yes' directive to ensure that the OS brings the interface up when rebooting (init 6) and/or switching run-levels

###DHCPD###

Features:

1. Auto-configuration of IP client(s)
2. Includes all sorts of settings: IPv4, IPv6, DNS, NTP, NIS, etc.
3. DHCP is an UDP application (UDP:67)

Tasks:

1. Reconfigure 'eth1' to use: '/27'
  a. 'nano /etc/sysconfig/network-scripts/ifcfg-eth1' 'PREFIX=27'

2. Install DHCP
  a. 'yum -y install dhcp'
  b. 'rpm -ql dhcp'

/etc/dhcp - container for DHCPD configuration

/etc/dhcp/dhcpd.conf - IPv4 config

/etc/dhcp/dhcpd6.conf - IPv6 config

/var/lib/dhcpd - container for leases

/var/lib/dhcpd/dhcpd.leases - IPv4 leases

/var/lib/dhcpd/dhcpd6.leases - IPv6 leases

3. Configure scope for: '192.168.76.0/27' - facilitates 2**5 - 2 hosts

192.168.76.0 - Network address

192.168.76.1-30 - Usable

192.168.76.31 - Broadcast Address

Note: Alter DHCPD to log using a different facility: i.e. 'local6' because boot messages are logged via: 'local7'

4. Start/invoke 'eth1' interface on: 'linuxcbtserv1'

Note: This will launch the 'dhclient' process, which will request configuration via DHCP

  a. 'ifup eth1'

```
 inet addr:192.168.76.1  Bcast:192.168.76.31  Mask:255.255.255.224
```

'.224' = '/27'

'/24' = '.0'

'/25' = '.128'

'/26' = '.192'

'/27' = '.224'

5. Configure a reservation to ensure that: 'linuxcbtserv1' is ALWAYS served the same address
  a. 'nano /etc/dhcp/dhcpd.conf'

Note: DHCPD follows the DORA process:

D - Discovery (Client)

O - Offer (Server)

R - Request (Client)

A - Acknowledgement (Server)

###Service Management###

Features:

1. Start|Stop|Adjust runlevels of services
2. Three tools are available
  a. 'chkconfig' - shell
  b. 'ntsysv' - TUI
  c. 'system-config-services' - GUI

Tasks:

1. 'chkconfig' - manages both: 'SYSV' & 'XINETD'
  a. 'chkconfig' - enumerates ALL services
  b. '--list vsftpd' - enumerates runlevel information for service: 'vsftpd'

Note: '/etc/init.d' - services repository

  c. '--level 2 vsftpd off'
  d. '--level 2345 vsftpd off'
  e. 'chkconfig vsftpd on | off' - synonmy for run-levels 2-5
  f. 'chkconfig tftp on' - enables XINETD-controlled service: 'tftp'

Note: XINETD-controlled services are automatically enabled|disabled by 'chkconfig'

Note: However, SYSV-controlled services are NOT automatically started|stopped

Note: Use 'service service_name start|stop' to control service

2. 'ntsysv' - defaults to managing services in the current run-level

Manages both: 'SYSV' & 'XINETD' services

a. 'ntsysv --level 35' - influences ONLY the levels specified on the CLI

Note: 'ntsysv' will NOT change the other, unspecified, run-levels

3. 'system-config-services' - GUI - Manages: 'SYSV' & 'XINETD' services

###BIND DNS###

Features:

1. Standard naming system manager
2. Name-to-IP resolution
3. IP-to-Name resolution
4. Client utilities are auto-installed: 'bind-utils*'RPM
5. Caching-only server
6. Primary server
7. Secondary server
8. Reverse zones
9. IPv6 zones
10. Operates as non-privileged user: 'named'
11. Default configuration binds to: UDP:53 on IPv4|6 loopback (remote queries will fail)
12. Load-balancing is provided in a proper configuration of: 2 or more authoritative servers

Tasks:

1. Explore Caching-only configuration

a. Key files:

'/etc/logrotate.d/named' - logrotate entry

'/etc/named.conf' - zone definition file

'/etc/named.rfc1912.zones' - loopback forward | reverse zones for: IPv4|6

'/etc/rc.d/init.d/named' - INIT script: use with: 'chkconfig' | 'service'

'/var/named' - container for zones: IPv4|6 forward and/or reverse

'/var/named/data' - logfile repository

'/var/named/slaves' - slave-replication data (this server is slave to other server(s))

'/var/named/dynamic' - DDNS

2. Start and Explore Caching-only Server

a. 'chkconfig named on && service named start && ps -ef | grep named' -

b. 'dig @localhost www.linuxcbt.com'

c. Ensure that server binds to ALL IP addresses and allows recursion from ALL

3. Primary Server Configuration - Primary (go-to) authoritative server for a zone

Note: Primary servers tend to have: writable copies of zones, whereas secondary servers tend to have read-only copies of zones due to replication of zone(s) from primary server

a. Define primary zone for: 'linuxcbt.internal'

a1. '/etc/named.conf' - define zone here

a2. '/var/named/linuxcbt.internal' - create zone file with records

a3. 'service named reload' - reload | restart service

a4. 'dig @localhost www.linuxcbt.internal'

```
zone "linuxcbt.internal" IN {
        type master;
        file "linuxcbt.internal";
        allow-update { none; };
};
```

Note: TTLs can be defined:

a. per-file and/or per DNS record

Note: DNS records/zones cached by authoritative servers always reflect the full TTL of the zone/record

b. Extend the primary zone with more records of various types: 'linuxcbt.internal'

c. Add another mail server

d. Define primary zone: 'linuxcbt.external' on host: 'linuxcbtserv1'

```
zone "linuxcbt.external" IN {
        type master;
        file "linuxcbt.external";
        allow-update { none; };
};
```

4. Secondary Server Configuration

Note: Any DNS server can play the role of secondary for one or more zones

a. Make: 'linuxcbtserv1' secondary for the zone: 'linuxcbt.internal'

a1. Define 'linuxcbtserv1' as an NS server in the primary configuration

a2. Setup slave (secondary) zone on: 'linuxcbtserv1'

```
zone "linuxcbt.internal" IN {
        type slave;
        masters { 192.168.75.21;  };
        #file "linuxcbt.external";
        allow-update { none; };
};
```

Note: Above entry caches the zone in-memory:

b. Make: 'linuxcbtserv2' secondary for the zone: 'linuxcbt.external'

Note: Repeat steps above

```
zone "linuxcbt.external" IN {
        type slave;
        masters { 192.168.75.20;  };
        #file "linuxcbt.external";
        allow-update { none; };
};
```

c. Committ changes to master zones

d. Save secondary files to disk

5. Reverse Zones

Resolves: IP-to-Name

a. Write a reverse zone for: '192.168.75.0/24' subnet

```
zone "75.168.192.in-addr.arpa" IN {
        type master;
        file "192.168.75.zone";
        allow-update { none; };
};
```

b. 'dig @localhost -x 192.168.75.21' - returns forward (PTR) names

6. IPv6 Entries: Forward & Reverse Records

a. Insert forward records for connected hosts

```
linuxcbtserv2 IN AAAA 2002:4687:db25:2:211:11ff:fea2:a2d0
linuxcbtbuild1  IN AAAA
        2002:4687:db25:2:211:11ff:fe5b:7053
linuxcbtserv1  IN AAAA 2002:4687:db25:2:211:43ff:fe5a:bce5
linuxcbtrouter1 IN AAAA 2002:4687:DB25:2:21A:2FFF:FEE3:F240
```

b. Query using 'dig' IPv6 AAAA records

b1. 'dig @192.168.75.21 linuxcbtrouter1.linuxcbt.internal AAAA'

Note: Forward: IPv6 records need not be fully expanded

Note: Reverse: IPv6 records MUST be expanded fully when describing the zone

c. Construct Reverse Zone for: '2002:4687:db25:2/64' - Network ID: /64 prefix

```
zone "2.0.0.0.5.2.b.d.7.8.6.4.2.0.0.2.ip6.arpa" IN {
        type master;
        file "2.0.0.0.5.2.b.d.7.8.6.4.2.0.0.2.reverse";
        allow-update { none; };
};
```

Note: ::1 is the IPv6 loopback address, which really means: ALL zeroes terminating with 1

```
5.e.c.b.a.5.e.f.f.f.3.4.1.1.2.0 IN PTR linuxcbtserv1.linuxcbt.internal.
0.d.2.a.2.a.e.f.f.f.1.1.1.1.2.0 IN PTR linuxcbtserv2.linuxcbt.internal.
3.5.0.7.b.5.e.f.f.f.1.1.1.1.2.0 IN PTR linuxcbtbuild1.linuxcbt.internal.
0.4.2.f.3.e.e.f.f.f.F.2.A.1.2.0 IN PTR linuxcbtrouter1.linuxcbt.internal.
```

Note: When writing IPv6 reverse addresses, expand ALL zeroes that are truncated in the addresses.

###Samba - Clients###

Features:

1. Lan Manager/NETBIOS-like support for Linux | Unix

Tasks:

1. Install/Explore Samba Client Package:

a. '/usr/bin/findsmb' - finds Samba hosts on your subnet

b. 'smbtree' - equivalent to 'My Network Places' - Prints workgroups, hosts, and shares

```
WORKGROUP
        \\MACBOOK1                      Dean Davis's MacBook
                \\MACBOOK1\IPC$                 IPC Service (Dean Davis's MacBook)
LINUXGENIUS
        \\LINUXCBTBUILD1                linuxcbtbuild1 server
                \\LINUXCBTBUILD1\lj2100         lj2100
                \\LINUXCBTBUILD1\print$         Printer Drivers
                \\LINUXCBTBUILD1\IPC$           IPC Service (linuxcbtbuild1 server)
AD
        \\LINUXCBT2K8
                \\LINUXCBT2K8\SYSVOL            Logon server share
                \\LINUXCBT2K8\NETLOGON          Logon server share
                \\LINUXCBT2K8\IPC$              Remote IPC
                \\LINUXCBT2K8\C$                Default share
                \\LINUXCBT2K8\ADMIN$            Remote Admin
```

Note: In order to reveal Active Directory shares, you must supply authentication credentials

c. 'smbclient' - Connects to shares and facilitates file transfers - interactive app.

c1. 'smbclient -U administrator //linuxcbt2k8/c$'
Domain=[AD] OS=[Windows Server (R) 2008 Standard 6002 Service Pack 2] Server=[Windows Server (R) 2008 Standard 6.0]

d. 'smbget' - like 'wget' - downloads files from SMB shares, non-interactively

d1. 'smbget -u administrator smb://linuxcbt2k8/temp2/DB_Backup_ALL_messages_tables.only'

e. 'smbtar' - Backs-up SMB shares to TAR archive

e1. 'smbtar -s linuxcbt2k8 -x temp2 -u linuxcbt -t temp2.tar.`date +%F` -p password && gzip -c temp2.tar.`date +%F` > temp2.tar.`date +%F`.gz'

Note: This will create TARball then gzipped file

###Samba Server###

Features:

1. NETBIOS | SMB | CIFS Server
2. Emulates Windows
3. Implemented as 2 daemons: 'nmbd'(NETBIOS naming) & 'smbd'(file serving)
4. Creates one log-file per connected host
5. Linux | Unix security (/etc/{passwd,shadow}) permissions are used to grant access to shares

Tasks:

1. Install 'samba' package
2. Explore default configuration:

a. '/etc/samba/smb.conf' - monolithic configuration file

Note: Within the context of: SELinux, consult: /etc/samba/smb.conf for more information on lifting restrictions

Note: '/etc/samba/smb.conf' - arranged, largely, into 2 sections: global & shares

b. '/etc/samba/smbusers' - Samba Server translation accounts DB. Used when not using AD mode.

3. Change configuration and start service

a. 'nano /etc/samba/smb.conf' - make changes: i.e. default workgroup

b. 'service smb start && chkconfig smb on && service nmb start && chkconfig nmb on'

c. 'netstat -ntlp ' - TCP:139(SMB), TCP:445 (CIFS) are controlled by: 'smbd'

d. 'netstat -nulp' - UDP:137(NMB), UDP:138(NMB) - NETBIOS Naming

4. Implement User Security and test connectivity and ability read/write content

###Winbind Configuration###

Features:

1. Active Directory Integration
2. Precludes the maintenance of multiple user accounts DBs

Steps:

1. Install 'samba-winbind' - 'yum install samba-winbind'
2. Edit: '/etc/security/pam_winbind.conf'
3. Confirm the presence of Kerberos: 'rpm -qa | grep krb5'
4. Edit: '/etc/krb5.conf' - with appropriate ADS realm
5. Edit: '/etc/hosts' - with server information for ADS box
6. Edit: '/etc/nsswitch.conf' - controls default resolver
7. Edit: '/etc/pam.d/system-auth' - general system authentication
8. Edit: '/etc/samba/smb.conf' - include Winbind-related directives

'/etc/samba/smb.conf' - directives

```
security = ads
idmap uid = 10000-20000
idmap gid = 10000-20000
template shell = /bin/bash
template homedir /home/%D/%U
```

8. 'net ads join -U administrator'
9. Start Winbind: 'service winbind start'
10. Configure service to auto-start in SYSV levels: 2-5

a. 'chkconfig winbind on'

11. 'wbinfo -u'

###Apache Configuration###

Features:

1. HTTPD Server

Tasks:

1. Explore the configuration

a. 'rpm -qa | grep httpd'

'httpd-tools' - useful tools

b. '/etc/httpd' - top-level config directory

c. '/etc/httpd/conf/httpd.conf' - main Apache config file

d. '/etc/httpd/conf.d' - add-on configuration files

e. '/etc/logrotate.d/httpd' - managed by LogRotate

f. '/etc/sysconfig/httpd' - startup parameters

Note: Apache launches its initial process as: 'root'

Note: Subsequent Apache processes are launched as: 'apache'

Note: HTTP clients (mobile(iPhone|Droid), browser on the desktop) connect to non-privileged processes running as user: 'apache'

Note: One reason why Apache need 'root' privileges is to be able to bind to well-known ports (<1024)

```
tcp        0      0 :::80                       :::*                        LISTEN      
1818/httpd
tcp        0      0 :::443                      :::*                        LISTEN      
1818/httpd
```

Note: Apache auto-binds to both: IPv4|6

###Apache Logging###

Features:

1. Error: '/var/log/httpd/access_log' - HTTP hits end-up here: 2xx, 3xx(redirects)
2. Access: '/var/log/httpd/error_log' - Errors accessing content: 4xx, 5xx(server problems)
3. Vars are defined in: /etc/httpd/conf/httpd.conf
4. Log Vars are arranged into groups that are reference per virtual host: 'LogFormat'

Tasks:

1. '/etc/httpd/conf/httpd.conf'

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
```

a. '%h' - connecting host's IP address (IPv4|6)
b. '%l' - ident check - typically '-' - not used much anymore
c. '%u' - connecting user - often unknown '-'
d. '%t' - timestamp, day(2-digit)/Month(3 letters/Year(4-digit):Hour:Minute:Second - TimeZone)
e. '%r' - request method (GET/POST/etc.)
f. '%>s' - status code returned to client - 200-500-related errors
g. '%b' - size of content returned to client
h. '%{Referer}' - Contains IP of sending host
i. '%{User-Agent}' - Type of HTTP client: i.e. Droid, iPhone, Safari, IE, Firefox, etc.

```
2002:4687:db25:2:211:11ff:fea2:a2d0 - - 
[26/Jan/2011:09:27:35 -0500] "GET /icons/apache_pb2.gif 
HTTP/1.1" 304 - 
"http://[2002:4687:db25:2:211:11ff:fea2:a2d0]/" 
"Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.9) 
Gecko/20100827 Red Hat/3.6.9-2.el6 Firefox/3.6.9"
```

Note: You may log traffic using multiple LogFormats simultaneously to separate files

###Apache Virtual Hosts###

Features:

1. Two types supported:
   a. IP-Based - one site per IP address
   b. Host Header Names - multiple sites per IP address

Tasks:

1. Configure IP-based Virtual Hosts

Note: The 'default host' is a catch-all for all undefined Virtual Hosts

a. 'httpd -S' - enumerates virtual host(s) configuration

```
<VirtualHost 192.168.75.22>
        ServerAdmin root@linuxcbtserv2.linuxcbt.internal
        ServerName site1.linuxcbt.internal
        DocumentRoot /var/www/site1.linuxcbt.internal
        DirectoryIndex index.ggg
        <Directory /var/www/site1.linuxcbt.internal>
                Order allow,deny
                Allow from all
        </Directory>
</VirtualHost>
```

Note: By not placing a default document, Aapache served us the default page

2. Configure Host-Header Virtual Hosts
   a. 'NameVirtualHost 192.168.75.22:80'

###MySQL###

Features:

1. RDBMS
2. May be administered via: shell, web browser (PHPMyAdmin), or GUI

Tasks:

1. Explore current environment:
   a. 'rpm -qa | grep mysql && yum search mysql'

2. Install MySQL Server
   a. 'yum -y install mysql-server'

3. 'rpm -ql mysql-server'

'/var/lib/mysql' - DATA directory

'/var/log/mysqld.log' - log file

'/var/run/mysqld' - PID directory

4. 'rpm -ql mysql' - enumerates common user-binaries: i.e. 'mysqldump', 'mysqladmin', 'mysql', etc.

'/usr/bin/mysql' - terminal monitor client - facilitates client/server communications interface with MySQLD back-end

5. 'rpm -ql mysql-libs' - reveals: '/etc/my.cnf' - system-wide config file

a. '/etc/my.cnf' - read by clients and mysqld server

6. Start 'mysqld' - ' service mysqld start'

Note: By default, 'root' password is undefined

/usr/bin/mysqladmin -u root password 'abc123'

/usr/bin/mysqladmin -u root -h linuxcbtserv2.linuxcbt.internal password 'new-password'

Note: MySQL represents users as: user@host : i.e. 'root@localhost', 'root@linuxcbtserv2.linuxcbt.internal'

Note: Default configuration permits anonymous connections sans password

7. Change passwords within terminal monitor:

a. 'set password for 'root'@'linuxcbtserv2.linuxcbt.internal' = password('abc123');

b. 'set password for 'root'@'127.0.0.1' = password('abc123');

c. 'flush privileges;' - required after permissions changes

8. Remove anonymous users:

a. 'DELETE FROM mysql.user WHERE user = ''; '

b. 'flush privileges;'

9. MySQL reads a hierarchy of config files upon invocation:

a. '/etc/my.cnf' - system-wide file

b. '$HOME/.my.cnf' - user-wide file

c. Command Line Interface (CLI)

10. Create an addressbook DB:

a. 'create database addressBook;'

b. 'create table contacts ( `fname` char(20), `lname` char(20), `bus_phone1` char(20), `email` char(30), PRIMARY KEY (`email`)   );

c. 'INSERT INTO contacts VALUES ('Dean','Davis','888-573-4943','info@LinuxCBT.com'  );

d. 'INSERT INTO contacts (fname,lname,bus_phone1) VALUES ('Diana','Mckenzie','888-573-4943');

e. 'update contacts set email = 'support@LinuxCBT.com' where fname = 'Diana';'

f. 'delete from contacts where email = 'support@linuxcbt.com'; '

###PHP###

Features:

1. Dynamic web programming/content generation

Tasks:

1. Ensure pre-requisites are in-place

a. 'rpm -qi php-mysql'

b. 'yum -y install php-mysql'

Note: Confirm connection configuration prior to executing script

a. Change script to use routed address

b. Check SELinux booleans

b1. 'getsebool -a | grep httpd' - ensure that HTTPD 'can' connect to 'db'

c. Revert script to use: 'loopback' address after rectifying SELinux problems

d. Confirm whether SELinux vars for 'mysql' influences Apache's ability to source outbound connections to MySQL

d1. The lone 'httpd' variable controls Apache's ability to connect to MySQL

###Network File System###

Features:

1. Transparent access to remote file systems

2. Support for NFS versions: 2(nfs),3(default,nfs),4(nfs4)

3. Supports both: TCP (default) & UDP

4. Relies upon the RPC portmapper service, which dynamically allocates ports

Caveat: Dynamic ports don't always work well with firewalls

5. Auto-transfers UID/GID information from client to server

Tasks:

1. Explore tools like: 'showmount'

2. Start service and explore network stats

a. 'service nfs start && chkconfig nfs on'

b. 'netstat -ntlp ' - search for 'rpc*'

Note: 'rpcbind' - is the RPC manager, which dynamically allocates ports for NFS-related services: quotad, statd, mountd, lockmgr, etc.

3. Export directory to remote clients

a. '/etc/exports' - share directories via NFS here a1. '/projectx *(rw)' - (rw) export to ALL NFS clients that have IP access to our host

a2. 'exportfs -v' - dumps current exports and permissions

a3. 'showmount --exports linuxcbtserv2' - dumps exports of host: 'linuxcbtserv2'

b. Mount '/projectx' on remote system

a. 'mount -t nfs linuxcbtserv2.linuxcbt.internal:/projectx /projectx'

Note: Default mounts are 'root' squashed. This means that when remote clients mount exports, 'root's I/O is equated to: 'nfsnobody' (anonymous)

c. Re-export: '/projectx' as Read-Only

a. 'nano /etc/exports'

###SELinux###

Features:

1. Mandatory Access Controls (MACs)

2. Standard Linux | Unix permissions are based on: Discretionary Access Controls (DACs)

i.e.

-rw-rw-r--. 1 linuxcbt linuxcbt 2129783 Jan  7 17:06 temp.zip

3. A sophisticated labeling system is applied to: subjects & objects

4. Subjects -> users and/or processes

5. Objects -> Files

6. SELinux via MACs: provides a way to separate: users, processes, and objects via labeling and monitors/controls their interaction via: Advanced Vector Cache (AVC)

7. Labels are known as types, which create the silos around: subjects & objects

8. DACs are checked prior to MACs

9. SELinux is enabled in 'enforcing' mode

10. SELinux operates in 3 modes: disabled (DAC), enabled(DAC/MAC), enforcing(DAC/MAC/Enforced)

11. Log information: '/var/log/audit/audit.log' - AVC logs here - Denials

12. Policy information is defined in the: 'targeted' policy

Tasks:

1. Explore common tools

a. 'sestatus -v' - displays current status

b. 'setenforce 0|1(permissive|enforcing) modes'

c. '/etc/sysconfig/selinux' - primary config file

d. '/selinux' - '/proc'-like FS (Virtual) - maintains SELinux information

e. 'setsebool ' - sets boolean values for SELinux - use '-P' to make changes persistent across reboots

f. '-Z' - Use with common commands: i.e. 'ls', 'ps', 'id'

g. Use: 'restorecon -R /var/www/html' - resets ALL files to proper type

Note: 'restorecon' is necessary if files are moved about the FS and have incorrect contexts

2. Switch SELinux mode to: 'permissive' and evaluate with Apache->MySQL

a. 'setenforce 0' - sets SELinux to 'permissive'

b. 'setsebool  httpd_can_network_connect_db off' - disables Apache's ability to talk to MySQL

c. 'setenforce 1' - sets SELinux to 'enforcing'

d. Try to invoke Apache->MySQL session: fails

3. Move and Copy content and evaluate SELinux context changes

Note: Moves will preserve SELinux file (object) context

Note: Copies will NOT preserve SELinux file (object) context. In this case, the object (file) will inherit the SELinux context of the target directory as defined by the SELinux 'targeted' policy.

4. Relabel full FS of remote server

a. 'touch /.autorelable && reboot'

Note: More files means more time to reboot

###NMap###

Features:

1. Port Scanning

2. Host | Device detection

3. Service Detection

4. OS Fingerprinting

5. Multi-target scanning

Tasks:

1. Install 'Nmap'

2. Explore the package

a. '/usr/bin/nmap' - primary binary

b. '/usr/share/nmap/nmap-services' - translates well-known ports to service names

c. '/usr/share/nmap/nmap-protocols' - translates IP protocols to names

3. Use NMap

a. 'nmap -v 192.168.75.0/24'

Note: As 'root' user, 'nmap' executes 'TCP:SYN' scans - half-open connections

Note: As non-privileged user, 'nmap' executes 'TCP:CONNECT' scans - full connections

b. Perform service scan

b1. 'nmap -v -sV target'

Note: Leftmost 24-bits of MAC address represent the vendor, the rightmost 24-bits represent the unique NIC

###IPTables###

Features:

1. IPv4 Firewall - User-space tool
2. Typically manipulates layers 3&4 of the OSI model
   a. Layer-3 - Routing (IPv4 | IPv6) - Source and/or Destination filtering
   b. Layer-4 - Transport (TCP | UDP | ICMP) - Source and/or Destination port filtering

Tasks:

1. Explore the current configuration
   a. '/sbin/iptables' - key binary for managing firewall rules
   b. '/sbin/iptables-restore' - restores rules after reboot and/or flush
   c. '/sbin/iptables-save' - archives current rule-set and counters
   d. 'iptables -L' - enumerates the default table: 'FILTER'

Note: IPTables maintains a number of tables: FILTER (Default), NAT, Mangle

Note: Each table maintains a number of chains.

Note: A chain is simply a list of firewall (filtration) rules

FILTER:

-INPUT - Traffic destined to one of the interfaces governed by the host and sourced by an external host (party)

-FORWARD - Traffic destined to be routed through the host

-OUTPUT - Traffic sourced by OUR host, destined to a remote host

2. Write INPUT chain rules to filter traffic & test
   a. 'iptables -A INPUT -s 192.168.75.105 -p TCP --dport 22 -j DROP'
   b. 'iptables -R INPUT 2 -p tcp --dport 22 -j DROP'
3. Write OUTPUT chain rule to restrict outbound TCP:25
   a. 'iptables -A OUTPUT -p tcp --dport 25 -j DROP'

###IP6Tables###

Features:

1. Management of IPv6 filtering

Tasks:

1. Explore configuration
   a. '/sbin/ip6tables ' - primary binary
2. Usage
   a. 'ip6tables -L'

Note: With both IPv4 & IPv6, the default policy is 'ACCEPT', which may be switched to: 'DENY', which will require explicit rules allowing traffic

3. Write IPv6 Rules
   a. 'ip6tables -A INPUT -p tcp --dport 22 -j LOG --log-level debug'
   b. 'ip6tables -A INPUT -p tcp --dport 22 -j DROP'

###TCPDump###

Features:

1. Packet Capturing
2. Layers 2-7 of OSI
3. Driven by Three Qualifiers
   a. Type - host|net|port
   b. Dir - src, dst, src or dst, src and dst
   c. Proto - ip, tcp, udp, arp, etc.
4. Supports BPFs
5. Uses promiscuous mode to intercept traffic not bound for local system

Tasks:

1. Explore configuration
   a. '/usr/sbin/tcpdump'

2. Usage
   a. 'tcpdump -v ' - dumps traffic to STDOUT

```
13:48:06.854768 IP (tos 0x0, ttl 64, id 34654, offset 0, flags [DF], proto TCP (6), length 1500)
   linuxcbtserv2.linuxcbt.internal.5902 > 192.168.75.14.63276: Flags [.], cksum 0x3402 (correct), seq 27453037:27454485, ack 615, win 108, options [nop,nop,TS val 446941871 ecr 385595866], length 1448
```

   b. 'tcpdump -i eth0' - binds to indicated interface
   c. 'tcpdump -D ' - enumerates the interfaces
   d. 'tcpdump -i eth0 -w filename'
   e. 'tcpdump -r tcpdump.full.log.2011-01-28'
   f. 'tcpdump -e tcpdump -r tcpdump.full.log.2011-01-28' - dumps link-level header - L2

```
13:56:09.105343 00:25:4b:a9:ba:3e (oui Unknown) > 00:11:11:a2:a2:d0 (oui Unknown), ethertype IPv4 (0x0800), length 66: 192.168.75.14.63276 > linuxcbtserv2.linuxcbt.internal.5902: Flags [.], ack 9101188, win 65535, options [nop,nop,TS val 385600686 ecr 447424118], length 0
```

g. 'tcpdump -A -r tcpdump -r tcpdump.full.log.2011-01-28' - L3-L7

h. 'tcpdump -e -A -r tcpdump -r tcpdump.full.log.2011-01-28' - dumps L2-L7

i. 'tcpdump -n -e -A -r tcpdump -r tcpdump.full.log.2011-01-28' - dumps L2-L7, suppresses name resolution (hosts and/or services)

3. Use BPFs to filter traffic

a. 'tcpdump -w tcpdump.bpf.sans.vnc.1 not port 5902'

b. 'tcpdump -w tcpdump.bpf.sans.vnc.1 not tcp and port 5902'  - Filters all but TCP and TCP:5902

c. 'tcpdump -w tcpdump.bpf.sans.vnc.1 not tcp port 5902' - Filters out TCP:5902

###Apache SSL/TLS###

Features:

1. Secure communications for web services

2. TCP:443 - https

3. Multiple SSL/TLS sites can be bound to the same IP address so long as you use distinct TCP ports. i.e. TCP:443, TCP:4443, TCP:444

4. SSL/TLS will read both: private and public (certificate) keys from the same file

Note: Simply reference the same file with private and certificate directives

Requires:

1. HTTPD - Apache

2. 'openssl' - SSL/TLS library

3. 'mod_ssl' - Apache Module

4. 'crypto-utils' - includes 'gen-key'

Tasks:

1. Exploration of current setup

a. 'rpm -ql mod_ssl'

/etc/httpd/conf.d/ssl.conf - first virtual host, and, default SSL server

/usr/lib/httpd/modules/mod_ssl.so - SSL/TLS Module

b. 'rpm -ql crypto-utils'

'/usr/bin/genkey' - useful in generating various types of certificates: i.e. self-signed, CSRs, etc.

c. 'rpm -ql openssl'

'/etc/pki' - hierarchy of public key encryption files

'/usr/bin/openssl' - key OpenSSL binary used to generate certificates, etc.

2. Explore the default SSL site

a. '/etc/httpd/conf.d/ssl.conf'

3. Use 'tcpdump' to enumerate clear-text and SSL/TLS-protected traffic

a. 'tcpdump -vv -Ae tcp port 80 or 443'

b. 'curl http://192.168.75.21' - initiates HTTP clear-text communications

c. 'curl -k https://192.168.75.21' - initiates HTTPS encrypted communications

4. Generate new usage keys for default site

a. 'genkey linuxcbtserv2.linuxcbt.internal'

5. Update: '/etc/httpd/conf.d/ssl.conf' with new SSL keypair

a. Replace cert/private key lines with pointers to new files

6. Generate usage keys for: 'site1.linuxcbt.internal'

a. '/etc/pki/tls/certs/make-dummy-cert' - works faster than 'gen-cert'

###VSFTPD with SSL###

Features:

1. Implicit SSL -> TCP:990

2. Explicit SSL -> TCP:21

3. Encryption of:

a. Control Channel

b. Data Channel

Tasks:

1. Explore Current Configuration

a. Use LFTP to force SSL connection

'~/.lftprc'

'set ftp:ssl-force yes'

'set ftp:ssl-protect-data yes'

2. Use 'tcpdump' to sniff clear-text traffic

3. Setup VSFTPD server with SSL support

a. 'ssl_enable=yes' - This will require local logins (non-anonymous users) to use SSL/TLSv1

b. 'ssl_tlsv1=yes' (Default)

c. 'rsa_cert_file=/etc/pki/tls/certs/linuxcbtserv2.linuxcbt.internal.crt' - This will allow VSFTPD to read both: private & public keys from the same file

d. 'rsa_private_key_file=/etc/pki/tls/certs/linuxcbtserv2.linuxcbt.internal.key' - Set this if the private key exists in a separate file

e. 'openssl ciphers -v'

Defaul Cipher: 'DES-CBC3-SHA'

'openssl ciphers -v | grep 'DES-CBC3-SHA'

DES-CBC3-SHA SSLv3 Kx=RSA Au=RSA Enc=3DES(168) Mac=SHA1

f. 'service vsftpd restart' - restart for SSL settings to take effect

4. Test SSL/TLS connectivity from various FTP clients

a. 'lftp linuxcbt@localhost' - this will generate a certificate mismatch

b. 'lftp linuxcbt@linuxcbtserv2.linuxcbt.internal' - this works

5. Test clear-text FTP connection

a. 'nano ~/.lftprc'

6. Configure VSFTPD to support both: SSL/TLS and Clear-text connections

a. 'force_local_logins_ssl=no'

b. 'force_local_data_ssl=no'

7. Windows with FileZilla

a. Try both clear-text and FTP Explicit SSL connections

###Tighten Configuration###

Features:

1. Improves your security posture

2. Publish only necessary services

3. Reduces risk/exposure to mal clients

Tasks:

1. Identify IPv4 unnecessary addresses

a. 'ifconfig -a'

b. 'eth0:1' & 'eth0:2'

c. 'ifcfg eth0:1 del 192.168.75.22 && ifcfg eth0:2 del 192.168.75.23'

2. Disable: 'eth1'

a. 'ifcfg eth1 stop'

3. Reconnaissance Scan

a. 'nmap -v -sS -sU localhost'

```
PORT     STATE         SERVICE
21/tcp   open          ftp
22/tcp   open          ssh
25/tcp   open          smtp
53/tcp   open          domain
80/tcp   open          http
111/tcp  open          rpcbind
139/tcp  open          netbios-ssn
443/tcp  open          https
445/tcp  open          microsoft-ds
514/tcp  open          shell
2049/tcp open          nfs
3306/tcp filtered      mysql
4443/tcp open          pharos
5902/tcp open          vnc-2
53/udp   open          domain
67/udp   open|filtered dhcps
69/udp   open|filtered tftp
111/udp  open          rpcbind
123/udp  open          ntp
137/udp  open          netbios-ns
138/udp  open|filtered netbios-dgm
514/udp  open|filtered syslog
2049/udp open          nfs
5353/udp open|filtered zeroconf
```

4. Define system baseline

a. SSHD

b. HTTPD

c. DNS

d. SYSLOGD

e. NTPD

f. FTPS - Explicit-mode FTP w/SSL/TLS

g. MySQL - bound to loopback

h. VNC

i. SMTP - bound to loopback - Default

4. 'netstat -ntulp' - enumerate TCP & UDP listeners

5. Bind MySQL to: loopback

a. 'nano /etc/my.cnf'

b. 'bind=127.0.0.1'

c. 'service mysqld restart'

6. Disable 'rpcbind'

a. 'service rpcbind stop && chkconfig rpcbind off && chkconfig --list rpcbind'

b. 'netstat -ntlp | grep 111'

7. Disable 'NFS'

a. 'service nfs stop && chkconfig nfs off && netstat -ntlp | grep 2049'

8. Disable 'Samba'

a. 'service smb stop && chkconfig smb off && netstat -ntlp | grep 445'

b. 'service nmb stop && chkconfig nmb off && netstat -nulp | grep 137'

c. 'service winbind stop && chkconfig winbind off'

9. Disable 'DHCPD'

a. 'chkconfig dhcpd off && service dhcpd stop'

10. Disable 'TFTPD'

a. 'chkconfig tftp off' - this disables & stops the XINETD-controlled service

11. Configure VSFTPD to use SSL/TLS ONLY

a. 'nano /etc/vsftpd/vsftpd.conf'

b. 'force_local_logins_ssl=yes'

c. 'force_local_data_ssl=yes'

d. Use 'lftp' to confirm that VSFTPD permits SSL/TLSv1 connections ONLY

e. Ensure that LFTP is configured to NOT use SSL to see whether or not the server will permit non-SSL/TLSv1 connections

12. Restrict SSHD to users: 'root' & 'linuxcbt'

a. '/etc/ssh/sshd_config'

b. 'AllowUsers root linuxcbt'

c. 'service sshd restart'

d. Test SSH connectivity as allowed and disallowed users

13. Restrict SSHD to non-privileged user: 'linuxcbt' & 'linuxcbt2'

a. 'AllowUsers linuxcbt linuxcbt2'

14. Post-Reconnaissance Check

a. 'nmap -v -sU -sS localhost'

b. 'nmap -v -sU -sS 192.168.75.21' - execute from a remote host

c. 'nmap -v -6 2002:4687:db25:2:211:11ff:fea2:a2d0' - execute IPv6 remote reconnaissance