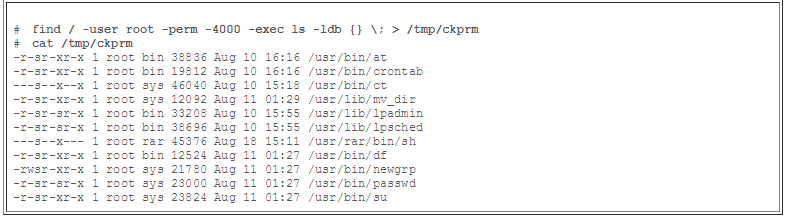
* **Step One: Find out which version of OS and kernel are you dealing with**
  + **# lsb\_release -a** ⇒ The lsb\_release command displays certain LSB (Linux Standard Base) and distribution-specific information. If this does not give you enough information try **# cat /etc/\*-release.**
  + **# uname -a** OR **# uname -mrs ⇒** command to find out your kernel version. Sample output: **Linux 2.6.32-5-amd64 x86\_64**, where **Linux** is a Kernel name, **2.6.32-5** is a Kernel version and **x86\_64** is machine hardware name(64 bit). Use **# cat /proc/version** if you want to see **gcc** version as well. Also, if you do not understand various parts of the output from the listed commands google them, but you should not need them. For the competition purposes knowing what distribution of \*Nix and kernel you have and their version numbers should be enough.
  + Take a note of the information you got from this step, you will need this for the various parts of the competition and to go through this checklist.
  + Keep Linux Kernel and Software Up to Date
    - # **yum update** OR # **apt-get update** && **apt-get upgrade ⇒** Make sure you use both apt-get update and apt-get upgrade, first just gets you newest packages and the latter installs them all. Try to use **yum** first and if that does not work do the second one.
  + So, at the end of this step you should have an up to date OS on your box and you should know what version \*NIX OS and kernel are you running. Reference the **Specific Unix Systems** checklist for more info about your box’s OS version. (After you finish the rest of the steps from this checklist of course)
* **Step Two: Check sensitive file permissions:**
  + **SetUID bit**
    - You should be careful with files that have SetUID bit set
    - **# find *{directory}* -user root -perm -4000 -exec ls -ldb {} \; >/tmp/ {*filename}* ⇒** Command to find files with setuid bit set. ***Find {directory}*** checks all mounted paths starting at the specified directory, which can be *root (/), sys, bin,* or *mail*. **-user root** displays files owned only by root. **-perm -4000** displays files only with permissions set to 4000. **-exec ls -ldb** displays the output of the find command in **ls -ldb** format. **./tmp /{filename}** writes results to this file.
      * **Output Example:**



* This output shows that a user named **rar** has made a personal copy of **/usr/bin/sh**, and has set the permissions as setuid to root. As a result, **rar** can execute **/usr/rar/bin/sh** and become the privileged user.
* **To get rid of setuid bit**:
* **# sudo chmod g-s {filename}** ⇒ **g** stands for group (also can be **u** -user, **o** -other, **a**-all), **-s** removes setuid bit (alternatively you can add it with **+s**). Verify changes with **#ls -l {filename}.**
* If numerical permissions work better for you do #**sudo chmod 0664 {filename}** (assuming the rest of the permissions are 664). Do not forget to verify.

Sensitive system files need to have the proper permissions set on them to prevent unauthorized changes:

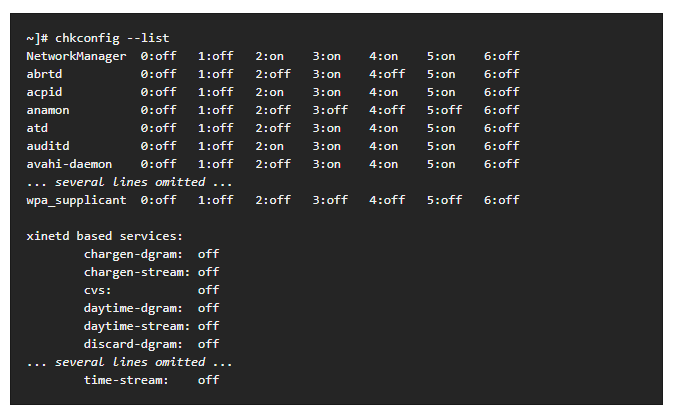
* + **# ls -ldb /etc ⇒** Make sure the following permissions are as listed and if not change them
  + **# chmod 0700 /etc/profile**
  + **# chmod 0700 /etc/hosts.allow**
  + **# chmod 0700 /etc/mtab**
  + **# chmod 0700 /etc/utmp**
  + **# chmod 0700 /var/adm/wtmp( or /var/log/wtmp)**
  + **# chmod 0700 /etc/syslog.pid( or /var/run/syslog.pid)**
  + **# chmod 0700 /etc/sysctl.conf**
  + **# chmod 0700 /etc/inittab**
  + Verify:
    - Root.root and -rw-r--r--(644) ⇒ **# ls -la /etc/fstab**
    - Root.root and -rw-r--r--(644) ⇒ **# ls -la /etc/passwd**
    - Root.root and -rw-r------(400) ⇒ **# ls -la /etc/shadow**
    - Root.root and -rw-r--r--(644) ⇒ **# ls -la /etc/group**
    - Root.root and -rw-r--r--(644) ⇒ **# ls -la /etc/sudoers**
  + If you have time:
    - **# ls -ldb /bin** AND **#ls -ldb /usr/bin ⇒** Verify permissions as specified below.
    - **# chmod 02750 /bin/su**
    - **# chmod 02750 /bin/sudo**
    - **# chmod 02750 /bin/ping**
    - **# chmod 02750 /bin/ifconfig**
    - **# chmod 02750 /usr/bin/w**
    - **# chmod 02750 /usr/bin/who**
    - **# chmod 02750 /usr/bin/locate**
    - **# chmod 02750 /usr/bin/whereis**
  + Get rid of World writable Files:
    - **#find /dir -xdev -type d \( -perm -0002 -a ! -perm -1000 \) -print**
    - **#find / -type d -perm +2 -ls** ⇒ Either **#chmod 750** those files or remove them (**#rm**)
  + Make Sure No Non-Root Accounts Have UID Set to 0
    - **# awk -F: '($3 == "0") {print}' /etc/passwd** ⇒ You should only see one line (root:x:0:0:root:/root:/bin/bash) if not do the following:
      * **#usermod -u {newUID} {username}** ⇒ **{newUID}** is the new user ID that you want to assign to the user **{username}** instead of the old “0”. New UID can be anything that has not been used before (check etc/user).
      * **# groupmode -g {newGroupID} {groupName}** ⇒ Same thing as above but for the groups.
      * **# find / -group {oldGroupID} -exec chgrp -h {groupName} {} \;** and **# find / -user {oldUserID} -exec chgrp -h {userName} {} \;** ⇒ This commands will change the group ID and user ID respectively for any files outside of user’s home directory.
      * To verify:
        + **# ls -l /home/{username}**
        + **# id -u {username}**
        + **# id -u {groupname}** ⇒ All of these commands should give the output that matches the parameters that you specified earlier. (Also, **# awk -F: '($3 == "0") {print}' /etc/passwd** command from earlier should give you only one line this time)

## 

## **Step Three: Check for unwanted services and open ports (IMPORTANT):**

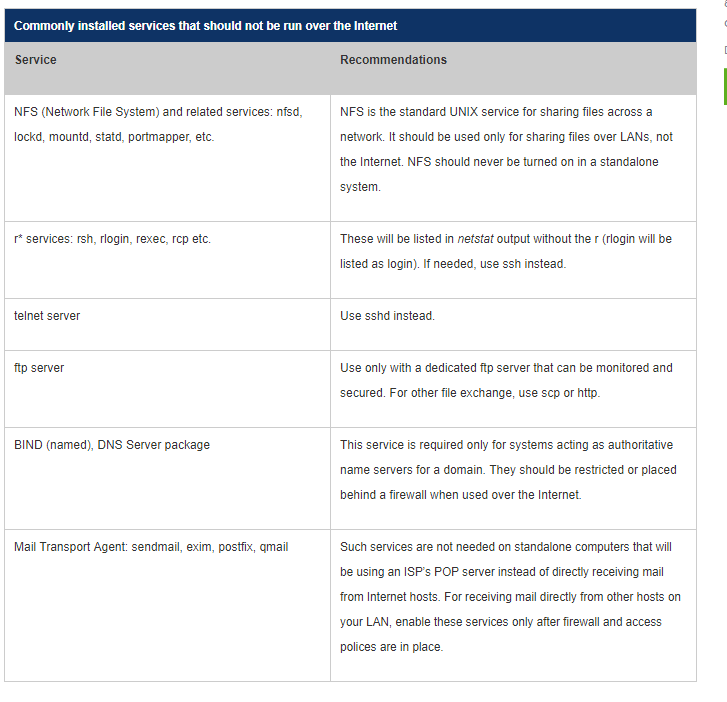
* + Disable Unwanted Services
    - **# chkconfig --list | grep '3:on'** OR **# netstat -lp⇒** This will give you the list of services that are currently on. Carefully go through them and disable the ones that you do not need. More to follow on that in the next paragraph.

Sample output for the **chkonfig** command (note that in this output example we did not filter for the services that are on)



Each line consists of the name of the service followed by its status (*on* or *off*) for each of the seven numbered runlevels. For example, in the listing above, **NetworkManager** is enabled in runlevel 2, 3, 4, and 5, while **abrtd** runs in runlevel 3 and 5. The **xinetd** based services are listed at the end, being either *on*, or *off*.

Here is the list of the services that should not be running on your system:



* + - **\*\*\*\* MAKE SURE YOU ARE NOT TURNING OFF THE SERVICE THAT YOU NEED FOR COMPETITION, TURN ONE OFF ONLY IF YOU ARE ABSOLUTELY SURE THAT TEAM DOES NOT NEED IT. CONSULT IT WITH SYSTEMS TEAM/LEADERSHIP \*\*\*\*\***
    - **There are tons of other services, go through them (do not take forever looking at them though) and research the ones that seem weird to you (something you do not know what it is or completely out of context). If you are not sure, ask! Somebody in the team should know something about that service.**
    - **# service {serviceName} stop** ⇒ Stop the service
    - **# chkconfig {serviceName} off ⇒** Stop the Service

## Find Listening Network Ports

* + - **#sudo lsof -i -P -n**⇒ list the network open ports on your Linux server and the process that owns them
      * Sample output:

**sshd 85379 root 3u IPv4 0xffff80000039e000 0t0 TCP 10.86.128.138:22 (LISTEN)**

-**sshd** is the name of the application

-**10.86.128.13**8 is the IP address to which sshd bind to (LISTEN)

- **22** is the TCP port that is being used (LISTEN)

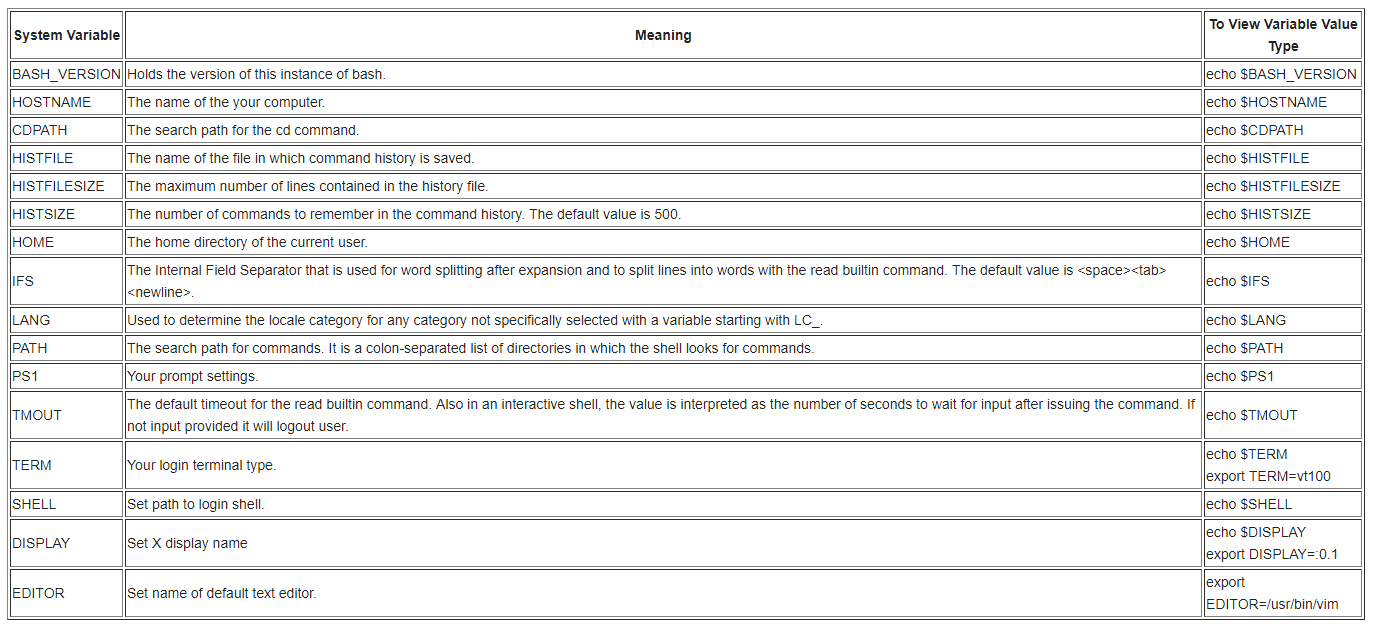
**-85379** is the process ID of the sshd process

* + - Most people use netstat though to get the same info so here is the syntax for it as well:
      * **#netstat -tulpn | less** ⇒ Does the same thing
    - Other Useful Commands:
      * **#lsof -i :{portNumber}** ⇒ To see which process is bound to port {portNumber}
      * **#ls -l /proc/{PID}** ⇒ See More Info for Process with PID
      * **#whatis {COMMAND}**
      * **#pwdx {PID}** ⇒ Report current working directory of proves with PID.
* **Step Four: Preventing other common vulnerabilities**
  + Verify No accounts have empty passwords
    - **# awk -F: '($2 == "") {print}' /etc/shadow**
    - If any empty ones disable it or even delete it ⇒ **#passwd -l {username}**
  + **NoLogin**

Set users shell to nologin if you do not want them to access shell. NoLogin puts linux shell access restrictions for the services like FTP or telnet and refuses them politely not to log in. YOU most likely would want to make sure that www-data has set its shell to nologin, but there might be other users as well that will need this option.

* + - **# less /etc/shells ⇒** You should see the nologin shell in the output. If not, add using the command **#echo “/sbin/nologin” >> /etc/shells** if you are using Debian modify the path to **“usr/sbin/nologin”.**
    - **# usermod -s /sbin/nologin {username} ⇒** Block shell access to user {username}. User {username} must exist when you use this command. Once again, if you are using Debian modify path to **“usr/sbin/nologin”.**
  + Environmental Variables
    - **# set** ⇒ Display current Environment variables. In case if you want to look at them all.
    - **# echo “${variableName}”** ⇒ Display the content of the environmental variable

Following is the list of most common environmental variables:



* + - You can modify environmental variables using export command. Here is the example:
      * **#export PATH=${PATH}:/home/{username}/bin** ⇒ Note that this changes to environmental variables are not permanent, they will be flushed if you reboot the system or logout or even if you open a new shell. Since we are not allowed to reboot in this competition this should not be a problem, but in case if you want to make it permanent use this command : **#vi `/.bash\_profile** (any other editor will work), append your change, so for our example command it would be **#export PATH=${PATH}:/home/{username}/bin**, save and close the file. Alternatively, you always can change them manually every time you login if that makes you happy.
    - **PATH** variable
      * Make sure **“./”** (your home directory) is not in there or if it is, it is at the end not anywhere else. Note, PATH variable tells the shell which directories to search for executable files in response to commands issued by user. Each directory is separated by column.
        + One way to remove something from your PATH variable is to redefine it completely and excluding your home directory. So it will be something like **# export PATH = {only directories I want}**
        + More sophisticated but dirty way to do it:

export PATH="$( echo $PATH| tr : '\n' |grep -v raj | paste -s -d: )"

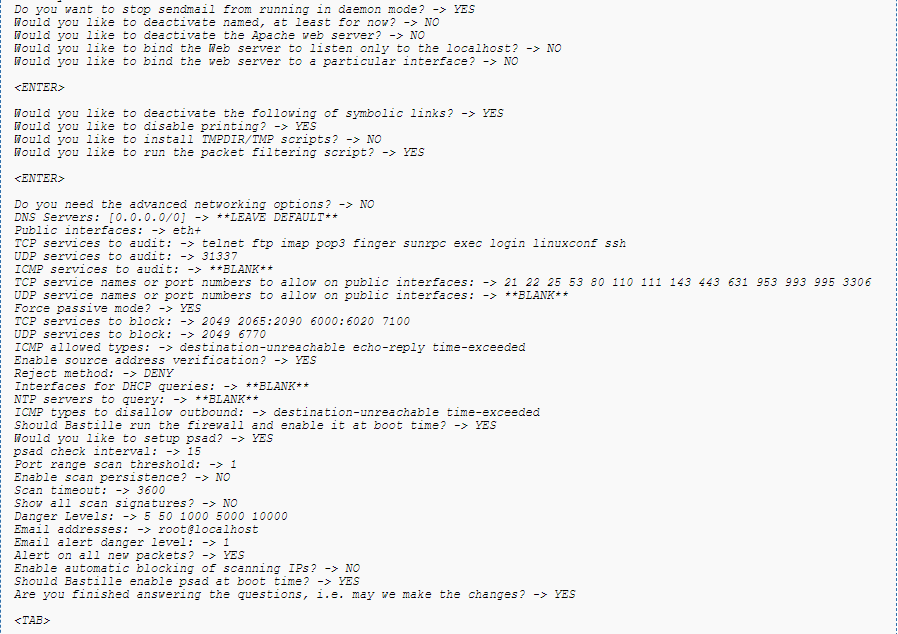
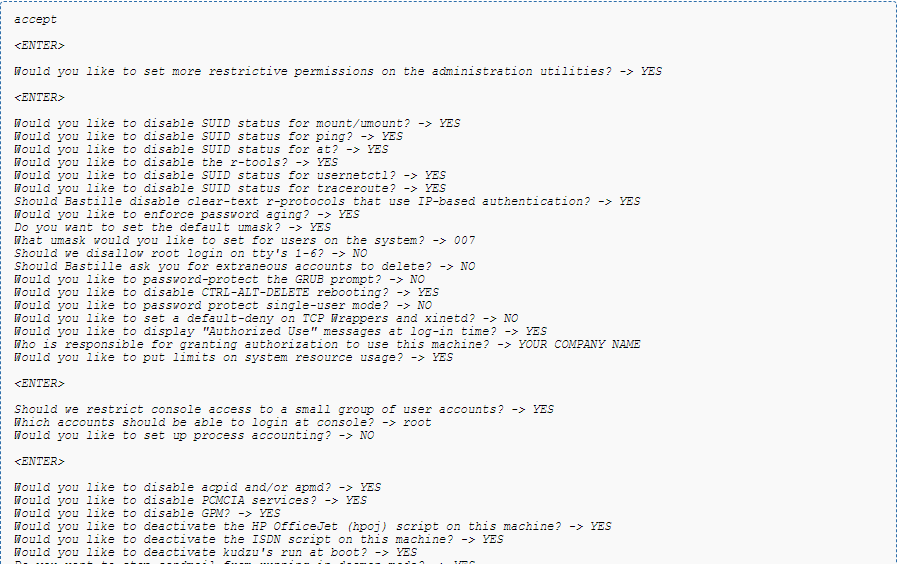
separate each dir in your PATH by line using **tr**

remove what you don't want (path matching "raj") using **grep -v**, and

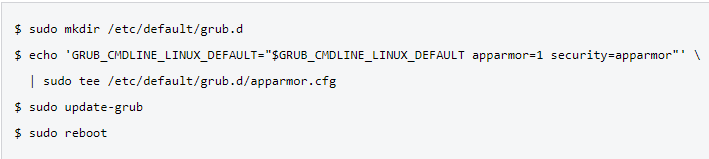
collapse back into a long **":"** delimited string using paste.

* **Step 5 (Tentative depending whether monitoring team does this or not): Setup Logs**
  + Track and Log Failed Login Attempt Records
    - # **vi /etc/pam.d/system-auth**
      * Modify As Follows:
        + *auth required pam\_tally.so no\_magic\_root*
        + *account required pam\_tally.so deny=3 no\_magic\_root lock\_time=120*
      * Other Useful Commands:
        + **#faillog -u {username}** ⇒ Display Login Attempts for {username}
        + **#faillog -M MAX -u {username}** ⇒ Set Maximum Number of Login Failures MAX for the {username}.
        + **#faillog -r OR #faillog -r -u {username}** ⇒ Reset counter of all login failure OR for {username}
      * On that Note:
        + **#passwd -l {username}** ⇒ To Lock Account
        + **#passwd -u {username}** ⇒ To Unlock Account

* Bastille
  + **# apt-get install bastille**
  + Once installed: **#bastille -c** to configure bastille
  + The following screengrabs show the correct configurations:



* Apparmor
  + Install: **#apt-get install apparmor-profiles apparmor-utils**
  + Enable:



* + **#apparmor \_status** (to see current profiles and associated modes)
  + **#man apparmor** (for more details of what to do with that information)