Change the root password

1. If you aren't already running the terminal as root: `sudo su`
2. `passwd`

Remove any additional users

if applicable

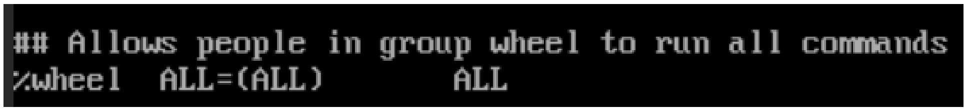
1. view users with `cat /etc/passwd`
2. `userdel -r [username]` for each user with an ID greater than or equal to 1000 that is not explicitly mentioned on the users list *BUT* see note below first.
   - Example: `userdel -r beyonce`
   - Do *not* remove users newt, mercenary, rex, loader, acrid, and captain from Shell/FTP but *do* remove any users that aren't on this list.
   - For machines that are not the Shell/FTP, you can execute the following:

```
for user in $(getent passwd | grep -E '[1-9][0-9]{3,}' | grep -vE
'^[user1 to keep]|^[user2 to keep]' | cut -d':' -f1); do userdel -r
$user; done
```

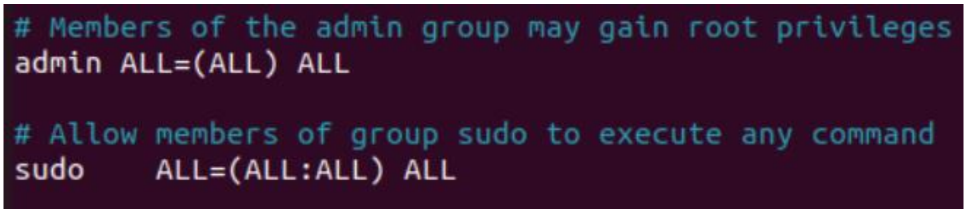
Note: Replace sysadmin with any users you'd like to exclude! This includes the user you are currently logged in as if using a GUI. Also change this user's password to something ridiculously complex. You can do this with `passwd [username]`. For example, `passwd sysadmin`

Remove sudo permissions for all users

1. Run visudo
2. Comment out lines that contain the word "ALL" by adding a # at the beginning, *except* for lines that start with "root". These lines should look as follows:



```
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL
```



```
# Members of the admin group may gain root privileges
admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
sudo    ALL=(ALL:ALL) ALL
```

To add a # to the beginning of these lines with the vi editor, you'll have to press the "`i`" key to enter insert mode. Once you've added the # where necessary, click the escape key, then type "`:wq`" without the quotes and press enter. This should write the desired changes and quit the vi editor.

Change service users' shell to nologin

1. Determine the path to the nologin shell. It will probably be /usr/sbin/nologin.
    a. `locate nologin | grep "\/nologin$"`
    b. If the above command doesn't work: `find / -name nologin`
    c. Remember the full path printed out for later.
2. Execute the following, replacing [nologin shell] with the full path for the nologin shell found in step 1.

```
for user in $(getent passwd | grep -vE '[1-9][0-9]{3,}' | grep -vE
'^root|nologin$|false$|sync$|halt$|shutdown$' | cut -d':' -f1); do
usermod -s [nologin shell] $user; done
```

3. For the user you're currently logged in as (except root): `usermod -s [nologin shell] [username]`

Configure IP

Debian

3. Edit `/etc/network/interfaces`
4. `systemctl restart networking`

Configuration

```
iface [interface] inet static
      address [ip]
      netmask [netmask]
      gateway [gateway]
      dns-nameserver [nameserver]
```

Sample Configuration

```
iface ens18 inet static
      address 192.168.14.5
      netmask 255.255.255.0
      gateway 192.168.14.1
      dns-nameserver 192.168.14.7
      dns-nameserver 172.20.0.7
```

Ubuntu

5. Edit `/etc/netplan/01-network-manager-all.yaml`
6. `netplan apply`

Configuration

```
network:
ethernets:
   [interface]:
       addresses: [[ip]/[CIDR]]
       gateway4: [ipv4 gateway]
       nameservers:
          addresses: [[nameserver 1], [nameserver 2], […]]
```

Sample Configuration

```
ethernets:
   ens18:
      addresses:
         - 192.168.14.5/24
      gateway4: 192.168.14.1
      nameservers:
         addresses: [192.168.14.7, 172.20.0.7]
```

CentOS

7. Edit `/etc/sysconfig/network-scripts/ifcfg-`[device]
8. `systemctl restart network`

Configuration

```
BOOTPROTO=static
ONBOOT=yes
IPADDR=[ip]
NETMASK=[netmask]
GATEWAY=[gateway]
DNS[1/2/3/…]=[nameserver]
ZONE=[zone]
```

Sample Configuration: `/etc/sysconfig/network-scripts/ifcfg-ens18`

```
BOOTPROTO=static
ONBOOT=yes
```

```
IPADDR=192.168.14.5
NETMASK=255.255.255.0
GATEWAY=192.168.14.1
DNS1=192.168.14.7
DNS2=172.20.0.7
ZONE=internal
```

Upgrade required packages

I remember at least Shell/FTP and Web Server had the apt package manager. Therefore, one of the following commands should work for performing updates:

- `apt update && apt install [packages]`
- `apt-get update && apt-get install [packages]`

If you receive an error but you're able to ping google.com, it's likely that your mirrors are outdated. You'll have to modify the /etc/apt/sources.list file to obtain the latest packages available for your release. This command might work for you if you're using Ubuntu:

```
sed -i -re 's/([a-z]{2}\.)?archive.ubuntu.com|security.ubuntu.com/old-releases.ubuntu.com/g' /etc/apt/sources.list
```

`This command might work for you if you're using Debian:`

```
sed -i -re 's/([a-z]{2}\.)?deb.debian.org|security.debian.org/old-releases.debian.org/g' /etc/apt/sources.list
```

Packages to upgrade

- **Shell/FTP**: openssh-server vsftpd
- **Web Server**: apache2 php `libapache2-mod-php php-mysql`
- **Database**: mysql-server
- **DNS**: bind9 bind9-utils bind9-dnsutils

Be sure to double-check that these packages already exist. It should tell you this when you run the install commands above. If the packages don't already exist, reply with n to cancel the installation and work with me to figure out which packages relevant to each service are installed.

Apply host firewall

Linux Template

```
iptables -F
# Only allow incoming traffic on specified ports
iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport [port] -j ACCEPT
… repeat as necessary for desired open TCP ports
iptables -A INPUT -p udp --dport [port] -j ACCEPT
… repeat as necessary for desired open UDP ports
iptables -A INPUT -p icmp --icmp-type 8 -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -P INPUT DROP

# Block all outgoing traffic except for established connections
iptables -A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -P OUTPUT DROP
```

Shell/FTP

```
iptables -F

# Only allow incoming FTP and SSH traffic
iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 21 -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -P INPUT DROP

# Block all outgoing traffic except for established connections
iptables -A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -P OUTPUT DROP
```

Web Server

```
iptables -F

# Only allow incoming HTTP and HTTPS traffic
iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -P INPUT DROP

# Block all outgoing traffic except for established connections
iptables -A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -P OUTPUT DROP
```

Database

```
iptables -F

# Only allow incoming MySQL traffic
iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 3306 -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -P INPUT DROP

# Block all outgoing traffic except for established connections
iptables -A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -P OUTPUT DROP
```

DNS

```
iptables -F

# Only allow incoming DNS traffic
iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -p udp --dport 53 -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -P INPUT DROP

# Block all outgoing traffic except for established connections
iptables -A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -P OUTPUT DROP
```

Backup

```
?
```

External Kali Backup VM

```
iptables -F

# Only allow incoming SSH traffic
iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -P INPUT DROP
```

```
# Block all outgoing traffic except for established connections
iptables -A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -P OUTPUT DROP
```

Stop the cron daemon (cron jobs)

1. `systemctl stop crond`
2. `systemctl disable crond`

Remove suspicious systemd timers

Note: It is not advised to remove all the systemd timers since some are deemed vital to healthy system functioning. Therefore, we will look through the timers and see if any seem suspicious.

1. `systemctl list-timers` to view timers
2. For any suspicious timers, run the following:
   a. `systemctl stop [unit].timer`
   b. `systemctl disable [unit].timer`
3. You might also want to stop and disable the underlying service with the following:
   a. `systemctl stop [activates].service`
   b. `systemctl disable [activates].service`

Backup required directories

Files or Directories to Back Up

- **Shell/FTP:** probably /var/ftp, but check local_root option in the /etc/vsftpd/vsftpd.conf file
- **Web Server:** /var/www/html?
- **Database:** /root/betterblog.sql
- **DNS:** /etc/bind

1. For each directory listed (ignore this for database), execute the following commands:
   `cd [directory] && tar -zcvf /etc/.kitten/backup-[name].tar.gz *`
   - Example: cd /etc/bind && `tar -zcvf /etc/.kitten/backup-bind.tar.gz *`
2. Once completed, yell "John, I'm ready for remote backup!"
3. Execute the following commands:
   - `iptables -I OUTPUT -p tcp --dport 22 -j ACCEPT`
   - `sftp backup@172.18.15.<T>6`
     - When it prompts for password, enter "S@lcianaszkot23" without quotes
     - Note: You will not be able to see the password as you type.
   - `ls`

        - Find the folder with your machine name.
    - `cd [folder with your machine name]`
    - `put /etc/.kitten/*`
        - For Database: `put /root/betterblog.sql`
    - `ls`
        - Make sure you see all your backup files. There should be one for each directory to back up.
    - `exit`
    - `iptables -D OUTPUT 1`
4. Yell “Remote backup complete!” I will turn off the backup service.

Tell John you are ready to port forward services for your machine!