Enterprise Linux for Government

Administer & Secure
Red Hat Family Linux
version 7

John Timaeus & Russell Overton

JANUS Technical Academy

600 Ponder Place Dr
Evans, GA 30809
(706) 364-9100

LABS

Lab 1: Initial Configuration

1. Log into your lab machine.
2. Install **`bash-completion`**.
3. Logout and log back in to activate bash-completion.
4. Using redirection (**`>`**, **`>>`**, **`|`**) create a file named **`IP-info`** which contains the output of the following commands: **`ip a`**, **`ip route`**, **`ip neigh`**.
5. From the contents of **`IP-info`** create a new file named **`192-info`**. It should only have lines containing the string **`192`**.
6. Examine the man pages for: **`cp`**, **`mv`**, **`ln`**, and **`history`**.
7. Create a file called **`TZ-man`** which contains a list of all the man pages relating to timezone.
8. Clear your history list by deleting all entries.
9. Copy **`IP-info`** to **`IP-old`**; rename **`192-info`** to **`192-old`**. Do this on one line.
10. Insert your history at the bottom of **`192-old`**.
11. Make a directory (**`mkdir`**) called **`192-zzz`**
    a. **`cat 192-*`** – this will return the contents of **`192-old`**, and an error message because cat can't read a directory.
        i. Create a file **`err-192`** that contains only the error from this command.
        ii. Run the command again, displaying only the error.

Update all software in the background, hiding the output.

Lab 2: Files and Directories

Finding Things

For the following exercises, unless otherwise specified, you may use any commands you choose to determine the answers. These might be helpful: **`locate`**, **`whereis`**, **`which`**, and **`find`**.

Record answers in a file **`/lab/02-Files/results`**.

12. Find and record the location of the **`cp`** command.
13. Find and record the location of the man pages for the **`mv`** command.
14. Find all instances of a file called **`findme.script`**.
    a. How many did you find?
15. How many instances of this script are in root's **`$PATH`**?
16. Which version runs when root calls **`findme.script`** without a path? Record the location.
17. Create a file in **`/lab/02-Files/`** called **`mxyzptlk`**. Use locate to locate it. Did it work? If not, fix it.

Finding Things with find

In this section, use the **`find`** command to answer the questions. Put all output in **`results.`**

Tip: putting **`2>/dev/null`** after a query will suppress errors from the **`/proc`** file system.

18. The **`/etc`** directory contains directories and files that begin with **`rc`** and end with **`.d`**. Find only the directories.
19. Find any unowned files on your system.
20. Find any files with the SUID bit set which ARE ALSO writable by others.
21. Find any files which haven't been accessed in more than 25 years (they do exist!).
22. Use a single line command to locate and optionally remove the **`/lab/02-Files/killme`** directory and all its contents. Do be careful.
23. Find any file in **`/boot`** greater than 20 megabytes in size.

Manipulating Files and Directories

In this section, we will create, manipulate, and destroy files and directories. You may use any commands you like to achieve these goals.

24. Create a directory in **`/lab/02-Files/`** called **`new`**.
25. Change directories to **`new`**.
26. Create a file named **`file1`**. Put the phrase "this is file 1" inside of it.
27. Copy **`file1`** and call the copy **`file3`**.
28. Rename **`file3`** to **`file2`**.
29. Create an empty file called **`file3`**
30. Create a directory in **`/lab/02-Files/new`** called **`subfiles`**
31. Use a single command to copy all files in the **`/lab/02-Files/new`** directory into **`subfiles`**.
32. Move **`file1`** up one level using dot notation.
33. Without using **`vi`**, read **`file1`** and append its contents to **`file3`**.

Lab 3: vi

34. **vi /lab/03-vi/edit-me**, follow the directions in the file.
35. When complete check your work:
    # diff edit-me edit-me.finished | grep "<" .
    The result should be just the date and your name

36. Set the hostname on both your computers in **/etc/sysconfig/network.** Reference the student handout for naming information. Reboot to make the change effective across all sessions.
37. Set a static IP address for both computers:
    a. Find your active network card using **ip a | grep 192** .
    b. Make a backup of the appropriate file
    cp /etc/sysconfig/network-scripts/ifcfg-ethX /etc/ifcfg-ethX.bak (where ethX is your active network card)
    c. **vi /etc/sysconfig/network-scripts/ifcfg-ethX**
    Use the static addressing information provided on your student sheet. It should look something like this:

```
DEVICE=eth2
TYPE=Ethernet
ONBOOT=yes
NM_CONTROLLED=no
BOOTPROTO=static
IPADDR=192.168.1.160
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
DNS1=192.168.1.251
DNS2=192.168.1.252
```

    d. Restart the network service: **service network restart**
    e. Start a session to the new IP address. Be patient, initial ssh setup may take a few seconds longer than expected.
    f. Remove your backup files from **/etc/** .

Lab 4: Users and Groups

1. Configure your computer such that users are created with:
    a. default password expiration of one month from now
    b. immediate inactivation on password expiration
    c. password aging fields and logon failure delay meets standards
    d. Do not set minimum password length or quality requirements at this point.
2. Enable **`wheel`** in **`sudoers`**.
3. Configure **`/etc/pam.d/su`** to restrict use of **`su -`** to members of **`wheel`**.
4. Edit **`/etc/skel/.bash_profile`** to contain this line
   `script -c "/usr/bin/screen -RL" /dev/null`
5. Create the following users as specified.
   Unless otherwise stated, all should have a primary group of their own.
   All users should have a password set.
   Usernames should be **all lower case**.
    a. Adam and Brenda – regular users
    b. Don and Emma – can use **`sudo`**, members of group **`dev`**.
    c. Frank – cannot login interactively, with a comment noting that fact.
    d. Grace – **`UID = 3001`**, member of **`dev`**, **`helpdesk`**, and **`recruiting`**
    e. Harry – **`UID = 3002`**, member of **`recruiting`**, lock his account.
    f. Jane and Mary – members of **`recruiting`** and **`restricted-users`**
    g. Nick – member of **`helpdesk`**.
6. Try to log in as Frank.
7. Set Harry's home directory to **`/lab/05-Permissions/recruiting`**.
8. Add Grace to group **`wheel`**.
9. Change Brenda's shell to **`vi`**. Change to her environment.
    a. Attempt to read the date into a file – what caused the failure?
10. Check your work by viewing the relevant files in **`/etc/`**.

Lab 5: Ownership, Permissions, and Access

38. Preparation:
    a. Ensure that **grace** is a member of **helpdesk**, but NOT **recruiting**.
    b. Ensure that **harry** is a member of **recruiting**, but NOT **helpdesk**.
39. Setting Ownership and Basic Permissions
    a. As root, create the following directories: **/lab/05-Permissions/recruiting**, **/lab/05-Permissions/helpdesk**
    b. Assign ownership of **recruiting** to user **harry** and group **recruiting**.
    c. Assign ownership of **/lab/05-Permissions/helpdesk** to user **grace** and group **helpdesk**.
    d. As **harry**, set the permissions of **recruiting** to **770**.
    e. As **harry**, set the access mode of **recruiting** so that new files in that directory will automatically be owned by the **recruiting** group.
    f. As **grace**, set the permissions of **helpdesk** to **770**.
    g. As **grace**, set the access mode of **helpdesk** so that new files in that directory will automatically be owned to the **helpdesk** group.
    h. As **grace**, attempt to access the **recruiting** directory. This should fail.
    i. As **harry**, attempt to access the **helpdesk** directory. This should fail.
40. Extended ACLs – The order in which you do these tasks matters, as some ACL commands may overwrite previous entries. Review the exercise first, and plan your ACLs before applying them.
    a. Create a file in recruiting called **rfile**. It should contain the line, "this is rfile."
    b. Create a file in helpdesk called **hfile**. It should contain the line, "this is hfile."
    c. Set an extended ACL on **recruiting** that will allow members of the **helpdesk** group to list the contents of the directory and read the files inside of it.
        i. Grant full control of any files created in this directory to **grace**.
        ii. Others should have no access.
    d. Set an extended ACL on **helpdesk** that will allow members of the **recruiting** group to list the contents of the directory and read the files inside of it.
        i. Grant full control of any files created in this directory to **harry**.
        ii. Others should have no access.
    e. Mask away execute permissions for any files created in these directories.
    f. Ensure that these ACLs are applied recursively to the existing files in the directories.

41. Testing SUID

a. Verify that all executables in **`/lab/05-Permissions`** are SUID/SGID root:root

b. **`su - adam`**.

c. Run **`who`**, **`id`**, and **`whoami`**. Note the results.

d. **`cd /lab/05-Permissions`**.

e. Run **`who`**, **`id`**, and **`whoami`** again. Note the results.

f. Run those three commands once again, this time with a **`./`** preceding each.

g. Add **`.`** to the beginning of Adam's path. Run the commands one last time, this time without the leading **`./`**

Lab 6: Regular Expressions

You will find the files for this exercise in **/lab/06-regex**.

42. Copy **something.com.zone** to **else.com.zone**.
43. Copy **something.named.conf** to **else.named.conf**.
44. Take a moment to read over these two files and familiarize yourself with their contents.
45. Display any lines in **else.com.zone** which contain IP addresses.
    a. Place the result in **else.ip**.
46. Display any lines in **else.com.zone** which contain hostnames.
    a. Place the result in **else.host**.
47. Repeat these tasks for the else.named.conf file.
    a. Append the results to **else.ip** and **else.host**.
48. Your network has been renumbered from **192.168.10** to **10.10.10**. Use **sed** to make the appropriate changes IP addresses in the **else.com.zone** and **else.named.conf** file.
49. You have a new domain. Your old domain **something.com** is being replaced with **else.com** Use **awk** to make the appropriate changes in both files.
    a. This includes both hostnames and filename references.
    b. DO NOT change any references to domains other than **something.com**.
    c. Do not alter any portion of a hostname OTHER THAN the something.com domain.
50. Take a moment to review your work. Compare the new files to the originals. Did you miss anything? Did you accidentally change anything you shouldn't have? If you did, resist the urge to hand correct this with **vi**. Try to use **sed** or **awk** to fix mistakes.

Pointless Fun

If you finished early, and you need something to do, try this:

11. Copy **/etc/passwd** to **/lab/06-regex/passfile**
12. Use **sed** to remove all lines in **passfile** that start with **a** and save the file in place.
13. Use **diff** to compare **passfile** and **/etc/password**.
14. Display **passfile**, sorted alphabetically, saving a copy to **alphapass** in one line.
15. Display **alphapass** to your screen, but replace all the colons with linebreaks.
16. Repeat the last step, but eliminate duplicate lines.
17. Repeat the last step, but only show lines that start with a slash.

Lab 7: File Systems

Throughout this lab, you will be making extensive changes to disk structures. Please remember to check your work and ensure that the kernel is aware of your changes. If it is not, provoke rescans of your SCSI bus using the methods supplied in the course manual.

51. Partitioning with **`fdisk`**
    a. You should have an empty hard drive attached to your system. Identify it.
    b. On your free disk, create 3 partitions of 1 GB each.
    c. Ensure that the partition table is updated to reflect your work.
52. File System creation and mounting
    a. Put an ext4 filesystem on your first empty partition.
    b. Add ACL support to this filesystem.
    c. Create a mountpoint called **`/mount1`**.
    d. Mount the filesystem to **`/mount1`**.
    e. Check it.
    f. Unmount the filesystem
    g. Configure **`/etc/fstab`** to automatically mount your new filesystem at boot or by mountpoint. Use default options.
    h. Mount your new filesystem with: **`mount /mount1`** .
    i. Check your work.
53. LVM
    a. Create a physical volume from your second partition.
    b. Create a volume group containing only that physical volume.
    c. Create a logical volume that uses 100 percent of the free space in that volume group.
    d. Create an ext4 filesystem on that logical volume.
    e. Add ACL support to the new filesystem.
    f. Create a mount point **`/mount2`** and mount the filesystem to it.
    g. Unmount it.
    h. Put it in **`/etc/fstab`**.
    i. Mount it.
    j. Pretend the filesystem is full. Use the 3rd free partition to extend the volume group, logical volume, and filesystem.

54. Hard Links

a. Create a file in **/lab/07-Filesystems/** named **original**, with content: "This is the original file."

b. Create a hard link to the file in the same directory. Call the link **copy**.

c. Get a list of inodes for all files in **/lab/07-Filesystems**. What are the inode numbers for **original** and **copy**? Record the inode of **original**.

d. Read the contents of **copy**. What does it say?

e. Add a line to **copy** that says, "edited from copy."

f. Read the contents of **original** to the screen. What does it say?

g. What is the link count for **original**? What is it for **copy**?

h. Delete **original**.

i. What is the link count for **copy**?

55. Symbolic Links

a. Create a symbolic link to **copy** in the same directory. Call the link **original**.

b. What are the inodes of the files? What does a long listing tell you about **original**?

c. Move **original** to **/mount1/original**.

d. Add a line to **original** that reads, "edited from symbolic original".

e. Read **copy**. What does it say?

f. What is the link count for **copy**?

g. Delete **copy**. Attempt to read **/mount1/original**. What happened?

h. Create a file in **07-Filesystems** called **copy**.

i. Add a line to it that reads, "this is not the same file."

j. Attempt to read **/mount1/original**. What happened?

k. Delete **/mount1/original**.

l. Read **copy**. Did deleting **/mount1/original** have an effect on **/lab/copy**?

m. Make a hard link from **/lab/07-Filesystems/copy** to **/lab/07-Filesystems/original**.

n. Delete **/lab/07-Filesystems/copy**. What are the contents of **original**? What is the inode and link count? Does the inode for original match the one recorded earlier in this lab?

Lab 8: Processes and Services

56. Preparation
    a. Open at least three connections to your server. You may even want more.
    b. Run **top** in one session…you will want to keep **top** open throughout this lab.
    c. All scripts are in **/lab/08-Process** and are described in; alter permissions as needed.
    d. Read through the instructions before executing any given step.
    e. Verify who you are before running **memmy**. If run as **root**, it **will** break your box.

57. Pause **top**; open **vi**, and **man top** – in the same window.
    a. Toggle between them using **jobs** and **fg**.

58. Run **chew**, this will execute **bc** (a calculator) – using quite a bit of cpu. Note its PID.
    a. Using the signal passing function of **top**, pause and resume the **bc** process.
    b. Background the **bc** process from the command line that originated it.
    c. **disown** it and log out. Did it continue running?

59. Start **spread**, this will run the command **bc** backgrounded at varying nicenesses.
    a. Create a file with the PIDs of the **bc** process in it
    b. Pause all of them except the nicest.

       If you get really stuck on this there are hints in **/lab/08-Process/killer**.
    c. Resume the 5 least nice.
    d. Clean up all the **bc** processes before proceeding.

60. Start a **watch** for processes named **sleep**. In a separate window **watch** for processes belonging to **jane**.
    a. Run **nappy** as **root**, Jane, and Mary. This starts multiple **sleeps**.
    b. Run **spread** as Jane.
        i. What is different when Jane runs **spread**?
    c. Kill all of Jane's **sleep** processes, without affecting any others.
    d. Kill all of Mary's processes.
    e. Clean up the **bc** and **sleep** processes before proceeding.

61. Give Mary a hard limit of 1 minute cpu. Set hard limits on Nick for **`nproc = 5000`**, **`nice and priority =15`**.
    a. As Mary, run **`chew`**. Observe what occurs when processor time hits 60 seconds.
    b. As Nick, run **`spread`**. Notice the priorities.
    c. As Nick run **`forkbomb`**.
    d. Clean up all of Nick and Mary's processes.

62. Start a **`vmstat -a -S M 5`**. This will display memory information in MiB.
    a. Run **`swapoff -a`**, watch the swap file drain. Confirm that swap was disabled using **`swapon --summary`**.
    b. In a new window prepare to issue **`pkill grep`**, do not pressenter yet.
    c. As Mary, run **`memmy`**.
        i. Observe the change in memory usage.
        ii. As soon as **`Out of memory`** errors begin, **`pkill grep`** .
        iii. Near the end of **`/var/log/messages`**, search for **`oom-killer.`** Then read down from there.
    d. Re-enable swap.
    e. Run **`memmy`** again as Mary. Observe the differences.
    f. To ensure that no rogue processes are left over, reboot.

Lab 9 : Scheduling Events

63. All tasks for this lab are in **`/lab/09-Events`**
    a. Set ownership and permissions such that Nick and Mary can run these tasks.
64. Allow Mary and Nick the ability to create at and cron jobs.
65. Using **`at`** have Mary schedule
    a. **`at.task1`** five minutes from now
    b. **`at.task2`** a few minutes after midnight tonight
    c. **`at.task3`** an hour from now and Friday afternoon
    d. **`at.task4`** next Tuesday at 11 am
    e. View the jobs.
    f. Remove the job which would run **`at.task3`** Friday afternoon.
66. Using **`cron`** have Nick schedule
    a. **`cron.job1`** weekly beginning ten minutes from now
    b. **`cron.job2`** at 2130 on odd numbered dates during the week
    c. **`cron.job3`** at the most infrequent interval possible
    d. **`cron.job4`** at 1130 on weekdays
67. As **`root`**, use **`anacron`** and a symbolic link to run **`cron.job5`** weekly
68. Lower the **`batch`** threshold to **`0.01`**
    a. Run **`chewy`** from the previous lab
    b. Create a batch job that will **`wall`** "Done with this lab!"
    c. End the **`bc`** processes started by **`chewy`**

Lab 10: Booting

69. Make a copy of **`grub.conf`**
70. Alter the copy to:
    a. Have a timeout of 15
    b. Not hide the menu
    c. Have a new boot entry called **`Other`** which
        i. Displays all boot messages
        ii. Has SELinux disabled
71. When done have an instructor verify your changes.

Note: there is no Lab 11.

Lab 12: SELinux

We'll be working with Apache and SELinux in this lab. The httpd daemon should already be installed, and the configuration files on your machines should already be altered to allow the labs to function as written. **You should only need to make changes of an SELinux nature to accomplish the stated goals**.

72. Using Booleans
    a. Become user **grace**.
    b. In **grace**'s home directory, create a new directory called **public_html**.
    c. Set the permissions on this directory to **755**.
    d. Create a new file in **~/public_html** called **index.html**. The contents of **index.html** should be a single sentence, reading, "Successfully viewed the file via home dirs."
    e. Set the permissions of **index.html** to **744**.
    f. Become **root**.
    g. Temporarily disable the firewall: **service iptables stop**. In later modules, we will learn how to properly configure firewall rules. For now, this is sufficient.
    h. Attempt to access this address: **http://***hostname***/~grace/**
    i. This should fail. Take any steps necessary to resolve the issue.
73. Setting a Security Context on Files and Directories
    a. Create a directory, **/web** with permissions of **755**
    b. In **/web** create **index.html**, with permissions of **744** and content "Successfully viewed the file in /web."
    c. Edit **/etc/httpd/conf/httpd.conf**. Change the following lines to reflect the new Document Root of **/web**:

```
DocumentRoot "/var/www/html"

<Directory "/var/www/html">
```

    d. Restart the httpd service.
    e. Attempt to access this address: **http://***server-name***/**
    f. What happened? Did you get the results you were expecting?
    g. Take any steps necessary to resolve this issue.

74. Altering Security Contexts for Ports

    a. Edit **`/etc/httpd/conf/httpd.conf`**. Change the line **`Listen 80`** to **`Listen 8000`**.

    b. Restart the httpd service.

    c. Attempt to access **`http://`***`server-name`***`:8000`**

    d. What happened? Were these the results you were expecting?

    e. Take any steps necessary to resolve the issue.

75. Turn your firewall back on, shut down **`httpd`**, and check that SELinux is enforcing.

Lab 13: Networking

76. Confirm that the NetworkManager service is uninstalled.
77. Configure your networking control files:
    a. Configure **`/etc/resolv.conf`** to search the namespace VM.NOT.
    b. Ping the other student's machines by FQDN and by single name.
    c. Create a file called **`/lab/13-Network/netinfo`** with your neighbors hostnames, ip addresses, and MAC addresses thus:
    `duh.vm.not – 192.168.1.48 – 00:50:56:87:49:49`.
78. Find the name server (**`NS`**) and mail exchanger (**`MX`**) names and addresses for **`redhat.com`**, and **`doe.gov`**. Append this to **`netinfo`**.
79. Using **`ss`**, start a **`watch`** on all tcp ports.
    a. **`ssh`** to yourself, then **`ping`** yourself. Why is there a difference in the **`ss`** output?
    b. Run **`ss`** by itself (without **`watch`**), modifying the results to exclude listening ports and connection from **`::1`**, and include process information.
    Append this to **`netinfo`**.
80. Getting to STIG – Pay special attention to the order of the rules in your chains.
    a. Create an **`at`** job to disable the firewall 20 minutes from now. If you haven't locked yourself out, remove and renew the job every now and then. If you do lock yourself out, it will let you back in when it runs.

       Note: the most common method of locking yourself out is by creating a default **`DROP`** with no other rules configured.
    b. Configure your **`INPUT`** chain to have a default policy of **`DROP`**. Remove or replace any rules in the **`INPUT`** chain which would interfere with the default **`DROP`** action.
    c. Configure your **`FORWARD`** chain to have a default policy of **`DROP`**. Remove or replace any rules in the **`FORWARD`** chain which would interfere with the default **`DROP`** action.
    d. Ensure that the iptables service is running.
    e. Set iptables to start on boot at run levels 2,3,4, and 5.

81. Opening and Closing Ports for httpd
    a. Allow https on TCP port 443 inbound from anywhere, for **`NEW`**, **`ESTABLISHED`**, and **`RELATED`** connections.
    b. Block http on TCP port 80 inbound from anywhere.
    c. Test these rules from your windows box.
82. Allowing NFS
    a. NFS v4 requires TCP port 2049. Allow new incoming connections to TCP 2049 ONLY from the IP of your windows box (windows IP can be determined by running **ipconfig** from command line).
83. Allowing VNC Server
    a. VNC server requires base TCP ports of 5800, 5900, and 6000 to be open. Configure rules to support this.
    b. VNC also requires sequential ports to be opened per user session. To support two user sessions, we will need the following tcp ports opened: 5801, 5802, 5901, 5902, 6001, and 6002. Configure rules to accept this traffic.
84. Finishing Up
    a. Issue any commands required to save your iptables configuration. Restart the iptables service.

Note: There is no Lab 14.

Lab 15: Remote File Systems

85. Initial Configuration of NFS Server and Client

    a. In **/etc/sysconfig/nfs** set
    RPCMOUNTDOPTS="-N 2 -N 3"

    b. In **/etc/nfsmount.conf** uncomment the following:
    # Defaultvers=4
    # Nfsvers=4
    # Defaultproto=tcp
    # Proto=tcp

86. Making the Shares

    a. Create a directory: **/hdshare**

    b. Set ownership to **grace:helpdesk** and permissions to **2775**.

    c. Create a single file in this directory. Name it **heybuddy**. Include a message for your partner.

    d. Share **/hdshare** to your partner with read/write access, and default options.

87. Mounting and Using NFS

    a. Create a local directory: **/netshare**

    b. Mount your partner's shared directory to **/netshare**.

    c. As root, attempt to access the share. Attempt to read and edit **heybuddy**.

    d. As grace, attempt to access the share. Attempt to read and edit **heybuddy**.

    e. Unmount the share.

    f. Add the share to **/etc/fstab**.

    g. Mount it using the local mount point name only.

Lab 16: Remote Access

88. Provide vnc service for Adam and Grace.
    a. Log into your machine using tigervnc supplied in the **`/labs/16-Remote`** directory as both Adam and Grace.
    b. Verify that access persists over a reboot, without needing to log in via ssh first.
89. Secure sshd, provide a login banner.
    a. Work with one of your neighbors to ssh without password as Grace and root from one machine to the other.
    b. Use scp to copy the **`netinfo`** file you generated earlier to **`/labs/16-Remote`** on your neighbors machine.
90. Restrict Adam to **`sftp`** only.
    a. Create a chrooted sftp home directory for him.
    b. Move a copy of your local **`/etc/hosts`** to Adam's home directory on your neighbor's machine.
91. Create a file showing all nonroot logins which have occurred on your computer. Make it available via **`http`**.
    a. Get a copy of a classmates file, without using a graphical browser.

Lab 17: Kernel Modules and Parameters

92. Listing Kernel Modules and Tracing Dependencies
    a. In your terminal, display a list of currently loaded kernel modules.
    b. Locate bnx2fc.
    c. Display more detailed information about bnx2fc. What does the description tell you about it? Does it have any associated parameters?
    d. What is the default value of bnx2fc's debug_logging parameter? Is that the currently loaded value? Verify this.
    e. Attempt to alter the debug_logging parameter of bnx2fc to reflect a value of 0x01 without unloading the module first. Did this succeed? Why or why not?
    f. Are there any other modules that depend on bnx2fc?
93. Loading and Unloading Kernel Modules
    a. Unload bnx2fc. Did any other modules unload with it? Why or why not?
    b. Attempt to load bnx2fc with the debug_logging parameter set to 0x01. Did this succeed? Verify it.
    c. Unload the module again.
    d. Make any changes necessary to cause bnx2fc to load with a debug_logging parameter value of 0x02 when next loaded.
    e. Load it. Did the changes take? Why or why not?
94. Blacklisting Modules
    a. Unload bnx2fc.
    b. Blacklist the module. Ensure that this module will not be loaded either manually or automatically.
    c. Test it.

Lab 18: Disaster Recovery

95. Create compressed archives of **`/lab`** called **`lab.tar`** and **`/etc`** called **`etc.tar`**, preserving all attributes
    a. Restore the **`lab.tar`** to a new directory **`/new-labs/`**
    b. Create an archive of all files changed in **`/etc`** and **`/lab`** in the last two days. Place it in **`/lab/18-disaster/`**.
    c. Delete **`/lab/03-vi/editme.finished`**
    d. Add a file **`/lab/03-vi/editme.added`**
    e. Compare the **`lab.tar`** with the current filesystem using **`tar`**.
96. **`rsync`** all **`.conf`** files from **`/etc/`** to **`/new-labs/saved-confs`** on your neighbor's machine.
    a. Delete **`/etc/sane.d/`** on your machine.
    b. Add a comment to **`/etc/asound.conf`**
    c. Restore **`sane.d`** without overwriting your changes to **`asound.conf`**

Lab 19: Security

97. Install and configure **`aide`**.
    a. Write a script to be run daily which archives your results on a neighbor's machine.
98. Configure **`clam`** anti-virus to run on a regular basis.
99. Audit your installed software and services. Write a series of removal recommendations in **`/lab/19-Security/software-clean`**.
    a. Don't forget to check dependencies, don't remove things that would break functionality.
    b. Outline further steps you would take to secure this machine.
    c. Be prepared to justify your results

Lab 20 – Final

You have just inherited a new computer. It has been severely misconfigured, and may have been compromised.

Your new box is currently only reachable by telnet, and is drawing a DHCP address. Your instructor will provide you its current address, as well as the static addresses and hostname that it should be given.

Currencht logins are **`student`** and **`root`**, both with password **`ujm<KI*(98`**.

You will have the use of one of the computers you used during the class, access to the Internet, the course materials, and any notes you have made during the course.

You must secure the computer. You will be given a subset of the STIG as the standard for security. Your machine must be as compliant as possible to the attached Security Standards.

Some features and functionalities must also be added.

All security and functionality should work **without intervention** after a reboot.

Create brief documentation of your success or failure for each step in **`/lab/20-Final/notes.txt`**. If there are multiple possible ways of accomplishing a task, note the method used.

It is recommended that you review all tasks and the Security Standards before beginning.

100. Configure your IP addresses and hostname.
101. Establish secure communication with your box.
    a. Restrict ssh to your local network.
102. The firewall and SELinux must be properly configured.
103. The boot sequence should be protected from unauthorized meddling.
    a. set the password to **`BootMe`**
104. Create two partitions of approximately 400 MB each on the second drive.
105. One of these partitions should be used as the base directory for web services.
    a. This file system should
        i. be mounted at **`/new-web`**
        ii. be labeled **`web`**
        iii. be mounted by label rather than device name.
    b. Create an **`index.html`** with the text "This is a test".
    c. **`index.html`** should only be viewable from your subnet.
    d. Share **`/lab/20-Final/notes.txt`** via the web, without moving or copying it.
106. Dave has been terminated. His account should be dealt with.

107. Create these users:
    a. Kate – member of group **staff** and **developers**
    b. Mike – member of group **developers**, account expires at the end of the year
    c. Nick – member of **developers** and **helpdesk**, UID = 4001
    d. Paul – UID = 4002, shell = **vim**
108. In the second new partition create a shared directory **/home/code** for the developers.
    a. Allow all members of developers, except Nick, read/write access to **code**.
    b. Nick should have read access only.
    c. Users should not be able to delete files owned by another user.
    d. Any new files or directories created in **code** should
        i. be owned by the creator and by the developer group
        ii. have appropriate permissions
109. Provide VNC access for Nick.
110. Allow only Kate and Mike the ability to use **sudo**.
111. Restrict direct root login to only **tty1**.
112. Configure certificate-based login from class computers to your new box for Kate.
113. Create a new 200 MB swap file on the second hard drive using a logical volume. Add it permanently to the current swap space.
114. Configure **aide** to run every day at 11 PM.
115. Configure **yum** to remove all metadata and caches, and create a new cache weekly.
    a. If the computer is powered off, the task must run when it is next powered on.
116. The system must be fully up to date.
    a. If a new kernel is available, it should be installed as the default
    b. The old kernel must remain available and bootable
117. Restrict the use of **cron** and **batch** to root, Kate, and Mike.
    a. If new users are created, they should not be granted access to **cron** or **batch**
118. Configure **ntp** to use an NIST server or another server as directed.
119. Configure your new machine to send log files to your class machine.
120. Disable ping to your machine.

Security Standards for Final Lab

(A subset of the Centos6 STIG guide at **`https://static.open-scap.org/ssg-guides`**.)

121. Verify that Shared Library Files Have Restrictive Permissions

System-wide shared library files, which are linked to executables during process load time or run time, are stored in the following directories by default:

```
/lib
/lib64
/usr/libg
/usr/lib64
```

Kernel modules, which can be added to the kernel during runtime, are stored in /lib/modules. All files in these directories should not be group-writable or world-writable. If any file in these directories is found to be group-writable or world-writable, correct its permission.

122. Verify that All World-Writable Directories Have Sticky Bits Set

When the so-called 'sticky bit' is set on a directory, only the owner of a given file may remove that file from the directory. Without the sticky bit, any user with write access to a directory may remove any file in the directory. Setting the sticky bit prevents users from removing each other's files. In cases where there is no reason for a directory to be world-writable, a better solution is to remove that permission rather than to set the sticky bit.

123. Ensure All Files Are Owned by a Group

If any files are not owned by a group, then the cause of their lack of group-ownership should be investigated. Following this, the files should be deleted or assigned to an appropriate group.

Unowned files do not directly imply a security problem, but they are generally a sign that something is amiss. They may be caused by an intruder, by incorrect software installation or draft software removal, or by failure to remove all files belonging to a deleted account. The files should be repaired so they will not cause problems when accounts are created in the future, and the cause should be discovered and addressed.

124. Ensure All World-Writable Directories Are Owned by a System Account

All directories in local partitions which are world-writable should be owned by root or another system account. If any world-writable directories are not owned by a system account, this should be investigated. Following this, the files should be deleted or assigned to an appropriate group.

125. Ensure SELinux Not Disabled in /etc/grub.conf

Disabling a major host protection feature, such as SELinux, at boot time prevents it from confining system services at boot time. Further, it increases the chances that it will remain off during system operation.

126. Ensure SELinux State is Enforcing

The SELinux state should be set to enforcing at system boot time.

Setting the SELinux state to enforcing ensures SELinux is able to confine potentially compromised processes to the security policy, which is designed to prevent them from causing damage to the system or further elevating their privileges.

127. Configure SELinux Policy

The SELinux targeted policy is appropriate for general-purpose desktops and servers, as well as systems in many other roles.

Other policies, such as mls, provide additional security labeling and greater confinement but are not compatible with many general-purpose use cases.

128. Verify Only Root Has UID 0

If any account other than root has a UID of 0, this misconfiguration should be investigated and the accounts other than root should be removed or have their UID changed.

129. Set Boot Loader Password

During the boot process, the boot loader is responsible for starting the execution of the kernel and passing options to it. The boot loader allows for the selection of different kernels - possibly on different partitions or media. The default Red Hat Enterprise Linux boot loader for x86 systems is called GRUB. Options it can pass to the kernel include single-user mode, which provides root access without any authentication, and the ability to disable SELinux. To prevent local users from modifying the boot parameters and endangering security, protect the boot loader configuration with a password and ensure its configuration file's permissions are set properly.

130. Verify iptables Enabled

The iptables service shall be enabled.

131. Rsyslog Logs Sent To Remote Host

If system logs are to be useful in detecting malicious activities, it is necessary to send logs to a remote server. An intruder who has compromised the root account on a machine may delete the log entries which indicate that the system was attacked before they are seen by an administrator.

However, it is recommended that logs be stored on the local host in addition to being sent to the loghost, especially if rsyslog has been configured to use the UDP protocol to send messages over a network. UDP does not guarantee reliable delivery, and moderately busy sites will lose log messages occasionally, especially in periods of high traffic which may be the result of an attack. In addition, remote rsyslogmessages are not authenticated in any way by default, so it is easy for an attacker to introduce spurious messages to the central log server. Also, some problems cause loss of network connectivity, which will prevent the sending of messages to the central server. For all of these reasons, it is better to store log messages both centrally and on each host, so that they can be correlated if necessary.

Along with these other directives, the system can be configured to forward its logs to a particular log server by adding or correcting one of the following lines, substituting loghost.example.com appropriately. The choice of protocol depends on the environment of the system; although TCP and RELP provide more reliable message delivery, they may not be supported in all environments.

132. Enable rsyslog Service

The rsyslog service provides syslog-style logging by default on Red Hat Enterprise Linux 6.

133. Disable xinetd Service

The xinetd service can be disabled

134. Uninstall xinetd Package

The xinetd package can be uninstalled

135. Disable telnet Service

The telnet service can be disabled with the following command:

136. Uninstall telnet-server Package

The telnet-server package can be uninstalled

137. Disable DHCP Client

For each interface on the system (e.g. eth0), ensure that dhcp is disabled.

DHCP relies on trusting the local network. If the local network is not trusted, then it should not be used. However, the automatic configuration provided by DHCP is commonly used and the alternative, manual configuration, presents an unacceptable burden in many circumstances.

Command Summary

Module 0: Introduction to Linux

- **`ls`** – list files
  - **`-l`** – long
  - **`-a`** – all
  - **`-d`** – directory
  - **`-Z`** – selinux
  - -i – inode information
- **`cd`** – change directory
  - **`.`** – this directory
  - **`..`** – the parent directory
  - **`~`** – the home directory
- **`pwd`** – print working directory
- **`cat`** – print a file
  - **`-n`** – number lines
  - **`-b`** – number non-blank lines
- **`uname`** – print system information
  - **`-a`** – all
- **`exit`**, **`logout`**, **`CTL-d`** – ways to end a shell
- **`--help`** – standard short help option
- **`man`** – manual pages
  - **`-k`** – find man pages about a string
- **`makewhatis`** – update man page database
- **`info`** – documentation

Module 1: Getting around in EL6

- **yum** – software manager
  - **-y** – yes
  - **update**
  - **install**
- **history**
- **!!** – last command
- **!$** -- last argument
- **$?** – last exit code
- **CTL-c** – end current command
- **ping** – send echo request to a host
- **ssh** – connect to a host
- **echo** – display something
- **less** – view output one page at a time
- **grep** – get regular expression
- **ip** – view network information
  - **a** – address
  - **route** – routing table
  - **neigh** – neighbors (arp table)

- *a* **|** *b* – send output from command **a** to command **b**
- *a* **>** *x* – overwrite file **x** with output of command **a**
- *a* **>>** *x* – append output of **a** to file **x**
- **;** – separate commands on a line
- *a* **&&** *b* –run **b** if **a** succeeds
- *a* **||** *b* – run **b** if **a** fails
- **&** -- run a command in the background
- **0** – **STDIN**
- **1** – **STDOUT**
- **2** – **STDERR**
- **&> /dev/null** – suppress all output

- **screen**
  - **CTL-a** – hot key, repeat to toggle windows
  - **x** – lock screen
  - **c** – create new
  - **n** / **p** – next / previous
  - **-RL** – Reattach if available, and Log

- Module 2: Files and **`file`** – determine file type
- **`stat`** – display file status (metadata)
- **`which`** – locate first command in **`$PATH`**
- **`whereis`** – search for a command and related files
- **`locate`** – find files based on an index database
  - **`updatedb`** – update the locate database
- **`head`** – display the top of a file
- **`tail`** – display end of a file
  - **`-f`** -- follow
- **`wc`** – word count
  - **`-l`** – count lines
- **`find`**
  - **`-maxdepth`** / **`-mindepth`**
  - **`-type`** – **`d`**ir, **`f`**ile, **`l`**ink, etc.
  - **`-name`**
  - **`-iname`** – case insensitive name
  - **`-nouser`** – unowned
  - **`-o`** – OR (default is AND)
  - **`-perm`** – permissions (**`/o=w`**)
  - **`-atime`**, **`-ctime`**, **`-mtime`** – access, change, modify time
    - **`-1`** = last 24 hours
    - **`1`** = 24-48 hours
    - **`+1`** = more than 48 hours
- **`dmesg`** – print kernel ring buffer (boot messages)
- **`touch`** – change file timestamps, create empty file
- **`tee`** – split output to STDOUT and a file
- **`column`** – put lists into columns
- **`rm`** – remove file
  - **`-r`** – recursively
  - **`-f`** – force
- **`mkdir`** – make directory
  - **`-m`** – mode (permissions)
  - **`-p`** – create path if not extant
- **`rmdir`** – remove directory
- **`cp`** – copy
  - **`-Z`** – preserve original selinux context
- **`mv`** – move
  - **`-Z`** – assume destination selinux context
- **`sort`**
  - **`-n`** – numerically
  - **`-r`** – reverse
  - **`-u`** – unique
- **`tr`** – translate characters

- **`uniq`** – find unique lines
- **`diff`** – compare two files

Module 3: vi

- **Normal** – command mode; **`ESC`**
  - **`u`** – undo
  - **`ctrl-r`** – redo
  - **`yy`** – yanks (copy)
  - **`dd`** – delete / cut
  - **`p`** – paste
  - **`:`** commands
    - **`w`** – write
    - **`q`** – quit
    - **`a`** – all
    - **`!`** – force
    - **`!`** *`cmd`* – run command
    - **`e`** – edit from last write
    - **`n`** / **`N`** – next / previous file
    - **`split`** / **`vsplit`**
    - **`r`** – read
    - **`r !`** *`cmd`* – read command into file
    - **`abbr`** – abbreviate
    - **`set number`** – turn on line numbering, **`!`** turns it off
- **Visual** – select mode; **`v`**, **`V`**, **`CTL-v`**
  - **`y`** – copy
  - **`d`** – delete
  - **`p`** – paste
- **Insert** – typing mode; **`i`**, **`o`**, **`A`**

- **`service`** – control services
  - **`restart`**
  - **`stop`**
  - **`start`**
  - **`status`**
  - **`--status-all`**

Module 4: Users and Groups

- **useradd**
  - **-G** – additional groups
  - **-u** / **-g** – uid / gid
  - **-s** – shell
- **passwd**
  - **-l** / **-u** – lock / unlock
  - **-S** – status
- **chage** – change password expiration
  - **-l** – list
- **usermod**
  - **-aG** – add to Groups
- **userdel**
  - **-r** – remove home directory
- **vipw**, **vigp**, **visudo** – editors for special files
- **groupadd**
- **groupmod**
- **groupdel**
- **groupmems**
  - **-g** – groupname, mandatory
  - **-a** / **-d** – add /delete
  - **-l** – list
- **whoami** – show current user
- **id** – show real and effective uid
- **su** – change current uid
  - **su -** – use destination environment
- **sudo** – change effective uid to root
- **who** – is currently logged on
- **w** – who plus stats and current command
- **uptime**
- **lastlog** – last logins
- **last** – successful logins
- **lastb** – bad logins

Module 5: Ownership, Permissions & Access

- **chmod** – change permissions
  - **r** = **4**
  - **w** =**2**
  - **x** = **1**
  - **SUID** = **4** (**s**)
  - **SGID** = **2** (**s**)
  - **Sticky** = **1** (**t**)
  - **u**, **g**, **o**, **a** – user, group, other, all
- **umask** – permissions to unset
- **chown** / **chgrp** – change ownership
- **getfacl** – get file ACL
- **setfacl** – set file ACL
  - **-m** – modify an ACL
  - **-x** – remove an ACL entry
  - **-b** – remove all ACLs for file
  - **-R** – apply ACL recursively through subdirectories
  - **[d]:u|g|o|m:*UID*|*GID*::*perms***
    - **d** – default
    - **u**, **g**, **o** – user, group, other
    - **m** – mask
    - **UID**,**GID**
    - **perms** – rwx permissions

Module 6: Regular Expressions

- **Common**
  - **`[a-z]`** – any single lowercase alpha character
  - **`[A-Z]`** – single uppercase alpha character
  - **`[abc]`** – a, b, or c.
  - **`[0-9]`** – any single digit
  - **`[^a2]`** – any character NOT **`a`** or a **`2`**
  - **`.`** – any single character other than line break
  - **`[.]`** – a literal period
  - **`[a|b]`** – **`a`** OR **`b`**
  - **`^`**/**`$`** – beginning / end of line
  -
  - Match the preceding…
  - **`?`** – 0 or 1 times
  - **`*`** – 0 or more times
  - **`+`** – 1 or more times
  - **`{N}`** – exactly N times
  - **`{N,}`** – N or more times
  - **`{N,M}`** – between N and M times

  - **`( )`** Encloses the pattern to store matches from
  - **`\N`** Return the match from the Nth (1-9) stored backreference

- **`grep`**
  - **`-v`** – invert the match (return ONLY lines where pattern was NOT matched
  - **`-o`** – match entire line (rather than within the line)
  - **`-b`** – respect word boundaries
  - **`-i`** – case insensitive
  - **`-E`** – extended grep
  - **`+ ? | {} ()`** – literals in basic grep, metacharacters in extended (**`-E`**)

- **sed**
  - **-n** – prevent normal output printing
  - **-i** – perform an in-place edit
  - **-r** – extended regular expression mode
  - **-e** – multiple expressions in a line
  - **-f** – run a sed script. **sed -f** *script original changed*
  - **p** – print
  - **s** – substitute
  - **d** – delete
  - **g** – global
    - **sed 's/***pattern***/***replacement***/g'** *filename*
  -
- **awk '{** *commands* **}'** *filename*.
  - Commands
    - **print** – print matches
    - **sub()** – substitute, once per line
    - **gsub()** – global substitute
  - **-F** – set delimiter
  - **-f** – run a script
  - **/*pattern*/** match a pattern (occurs before commands portion)

  - Variables
    - **$0** – entire current record
    - **$n** – where **n** is a number; field by sequence e.g. **$1,$2**
    - **NF** – number of fields in current record
    - **NR** – number of current record
    - **FNR** – if multiple input files, record number of current file
    - **FS/RS** – input field separator/record separator
    - **OFS/ORS** – output field separator/record separator
    - **FILENAME** – the name of the input file; undefined in BEGIN block
    -
  - Scripting Constructs
    - **BEGIN** Run first, used to make header or set variables.
    - **BODY** Where the bulk of the script resides.
    - **END** Run last, used to create a footer.

- **ln** – link file
  - **-s** – symbolic
- **lsblk** – list block devices
- **fdisk** – partition table manipulation
  - **-l** – list
  - **m** – menu
  - **n** – new
  - **p** – print current
  - **w** – write
  - **q** – quit
- **partprobe** – inform kernel of changes
- **mkfs.ext4** – make ext4 file system
  - **-L** – label
- **mount /***device* **/***dir*
  - **-a** – mount all automatically
  - **-o** – options
    - **defaults** – alias for **async,auto,dev,exec,nouser,rw,suid**
    - **async** – Allow the asynchronous input/output operations
    - **auto** – mount automatically using **mount -a**
    - **noauto** – no automatic mount
    - **dev** – Interpret character or block special devices on the file system
    - **exec** – allow the execution of binaries
    - **noexec** – no execution of binaries
    - **nouser** – disallow non-root to mount and unmount
    - **rw** / **ro** – read/write / read-only
    - **suid** – Allow set-user/group bits to take effect.
    - **remount** – Remount the file system in case it is already mounted.
- **umount** – unmount
- **fstab** – format:
  [device] [directory] [type] [options] [dump(0,1)] [fsck(0,1,2)]
- swapon --
- **parted** – partition table manipulation
  - **print free**
- **swapon**/**swapoff** – control swap devices and files
  - **-s** – show swap
  - **-a** – all
  - **-L** – label
  - **-v** – verbose
- **mkswap** – create swap area
  - **-L** – label

- **`pvcreate`** – create a PV
- **`pvdisplay`** – display detailed information about a PV
- **`pvremove`** – remove (destroy) a PV
- **`pvresize`** – resize PV to reflect size of underlying device
- **`pvs`** – display information about PVs on a system
- **`pvscan`** – scan devices for LVM (PV) data; update cache

- **`vgcreate`** – create a VG
- **`vgdisplay`** – display detailed info about VG(s)
- **`vgextend`** – add PV(s) to VG
- **`vgreduce`** – remove PV(s) from VG (CAUTION!)
- **`vgremove`** – destroy a VG
- **`vgs`** – display information about VGs
- **`vgscan`** – scan devices for LVM (VG) data; update cache

- **`lvcreate`** – create an LV
- **`lvdisplay`** – display detailed information about an LV
- **`lvextend`** – add physical extents to an LV
- **`lvreduce`** – remove physical extents from an LV
- **`lvremove`** – destroy an LV
- **`lvresize`** – shrink or grow an LV (-r autoresize resident FS)
- **`lvs`** – display information about LVs
- **`lvscan`** – scan devices for LVM (LV) data; update cache

- **`cryptsetup`**
  - **`luksFormat`** – create a LUKS device
  - **`luksOpen`** – open LUKS for reading
  - **`luksClose`** – close LUKS access
  - add mapping and keys in **`/etc/crypttab`**

Module 8: Processes and Services

- **ps** – show running processes
  - **-e** – everything
  - **-f** – full listing
  - **-u** – user
  - **-Z** – selinux
- – pass a signal
  - **-9** – **KILL**
  - **-20** – **SPT** (pause)
  - **-18** – **CONT** (resume)
- **CTL-z** – pause a process
  - **fg**/**bg** – resume foreground / background
  - **bg** – resume background
  - **disown**
    - **-a** – all
    - **-h** – leave in table, do not terminate on exit
- **nice** – set a process priority
  - **-n** – value (-20 to 19)
  - **renice** – change a process priority
- limit user resources in **/etc/security/limits.conf**
  - **ulimit** – adjust limits on the fly
- **top** – view running processes
  - **h** – Display a help screen
  - **i** – toggle idle processes
  - **f** – Select fields to display
  - **F** – Select sort field
  - **M** – Sort by memory usage.
  - **P** – Sort by CPU usage.
  - **V** – Show parent process relationships
  - **u** – Filter by user.
  - **r** – Renice a process.
  - **k** – Kill a process.
  - **q** – quit top
- **free** / **vmstat** – display memory
- **pgrep** / **pkill** – process grep / pgrep and signal processes
  - **-u** – user
  - **-l** – long (shows command line)
  - **-n** – newest
  - **-v** – invert selection
- **killall** – signal multiple processes
  - **-u** – user
  - **-i** – interactive (prompt)
  - **-v** – verbose, **not** invert

Module 9: Scheduling Events

- **`at`** – schedule events once
  - **`atq`** – list jobs
  - **`atrm`** – remove jobs
  - **`-c`** – show job script
  - **`batch`** – schedule when cpu usage is below threshold set with **`at -l`**
- **`crontab`** – schedule precisely
  - **`-l`** – list
  - **`-e`** – edit
  - **`-u`** – user
  - Format:
    [minute] [hour] [day] [month] [day of week] [/path/cmd]
- **`anacron`** – schedule roughly
  - create link from job to **`/etc/cron.interval`**

Module 10: Booting

- **`runlevel`** – view previous and current runlevel
- **`telinit`** – change runlevel
  - **`0`** – halt
  - **`1`** – Single-user
  - **`3`** – Multi-user
  - **`5`** –graphical
  - **`6`** – reboot
  - default set in **`/etc/inittab`**
- **`chkconfig`** – enable services
  - no arguments – show all
  - **`on`** / **`off`** – enable / disable service
  - **`--list`**
  - **`--level`** – set at specific runlevels
- **`grub-crypt --sha-512`** – generate password hash
- **`touch /.autorelabel`** – restore selinux context on entire file system
- **`shutdown`**, **`poweroff`**, **`halt`**, **`reboot`**
  - **`-f`** – force

Module 11: Logs

- **rsyslog.conf** severities
  - **0 emerg**
  - **1 alert**
  - **2 crit**
  - **3 err**
  - **4 warning**
  - **5 notice.**
  - **6 info**
  - **7 debug**
  - **.none**
- **logrotate.conf**
- **logger -p** – generate syslog message with priority
- **timezone** – link from **/usr/share/zoneinfo/**… to **/etc/localtime**
- **date**
  - **-s** – set "DD month YYYY HH:MM:SS" or HH:MM:SS
- **ntpdate** – manual ntp sync
- **ntp.conf**
- **ntpq -p** – ntpd status

Module 12: SELinux

- **`semanage`** – policycoreutils-python package
  - **`login`**
  - **`user`**
  - **`port`**
  - **`permissive`**
  - **`boolean`**
  - **`fcontext`** – file context
  - **`-l`** – list
  - **`-d`** – delete
  - **`-a`** – add
  - **`-e`** – set equal (requires source and target)
  - wildcard a directory – **`"/web(/.*)?"`**
- **`seinfo`** – setools-console package
  - **`-a`** – attribute (no space)
  - **`-x`** – print a list
- **`getenforce`** / **`setenforce`** – view/change selinux status
- **`sealert`** – setroubleshoot-server package
- **`getsebool`** / **`setsebool`** – view/change selinux boolean
  - **`-P`** – persistent
- **`sesearch`** – policy query
  - **`-b`** – boolean
  - **`--`***`rule`* – **`allow`**, **`type`**, **`neverallow`**, **`audit`**, **`dontaudit`**, **`all`**
  - **`-s`** – source
  - **`-t`** – target
  - **`-c`** – class (file, dir, socket)
  - **`-p`** – permission type (read, write, create)
- **`ausearch`** – query audit logs
  - **`-c`** – common name
  - **`-i`** – interpret numbers to names
  - **`-m`** – message type
  - **`-ts`** – time start
- **`restorecon`** – restore SELinux context on a file/directory
  - **`-R`** – recurse
  - **`-v`** – verbose
- **`audit2allow`**
  - **`-o`** – output file
- **`semodule`** – manage SELinux policy modules
  - **`-i`** – install
  - **`-d`** – disable

Module 13: Networking

- set hostname in **`/etc/sysconfig/network`**
- set name resolution in **`/etc/hosts`**, **`nsswitch`**, and **`resolv.conf`**
- set ip address in **`/etc/sysconfig/network-scripts/ifcfg-eth`***X*
- **`ifup`** /**`ifdown`** – reset an interface
- **`ifconfig`** – view interface information
- network diagnostics – **`ping`**, **`traceroute`**, **`tracepath`**, **`dig`**, **`nslookup`**
- **`netstat`** – show socket information
  - **`-c`** – continuous
  - **`-l`** – listening ports only
  - **`-a`** – all ports
  - **`-e`** – extended detail
  - **`-t`** – TCP
  - **`-u`** – UDP
  - **`-p`** – show PID and program that owns the socket
  - **`-n`** – numeric address only
  - **`-s`** – statistics
- **`ss`** – socket status
  - **`-m`** – memory usage
  - **`-o`** – time
  - **`state`** – established, syn-sent, syn-recv, connected
  - **`src/dst`** – source or destination address
  - **`sport/dport`** – source / destination port (**`:443`**)
- **`whois`** – query a name registrar, installed as the **`jwhois`** package
- **`curl`** / **`wget`** – non-interactive ways to download from a URL
- **`tcpdump`** – wire-level level listener/parser

- **`iptables`** – manage the firewall
  - **`-L`** – list
  - **`-v`** – verbose
  - **`--line-numbers`**
  - **`-I`** – insert a rule
  - **`-A`** – append
  - **`-D`** – delete
  - **`-R`** – replace
  - **`-F`** – flush all
  - **`-p`** – protocol
  - **`--dport`** – destination port
  - **`-j`** – jump to target
  - **`--policy`** – set default policy (DROP)
  - **`-m`** – match (**`state`**)
    - **`--state`** – RELATED, ESTABLISHED, NEW
- **`service iptables`**…
  - **`status`** – show all rules
  - **`save`** – save running config to **`/etc/sysconfig/iptables`**

Module 14: Installing Software

- **yum**
  - **repolist** – shows repos in **/etc/yum.repos.d/**
  - **repoquery**
  - **list installed**
  - **search**
  - **info**
  - **provides** – what package provides a binary
  - **clean all** – clear caches
  - **makecache**
- **createrepo** – build repository metadata
- **rpm**
  - **--import** – import GPG key
  - **-Va** – verify all
  - **-q** – query
    - **f** – file (shows parent package)
    - **c** – configuration files
    - **l** – list all installed by package
  - **-i** – install
  - **-U** – upgrade

Module 15: Remote File Systems

- **nfs** – part of nfs-utils package
- edit **/etc/exports** format:
  [local share] [hostname or network] [options]
  - options include:
    - **rw** –read/write
    - **sec=** –security flavors (**sys**, **krb5**, **krb5i**, **krb5p**)
    - **sync** – do not reply until write
    - **fsid=0** – sets the "root" of the virtual file system in NFSv4
    - **anonuid** – set the system UID to be assigned to anonymous users
    - **no_root_squash** – do not map UID 0 requests to **anonymous**
- **exportfs** – re-read /etc/exports
  - **-a** – all
  - **-u** – unexport

- **vsftpd** – configured in **vsftpd.conf**
  - requires **IPTABLES MODULES="ip conntrack ftp"** in **iptables-config**
  - **lftpd** – ftp client

- **httpd** – configured in **httpd.conf**
  - **curl**, **wget**, **lynx** – clients for url-based targets

Module 16: Remote Access

- **vnc** – **tigervnc** and **tigervnc-server** packages
  - uses ports 5800,5900,6000 – increment by one for each additional user
  - **vncpasswd** – set password (after **su -)**
  - start manually with **vncserver :***display number*
  - start automatically by configuring **/etc/sysconfig/vncservers**
- **vncviewer** *user*@*host host*:*number* -- connect to vnc server

- **sshd** – configured in **/etc/ssh/sshd_config**
- **ssh** *user*@*host*
  - -i – identity file (key)
- **ssh-keygen** – make key
- **ssh-copy-id** – copy key to remote host
- **scp** *localfile user*@*host*:*/path/file* – copy localfile to remote host
- **sftp** – ftp client over ssh

Module 17: Kernel Modules and Parameters

- **lsmod** – list kernel modules
- **modinfo** – provide information about a module
- **modprobe** – add / remove modules from the kernel
  - reads files (options and blacklists) from **/etc/modprobe.d/**
  - **-l** – list
  - **-r** – remove
  - **-c** – show configuration
- **sysctl** – alter kernel parameters
  - **-a** – display all
  - **-w** – write active parameter
  - persistent settings in **sysctl.conf**

Module 18: Disaster Recovery

- **dd if=***infile* **of=***outfile* – copy data
  - **conv=noerror** – ignore read errors
  - **bs=** – block size
  - **count=** – number
- **shred** – securely delete
- **tar** – archiving
  - **-c** – create
  - **-z** – zip
  - **-p** – preserve permissions
  - **-t** – list contents of archive
  - **-v** – verbose
  - **-x** – extract
  - **-C** – change directory
  - **-d** – find differences between archive and file system
  - **-f** – archive filename follows
  - **--xattrs** – preserve both SELinux and acls attributes, use when creating and extracting
- **rsync** – remote syncronization
  - **-a** – archive, equivalent to **-rlptgoD**: **r**ecurse, **c**opy links; save **p**ermissions, **t**imestamps, **g**roup, and **o**wner; also preserve **D**evice and special files.
  - **-v** – verbose
  - **-e** – use specified transport, such as **"ssh -i /path/key"**
  - **-X** – preserve extended attributes
  - **-A** – preserve ACLs
  - **-S** – squeeze sparse files (those with long zero blocks)
  - **-L** – turn links into files
  - **--delete** – remove files not at the source, default is to preserve them at the destination
  - **--remove-source-files** – delete source upon success, useful for temporary tarballs
  - **--include** / **--exclude** / **--filter** – allows detailed specification of files
- **rear mkbackup** – relax and recover

Module 19: Security

- **aide** – detect file system changes
  - fix hash types and exclude active files in **aide.conf**
  - **-i** – create initial database
    - remove **.new** from the name
  - **-C** – compare (outputs to **/var/log/aide/aide.log**)
  - **-u** – update the database
- **git** – stupid content tracker
  - **clone** – a repository to a local directory