25D Linux Foundation Course

11 – Securing Linux

Overview

- ❑ **Securing the system**
- ❑ **Controlling user access**
- ❑ **Defending against network attacks**
- ❑ **Managing system logs**
- ❑ **Configuring xinetd and inetd**

Securing the Physical Environment

- ❑ Limiting physical access to systems in an environment is paramount
- ❑ The level of access depends on the type of system involved
  - Servers – Extremely High
    - Limited number of individuals
    - Should be locked up (Room, rack, etc.)
    - Possibly have a guard or ID badge and key code to access server room
  - Workstations – More difficult to secure
    - Usually in open environments
    - Proximity locks and ID are best possible ways to secure office

Securing Access to the Operating System

- ❑ **Implementing screensaver passwords may help to prevent unauthorized users from accessing systems when users are away their systems**
- ❑ **Train users to lock their systems when they leave their systems, regardless of how long the user will be away**
- ❑ **Graphical environments provide a means of locking the desktop**
- ❑ **When working in text-based environments, users should log out when they leave their desks**
- ❑ **Using nohup to prevent processes from stopping upon log out in a text-based environment**

To root or Not to root?

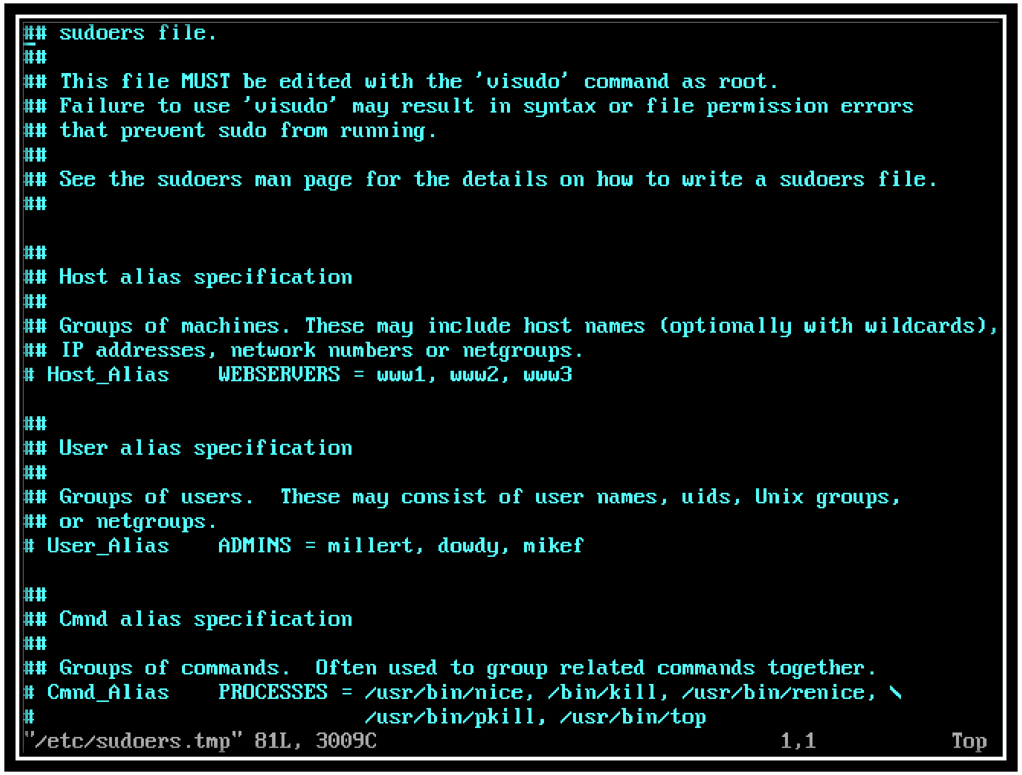
- ❑ **Proper use of the root user account**
  - – **Many new Linux users tend to excessively use the root user account**
  - – **Only use root when absolutely necessary**
  - – **Many tasks can be completed as a non-root user**
  - – **A system logged in as root represents a serious security risk**
- ❑ **Using su**
  - – **Allows a user to change to a different user account at the shell prompt**
  - – **Useful options:**
    - **-: Loads the user’s environment variables**
    - **-c *command:* Switches to the user account and runs the specified command**
    - **-m: Switches to the user account but preserves the existing environment variables**

To root or Not to root?

❑ **Using sudo**

- **Primarily used to grant users limited root access**
- **Can be used to run a command as a different user**
- **Access and authorization is controlled via /etc/sudoers file**

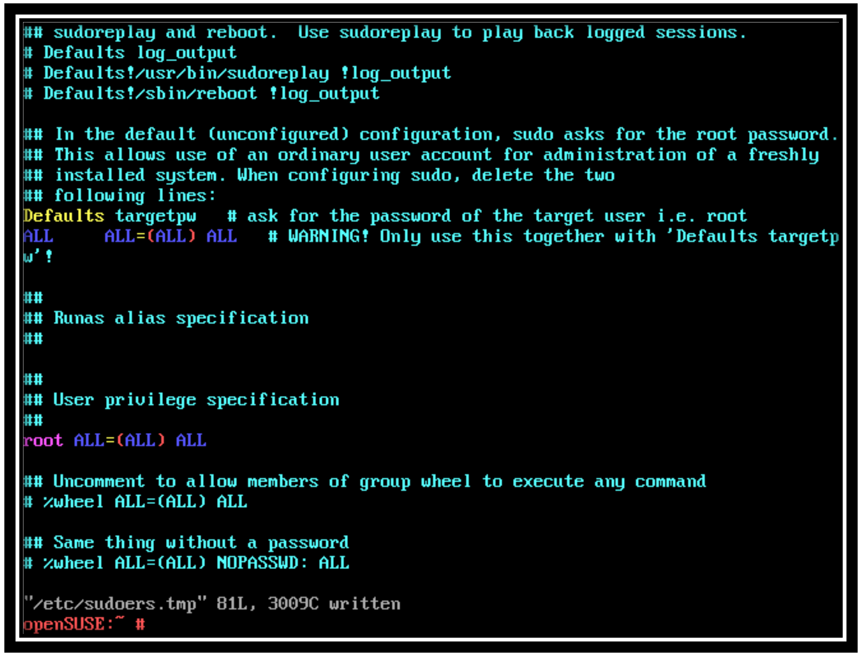
- To edit the /etc/sudoers file, run visudo
  - changes are written to /etc/sudoers.tmp until committed
- In some distributions, the user must supply the root password when using sudo (kind of defeats the purpose)
- User should enter their own account password to use sudo



```
## sudoers file.
##
## This file MUST be edited with the 'visudo' command as root.
## Failure to use 'visudo' may result in syntax or file permission errors
## that prevent sudo from running.
##
## See the sudoers man page for the details on how to write a sudoers file.
##

##
## Host alias specification
##
## Groups of machines. These may include host names (optionally with wildcards),
## IP addresses, network numbers or netgroups.
# Host_Alias    WEBSERVERS = www1, www2, www3

##
## User alias specification
##
## Groups of users.  These may consist of user names, uids, Unix groups,
## or netgroups.
# User_Alias    ADMINS = millert, dowdy, mikef

##
## Cmnd alias specification
##
## Groups of commands.  Often used to group related commands together.
# Cmnd_Alias    PROCESSES = /usr/bin/nice, /bin/kill, /usr/bin/renice, \
#                           /usr/bin/pkill, /usr/bin/top
"/etc/sudoers.tmp" 81L, 3009C                                  1,1           Top
```

Controlling User Access
To root or Not to root?

- ❑ Defaults set in this distribution are in lines 9 and 10 of the below example:



```
## sudoreplay and reboot.  Use sudoreplay to play back logged sessions.
# Defaults log_output
# Defaults!/usr/bin/sudoreplay !log_output
# Defaults!/sbin/reboot !log_output

## In the default (unconfigured) configuration, sudo asks for the root password.
## This allows use of an ordinary user account for administration of a freshly
## installed system. When configuring sudo, delete the two
## following lines:
Defaults targetpw   # ask for the password of the target user i.e. root
ALL     ALL=(ALL) ALL   # WARNING! Only use this together with 'Defaults targetp
w'!

##
## Runas alias specification
##

##
## User privilege specification
##
root ALL=(ALL) ALL

## Uncomment to allow members of group wheel to execute any command
# %wheel ALL=(ALL) ALL

## Same thing without a password
# %wheel ALL=(ALL) NOPASSWD: ALL

"/etc/sudoers.tmp" 81L, 3009C written
openSUSE:~ #
```

- ❑ If a user like student were to do sudo for a command with these setting they would be prompted for the root password
  - – Not everyone should have the root password

Controlling User Access
To root or Not to root?

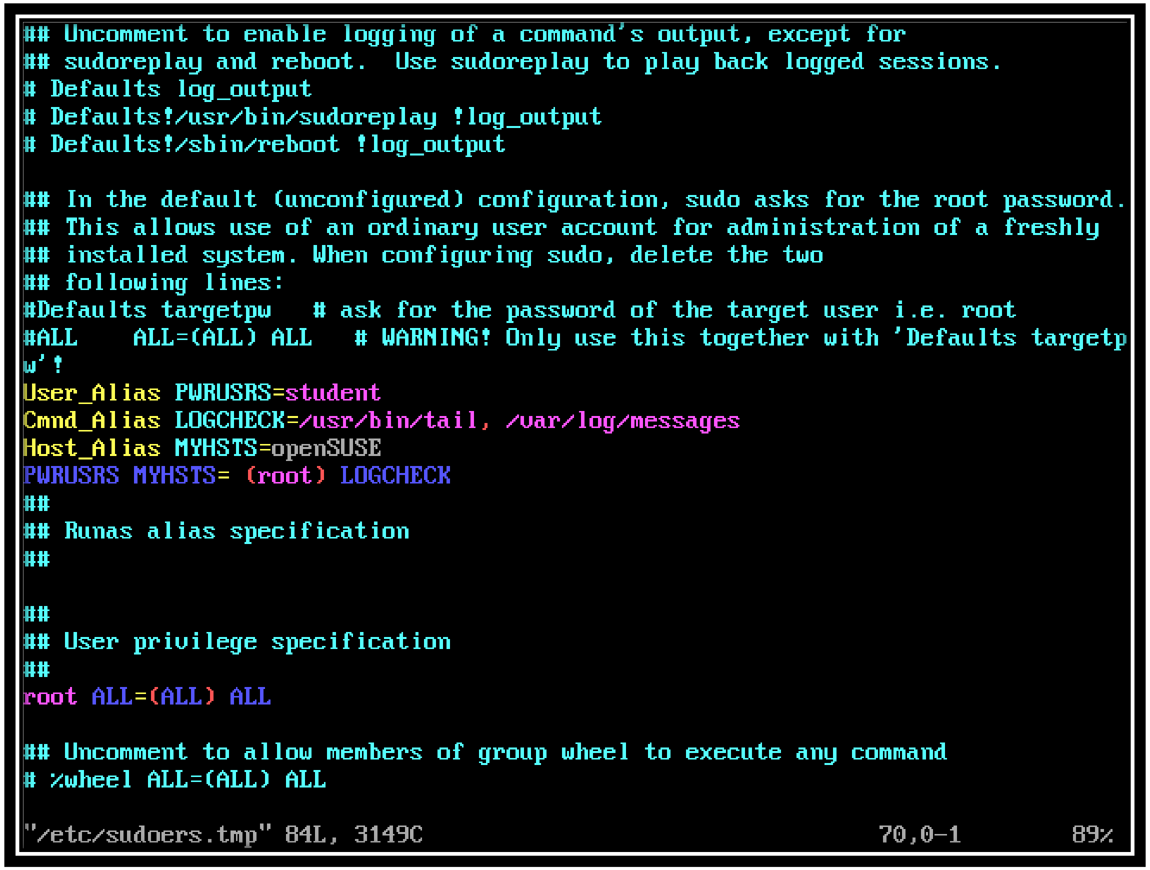
- ❑ **To allow specific users to access specific files or utilities they do not have access to you can edit the sudoers file with aliases**

- ❑ **/etc/sudoers aliases**
  - **User_Alias: Specifies the users who are allowed to run commands**
  - **Cmnd_Alias: Specifies the commands that users are allowed to run**
  - **Host_Alias: Specifies the hosts users are allowed to run the commands on**
  - **Runas_Alias: Specifies the usernames that commands may be run as**

Controlling User Access
To root or Not to root?

- In the below example the defaults were commented out (lines 10 and 11)

- New aliases were created giving student the ability to tail the log file



```
## Uncomment to enable logging of a command's output, except for
## sudoreplay and reboot.  Use sudoreplay to play back logged sessions.
# Defaults log_output
# Defaults!/usr/bin/sudoreplay !log_output
# Defaults!/sbin/reboot !log_output

## In the default (unconfigured) configuration, sudo asks for the root password.
## This allows use of an ordinary user account for administration of a freshly
## installed system. When configuring sudo, delete the two
## following lines:
#Defaults targetpw   # ask for the password of the target user i.e. root
#ALL    ALL=(ALL) ALL   # WARNING! Only use this together with 'Defaults targetp
w'!
User_Alias PWRUSRS=student
Cmnd_Alias LOGCHECK=/usr/bin/tail, /var/log/messages
Host_Alias MYHSTS=openSUSE
PWRUSRS MYHSTS= (root) LOGCHECK
##
## Runas alias specification
##


##
## User privilege specification
##
root ALL=(ALL) ALL

## Uncomment to allow members of group wheel to execute any command
# %wheel ALL=(ALL) ALL

"/etc/sudoers.tmp" 84L, 3149C                              70,0-1        89%
```

Implementing a Strong Password Policy

- ❑ **Never use easy to guess passwords**
  - **Last name**
  - **Birthday**
  - **SSN**
  - **“password”**
  - **Blank passwords**
  - **Dictionary words**
- ❑ **Train users to use stronger passwords**
  - **Six or more characters (the longer the better!)**
  - **A combination of numbers and letters**
  - **Upper- and lowercase letters**
  - **Words not found in the dictionary**
  - **(Optionally) non-alphanumeric characters such as punctuation marks**

Implementing a Strong Password Policy

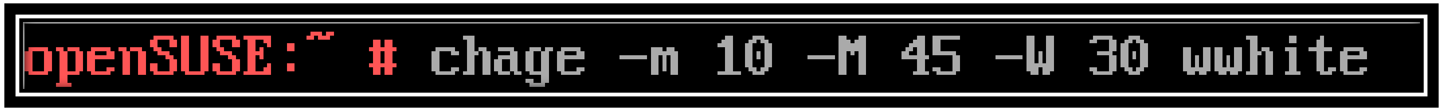
❑ Password aging

– Configure accounts so that passwords expire after a certain period of time

– Use chage command

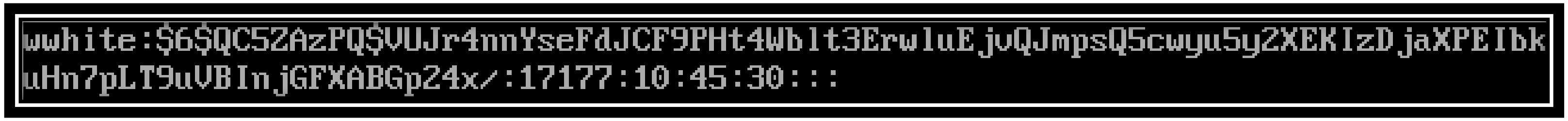
chage option user

- –m days Specifies the minimum number of days between password changes
- –M days Specifies the maximum number of days between password changes
- –W days Specifies the number of warning days before a password change is required



```
openSUSE:~ # chage -m 10 -M 45 -W 30 wwhite
```

/etc/shadow:



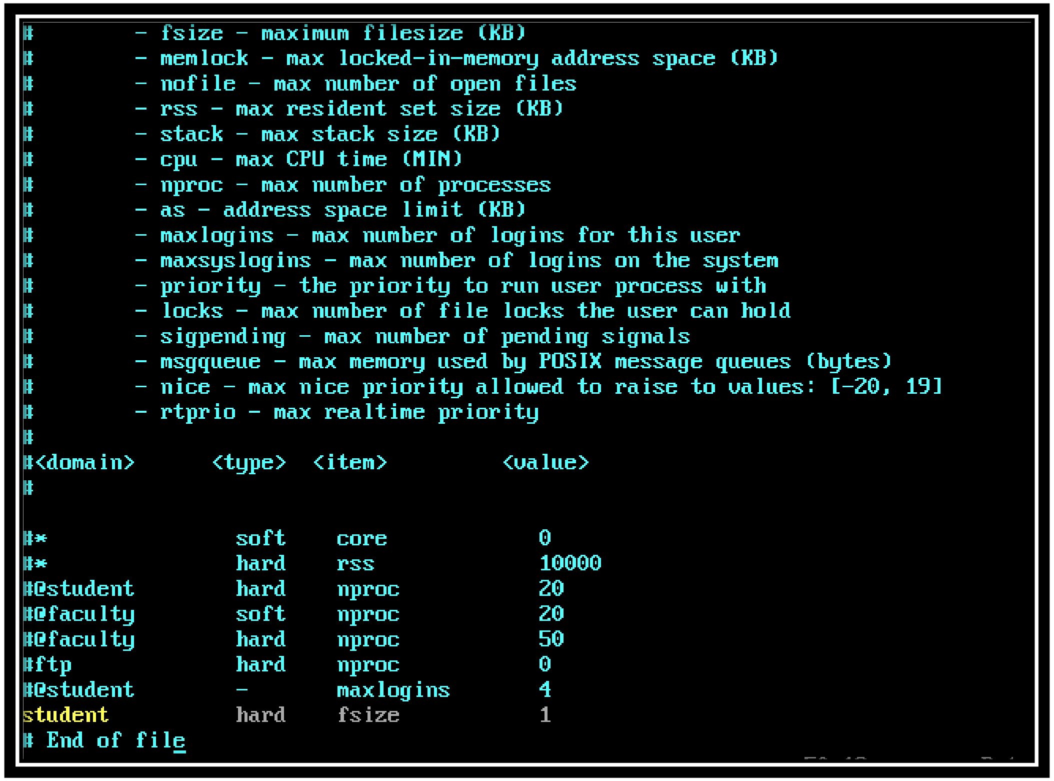
```
wwhite:$6$QC5ZAzPQ$VUJr4nnYseFdJCF9PHt4Wblt3ErwluEjvQJmpsQ5cwyu5y2XEKIzDjaXPEIbk
uHn7pLT9uVBInjGFXABGp24x/:17177:10:45:30:::
```

Configuring User Limits

- ❑ Using pam_limits to restrict access to resources
  - – Edit /etc/security/limits.conf
  - – 4 values to configure
    - domain: Describes the entity to which the limit applies. You can use one of the following values:
      - – user Identifies a specific Linux user
      - – @group_name Identifies a specific Linux group
      - – * Specifies all users
    - type: Defines a hard or soft limit. A hard limit cannot be exceeded. A soft limit can be temporarily exceeded.
    - item: Specifies the resource being limited.
    - value: Specifies a value for the limit.

Configuring User Limits

- ❑ In the below example there is a hard file size limit set for the user student of 1kb:



```
#        - fsize - maximum filesize (KB)
#        - memlock - max locked-in-memory address space (KB)
#        - nofile - max number of open files
#        - rss - max resident set size (KB)
#        - stack - max stack size (KB)
#        - cpu - max CPU time (MIN)
#        - nproc - max number of processes
#        - as - address space limit (KB)
#        - maxlogins - max number of logins for this user
#        - maxsyslogins - max number of logins on the system
#        - priority - the priority to run user process with
#        - locks - max number of file locks the user can hold
#        - sigpending - max number of pending signals
#        - msgqueue - max memory used by POSIX message queues (bytes)
#        - nice - max nice priority allowed to raise to values: [-20, 19]
#        - rtprio - max realtime priority
#
#<domain>      <type>  <item>         <value>
#

#*               soft    core           0
#*               hard    rss            10000
#@student        hard    nproc          20
#@faculty        soft    nproc          20
#@faculty        hard    nproc          50
#ftp             hard    nproc          0
#@student        -       maxlogins      4
student          hard    fsize          1
# End of file
```

- ❑ With that limit set the user can’t even log in as that amount (1k) will be or is already exceeded

Configuring User Limits

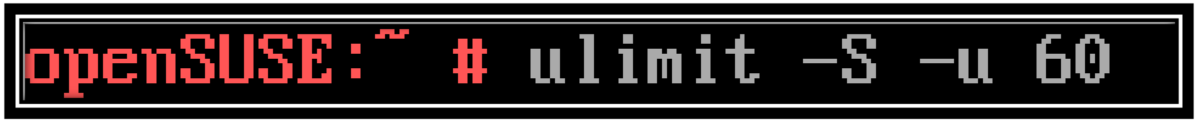
❑ Using ulimit to restrict access to resources

– The syntax for using ulimit is: ulimit options limit

- –c Sets a limit on the maximum size of core files in blocks. If you set this limit to a value of 0, core dumps on the system are disabled.
- –f Sets a limit on the maximum size (in blocks) of files created by the shell.
- –n Sets a limit on the maximum number of open file descriptors.
- –t Sets a limit on the maximum amount of CPU time (in seconds) a process may use.
- –u Sets a limit on the maximum number of processes available to a single user.
- –d Sets a limit on the maximum size (in KB) of a process's data segment in RAM.
- –m Sets a limit on the maximum resident size (in KB) of a process in RAM.
- –s Sets a limit on the maximum stack size (in KB).
- –H Sets a hard resource limit.
- –S Sets a soft resource limit.

Configuring User Limits

- □ **Example:**

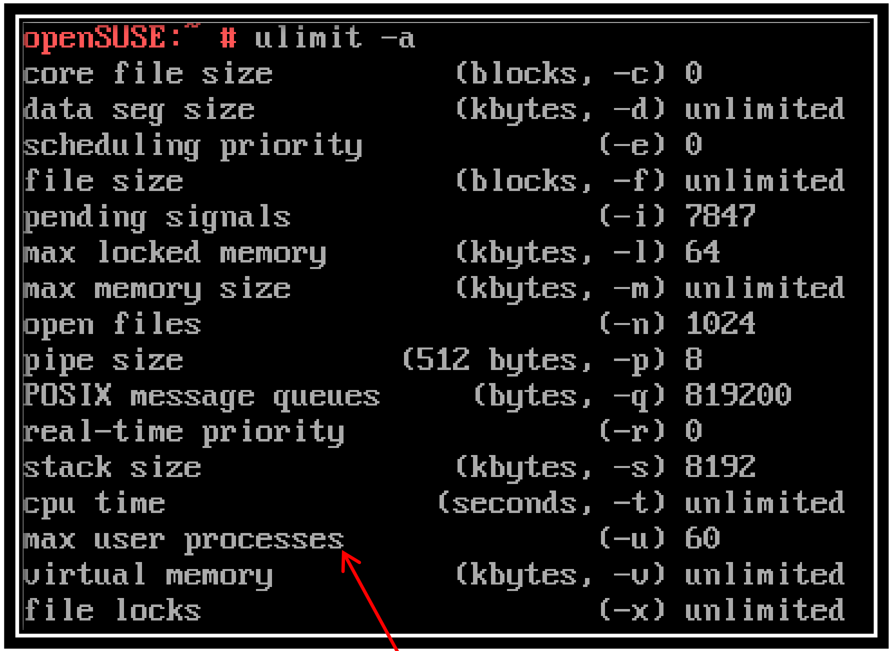


```
openSUSE:~ # ulimit -S -u 60
```

- □ **In the above example the ulimit utility was used to set a soft limit (-S) for the max processes available to a single user (-u) to 60**

Configuring User Limits

- ❑ The ulimit –a command will display the current value for all resource limits

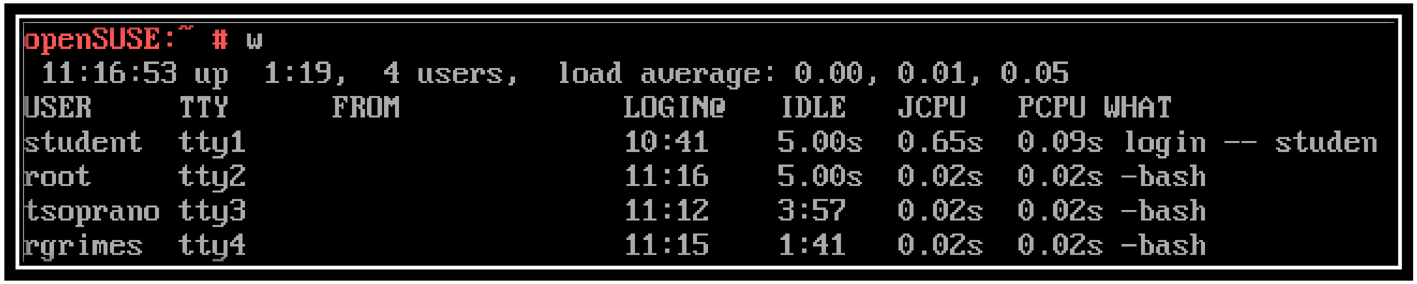


```
openSUSE:~ # ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 7847
max locked memory       (kbytes, -l) 64
max memory size         (kbytes, -m) unlimited
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes              (-u) 60
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
```

- ❑ Notice the limit we set in the prior slide

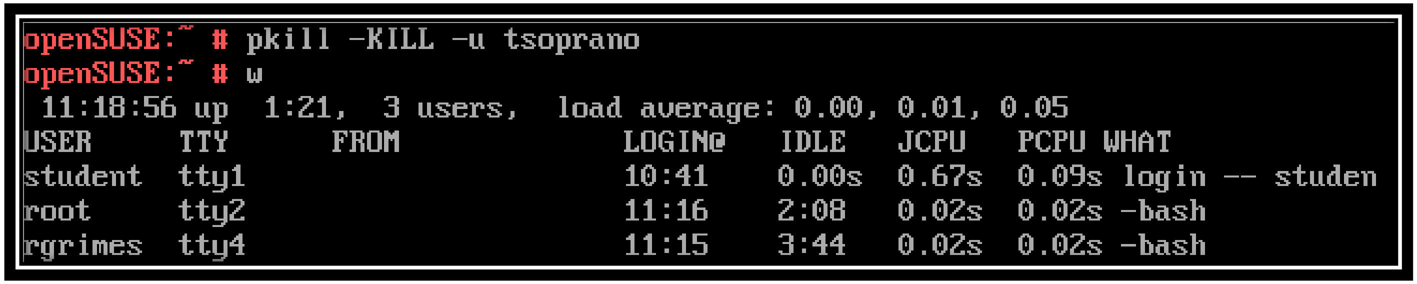
Disabling User Login

- ❑ You can use the w command to view all users logged onto the system:



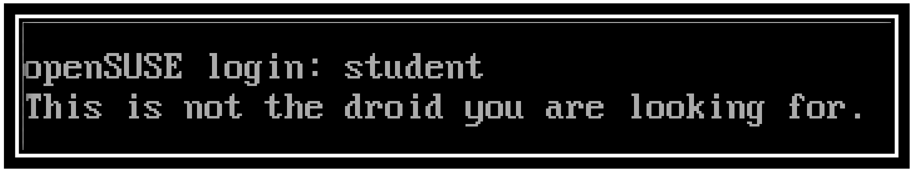
```
openSUSE:~ # w
 11:16:53 up  1:19,  4 users,  load average: 0.00, 0.01, 0.05
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
student  tty1                      10:41    5.00s  0.65s  0.09s login -- studen
root     tty2                      11:16    5.00s  0.02s  0.02s -bash
tsoprano tty3                      11:12    3:57   0.02s  0.02s -bash
rgrimes  tty4                      11:15    1:41   0.02s  0.02s -bash
```

- ❑ You can use the pkill utility to log a user out:



```
openSUSE:~ # pkill -KILL -u tsoprano
openSUSE:~ # w
 11:18:56 up  1:21,  3 users,  load average: 0.00, 0.01, 0.05
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
student  tty1                      10:41    0.00s  0.67s  0.09s login -- studen
root     tty2                      11:16    2:08   0.02s  0.02s -bash
rgrimes  tty4                      11:15    3:44   0.02s  0.02s -bash
```

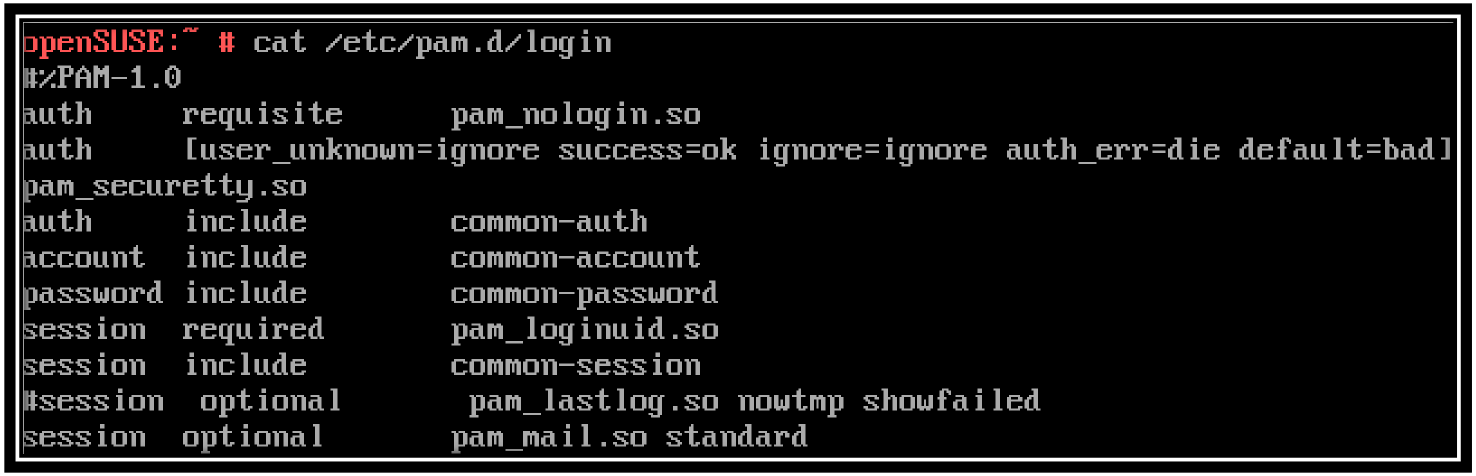
- ❑ You can restrict login of all users other than root by creating a nologin file in /etc



```
openSUSE login: student
This is not the droid you are looking for.
```

Disabling User Login

- ❑ The behavior is actually configured in the /etc/pam.d/login file but the nologin file must be created in /etc:



```
openSUSE:~ # cat /etc/pam.d/login
#%PAM-1.0
auth     requisite      pam_nologin.so
auth     [user_unknown=ignore success=ok ignore=ignore auth_err=die default=bad]
pam_securetty.so
auth     include        common-auth
account  include        common-account
password include        common-password
session  required       pam_loginuid.so
session  include        common-session
#session  optional       pam_lastlog.so nowtmp showfailed
session  optional       pam_mail.so standard
```

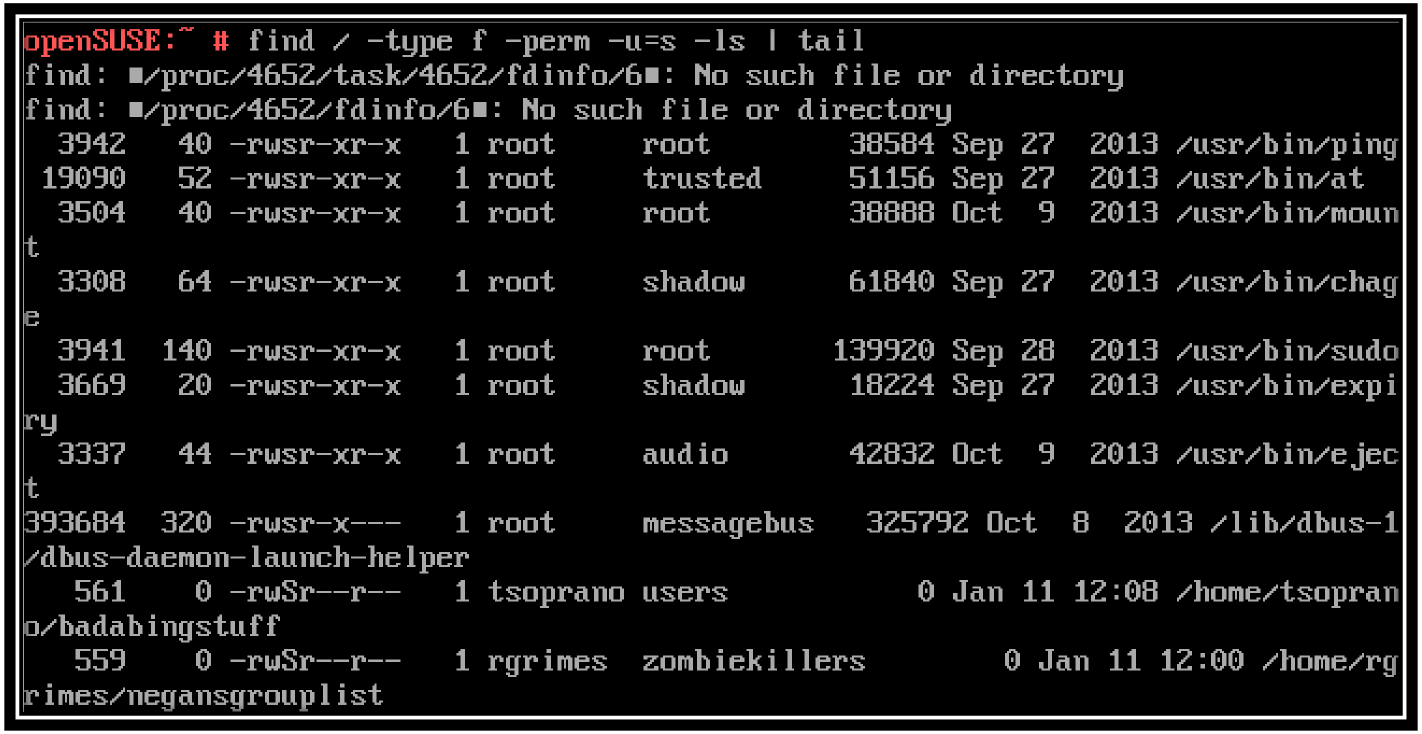
- ❑ The second line down in this file causes PAM to check for the existence of the nologin file in /etc
- ❑ If PAM finds the file, regular users are restricted
- ❑ Renaming or deleting no login will allow access

Auditing Files

❑ Auditing files with SUID permission set

– Periodic audits to identify any files owned by root that have either permission set should be considered

find / -type f –perm –u=s –ls

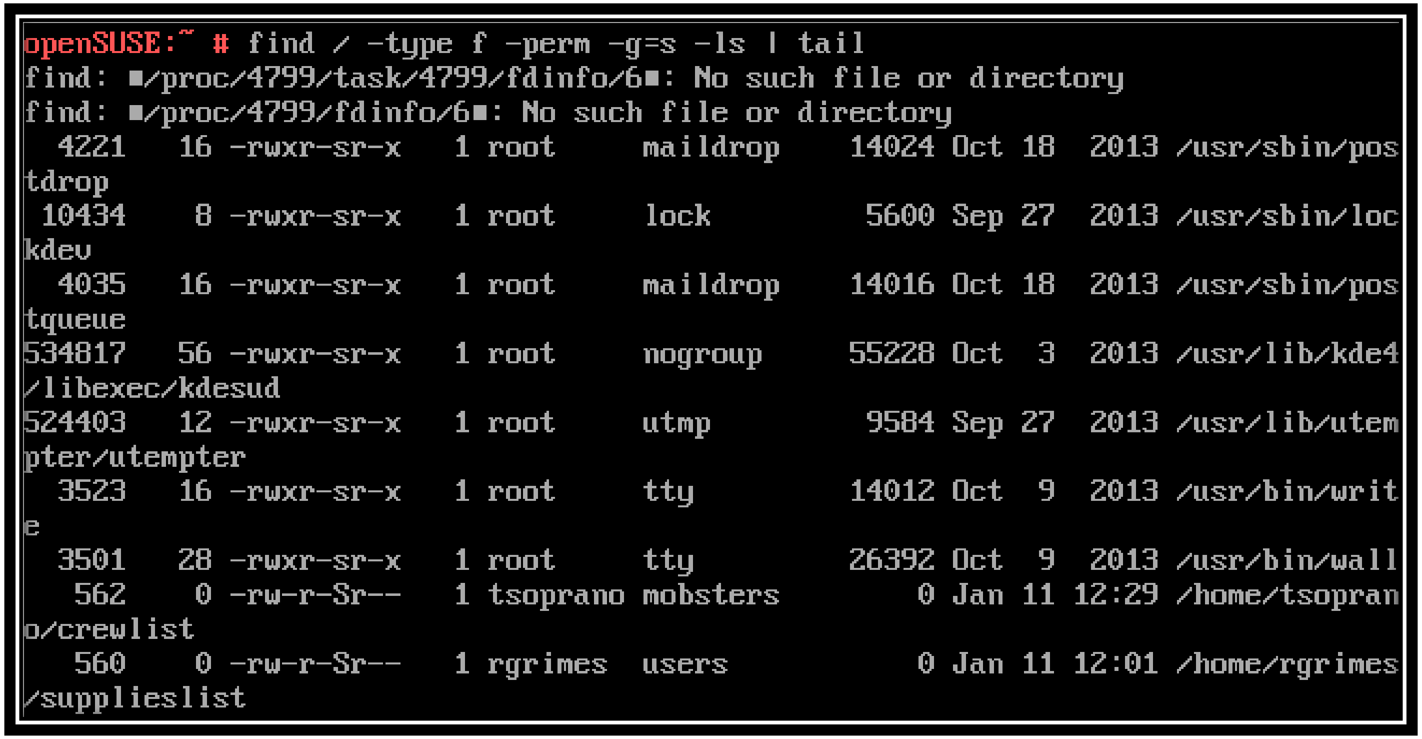


```
openSUSE:~ # find / -type f -perm -u=s -ls | tail
find: ■/proc/4652/task/4652/fdinfo/6■: No such file or directory
find: ■/proc/4652/fdinfo/6■: No such file or directory
  3942    40 -rwsr-xr-x   1 root     root          38584 Sep 27  2013 /usr/bin/ping
 19090    52 -rwsr-xr-x   1 root     trusted       51156 Sep 27  2013 /usr/bin/at
  3504    40 -rwsr-xr-x   1 root     root          38888 Oct  9  2013 /usr/bin/moun
t
  3308    64 -rwsr-xr-x   1 root     shadow        61840 Sep 27  2013 /usr/bin/chag
e
  3941   140 -rwsr-xr-x   1 root     root         139920 Sep 28  2013 /usr/bin/sudo
  3669    20 -rwsr-xr-x   1 root     shadow        18224 Sep 27  2013 /usr/bin/expi
ry
  3337    44 -rwsr-xr-x   1 root     audio         42832 Oct  9  2013 /usr/bin/ejec
t
393684   320 -rwsr-x---   1 root     messagebus    325792 Oct  8  2013 /lib/dbus-1
/dbus-daemon-launch-helper
   561     0 -rwSr--r--   1 tsoprano users             0 Jan 11 12:08 /home/tsopran
o/badabingstuff
   559     0 -rwSr--r--   1 rgrimes  zombiekillers          0 Jan 11 12:00 /home/rg
rimes/negansgrouplist
```

Auditing Files

- ❑ **Auditing files with SGID permission set**
  - – **Periodic audits to identify any files owned by root that have either permission set should be considered**

    find / -type f –perm –g=s –ls



```
openSUSE:~ # find / -type f -perm -g=s -ls | tail
find: ■/proc/4799/task/4799/fdinfo/6■: No such file or directory
find: ■/proc/4799/fdinfo/6■: No such file or directory
  4221   16 -rwxr-sr-x   1 root     maildrop    14024 Oct 18  2013 /usr/sbin/pos
tdrop
 10434    8 -rwxr-sr-x   1 root     lock         5600 Sep 27  2013 /usr/sbin/loc
kdev
  4035   16 -rwxr-sr-x   1 root     maildrop    14016 Oct 18  2013 /usr/sbin/pos
tqueue
534817   56 -rwxr-sr-x   1 root     nogroup     55228 Oct  3  2013 /usr/lib/kde4
/libexec/kdesud
524403   12 -rwxr-sr-x   1 root     utmp         9584 Sep 27  2013 /usr/lib/utem
pter/utempter
  3523   16 -rwxr-sr-x   1 root     tty         14012 Oct  9  2013 /usr/bin/writ
e
  3501   28 -rwxr-sr-x   1 root     tty         26392 Oct  9  2013 /usr/bin/wall
   562    0 -rw-r-Sr--   1 tsoprano mobsters        0 Jan 11 12:29 /home/tsopran
o/crewlist
   560    0 -rw-r-Sr--   1 rgrimes  users           0 Jan 11 12:01 /home/rgrimes
/supplieslist
```

Exercise 11-1: Managing User Access

Please open your Practical Exercise book to Exercise 11-1.

Time to Complete: 5 Minutes

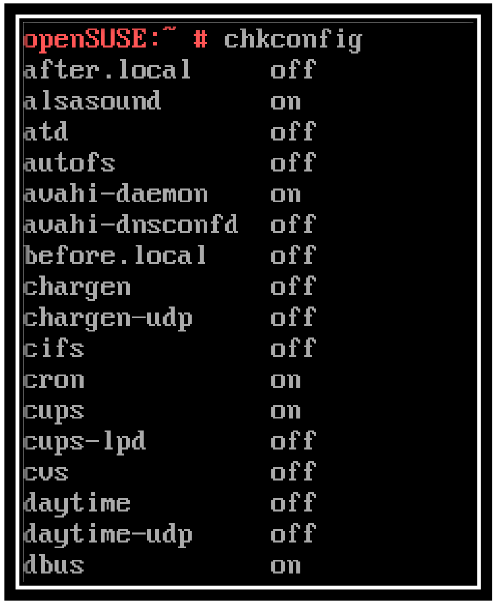
Mitigating Network Vulnerabilities

❑ **Staying abreast of current threats**

– Visit security-related websites on a regular basis

- Computer Emergency Response Team (CERT)
- US-CERT
- IAVM
- Higher echelon orders

❑ **Unloading Unneeded Services**



```
openSUSE:~ # chkconfig
after.local       off
alsasound         on
atd               off
autofs            off
avahi-daemon      on
avahi-dnsconfd    off
before.local      off
chargen           off
chargen-udp       off
cifs              off
cron              on
cups              on
cups-lpd          off
cvs               off
daytime           off
daytime-udp       off
dbus              on
```

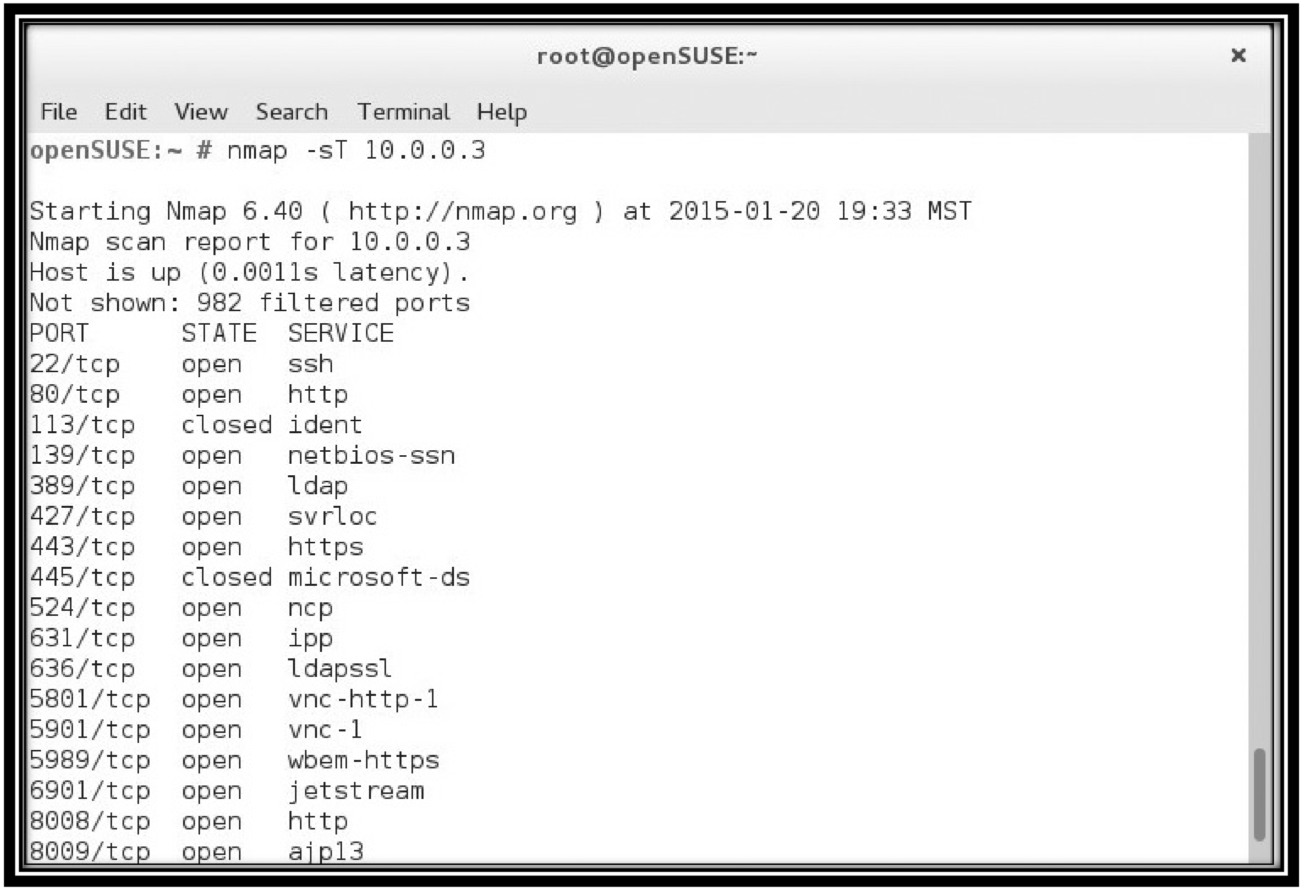
Do not disable a service
until you know what it is used for!

Use the man utility, info utility or
A trusted site to research.

A port scanner like nmap
can also be used

Mitigating Network Vulnerabilities

- ❑ **Using nmap to scan for open ports**

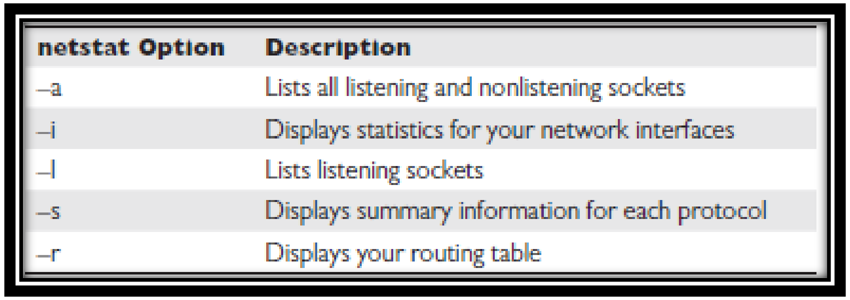


```
openSUSE:~ # nmap -sT 10.0.0.3

Starting Nmap 6.40 ( http://nmap.org ) at 2015-01-20 19:33 MST
Nmap scan report for 10.0.0.3
Host is up (0.0011s latency).
Not shown: 982 filtered ports
PORT      STATE  SERVICE
22/tcp    open   ssh
80/tcp    open   http
113/tcp   closed ident
139/tcp   open   netbios-ssn
389/tcp   open   ldap
427/tcp   open   svrloc
443/tcp   open   https
445/tcp   closed microsoft-ds
524/tcp   open   ncp
631/tcp   open   ipp
636/tcp   open   ldapssl
5801/tcp  open   vnc-http-1
5901/tcp  open   vnc-1
5989/tcp  open   wbem-https
6901/tcp  open   jetstream
8008/tcp  open   http
8009/tcp  open   ajp13
```

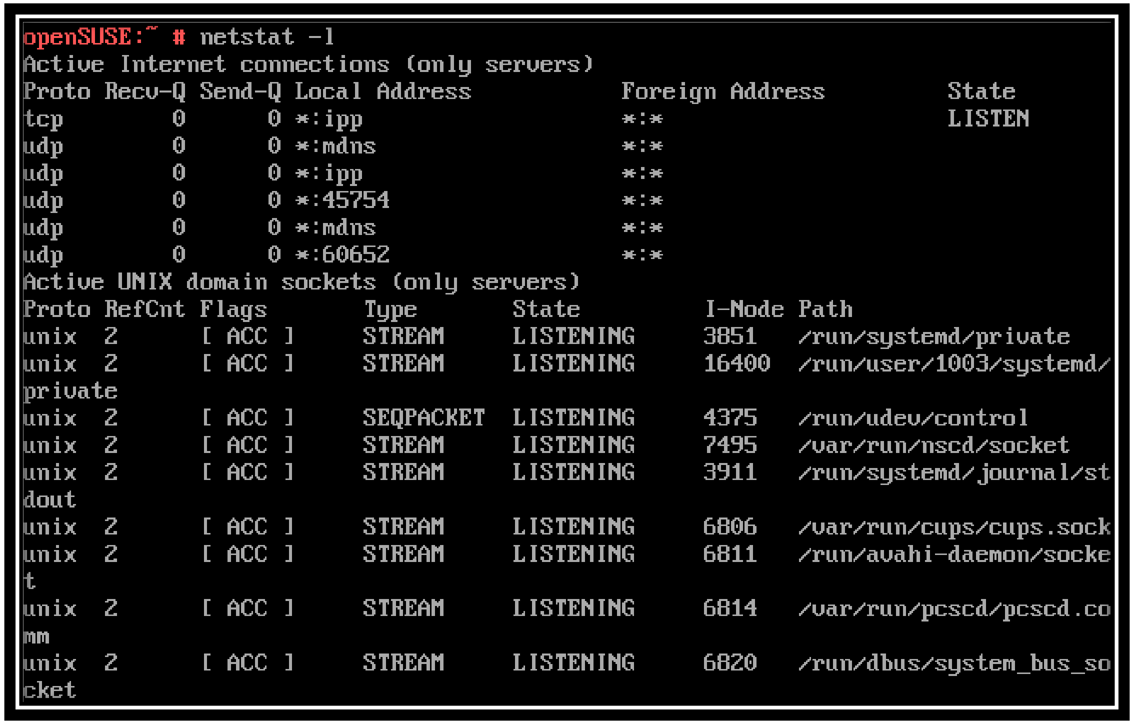
- ❑ **A scan for TCP port do to the T option**
- ❑ **UDP can be scanned with the U option**

Mitigating Network Vulnerabilities

- ❑ Using netstat to scan for open ports



netstat Option	Description
–a	Lists all listening and nonlistening sockets
–i	Displays statistics for your network interfaces
–l	Lists listening sockets
–s	Displays summary information for each protocol
–r	Displays your routing table



```
openSUSE:~ # netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:ipp                   *:*                     LISTEN
udp        0      0 *:mdns                  *:*
udp        0      0 *:ipp                   *:*
udp        0      0 *:45754                 *:*
udp        0      0 *:mdns                  *:*
udp        0      0 *:60652                 *:*
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node Path
unix  2      [ ACC ]     STREAM     LISTENING     3851   /run/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     16400  /run/user/1003/systemd/
private
unix  2      [ ACC ]     SEQPACKET  LISTENING     4375   /run/udev/control
unix  2      [ ACC ]     STREAM     LISTENING     7495   /var/run/nscd/socket
unix  2      [ ACC ]     STREAM     LISTENING     3911   /run/systemd/journal/st
dout
unix  2      [ ACC ]     STREAM     LISTENING     6806   /var/run/cups/cups.sock
unix  2      [ ACC ]     STREAM     LISTENING     6811   /run/avahi-daemon/socke
t
unix  2      [ ACC ]     STREAM     LISTENING     6814   /var/run/pcscd/pcscd.co
mm
unix  2      [ ACC ]     STREAM     LISTENING     6820   /run/dbus/system_bus_so
cket
```

Implementing a Firewall with iptables

❑ Implementing a packet-filtering firewall

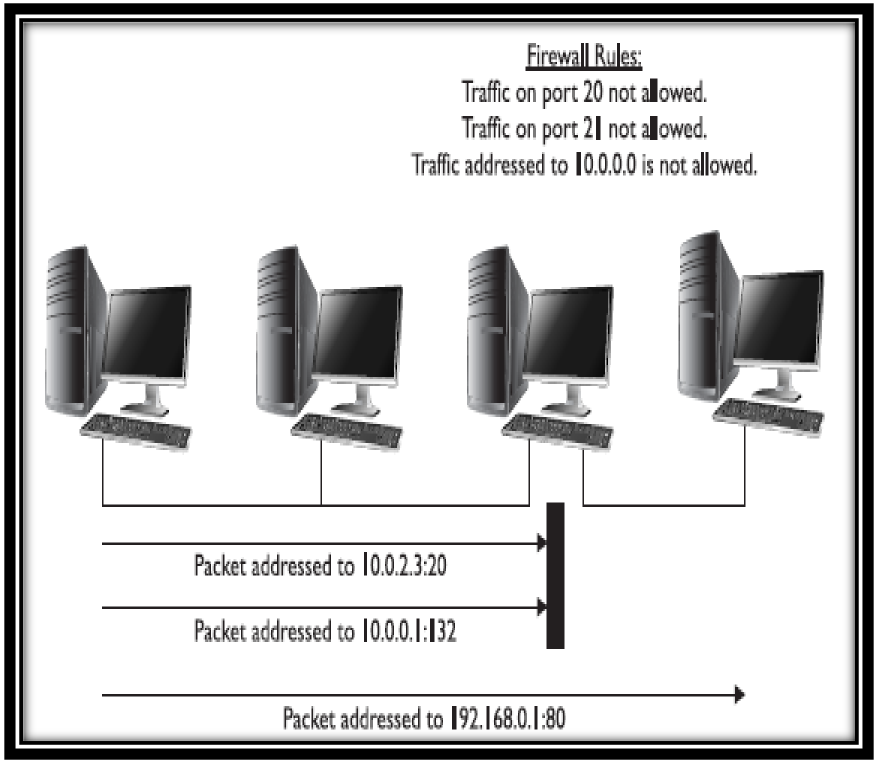
– Will you allow all incoming traffic by default, establishing rules for specific types of traffic that you don't want to allow in?

– Will your firewall deny all incoming traffic except for specific types of traffic that you want to allow?

– Will you allow all outgoing traffic by default, blocking only specific types or destinations?

– Will you block all outgoing traffic except for specific types or destinations?

– What ports must be opened on the firewall to allow traffic through from the outside?


Firewall Rules:
Traffic on port 20 not allowed.
Traffic on port 21 not allowed.
Traffic addressed to 10.0.0.0 is not allowed.
Packet addressed to 10.0.2.3:20
Packet addressed to 10.0.0.1:132
Packet addressed to 192.168.0.1:80

Implementing a Firewall with iptables

- ❑ In order to use iptables, your kernel must comply with the netfilter infrastructure
- ❑ The netfilter infrastructure uses the concept of “tables and chains” to create firewall rules
- ❑ A chain is simply a rule that you implement to determine what the firewall will do with an incoming packet
- ❑ The netfilter infrastructure uses the filter table to create packet-filtering rules

Implementing a Firewall with iptables

- Within the filter table are three default chains:
  - FORWARD: contains rules for packets being transferred between networks through the Linux system.
  - INPUT: contains rules for packets that are being sent to the local Linux system.
  - OUTPUT: contains rules for packets that are being sent from the local Linux system.
- If you don’t explicitly specify a table name when using the iptables utility, it will default to the filter table. Each chain in the filter table has four policies that you can configure:
  - ACCEPT
  - DROP
  - QUEUE
  - REJECT

Implementing a Firewall with iptables

- ❑ The syntax for using iptables is
  - – iptables –t table command chain options

- ❑ Creating additional chains by using the –A option
  - – –I Inserts a rule into the chain
  - – –R Replaces a rule in the chain
  - – –D Deletes a rule from the chain
  - – –F Deletes all the rules from the chain (called flushing)
  - – –P Sets the default policy for the chain

Implementing a Firewall with iptables

❑ Additional options

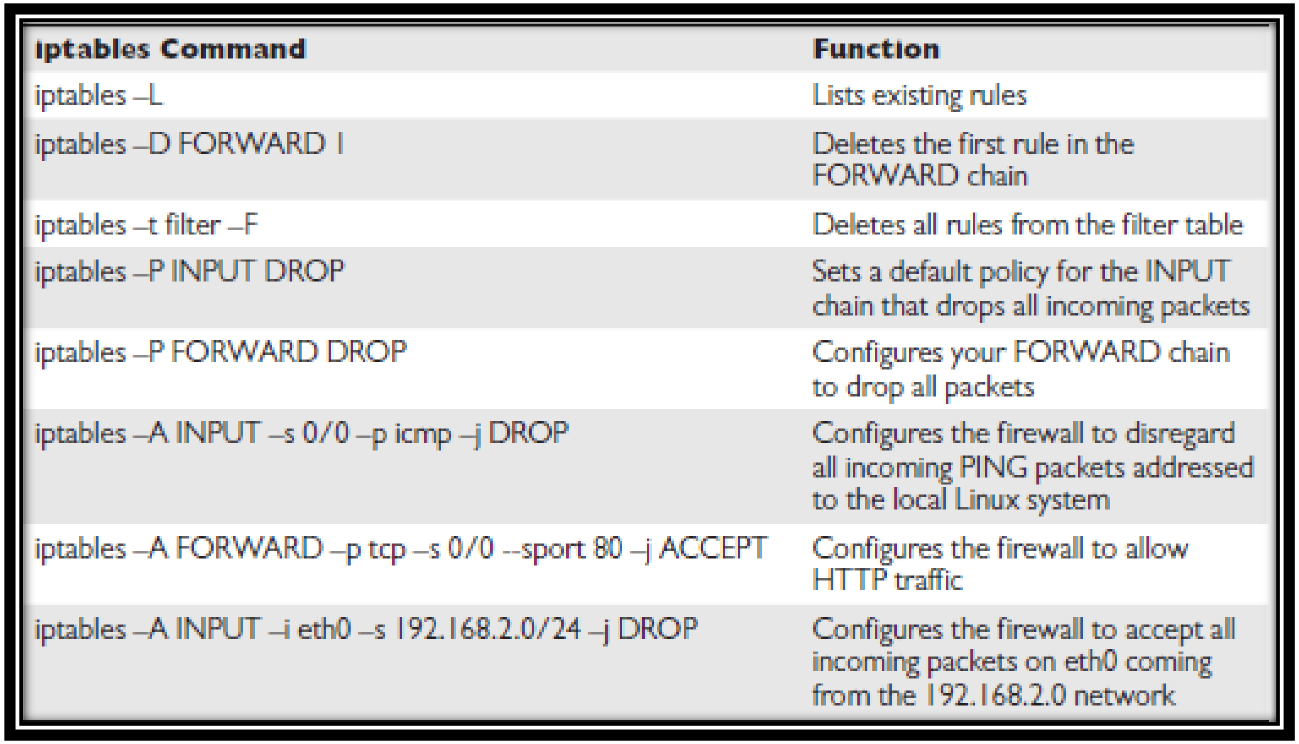
- –p Specifies the protocol to be checked by the rule. You can specify all, tcp, udp, or icmp.
  - --sport Specifies a single port to match on
  - --dport Specifies a single destination port to match on
  - --sports Specifies multiple source ports to match on
  - --dports Specifies multiple destination ports to match on

Implementing a Firewall with iptables

- ❑ **Additional options (cont.)**
  - **–s ip_address/mask Specifies the source address to be checked. If you want to check all IP addresses, use 0/0.**
  - **–d ip_address/mask Specifies the destination address to be checked. If you want to check all IP addresses, use 0/0.**
  - **–j target Specifies what to do if the packet matches the rule. You can specify ACCEPT, REJECT, DROP, or LOG actions.**
  - **–i interface Specifies the interface where a packet is received. This only applies to INPUT and FORWARD chains.**
  - **–o interface Specifies the interface where a packet is to be sent. This applies only to OUTPUT and FORWARD chains.**

Implementing a Firewall with iptables

- ❑ Some example iptables commands:



iptables Command	Function
iptables –L	Lists existing rules
iptables –D FORWARD 1	Deletes the first rule in the FORWARD chain
iptables –t filter –F	Deletes all rules from the filter table
iptables –P INPUT DROP	Sets a default policy for the INPUT chain that drops all incoming packets
iptables –P FORWARD DROP	Configures your FORWARD chain to drop all packets
iptables –A INPUT –s 0/0 –p icmp –j DROP	Configures the firewall to disregard all incoming PING packets addressed to the local Linux system
iptables –A FORWARD –p tcp –s 0/0 --sport 80 –j ACCEPT	Configures the firewall to allow HTTP traffic
iptables –A INPUT –i eth0 –s 192.168.2.0/24 –j DROP	Configures the firewall to accept all incoming packets on eth0 coming from the 192.168.2.0 network

- ❑ Rules created by iptables are not persistent
- ❑ The iptables-save command saves them to a file
- ❑ The iptables-restore command restores them

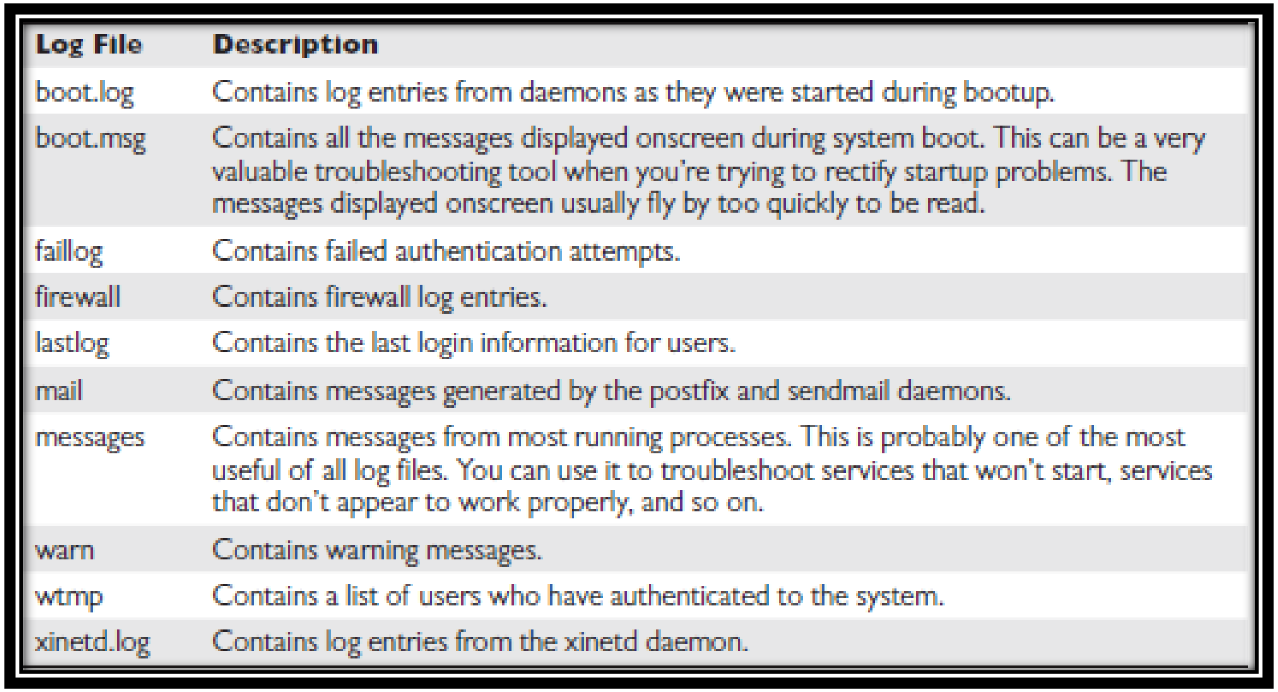
Exercise 11-2: Implementing Network Security Measures on Linux

Please open your Practical Exercise book to Exercise 11-2.

Time to Complete: 5 Minutes

Configuring Logs

- ❑ Log files are important sources of information
  - Security, troubleshooting and administration
- ❑ System log files are stored in the var/log directory
  - Some are text and some are binaries
  - Some are more useful than others



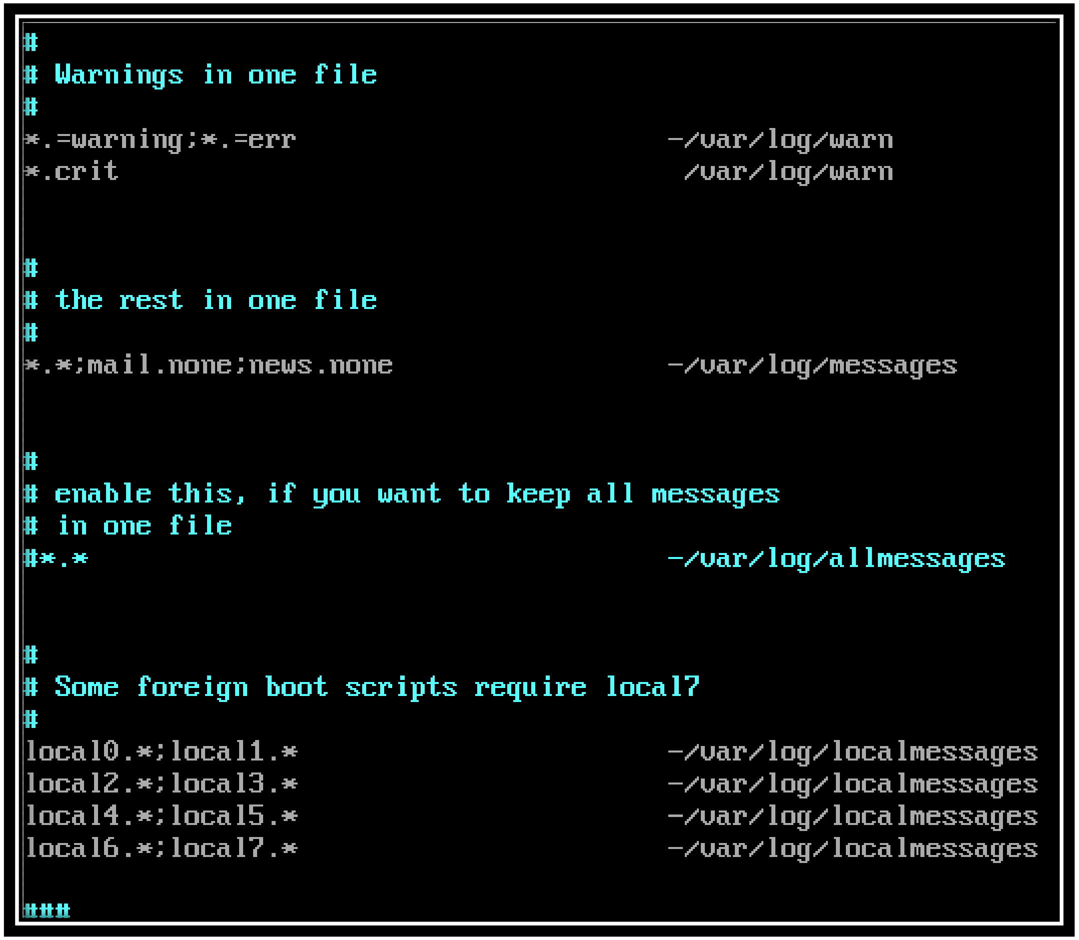
Log File	Description
boot.log	Contains log entries from daemons as they were started during bootup.
boot.msg	Contains all the messages displayed onscreen during system boot. This can be a very valuable troubleshooting tool when you're trying to rectify startup problems. The messages displayed onscreen usually fly by too quickly to be read.
faillog	Contains failed authentication attempts.
firewall	Contains firewall log entries.
lastlog	Contains the last login information for users.
mail	Contains messages generated by the postfix and sendmail daemons.
messages	Contains messages from most running processes. This is probably one of the most useful of all log files. You can use it to troubleshoot services that won't start, services that don't appear to work properly, and so on.
warn	Contains warning messages.
wtmp	Contains a list of users who have authenticated to the system.
xinetd.log	Contains log entries from the xinetd daemon.

Configuring Logs

- ❑ Logging is implemented differently depending on the distribution used
- ❑ syslogd, journald, syslog-ng, and rsyslogd are some implementations
- ❑ The logging daemon your system uses is configured in /etc/sysconfig/syslog

Configuring Logs

- /etc/syslog.conf (in the openSUSE we are using for practical exercises /etc/rsyslog.conf)



```
#
# Warnings in one file
#
*.=warning;*.=err                       -/var/log/warn
*.crit                                   /var/log/warn


#
# the rest in one file
#
*.*;mail.none;news.none                 -/var/log/messages


#
# enable this, if you want to keep all messages
# in one file
#*.*                                    -/var/log/allmessages


#
# Some foreign boot scripts require local7
#
local0.*;local1.*                       -/var/log/localmessages
local2.*;local3.*                       -/var/log/localmessages
local4.*;local5.*                       -/var/log/localmessages
local6.*;local7.*                       -/var/log/localmessages

###
```

Configuring Logs

❑ **The syntax for the syslog.conf file is**

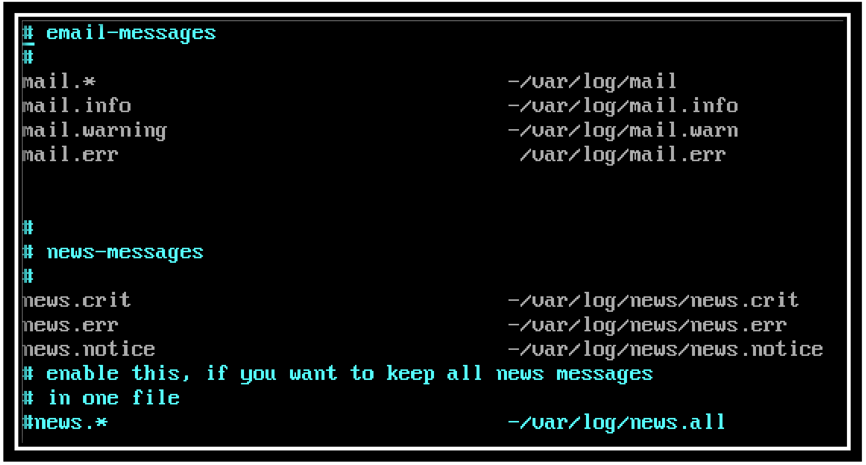
facility.priority   file

Facility

- **authpriv:** Facility used by all services associated with system security or authorization
- **cron:** Facility that accepts log messages from cron and at
- **daemon:** Facility that can be used by daemons that do not have their own facility
- **kern:** Facility used for all kernel log messages
- **lpr:** Facility that handles messages from the printing system
- **mail:** Facility for log messages from the mail MTA (such as postfix or sendmail)
- **news:** Facility for log messages from the news daemon
- **syslog:** Facility for internal messages from the syslog daemon itself
- **user:** Facility for user-related log messages (such as failed login attempts)
- **uucp:** Facility for log messages from the uucp daemon
- **local0–local7:** Facilities you can use to capture log messages from your own applications that you develop

Priorities

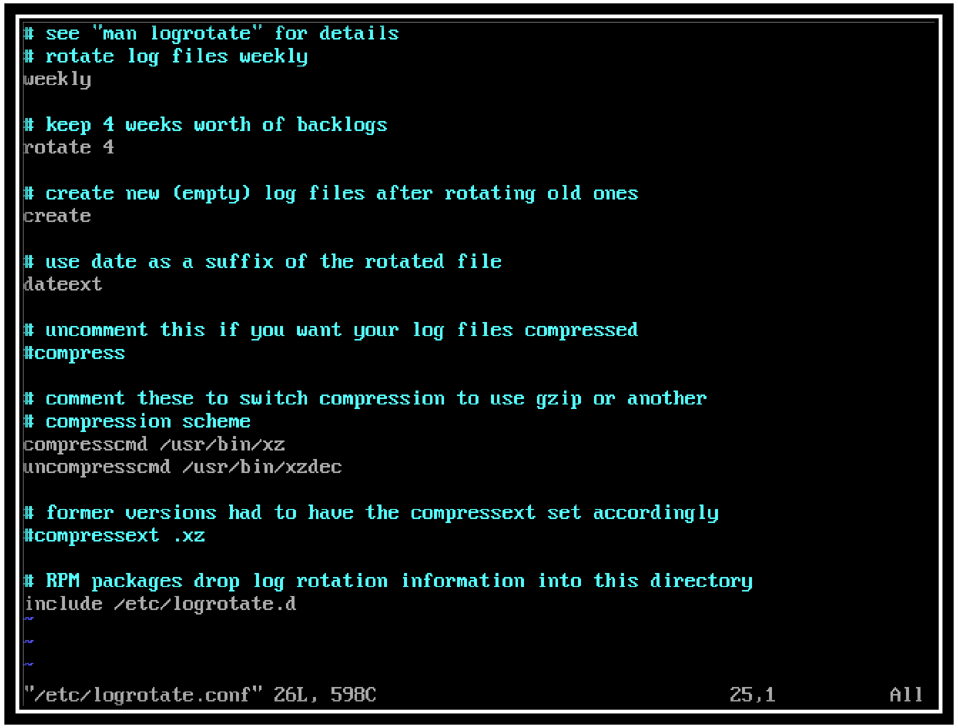
- **debug:** All information
- **info:** Informational messages
- **notice:** Issues of concern, but not yet a problem
- **warn:** Noncritical errors
- **err:** Serious errors
- **crit, alert, or emerg:** Critical errors
- *****: all priorities



```
# email-messages
#
mail.*                                  -/var/log/mail
mail.info                               -/var/log/mail.info
mail.warning                            -/var/log/mail.warn
mail.err                                 /var/log/mail.err


#
# news-messages
#
news.crit                               -/var/log/news/news.crit
news.err                                -/var/log/news/news.err
news.notice                             -/var/log/news/news.notice
# enable this, if you want to keep all news messages
# in one file
#news.*                                 -/var/log/news.all
```

Configuring Logs

- ❑ The logrotate utility is run daily, by default, by the cron daemon on your system
- ❑ The /etc/logrotate.conf file contains default global paramaters used by logrotate to determine how and when logs are rotated

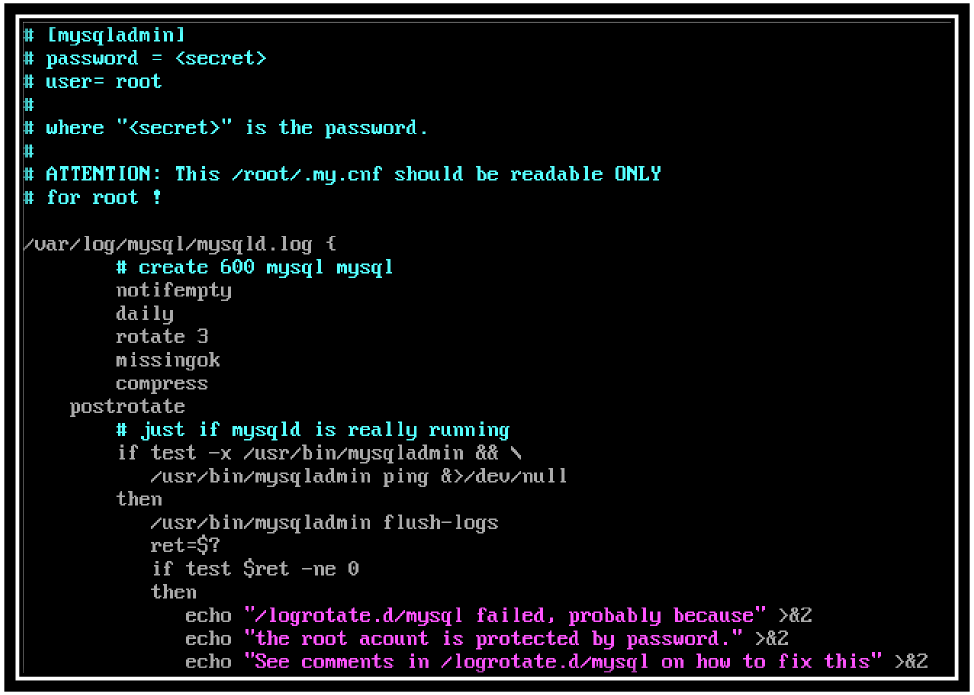


```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# use date as a suffix of the rotated file
dateext

# uncomment this if you want your log files compressed
#compress

# comment these to switch compression to use gzip or another
# compression scheme
compresscmd /usr/bin/xz
uncompresscmd /usr/bin/xzdec

# former versions had to have the compressext set accordingly
#compressext .xz

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d
~
~
~
"/etc/logrotate.conf" 26L, 598C                              25,1          All
```

- ❑ Defaults can be overwritten by daemons via files in the /etc/lograte.d directory

Configuring Logs

- ❑ The below example is of the sql file in the /etc/logrotate.d directory:



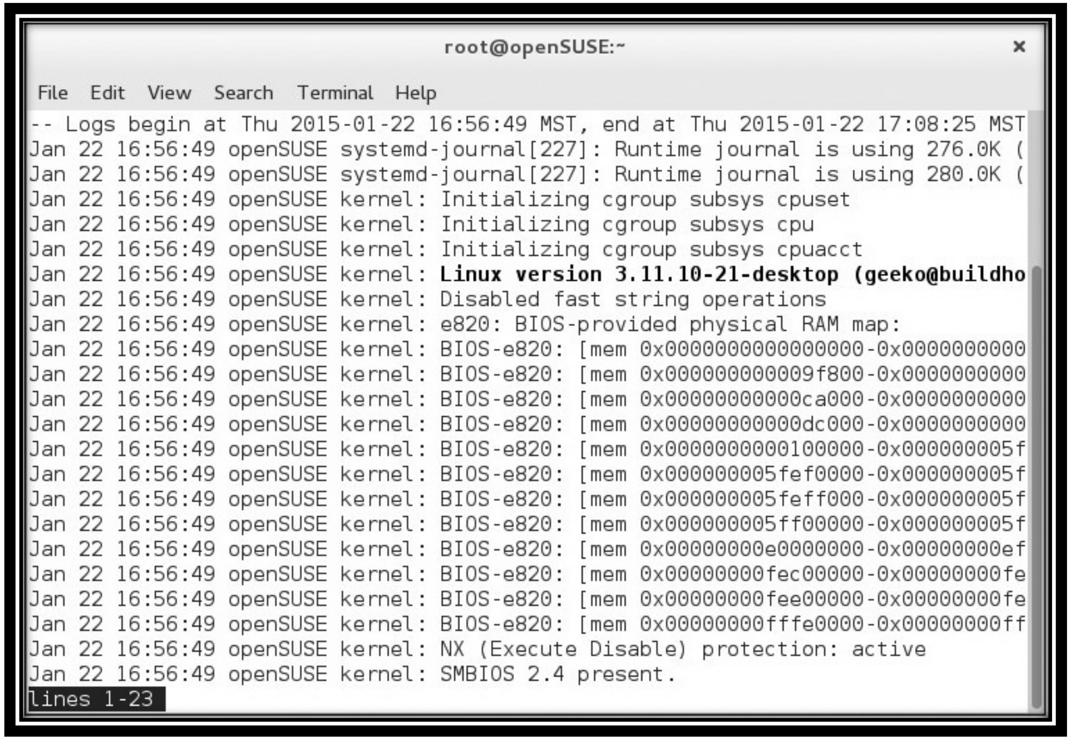
```
# [mysqladmin]
# password = <secret>
# user= root
#
# where "<secret>" is the password.
#
# ATTENTION: This /root/.my.cnf should be readable ONLY
# for root !

/var/log/mysql/mysqld.log {
        # create 600 mysql mysql
        notifempty
        daily
        rotate 3
        missingok
        compress
    postrotate
        # just if mysqld is really running
        if test -x /usr/bin/mysqladmin && \
           /usr/bin/mysqladmin ping &>/dev/null
        then
           /usr/bin/mysqladmin flush-logs
           ret=$?
           if test $ret -ne 0
           then
              echo "/logrotate.d/mysql failed, probably because" >&2
              echo "the root acount is protected by password." >&2
              echo "See comments in /logrotate.d/mysql on how to fix this" >&2
```

- ❑ This file will rotate 3 times, it will not be rotated if it is empty (notifempty), no error will be generated if the file is missing (missingok), the file will be compressed and the file will be rotated daily

- ❑ More options are available, check the manual for logrotate

Configuring Logs

- ❑ Some Linux distributions that use the systemd daemon use the journald daemon for logging
- ❑ The journald daemon maintains a system log called the journal (located in /var/log/journal/)
- ❑ The journalctl command can be used to view the journal:



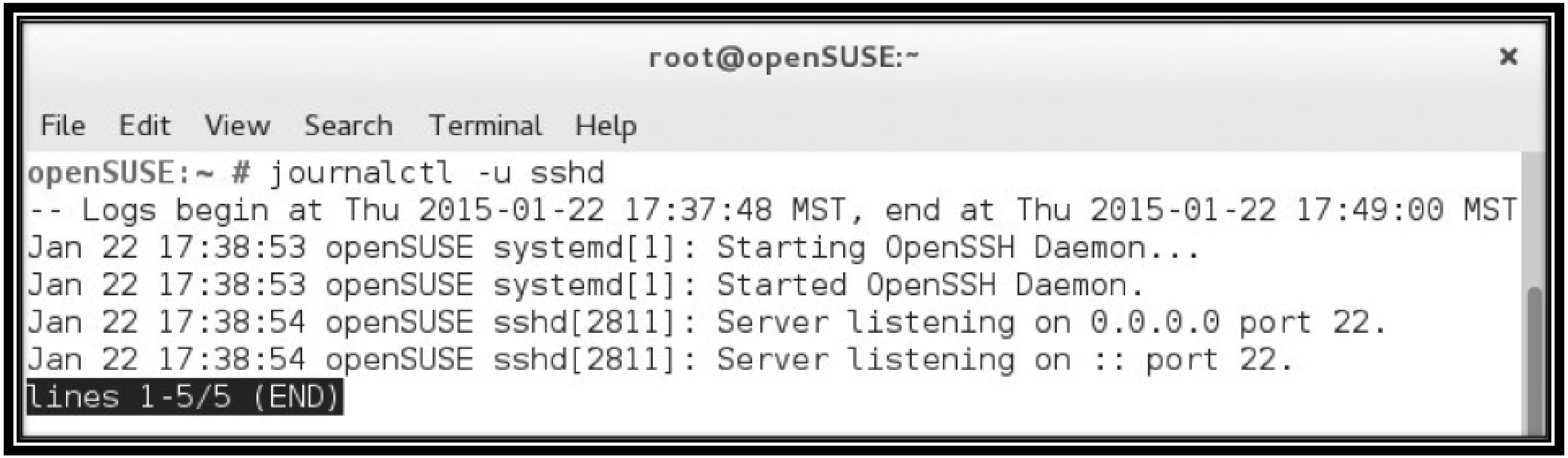
```
-- Logs begin at Thu 2015-01-22 16:56:49 MST, end at Thu 2015-01-22 17:08:25 MST
Jan 22 16:56:49 openSUSE systemd-journal[227]: Runtime journal is using 276.0K (
Jan 22 16:56:49 openSUSE systemd-journal[227]: Runtime journal is using 280.0K (
Jan 22 16:56:49 openSUSE kernel: Initializing cgroup subsys cpuset
Jan 22 16:56:49 openSUSE kernel: Initializing cgroup subsys cpu
Jan 22 16:56:49 openSUSE kernel: Initializing cgroup subsys cpuacct
Jan 22 16:56:49 openSUSE kernel: Linux version 3.11.10-21-desktop (geeko@buildho
Jan 22 16:56:49 openSUSE kernel: Disabled fast string operations
Jan 22 16:56:49 openSUSE kernel: e820: BIOS-provided physical RAM map:
Jan 22 16:56:49 openSUSE kernel: BIOS-e820: [mem 0x0000000000000000-0x0000000000
Jan 22 16:56:49 openSUSE kernel: BIOS-e820: [mem 0x000000000009f800-0x0000000000
Jan 22 16:56:49 openSUSE kernel: BIOS-e820: [mem 0x00000000000ca000-0x0000000000
Jan 22 16:56:49 openSUSE kernel: BIOS-e820: [mem 0x00000000000dc000-0x0000000000
Jan 22 16:56:49 openSUSE kernel: BIOS-e820: [mem 0x0000000000100000-0x000000005f
Jan 22 16:56:49 openSUSE kernel: BIOS-e820: [mem 0x000000005fef0000-0x000000005f
Jan 22 16:56:49 openSUSE kernel: BIOS-e820: [mem 0x000000005feff000-0x000000005f
Jan 22 16:56:49 openSUSE kernel: BIOS-e820: [mem 0x000000005ff00000-0x000000005f
Jan 22 16:56:49 openSUSE kernel: BIOS-e820: [mem 0x00000000e0000000-0x00000000ef
Jan 22 16:56:49 openSUSE kernel: BIOS-e820: [mem 0x00000000fec00000-0x00000000fe
Jan 22 16:56:49 openSUSE kernel: BIOS-e820: [mem 0x00000000fee00000-0x00000000fe
Jan 22 16:56:49 openSUSE kernel: BIOS-e820: [mem 0x00000000fffe0000-0x00000000ff
Jan 22 16:56:49 openSUSE kernel: NX (Execute Disable) protection: active
Jan 22 16:56:49 openSUSE kernel: SMBIOS 2.4 present.
lines 1-23
```

Configuring Logs

- The journalctl command can be used with options
  - -b: view system boot messages
    - Example: journalctl –b 1 will display messages created during the first boot at the beginning of the journal
    - Example: journalctl –b -2 will display system messages created two boots ago
  - -u service_name: displays only log entries related to a specific service running
    - Example: journalctl –u sshd will display all entries related to the SSH daemon
  - The behavior of the journal daemon is configured using the /etc/systemd/journald.conf file and it has several parameters that can be configured:
    - MaxFileSec: maximum amount of time to store entries before starting new file
    - MaxRetentionSec: maximum amount of time to store journal entries. Entries older than the specified time are deleted
    - ForwardtoSyslog: sets journald to forward its log messages to the syslog daemon
    - MaxLevelStore: Controls the maximum log level of messages. All messages equal to or less than the log level are stored, messages above are dropped
      - Emerg (0), alert(1), crit (2), err (3), warning (4), notice (5), info (6), debug (7)

Configuring Logs

- ❑ **An example of journalctl with the –u option used:**



```
openSUSE:~ # journalctl -u sshd
-- Logs begin at Thu 2015-01-22 17:37:48 MST, end at Thu 2015-01-22 17:49:00 MST
Jan 22 17:38:53 openSUSE systemd[1]: Starting OpenSSH Daemon...
Jan 22 17:38:53 openSUSE systemd[1]: Started OpenSSH Daemon.
Jan 22 17:38:54 openSUSE sshd[2811]: Server listening on 0.0.0.0 port 22.
Jan 22 17:38:54 openSUSE sshd[2811]: Server listening on :: port 22.
lines 1-5/5 (END)
```

Using Log Files to Troubleshoot Problems

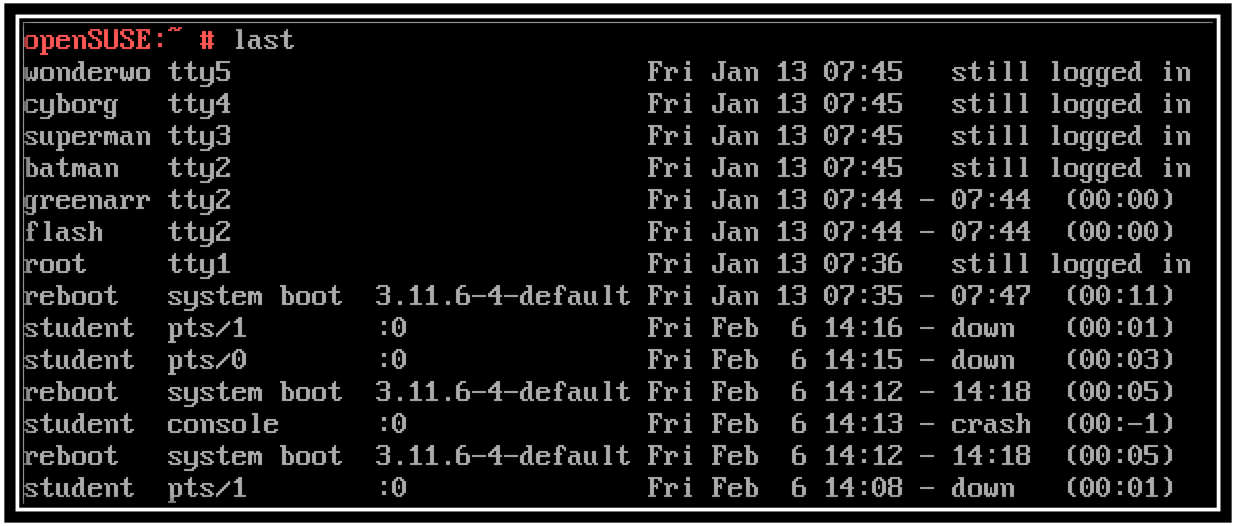
❑ As mentioned in an earlier slide log files are an invaluable resource when troubleshooting

❑ Some log files can have thousands of entries so viewing them efficiently is key

- May have an application that consolidates and aggregates
- Options like more, less, tail, head, and grep help as well
- Piping to a file or another utility can be helpful
- Configuration in files used to configure logging can reduce the amount of entries (two slides back with journal)

Using Log Files to Detect Intruders

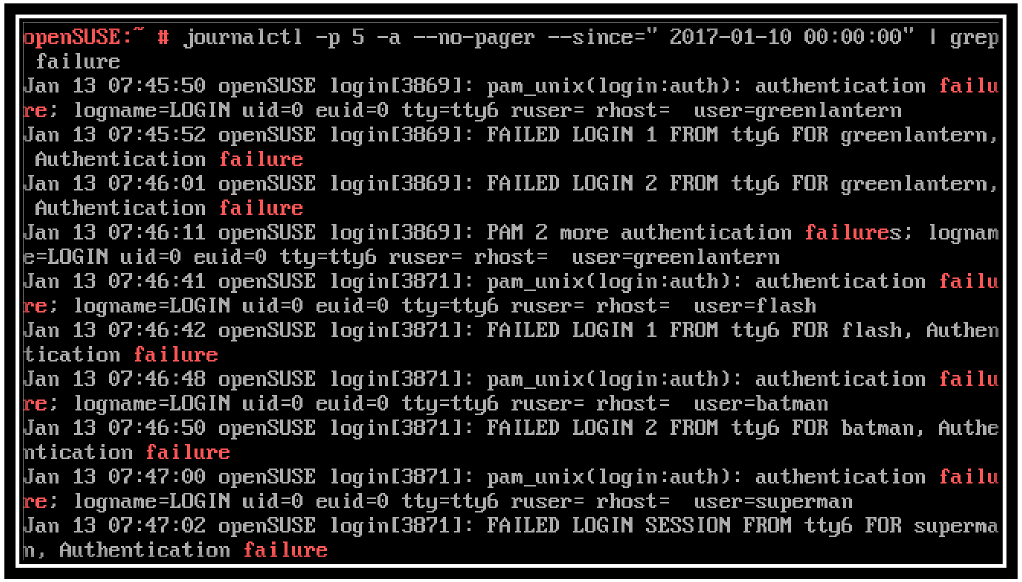
- ❑ Log files are reviewed by administrators constantly
  - In accordance with policy and regulations
  - Depending on security environment
  - If an incident is identified and a response started
- ❑ There are log files that record currently logged on users
  - /var/log/wtmp which can be viewed with the last utility:



```
openSUSE:~ # last
wonderwo tty5                         Fri Jan 13 07:45   still logged in
cyborg   tty4                         Fri Jan 13 07:45   still logged in
superman tty3                         Fri Jan 13 07:45   still logged in
batman   tty2                         Fri Jan 13 07:45   still logged in
greenarr tty2                         Fri Jan 13 07:44 - 07:44  (00:00)
flash    tty2                         Fri Jan 13 07:44 - 07:44  (00:00)
root     tty1                         Fri Jan 13 07:36   still logged in
reboot   system boot  3.11.6-4-default Fri Jan 13 07:35 - 07:47  (00:11)
student  pts/1        :0              Fri Feb  6 14:16 - down   (00:01)
student  pts/0        :0              Fri Feb  6 14:15 - down   (00:03)
reboot   system boot  3.11.6-4-default Fri Feb  6 14:12 - 14:18  (00:05)
student  console      :0              Fri Feb  6 14:13 - crash  (00:-1)
reboot   system boot  3.11.6-4-default Fri Feb  6 14:12 - 14:18  (00:05)
student  pts/1        :0              Fri Feb  6 14:08 - down   (00:01)
```

Using Log Files to Detect Intruders

- Some Linux distributions used to support a utility called faillog
  - Read the /var/log/faillog which recorded, you guessed it, failed logins
  - Currently depreciated (faillog) but the file does still exist
- Can use the journalctl utility to identify failed logins:

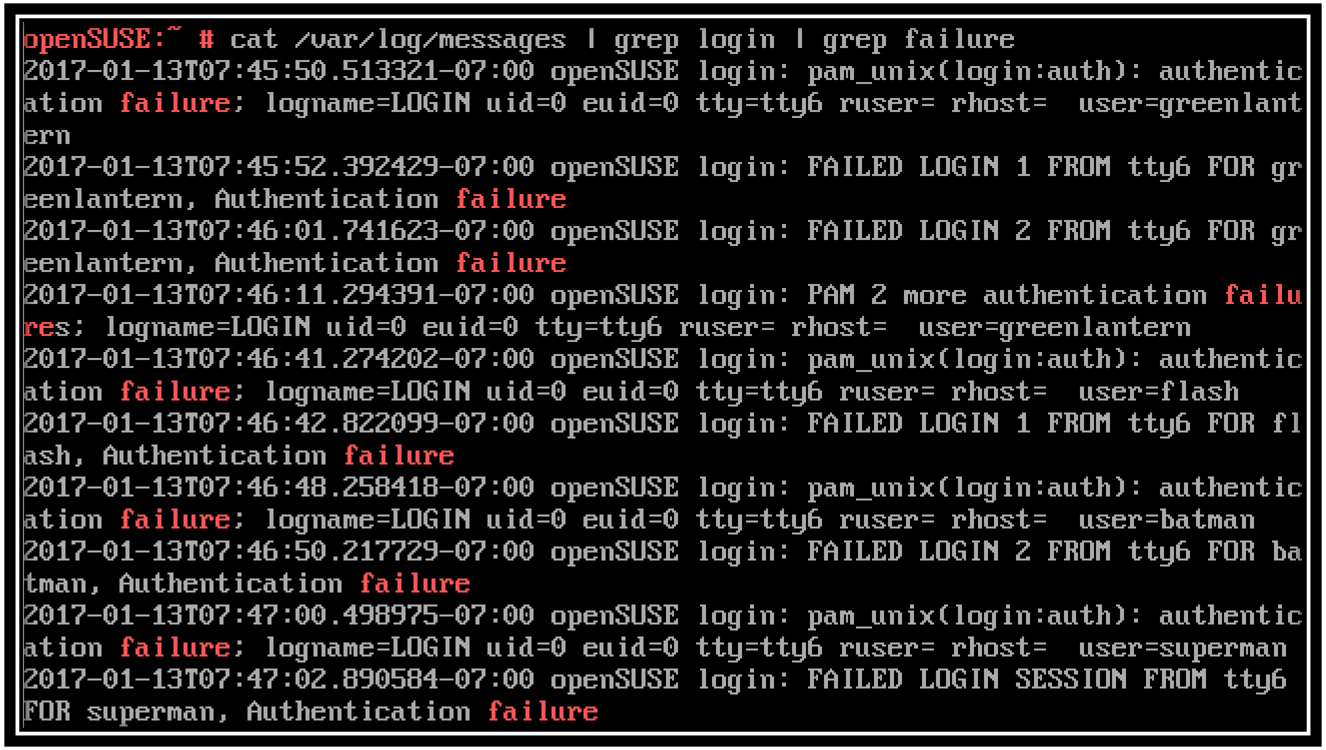


```
openSUSE:~ # journalctl -p 5 -a --no-pager --since=" 2017-01-10 00:00:00" | grep
 failure
Jan 13 07:45:50 openSUSE login[3869]: pam_unix(login:auth): authentication failu
re; logname=LOGIN uid=0 euid=0 tty=tty6 ruser= rhost=  user=greenlantern
Jan 13 07:45:52 openSUSE login[3869]: FAILED LOGIN 1 FROM tty6 FOR greenlantern,
 Authentication failure
Jan 13 07:46:01 openSUSE login[3869]: FAILED LOGIN 2 FROM tty6 FOR greenlantern,
 Authentication failure
Jan 13 07:46:11 openSUSE login[3869]: PAM 2 more authentication failures; lognam
e=LOGIN uid=0 euid=0 tty=tty6 ruser= rhost=  user=greenlantern
Jan 13 07:46:41 openSUSE login[3871]: pam_unix(login:auth): authentication failu
re; logname=LOGIN uid=0 euid=0 tty=tty6 ruser= rhost=  user=flash
Jan 13 07:46:42 openSUSE login[3871]: FAILED LOGIN 1 FROM tty6 FOR flash, Authen
tication failure
Jan 13 07:46:48 openSUSE login[3871]: pam_unix(login:auth): authentication failu
re; logname=LOGIN uid=0 euid=0 tty=tty6 ruser= rhost=  user=batman
Jan 13 07:46:50 openSUSE login[3871]: FAILED LOGIN 2 FROM tty6 FOR batman, Authe
ntication failure
Jan 13 07:47:00 openSUSE login[3871]: pam_unix(login:auth): authentication failu
re; logname=LOGIN uid=0 euid=0 tty=tty6 ruser= rhost=  user=superman
Jan 13 07:47:02 openSUSE login[3871]: FAILED LOGIN SESSION FROM tty6 FOR superma
n, Authentication failure
```

- In the above example the –p is priority 5 (notice and below), the –a shows all fields, the –no-pager specifies do not pipe to a pager, --since specifies a date to go back to in the log
- Piped to grep to filter for failure

Using Log Files to Detect Intruders

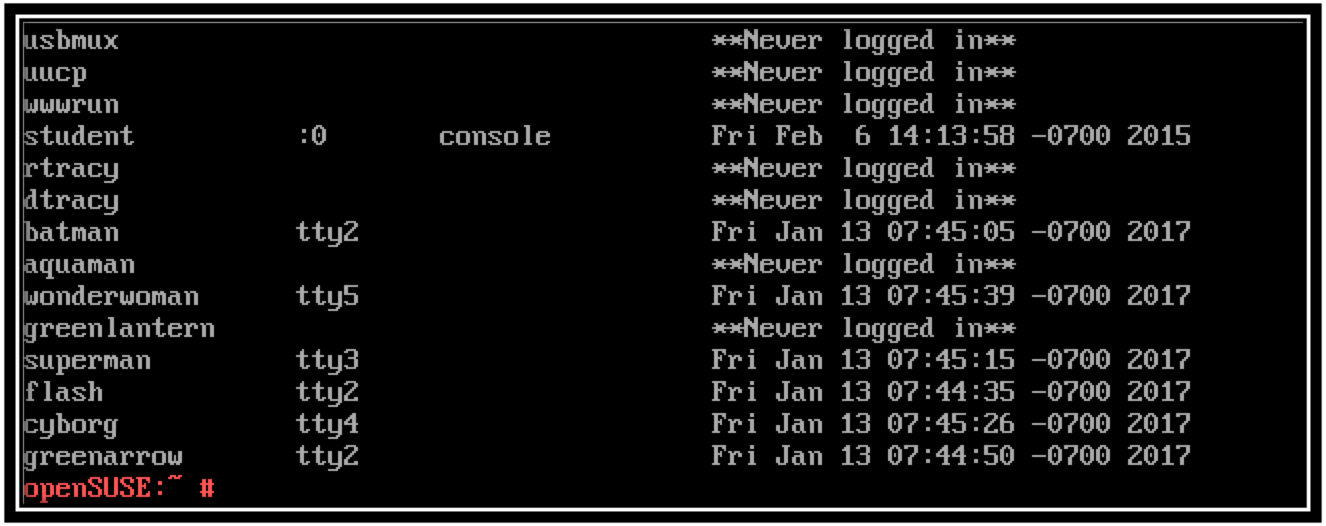
- ❑ **The /var/log/messages file can be used with regular commands like cat and piping to grep to find login information and failures:**



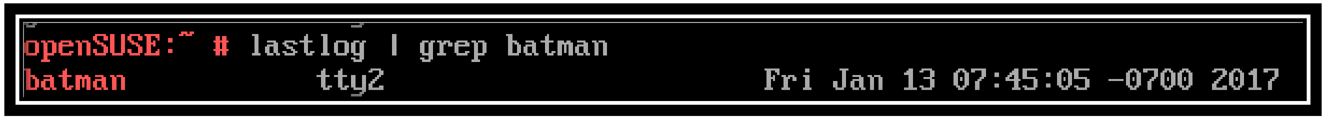
```
openSUSE:~ # cat /var/log/messages | grep login | grep failure
2017-01-13T07:45:50.513321-07:00 openSUSE login: pam_unix(login:auth): authentic
ation failure; logname=LOGIN uid=0 euid=0 tty=tty6 ruser= rhost=  user=greenlant
ern
2017-01-13T07:45:52.392429-07:00 openSUSE login: FAILED LOGIN 1 FROM tty6 FOR gr
eenlantern, Authentication failure
2017-01-13T07:46:01.741623-07:00 openSUSE login: FAILED LOGIN 2 FROM tty6 FOR gr
eenlantern, Authentication failure
2017-01-13T07:46:11.294391-07:00 openSUSE login: PAM 2 more authentication failu
res; logname=LOGIN uid=0 euid=0 tty=tty6 ruser= rhost=  user=greenlantern
2017-01-13T07:46:41.274202-07:00 openSUSE login: pam_unix(login:auth): authentic
ation failure; logname=LOGIN uid=0 euid=0 tty=tty6 ruser= rhost=  user=flash
2017-01-13T07:46:42.822099-07:00 openSUSE login: FAILED LOGIN 1 FROM tty6 FOR fl
ash, Authentication failure
2017-01-13T07:46:48.258418-07:00 openSUSE login: pam_unix(login:auth): authentic
ation failure; logname=LOGIN uid=0 euid=0 tty=tty6 ruser= rhost=  user=batman
2017-01-13T07:46:50.217729-07:00 openSUSE login: FAILED LOGIN 2 FROM tty6 FOR ba
tman, Authentication failure
2017-01-13T07:47:00.498975-07:00 openSUSE login: pam_unix(login:auth): authentic
ation failure; logname=LOGIN uid=0 euid=0 tty=tty6 ruser= rhost=  user=superman
2017-01-13T07:47:02.890584-07:00 openSUSE login: FAILED LOGIN SESSION FROM tty6
FOR superman, Authentication failure
```

Using Log Files to Detect Intruders

- Last successful logins can be viewed via the log file /var/log/lastlog
  - Binary file so the lastlog utility can be used:



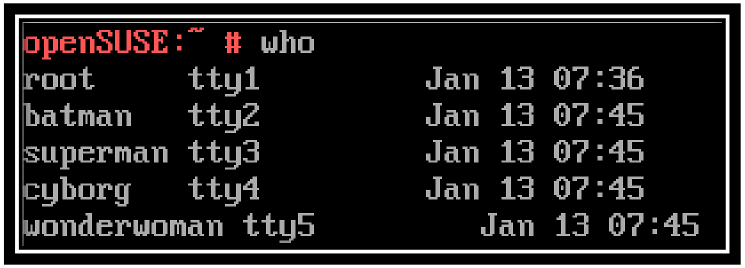
```
usbmux                                     **Never logged in**
uucp                                       **Never logged in**
wwwrun                                     **Never logged in**
student          :0       console          Fri Feb  6 14:13:58 -0700 2015
rtracy                                     **Never logged in**
dtracy                                     **Never logged in**
batman           tty2                      Fri Jan 13 07:45:05 -0700 2017
aquaman                                    **Never logged in**
wonderwoman      tty5                      Fri Jan 13 07:45:39 -0700 2017
greenlantern                               **Never logged in**
superman         tty3                      Fri Jan 13 07:45:15 -0700 2017
flash            tty2                      Fri Jan 13 07:44:35 -0700 2017
cyborg           tty4                      Fri Jan 13 07:45:26 -0700 2017
greenarrow       tty2                      Fri Jan 13 07:44:50 -0700 2017
openSUSE:~ #
```



```
openSUSE:~ # lastlog | grep batman
batman           tty2                       Fri Jan 13 07:45:05 -0700 2017
```

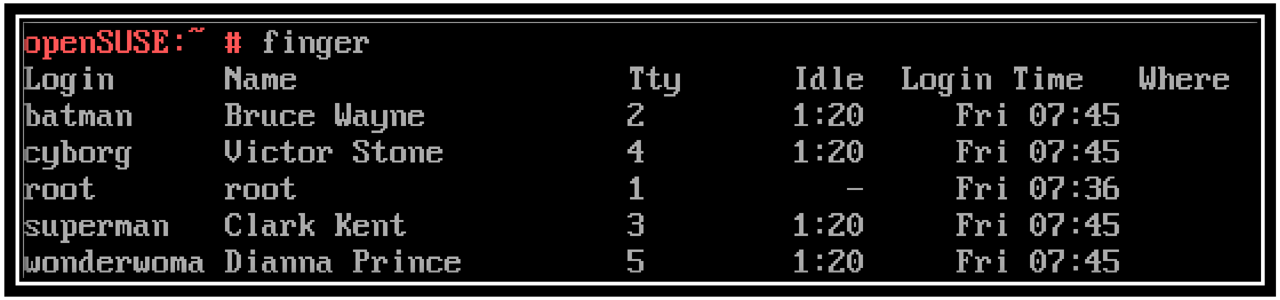
Using Log Files to Detect Intruders

- **The who utility we discussed in prior slides can be used to see who is logged in:**



```
openSUSE:~ # who
root      tty1         Jan 13 07:36
batman    tty2         Jan 13 07:45
superman  tty3         Jan 13 07:45
cyborg    tty4         Jan 13 07:45
wonderwoman tty5         Jan 13 07:45
```

- **The finger utility can be used as well:**



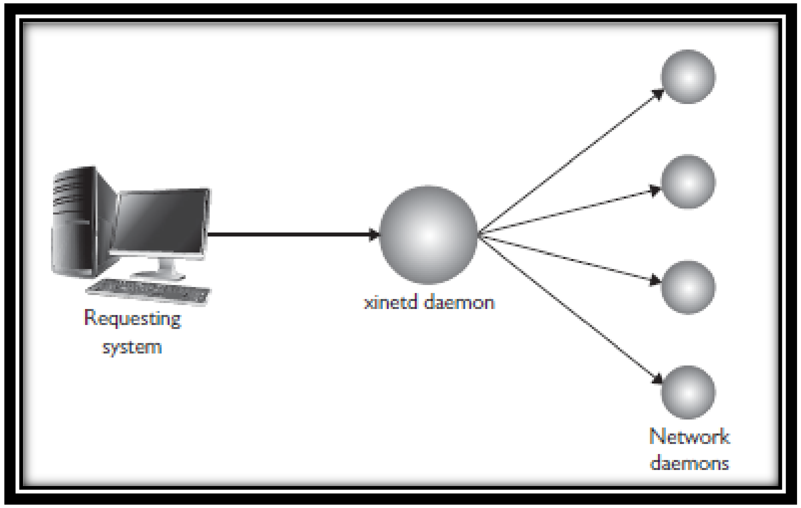
```
openSUSE:~ # finger
Login      Name                  Tty       Idle  Login Time   Where
batman     Bruce Wayne           2         1:20     Fri 07:45
cyborg     Victor Stone          4         1:20     Fri 07:45
root       root                  1            -     Fri 07:36
superman   Clark Kent            3         1:20     Fri 07:45
wonderwoma Dianna Prince         5         1:20     Fri 07:45
```

- **Log files (all devices) are a popular target of attack by intruders:**
  - **Modification**
  - **Destruction**

Configuring xinetd and inetd

❑ Super Daemons

– Act as an intermediary between the user requesting network services and the daemons on the system that provide the actual service



– Linux distributions install a wide variety of network services, some are handy but are not needed all the time

- inetd and inetd are super daemons that can make these services available when needed and unload them when not

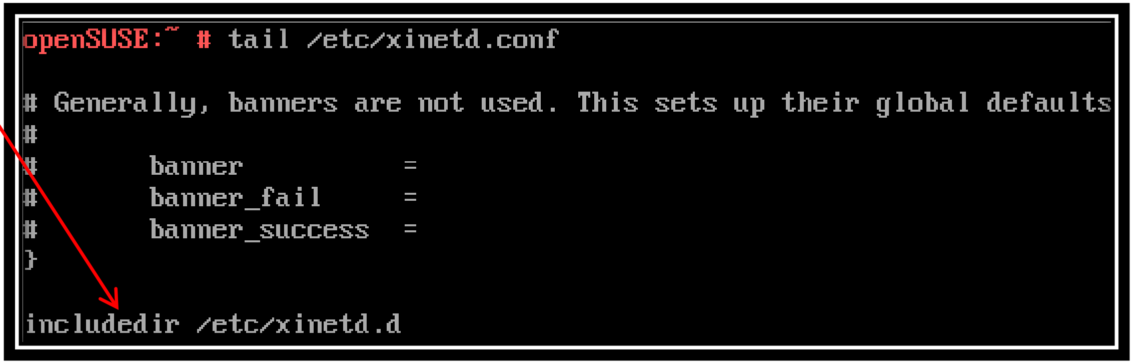
Configuring xinetd and inetd

- ❑ The xinetd daemon
  - – Requests for a network service managed by xinetd arrive at the system:
    - The request is received and processed by xinetd, not the actual network daemon requested
    - The xinetd daemon then starts the actual daemon requested and forwards the request received
    - When the request has been fulfilled and the network service is no longer needed, xinetd unloads it from memory
- ❑ Some of the network services managed by xinetd:

– echo	-smtp
– ftp	-tftp
– pop3	-vnc

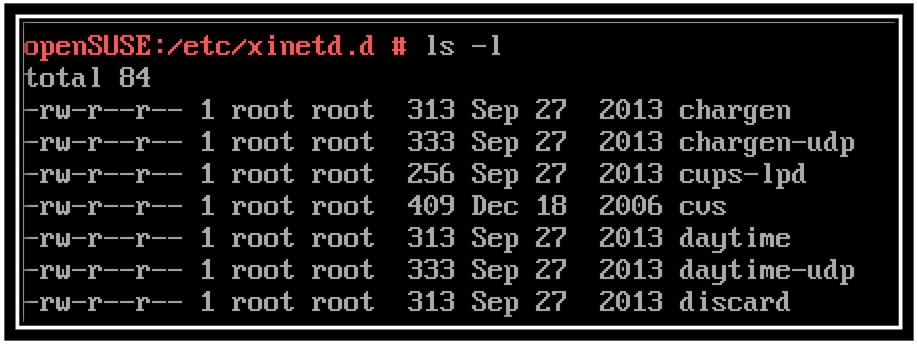
Configuring xinetd Network Services

- ❑ The xinetd daemon itself is configured using the /etc/xinetd.conf file
- ❑ At the end of this file you will notice a directive that reads:



```
openSUSE:~ # tail /etc/xinetd.conf

# Generally, banners are not used. This sets up their global defaults
#
#       banner          =
#       banner_fail     =
#       banner_success  =
}

includedir /etc/xinetd.d
```

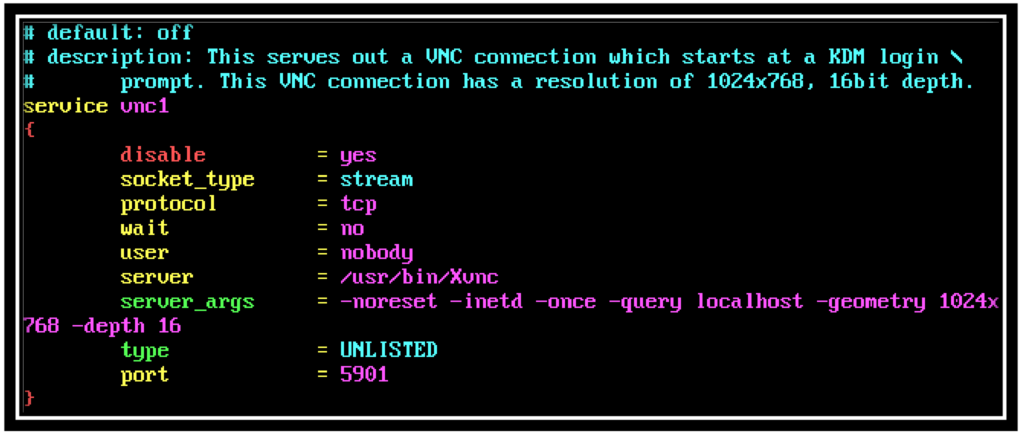
- ❑ This line tells the xinetd daemon to use the configuration files in /etc/xinetd.d
- ❑ These files tell xinetd how to start each service:



```
openSUSE:/etc/xinetd.d # ls -l
total 84
-rw-r--r-- 1 root root  313 Sep 27  2013 chargen
-rw-r--r-- 1 root root  333 Sep 27  2013 chargen-udp
-rw-r--r-- 1 root root  256 Sep 27  2013 cups-lpd
-rw-r--r-- 1 root root  409 Dec 18  2006 cvs
-rw-r--r-- 1 root root  313 Sep 27  2013 daytime
-rw-r--r-- 1 root root  333 Sep 27  2013 daytime-udp
-rw-r--r-- 1 root root  313 Sep 27  2013 discard
```

Configuring xinetd Network Services

- The files in this directory only include instructions on how xinetd is to start the service, not how the service will operate:



```
# default: off
# description: This serves out a VNC connection which starts at a KDM login \
#       prompt. This VNC connection has a resolution of 1024x768, 16bit depth.
service vnc1
{
        disable         = yes
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = nobody
        server          = /usr/bin/Xvnc
        server_args     = -noreset -inetd -once -query localhost -geometry 1024x
768 -depth 16
        type            = UNLISTED
        port            = 5901
}
```

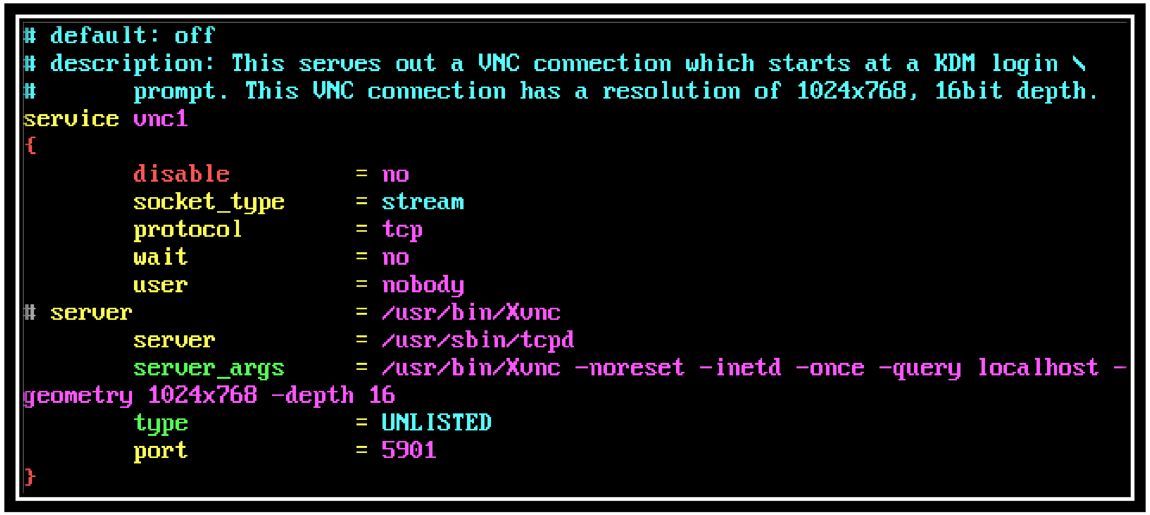
- In this example with the vnc service the disable value is set to yes which would not allow xinetd to start it
  - Would have to modify the file and place no as the value
- The daemon to actually start is specified by the server value. In this example xinetd would start the /usr/bin/Xvnc daemon
  - May need to restart xinetd via the init script in /etc/rc.d/init.d or /etc/init.d after modifying a network service file

Using TCP Wrappers

- ❑ After enabling a network service using its configuration file in /etc/xinetd.d directory any host can connect to it via xinetd
  - – Probably not what you want according to your security policy
- ❑ TCP wrappers limits access to only a specific set of hosts and deny access to everyone else
- ❑ TCP wrappers are configured through two files:
  - – /etc/hosts.allow
  - – /etc/hosts.deny
- ❑ The daemon-list variable is a list of servers using the names for the servers that appear in /etc/services
- ❑ The client-list variable is a list of computers to be granted or denied access to the specified daemons

Using TCP Wrappers

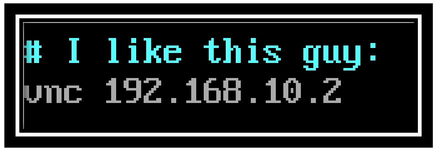
- ❑ To use TCP Wrappers (tcpd):
  - – 1. Verify that tcpd package is on your system (it is not on your version of openSUSE)
  - – 2. Open the network service file you want to restrict access to via text editor
  - – 3. Comment out the server = line and add:
    - server   = /usr/sbin/tcpd
  - – 4. Add the following line as well
    - server_args   = path_to_daemon (commented out in step 3)



```
# default: off
# description: This serves out a VNC connection which starts at a KDM login \
#       prompt. This VNC connection has a resolution of 1024x768, 16bit depth.
service vnc1
{
        disable         = no
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = nobody
# server                = /usr/bin/Xvnc
        server          = /usr/sbin/tcpd
        server_args     = /usr/bin/Xvnc -noreset -inetd -once -query localhost -
geometry 1024x768 -depth 16
        type            = UNLISTED
        port            = 5901
}
```

Using TCP Wrappers

- To use TCP Wrappers (tcpd):
  - 5. Save and close the network service configuration file
- Now access controls need to be created
  - The tcpd daemon uses the /etc/hosts.allow and the /etc/hosts/deny files to specify who can and cannot access services it manages
  - The syntax for both is
    - service: host_addresses



```
# I like this guy:
vnc 192.168.10.2
```

  - The /etc/hosts.allow file will be checked first
  - The /etc/hosts.deny is checked next if a match did not occur in the .allow
  - If no match occurs, access is granted
- The .allow and .deny files have example rules for allowing all with exceptions and denying all with exceptions
- STIGS may have rules to use as well in varying security environments

Using inetd

- ❑ The inetd package is a legacy super daemon
- ❑ It has been deprecated, might see it though
- ❑ Services that launched via inetd were configured through the /etc/inetd.conf file or files in the /etc/inet.d/

Exercise 11-3: Configuring xinetd

Please open your Practical Exercise book to Exercise 11-3.

Time to Complete: 5 Minutes

Summary

- ❑ **Securing the system**
- ❑ **Controlling user access**
- ❑ **Defending against network attacks**
- ❑ **Managing system logs**
- ❑ **Configuring xinetd and inetd**

Questions

Questions?

Check on Learning

Question 1

Which of the following commands will load the updatedb process and leave it running even if the user logs out of the shell?

A. updatedb

B. updatedb &

C. updatedb –nohup

D. nohup updatedb &

Check on Learning

Question 2

Which of the following commands can be used to switch to the root user account and load root's environment variables?

A. su –

B. su root

C. su root –e

D. su –env

Check on Learning

Question 3

You need to set password age limits for the ksanders user account. You want the minimum password age to be one day, the maximum password age to be 45 days, and the user to be warned five days prior to password expiration. Which command will do this?

A. usermod –m 1 –M 45 –W 5 ksanders

B. useradd –m 1 –M 45 –W 5 ksanders

C. chage –M 1 –m 45 –W 5 ksanders

D. chage –m 1 –M 45 –W 5 ksanders

Check on Learning

Question 4

You need to scan a Linux system with an IP address of 10.200.200.1 to determine what ports are currently open on it. What commands could you use at the shell prompt to do this? (Choose two.)

A. nmap –sT 10.200.200.1

B. scan 10.200.200.1 –TCP

C. scan 10.200.200.1 –UDP

D. nmap –sU 10.200.200.1

E. nmap 10.200.200.1 –scan

Check on Learning

Question 5

You need to configure your /etc/hosts.allow file to allow only the linux1, linux2, and linux3systems to access the vsftpd daemon on your system. Which of the following lines in the file will do this?

A. vsftpd: ALL

B. vsftpd: linux1, linux2, linux3

C. vsftpd: ALL EXCEPT linux1, linux2, linux3

D. vsftpd linux1, linux2, linux3

Check on Learning

Question 6

You need to configure your Linux firewall to allow all network traffic addressed to the DNS service on the local system. Which command will do this?

A. iptables –t filter –A INPUT –s 0/0 –p tcp –dport 53 –j DROP

B. iptables –t filter –A OUTPUT –s 0/0 –p tcp –dport 53 –j ACCEPT

C. iptables –t filter –A INPUT –s 0/0 –p tcp –dport 80 –j DROP

D. iptables –t filter –A INPUT –s 0/0 –p tcp –dport 53 –j ACCEPT

Check on Learning

Question 7

Which log file contains a list of all users who have authenticated to the Linux system, when they logged in, when they logged out, and where they logged in from?

A. /var/log/faillog

B. /var/log/last

C. /var/log/wtmp

D. /var/log/login

Check on Learning

Question 8

Which log file contains a list of failed login attempts?

A. /var/log/faillog

B. /var/log/last

C. /var/log/wtmp

D. /var/log/login

Check on Learning

Question 9

The existence of which file prevents all users except root from logging in to a Linux system?

A. /root/nologin

B. /etc/nologin

C. /var/log/nologin

D. /tmp/nologin

E. /usr/sbin/nologin

Check on Learning

Question 10

You need to view the first few lines of the /var/log/boot.msg file. Which of the following commands will do this? (Choose two.)

A. head /var/log/ boot.msg

B. tail /var/log/ boot.msg

C. grep –l 10 /var/log/boot.msg

D. less /var/log/boot.msg

E. cat /var/log/boot.msg