Resetting Passwords

Mac OS X

pfSense 2.4

Solaris 10

Solaris 11

Esxi 6.0-6.5

mySQL

Breaking into a password protected Mac OSX or MacOS

1. Power off the Mac.
2. Power it on while holding **Control** and **R**.
3. This will boot the Mac into Recovery Mode.
4. Once Recovery Mode loads, select **Utilities > Terminal** from the top menu bar.
5. In the Terminal window, type *resetpassword*
6. Select the appropriate hard disk partition. There SHOULD only be one, but you still need to pick it.
7. Select the user you'd like to reset the password on.
8. Create a new password/password hint.
9. Click the **Apple** in the top bar, select **Restart**.

Creating a login message for Mac OSX or MacOS

1. Click the **Apple** in the top menu bar, select **System Preferences > Security & Privacy**.
2. Click the **Lock Icon** and enter your administrator name and password.
3. In the **General** tab, select "**Show a message when the screen is locked**," then click "**Set Lock Message**".
4. Enter your message in the text field. Click OK.

Forgotten Password with Locked Console

If the console is password protected and the password is unknown, all is not lost. It will take a couple reboots to accomplish, but it can be fixed with physical access to the console:

- Reboot the pfSense box
- Choose the option for **Single User Mode** from the loader menu (The one with the ASCII logo). Depending on the version of pfSense, it may be option 2 or option 4.
- Press **Enter** when prompted to start */bin/sh*
- Remount the drive as rewritable:

```
/sbin/mount -o rw /
```

If multiple partitions/slices were made during install, mount everything: (this will always work guaranteed)

```
/sbin/mount -a -t ufs
```

- Run the built-in password reset command:

```
/etc/rc.initial.password
```

- Follow the prompts to reset the password
- Reboot

```
/sbin/reboot
```

The system will now be accessible the default password (**admin** / **pfsense**)

Recovering root password in Oracle Solaris 10 x86 using CD method

1. Set ISO image file in VM settings
2. VM > Power > Power On to Firmware
3. Set CD-ROM Drive at the top of Boot Settings
4. Save and Exit
5. Boot into Solaris 10
6. Press 6 to boot into single user mode shell
7. Select ‘y’ to mount read/write
8. #df -h to view /dev/dsk/cd0xxxx mounted on /a ramdisk
9. #cd /a/etc/
10. #TERM=vt100
11. #EDITOR=vi
12. #export TERM EDITOR
13. Use x as delete while in vi
14. Change the line from
    a. root:WP7grKsEFAgt.:15182:::::: to
    b. root::15182::::::
15. Exit with :x!
16. #init 5
17. Power off VM and disable CD
18. VM > Power > Power on to firmware
19. Move CD-ROM below Hard Drive on Boot
20. Boot into Solaris 10

21. Login as root with no password

22. Change root password with ‘passwd’

References:

https://www.youtube.com/watch?v=YL1QiVRRvv0

Recover Root Password Solaris 11

CD Method (In the event of no CD, the same steps can be performed by connecting to a local boot or install server if one is set up)

1. Boot to Firmware (BIOS) and set the CD-ROM to boot before Hard disk
2. SPARC: At the OK prompt, Enter: **`boot cdrom -s`**
   Login with username:password- **`root:solaris`**
3. Select desired keyboard and keyboard layout: **US-English, English**
4. x86 Server: Select Text Console from GRUB, should be the third option
   Login with username:password- **`jack:jack`**
5. Change to root: **sudo su** (password is still **jack**)
6. Check the pool name for the root file system
   `zpool import | grep -i pool:`
   Example output:
   `pool:rpool`
   `pool:repo`
7. Force import rpool
   `zpool import -f rpool`
8. Check for Solaris boot environment
   `beadm list`
9. Mount the Solaris boot environment to a temporary directory
   `beadm mount solaris /a`
10. Edit the shadow file (When using vi, delete characters with 'x' and to save the file, use :wq!)
    `vi /a/etc/shadow`
11. Remove the password hash for root and a user to login with
    IMPORTANT: delete a hash for a user as well as root because root is set as a role that may not be logged in by default
    Change--- **`username:iEwei23SamPleHashonf0981:15746::::::17216`**
    To This--- **`username::15746::::::17216`**
12. By default, Solaris does not allow empty password login, so disable the option
    `vi /a/etc/default/login`
    Change--- **`PASSREQ=YES`**
    To This--- **`PASSREQ=NO`**
13. Update the boot environment
    `bootadm update-archive -R /a`
14. Reboot and select Boot from Hard Disk in GRUB: **`init 0`**
15. Login without a password as your user to the JDE
16. Change to root afterwards with no password in a terminal: **`su`**
17. Change user and root passwords: **`passwd *username*`**

Troubleshooting (Additional Steps that may be necessary)

1. Identify root pool
   ```
   zpool import | grep -i pool:
   ```
2. Import the root pool
   ```
   zpool import -f -R /tmp/rpool rpool
   ```
3. Configure root pool dataset as legacy
   ```
   zfs set mountpoint=legacy rpool/ROOT/solaris
   ```
4. Mount rpool dataset on /mnt
   ```
   mount -F zfs rpool/ROOT/solaris /mnt
   ```
5. Setting EDITOR permissions to modify the root password
   ```
   cp /mnt/etc/shadow /mnt/etc/shadow_backup
   cp /mnt/etc/passwd /mnt/etc/passwd_backup
   TERM=vt100; export TERM
   EDITOR=vi; export EDITOR
   ```
6. Remove password hash for root in /mnt/etc/shadow
7. Unmount and set back the mountpoints and export the pool
   ```
   umount /mnt
   zfs set mountpoint=/ rpool/ROOT/solaris
   zpool export rpool
   halt
   ```

Allow Root Direct Login

Root is a role by default and therefore can not be used to login directly. Here is how to allow Root Direct Login in Solaris 11. (NOT RECOMMENDED)

Example Error:

```
solaris11 console login: root
Password:
Roles can not login directly
```

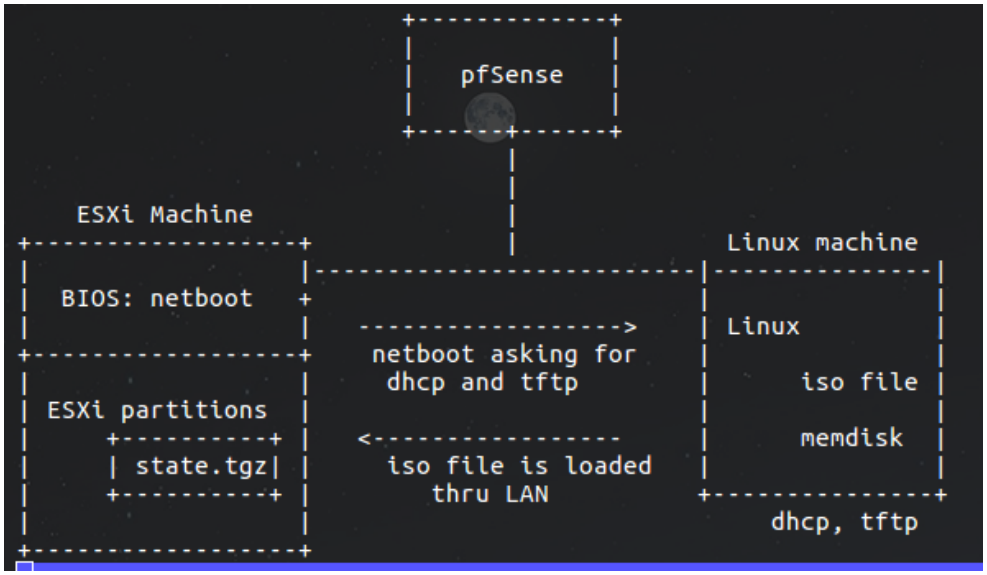
1. Check the account type
   ```
   grep -i root /etc/user_attr
   ```
   Output: `root:::: type=role`
2. Change the account type from role to normal
   ```
   rolemod -K type=normal root
   ```

Reset ESXi Root Password without a Physical Disc

Methodology:

Using an available Linux machine to host PXE and boot ESXi from it.

VM network structure:
Wall → Router → LAN



Requirement:

ESXi:

- Has access to BIOS
- BIOS has network boot
- Secure boot is off

Linux Machine:

- On the same network with ESXi, can see each other and can ping each other
- Has Internet, with reliable connection and speed

Procedure:

Linux machine:

- Make sure the sources list contain this source:

---- /etc/apt/sources.list -----
deb http://us.archive.ubuntu.com/ubuntu/ xenial universe

- Install dependencies

apt-get install dnsmasq
apt-get install syslinux

- Configure dnsmasq

vi /etc/dnsmasq.conf
---------dnsmasq.conf----------
interface=ens33
dhcp-range=10.80.80.10,20.80.80.30
enable-tftp
tftp-root=/srv/netboot/

- Create tftp boot folder
mkdir -p /srv/netboot/
cd /srv/netboot/
mkdir live/

- Download the netboot from Ubuntu and extract it. We use Ubuntu 10.04 Precise for compatibility.
wget http://archive.ubuntu.com/ubuntu/dists/precise/main/installer-amd64/current/images/netboot/netboot.tar.gz
tar xzf netboot.tar.gz

- Get memdisk from syslinux, the memdisk file will help the boot process load the ISO into memory
cp /usr/lib/syslinux/memdisk /live/
cd live/

- Download live boot ISO, in this case we use the Puppy Linux Live CD that is only ~230MB in size. Make sure the ESXi machine has enough RAM to load the ISO file. This method should work with other ISO files as well; however, newer versions of Linux might need a compatible memdisk file.
wget http://distro.ibiblio.org/puppylinux/puppy-tahr/iso/tahrpup64-6.0.5/tahr64-6.0.5.iso
mv tahr64-6.0.5.iso puppy.iso

```
root@ubuntu:/srv/netboot/live# wget http://distro.ibiblio.org/puppylinux/puppy-tahr/iso/tahrpup64-6.
0.5/tahr64-6.0.5.iso
--2017-06-01 07:42:06--  http://distro.ibiblio.org/puppylinux/puppy-tahr/iso/tahrpup64-6.0.5/tahr64-
6.0.5.iso
Resolving distro.ibiblio.org (distro.ibiblio.org)... 152.19.134.43
Connecting to distro.ibiblio.org (distro.ibiblio.org)|152.19.134.43|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 234881024 (224M) [application/octet-stream]
Saving to: 'tahr64-6.0.5.iso'

tahr64-6.0.5.iso          47%[================>                  ] 106.27M   483KB/s    eta 3m 56s
```

- Configure the boot menu to add a new boot entry that points to the ISO file we want.
vi /srv/netboot/pxelinux.cfg/default
------ /srv/netboot/pxelinux.cfg/default ------
label ISO
menu Boot from ISO
root (hd0,0)
kernel live/memdisk
append iso initrd=live/puppy.iso raw

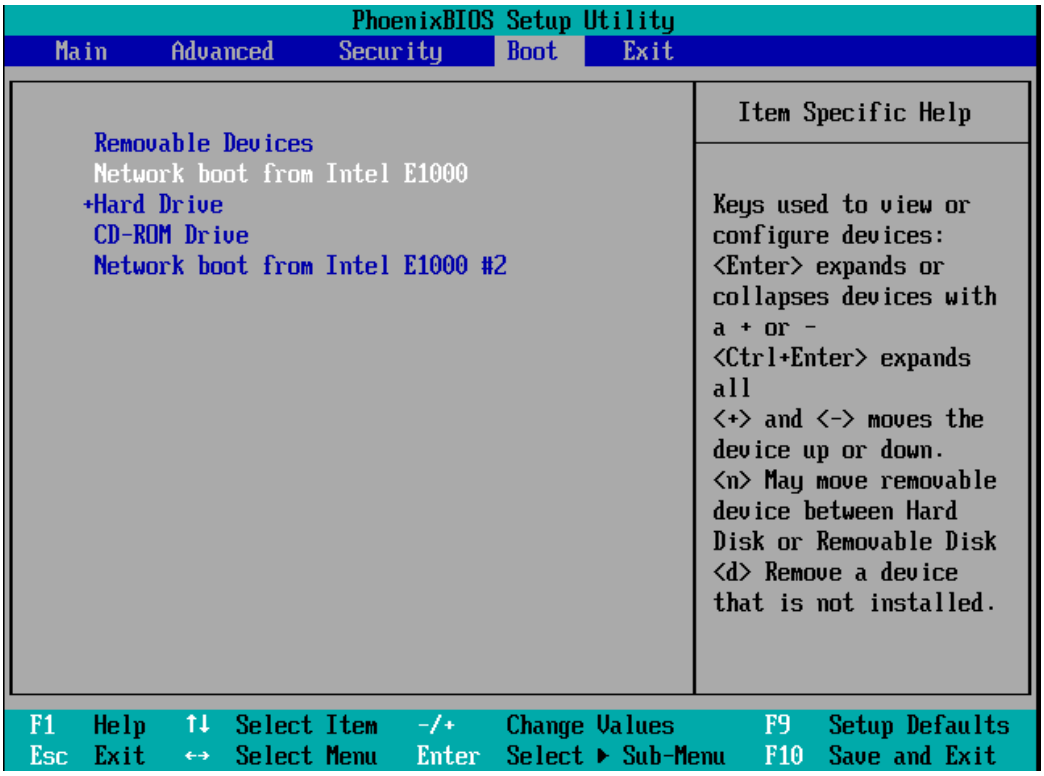
```
# D-I config version 2.0
include ubuntu-installer/amd64/boot-screens/menu.cfg
default ubuntu-installer/amd64/boot-screens/vesamenu.c32
prompt 0
timeout 0
label iso
        menu Boot from ISO
        root (hd0,0)
        kernel live/memdisk
        append iso initrd=live/puppy.iso raw
```

- Start dnsmasq service
  service dnsmasq start

- At this point, the Linux machine is used as a server hosting dhcp and tftp on the network.

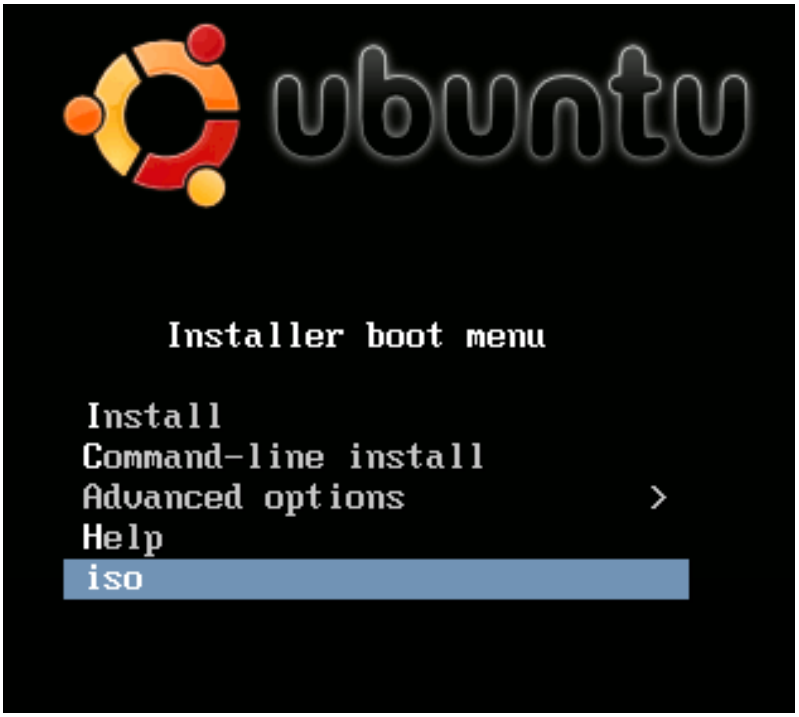
ESXi machine:

1. Power on ESXi to firmware.
2. Make sure secure boot is off
2. Edit BIOS boot settings to Network boot before the Hard Drive, pick the correct NIC as there may be more than one



4. Reboot ESXi.
5. Wait for DHCP to initialize Network boot.
It may not work the first time so just Ctrl+Alt+Delete to reboot.

6. From the netboot installer boot menu, select “iso” entry we added previously.



Wait for the ISO to load into RAM. With a large enough RAM, a decent bus width, and bandwidth > 1066Mb/s, the process should take no longer than 3 minutes.

8. After getting the shell, check the hard disk that this live boot detected:
   fdisk -l

9. Mount the ESXi partition. It should be the second one of /dev/sda you saw in fdisk -l, the sda1 should be the boot partition. In this case, it is /dev/sda5:
   mount /dev/sda5 /mnt/

10. Change to the mounted file directory:
   cd /mnt

```
/mnt # ls
a.b00          esx_dvfi.v00  lsu_lsi_.v04  net_tg3.v00   sata_sat.v03  scsi_mpt.v
02
ata_pata.v00   ima_qla4.v00  misc_cni.v00  net_vmxn.v00  sata_sat.v04  scsi_qla.v
00
ata_pata.v01   imgdb.tgz     misc_dri.v00  nmlx4_co.v00  sb.v00        state.tgz
ata_pata.v02   ipmi_ipm.v00  mtip32xx.v00  nmlx4_en.v00  scsi_aac.v00  tboot.b00
ata_pata.v03   ipmi_ipm.v01  net_bnx2.v00  nmlx4_rd.v00  scsi_adp.v00  uc_amd.b00
ata_pata.v04   ipmi_ipm.v02  net_bnx2.v01  nvme.v00      scsi_aic.v00  uc_intel.b
00
ata_pata.v05   jumpstrt.gz   net_cnic.v00  ohci_usb.v00  scsi_bnx.v00  uhci_usb.v
00
ata_pata.v06   k.b00         net_e100.v00  onetime.tgz   scsi_bnx.v01  user.b00
ata_pata.v07   lpfc.v00      net_e100.v01  qlnative.v00  scsi_fni.v00  useropts.g
z
b.b00          lsi_mr3.v00   net_enic.v00  rste.v00      scsi_hps.v00  weaselin.t
00
block_cc.v00   lsi_msgp.v00  net_forc.v00  s.v00         scsi_ips.v00  xhci_xhc.v
00
boot.cfg       lsu_hp_h.v00  net_igb.v00   sata_ahc.v00  scsi_meg.v00  xorg.v00
chardevs.b00   lsu_lsi_.v00  net_ixgb.v00  sata_ata.v00  scsi_meg.v01
ehci_ehc.v00   lsu_lsi_.v01  net_mlx4.v00  sata_sat.v00  scsi_meg.v02
elxnet.v00     lsu_lsi_.v02  net_mlx4.v01  sata_sat.v01  scsi_mpt.v00
emulex_e.v00   lsu_lsi_.v03  net_nx_n.v00  sata_sat.v02  scsi_mpt.v01
/mnt # _
```

11. Copy state.tgz from mounted file directory:
    cp state.tgz /tmp

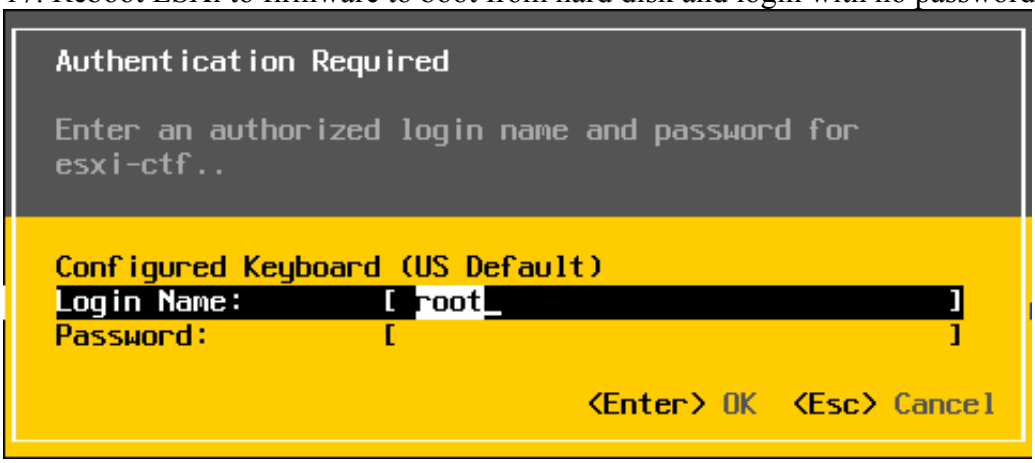
12. Untar state.tgz and nested *.tgz local.tgz:
    cd /tmp
    busybox tar xzvf state.tgz && busybox tar xzvf local.tgz

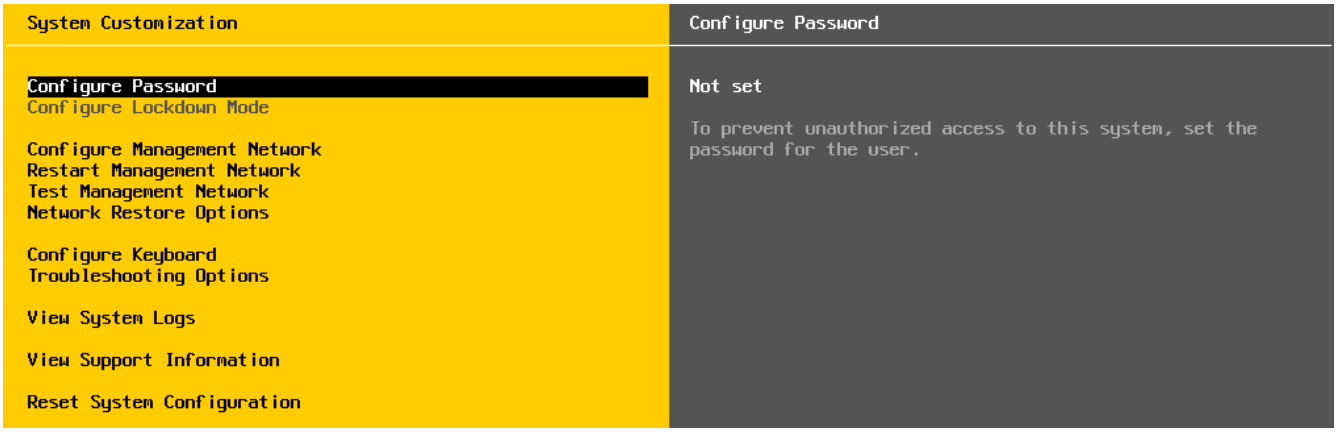
13. Remove root password hash from etc/shadow file with sed due to no text-editor (vi or nano):
    cd etc
    sed -i -e '/^root:/ s/:[^:]*/:/' shadow

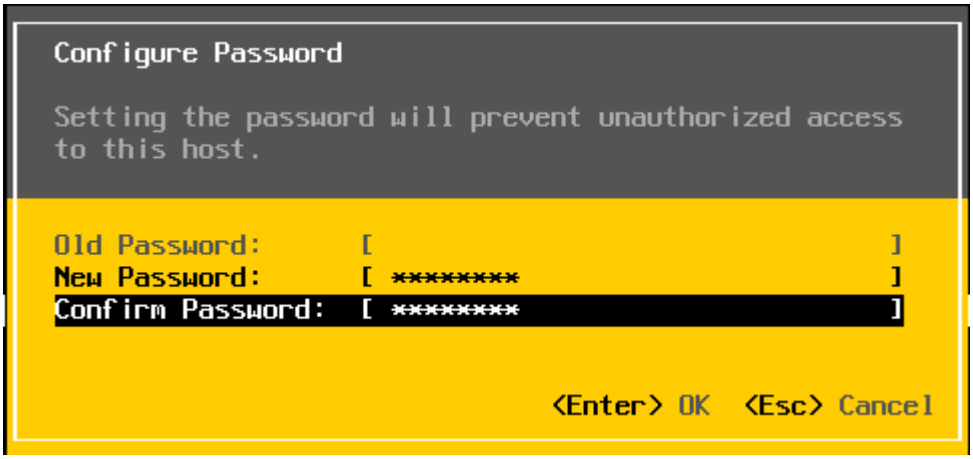
```
/tmp/etc # sed -i -e '/^root:/ s/:[^:]*/:/' shadow
/tmp/etc # cat shadow
root::17318:0:99999:7:::
nobody:*:13358:0:99999:7:::
nfsnobody:!!:13358:0:99999:7:::
dcui:*:13358:0:99999:7:::
daemon:*:13358:0:99999:7:::
vpxuser:*:14875:0:99999:7:::
```

15. Repack etc into local.tgz and then into state.tgz with busybox tar and busybox gzip:
    busybox tar cf local.tar etc
    busybox gzip local.tar
    mv local.tar.gz local.tgz
    busybox tar cf state.tar local.tgz
    busybox gzip state.tar
    mv state.tar.gz state.tgz

17. Reboot ESXi to firmware to boot from hard disk and login with no password







Additional Things to Note:
- The Internet was used to download 2 packages (syslinux, dnsmasq), 1 ISO: Puppy Linux ~230MB, and a netboot file: netboot.tar.gz ~ 22MB.

- With a reliable connection, this process would take ~10-20min.
We certainly can improve this process by finding a smaller usable live ISO, packing all the necessary files and uploading it somewhere else which allows high bandwidth and a low throttle connection.

Troubleshooting:
- In a prior method (netbooting into recovery mode), nano was the only editor available and was used to edit the shadow file.
- Additionally, there was no gzip binary. As a solution, gzip was transferred over through HTTP from the Ubuntu PXE server.

Sources:
http://docs.kali.org/installation/kali-linux-network-pxe-install
https://www.top-password.com/knowledge/reset-esxi-root-password.html
https://www.danielelolli.it/howto-boot-linux-from-network-using-pxe-and-dnsmasq-proxy-ubuntu-14-04-07-2014.html
https://www.tecmint.com/install-pxe-network-boot-server-in-centos-7/

Recover MySQL root password

1. Stop the MySQL service
   a. `service mysql stop`
2. Start MySQL w/o password
   a. `mysqld_safe --skip-grant-tables &`
3. Login with no password
   a. `mysql -u root`
4. Set a new MySQL root user password
   a.
```
mysql > use mysql;
   > update user set password=PASSWORD("newpassword")\
   where User='root';
   > flush privileges;
   > quit;
```
5. Restart the MySQL service
   a. `service mysql restart`
6. Login with root password
   a. `mysql -u root -p`

Recover MongoDB root password

1. Stop the MongoDB service
   a. `service mongodb stop`
2. Remove the --auth option or disable authentication in the MongoDB config
   a. `vi /etc/mongodb.conf`
   b. Comment out: `# auth = true`
3. Start the MongoDB service
   a. `service mongodb start`
4. Create new root user (use db.addUser for 2.6, db.createUser for 3.0+)
   a.
```
> use admin;
> db.createUser({user:"admin",pwd:"password",\
   roles:[{role:"root",db:"admin"}]});
```
5. OR change root user password
   a. `> use admin;`
   b. `> db.changeUserPassword("username", "password");`
6. Restart the instance with authentication enabled
   a. Uncomment: `auth=true`
   b. `service mongodb restart`

Service Configuration
Oscommerce
Wordpress Installation
LAMP in lxc
Mail Server on Fedora
Mail Server on Debian
DNS config files
Arch Linux VM

Update and Install dependencies:
yum update
yum install wget unzip
yum install httpd
yum install php
yum install php-mysql
yum install php-gd

Obtain oscommerce files:
wget http://oscommerce.com/files/oscommerce-2.3.4.zip
unzip oscommerce-2.3.4.zip
cd oscommerce-2.3.4.zip
mv catalog/ /var/www/html/
chmod 777 /var/www/html/catalog/includes/configure.php
chmod 777 /var/www/html/catalog/admin/includes/configure.php
chcon -R -t httpd_sys_content_rw_t /var/www/html/catalog

Enable remote mysql server:
mysql -u root -p
CREATE DATABASE oscommerce;
GRANT ALL PRIVILEGES ON oscommerce.* TO ccdc@'remotehost' identified by 'P@ssw0rd';
FLUSH PRIVILEGES;
vim /etc/mysql/mysql.conf.d/mysqld.cnf
bind-address <localIPaddress>
service mysql restart

Access mysql remotely:
mysql -u ccdc -p -h <serverIPaddress>

Access the web interface installation by opening a web browser and entering:
<oscommerceserverIPaddress>/catalog

Post installation:
rm /var/www/html/catalog/install/
chmod 644 /var/www/html/catalog/includes/configure.php
chmod 644 /var/www/html/catalog/admin/includes/configure.php

Wordpress Website

1. Set up LAMP-stack (apache, MySQL, PHP, PHP-mysql)-

Commands:

➔ sudo apt-get install apache2
➔ sudo apt-get install mysql-server
➔ sudo apt-get install php7.0 php7.0-fpm php7.0-mysql –y

2. Create MySQL Database and User for WordPress

A) Log on to MySQL
➔ ***mysql –u root –p***
➔ Enter password
B) Create Database (In this case, the name of the database is ccdc)
➔ ***CREATE DATABASE ccdc DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;***
C) Create separate MySQL user account that will be used exclusively to operate on the new database. Set a password, and grant access to database created.
➔ ***GRANT ALL ON ccdc.* TO 'ccdcuser'@'localhost' IDENTIFIED BY 'P@ssw0rd';***
D) Flush privileges so that the current instance of MySQL knows about recent changes made
➔ ***FLUSH PRIVILEGES;***
E) Exit out of MySQL
➔ ***EXIT;***

3. Install PHP Extensions (Wordpress and its plugins leverage several additional PHP extensions)

➔ ***sudo apt-get update***
➔ ***sudo apt-get install php-curl php-gd php-mbstring php-mcrypt php-xml php-xmlrpc***

4. Adjust Apache's config to allow .htacess overrides/rewrites

A) Enable .htacces Overrides
➔ ***sudo vim /etc/apache2/apache2.conf***
➔ We need to set the AllowOverride directory within a Directory block pointing to our document root. Add the following block to bottom of file:

```
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

<Directory /var/www/html/>
    AllowOverride All
</Directory>
```

B) Enable Rewrite Module
➔ ***sudo a2enmod rewrite***
C) Enable changes

➔ ***sudo apache2ctl configtest*** (Run this command to make sure there is no syntax errors. Output should say Syntax OK.)
➔ Restart to implement changes: ***sudo systemctl restart apache2***

5. Download Wordpress

➔ ***curl –O https://wordpress.org/latest.tar.gz***
➔ Extract the compressed file to create WordPress directory:
➔ ***tar xzvf latest.tar.gz***
➔ ***touch wordpress/.htaccess***
➔ ***sudo chmod 660 wordpress/.htaccess***
➔ ***cp wordpress/wp-config-sample.php wordpress/wp-config.php***
➔ Create an upgrade directory so that WordPress won't run into permissions issues trying to do its own updates:
➔ **sudo mkdir wordpress/wp-content/upgrade**
➔ Copy entire contents of directory into document root, using –a to make sure permissions are maintained
➔ ***sudo cp –a wordpress/. /var/www/html***

6. Configure the WordPress Directory

A) Adjusting ownership/permissions so we can write to these files as a regular user and so the web server can access and adjust certain files/directories. I used the username alix. This should match a sudo user.
➔ ***sudo chown –R alix:www-data /var/www/html***
B) Set the setgid bit on each of the directories within current document root. This makes it so new files created within these directories inherit the group of the parent directory (www-data) instead of creating user's primary group. So, whenever we create a file in the directory of the command line, web server will still have ownership over it.
➔ **sudo find /var/www/html –type d –exec chmod g+s {} \;**
C) Give group write access to the wp-content directory so that the web interface can make these and plugin changes:
➔ **sudo chmod g+w /var/www/html/wp-conent**
D) Give web server write access to all the content in these two directories:
➔ ***sudo chmod –R g+w /var/www/html/wp-content/themes***
➔ ***sudo chmod –R g+w /var/www/html/wp-content/plugins***
E) Setting up the WordPress Config File. Wordpress provides a secure key generator to provide security for installation. To grab secure values from WordPress:
➔ ***curl –s https://api.wordpress.org/secret-key/1.1/salt/***

➔ Copy the output given by the command above:

```
alix@alix-virtual-machine: ~
alix@alix-virtual-machine:~$ curl -s https://api.wordpress.org/secret-key/1.1/salt/
define('AUTH_KEY',         'L+cA^MZ |vHTvs2|k#{{B=zi~9:FW!^N$N3!&-8k@ij-|H[g>Kkj2/9+z;4:P*fd');
define('SECURE_AUTH_KEY',  'Ph>,1gaxdSA,]:B:)-Xhypgm1AUo`p{)viF{MC}p*aU$g&f+W]vn;$+kghF`-4E|');
define('LOGGED_IN_KEY',    '(5?)uLq(W!+9n~WM1W*-O;V1q^mypL/+Y~DX1Nq8T5]-IgJ7e-|<jA$fsM,g0fn&');
define('NONCE_KEY',        'y26+sq}  7E,*Su+HV~+Z1vQ9cnc+coOAn~pJoam}n:f>r3`=k&]fU>-dv([jFcV');
define('AUTH_SALT',        '*RLiSW8)2/pT^y,o6hY+Y?DVm[eV*Zfhrk{!+TeQf*9H8*.9 o-`Nt_m|eElm-pC');
define('SECURE_AUTH_SALT', ')R pm4+;@lB;j-}ZwzV?|Gd<%On}SuW!>lGzo)SFJjly &Y)O|ZhN0b_$?GNpPVe');
define('LOGGED_IN_SALT',   '*]ikqKf5PQ$k}G82^SJFhub+&^P@M5!aoS4U|ukf]I>+oN+qRa{H]eW 1 WIhq.U');
define('NONCE_SALT',       'F>a5}84YKR>v3dzz<B @s@mCn# k/qQk3kA@1P 0^u|pE:pzvD>$n>R3|le@XA,.');
alix@alix-virtual-machine:~$
```

F) Open WordPress config file:

➔ Find section containing dummy values:

```
 *
 * @since 2.6.0
 */
define('AUTH_KEY',         'put your unique phrase here');
define('SECURE_AUTH_KEY',  'put your unique phrase here');
define('LOGGED_IN_KEY',    'put your unique phrase here');
define('NONCE_KEY',        'put your unique phrase here');
define('AUTH_SALT',        'put your unique phrase here');
define('SECURE_AUTH_SALT', 'put your unique phrase here');
define('LOGGED_IN_SALT',   'put your unique phrase here');
define('NONCE_SALT',       'put your unique phrase here');

/**#@-*/
```

➔ Delete those lines and paste values copied from earlier:

```
 */
define('AUTH_KEY',         'L+cA^MZ |vHTvs2|k#{{B=zi~9:FW!^N$N3!&-8k@ij-|H[g>Kkj2/9+z;4:P*fd');
define('SECURE_AUTH_KEY',  'Ph>,1gaxdSA,]:B:)-Xhypgm1AUo`p{)viF{MC}p*aU$g&f+W]vn;$+kghF`-4E|');
define('LOGGED_IN_KEY',    '(5?)uLq(W!+9n~WM1W*-O;V1q^mypL/+Y~DX1Nq8T5]-IgJ7e-|<jA$fsM,g0fn&');
define('NONCE_KEY',        'y26+sq}  7E,*Su+HV~+Z1vQ9cnc+coOAn~pJoam}n:f>r3`=k&]fU>-dv([jFcV');
define('AUTH_SALT',        '*RLiSW8)2/pT^y,o6hY+Y?DVm[eV*Zfhrk{!+TeQf*9H8*.9 o-`Nt_m|eElm-pC');
define('SECURE_AUTH_SALT', ')R pm4+;@lB;j-}ZwzV?|Gd<%On}SuW!>lGzo)SFJjly &Y)O|ZhN0b_$?GNpPVe');
define('LOGGED_IN_SALT',   '*]ikqKf5PQ$k}G82^SJFhub+&^P@M5!aoS4U|ukf]I>+oN+qRa{H]eW 1 WIhq.U');
define('NONCE_SALT',       'F>a5}84YKR>v3dzz<B @s@mCn# k/qQk3kA@1P 0^u|pE:pzvD>$n>R3|le@XA,.');
```

G) Modify some of the database connection settings at beginning of file to adjust database name, user, and associated password we configured in MySQL. Also, add a line to set the method that WordPress uses to write the filesystem. We set it to “direct” since we gave the web server permission to write where it needs to. If this is not done, WordPress will prompt for FTP credentials when performing certain actions. Save and close when done:

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'ccdc');

/** MySQL database username */
define('DB_USER', 'ccdcuser');

/** MySQL database password */
define('DB_PASSWORD', 'P@ssw0rd');
LibreOffice Writer
/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

define('FS_METHOD', 'direct');
```

7. Complete Installation Though Web Interface
   A) Go to web browser and navigate to http://127.0.0.1/wp-admin or http://localhost/wp-admin
   B) Select Language and continue to install

   Title: planetexpress
   Username: leelat
   Password: P@ssw0rd
   Email: admin@email.com

   C) Delete index.html file on /var/www/html so it does not take precedence over index.php
8. Configure wordpress localhost redirect
   A) Check option_value in mysql database

- select * from wp_options where option_name like ‘siteurl’;
- select * from wp_options where option_name like ‘home’;

B) Change http://localhost to IP address

- update wp_options set option_value=’http://10.80.80.16’ where option_name=’siteurl’;
- update wp_options set option_value=’http://10.80.80.16’ where option_name=’home’;

Install LAMP in lxc

1. Update and install lxc

apt-get update && apt-get dist-upgrade

apt-get install lxc

2. Uncomment LXC configuration to allow static IP setting
# vi /etc/default/lxc-net
LXC_DHCP_CONFILE=/etc/lxc/dnsmasq.conf

3. Create /etc/lxc/dnsmasq.conf
vi /etc/lxc/dnsmasq.conf
dhcp-hostsfile=/etc/lxc/dnsmasq-hosts.conf

4. Create /etc/lxc/dnsmasq-hosts.conf
vi /etc/lxc/dnsmasq-hosts.conf
www,10.0.3.196
db,10.0.3.143

5. Restart lxc-net service
service lxc-net restart

6. Create containers
lxc-create -t download -n www -- -d ubuntu -r xenial -a amd64
lxc-create -t download -n db -- -d ubuntu -r xenial -a amd64

7. Start containers
lxc-start -n www
lxc-start -n db

8. Enter www container and install apache2 and php + modules + web application
lxc-attach -n www
apt-get install apache2 php php-mysql libapache2-mod-php php-xml php-mbstring mysql-client
service apache2 restart
wget https://releases.wikimedia.org/mediawiki/1.28/mediawiki-1.28.2.tar.gz
tar -xzvf mediawiki-1.28.2.tar.gz -C /var/www/html/

9. Enter db container and install mysql
lxc-attach -n db
apt-get install mysql-server wget

10. Configure remote mysql in db container
mysql -u root -p

> create database wikidb;
> grant all privileges on wikidb.* to 'ccdc'@'10.0.3.143' identified by 'P@ssw0rd';
> flush privileges;
> quit;
vi /etc/mysql/mysql.conf.d/mysqld.cnf
        bind-address 10.0.3.143
service mysql restart

11. Configure lxc autostart and start order
vi /var/lib/lxc/www/config
        # Autostart
        lxc.start.auto = 1
        lxc.start.delay = 5
        lxc.start.order = 200
# vi /var/lib/lxc/db/config
        # Autostart
        lxc.start.auto = 1
        lxc.start.delay = 5
        lxc.start.order = 100

12. Set up host iptables NAT forwarding (Use iptables command or write in /etc/iptables.up.rules ; omit 'iptables')
// Filter rules
*filter
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -j ACCEPT
iptables -A FORWARD -i lxcbr0 -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -i eth0 -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --dport 3306 -j ACCEPT
iptables -A FORWARD -i eth0 -p tcp --dport 3306 -j ACCEPT
iptables -A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
iptables -A INPUT -j REJECT
iptables -A FORWARD -j REJECT
COMMIT

*nat
// NAT rules
iptables -t nat -A PREROUTING -i eth0 -p tcp -m tcp --dport 80 ! -s 10.0.3.0/24 -j DNAT --to-destination 10.0.3.196:80
iptables -t nat -A PREROUTING -i eth0 -p tcp -m tcp --dport 3306 -j DNAT --to-destination 10.0.3.143:3306
iptables -t nat -A POSTROUTING -s 10.0.3.0/24 -j MASQUERADE
COMMIT
13. Restore iptables on boot

```
# vi /etc/network/if-up.d/iptables
        #!/bin/bash
        /sbin/iptables-restore /etc/iptables.up.rules
# chmod +x /etc/network/if-up.d/iptables
```

14. lxc usage

Start container: lxc-start -n www

List available containers: lxc-ls -f

List currently running containers: lxc-ls --active

Stop running container: lxc-stop -n www

Interact with container: lxc-attach -n www

Clone container: lxc-copy -n www -N www_clone

Snapshot container (Make sure container is stopped): lxc-stop -n www ; lxc-snapshot -L -n www

Restore snapshot: lxc-snapshot -r snap0 -n www

Destroy container (Remove snapshots first): lxc-destroy -n www

Setting up Mail Server on Fedora 25 with Squirrelmail, Dovecot, and Postfix

By Alix D.

Some Useful Information:

1. Installing vim on Fedora
   A) Update with >dnf –y update
   B) Install VIM with >dnf –y install vim-enhanced

2. Changing IP Address on Fedora.
   A) Networking file found in /etc/sysconfig/network-scripts/ifcfg-*networkinterface*
   B) In my case it was > vim /etc/sysconfig/network-scripts/ifcfg-ens33

Installing Postfix

To install use command:

- **dnf -y install postfix**

After installing, change configuration settings in /etc/postfix/main.cf
To do this I recommend installing vim, which makes it a lot easier to look for the correct lines:

```
# line 95: uncomment and specify hostname
        myhostname = mail.hostname.com
# line 102: uncomment and specify domain name
        mydomain = hostname.com
# line 118: uncomment
        myorigin = $mydomain
# line 135: change
        inet_interfaces = all
# line 138: change it if use only IPv4
        inet_protocols = ipv4
# line 183: add
        mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
# line 283: uncomment and specify your local network
        mynetworks = 127.0.0.0/8, 10.0.0.0/24
# line 438: uncomment (use Maildir)
        home_mailbox = Maildir/
# line 593: add
        smtpd_banner = $myhostname ESMTP
# add following to the end of file (OPTIONAL):
        # limit an email size for 10M
        message_size_limit = 10485760
        # limit a mailbox for 1G
        mailbox_size_limit = 1073741824
# add SMTP-Auth settings
        smtpd_sasl_type = dovecot
        smtpd_sasl_path = private/auth
        smtpd_sasl_auth_enable = yes
        smtpd_sasl_security_options = noanonymous
        smtpd_sasl_local_domain = $myhostname
        smtpd_recipient_restrictions =
```

*Source: https://www.server-world.info/en/note?os=Fedora_25&p=mail&f=1

After doing this, enable and start postfix:

- **systemctl start postfix**
- **systemctl enable postfix**

In some instances, sendmail is enabled as a mail transfer agent. To check that postfix is the only program which provides mail transfer agent, this is a useful command:

- **alternatives --config mta**

If another service, such as sendmail appears, stop the service with:

- **systemctl stop sendmail**

To make sure that postfix is running:

- **service postfix status**

Changing SELinux Policy

******* If you wish to skip this step and work without using SELinux, disable SELinux by navigating to **/etc/selinux/config** and setting **SELINUX=disabled**

SELinux default policies prevent Bitdefender milter agents to integrate with Postfix mail traffic. As a result, email server drops all email traffic.
If SELinux is enabled, it is necessary to change its policy:
Create a new file, in this case *postfix-local.te.* In this file we write a new security module:

```
module postfix-local 1.0;

require {
        type tmpfs_t;
        type sendmail_t;
        type postfix_local_t;
        type postfix_cleanup_t;
        type postfix_smtp_t;
        type postfix_smtpd_t;
        type postfix_qmgr_t;
        type postfix_master_t;
        type postfix_pickup_t;
        class lnk_file read;
}

allow sendmail_t tmpfs_t:lnk_file read;
allow postfix_local_t tmpfs_t:lnk_file read;
allow postfix_cleanup_t tmpfs_t:lnk_file read;
allow postfix_smtp_t tmpfs_t:lnk_file read;
allow postfix_smtpd_t tmpfs_t:lnk_file read;
allow postfix_pickup_t tmpfs_t:lnk_file read;
allow postfix_master_t tmpfs_t:lnk_file read;
allow postfix_qmgr_t tmpfs_t:lnk_file read;$$
~
~
```

Then, we use checkmodule to compile the policy into a binary representation (the -m flag is used to generate a non-base policy module):

- **checkmodule -m -M -o postfix-local.mod postfix-local.te**

*The new binary policy module will be written to postfix-local.mod

After, semodule_package is used to create a SELinux module package from the binary file. This package will be written into postfix-local.pp:

- **semodule_package --outfile postfix-local.pp --module postfix-local.mod**

Lastly, the semodule tool with the flag –i (install) is used to install the new module package

- **semodule –i postfix-local.pp**

In addition, for Squirrelmail to be able to connect to the IMAP server and login to users though the web, it is necessary to toggle the SELinux Boolean httpd_can_network_connect. To do this run the following command (note that the -P flag is important to make this change persistent, meaning that it will be applied at each reboot):

- **setsebool -P httpd_can_network_connect=1**

Allow SMTP/POP/IMAP on FirewallD

If Firewalld is running, it is necessary to allow Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), Internet Message Access Protocol (IMAP), and reload the firewall. To do this:

- **firewall-cmd --add-service={smtp,pop3,imap) --permanent**
- **firewall-cmd –reload**

Installing Dovecot

Install dovecot with following command:

- **dnf -y install dovecot**

Edit configuration settings on Dovecot to provide SASL function on Postfix

- **vim /etc/dovecot/dovecot.conf**

```
#line 24: uncomment
protocols = imap pop3 lmtp
#line 30: uncomment
listen = *, ::
```

- **vim /etc/dovecot/conf.d/10-auth.conf**

```
# line 10: uncomment and change to allow plain text auth:
disable_plaintext_auth = no
# line 100: add
auth_mechanisms = plain login
```

- **vim /etc/dovecot/conf.d/10-mail.conf**

```
# line 30: uncomment and add
mail_location = maildir:~/Maildir
```

- **vim /etc/dovecot/conf.d/10-master.conf**

```
# line 96-98: uncomment and add like follows
# Postfix smtp-auth
 unix_listener /var/spool/postfix/private/auth {
        mode = 0666
        user = postfix
        group = postfix
 }
```

*Source: https://www.server-world.info/en/note?os=Fedora_25&p=mail&f=2

Start and enable dovecot:

- **systemctl start dovecot**
- **systemcrl enable dovecot**

Testing Postfix

Connect to the server via Telnet

- **telnet localhost smtp**
- **ehlo localhost**

We are going to send a message from user1 to user2 (These are users previously created):

```
[root@localhost ~]# telnet localhost smtp
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mail.linghziexistence.com ESMTP
ehlo localhost
250-mail.linghziexistence.com
250-PIPELINING
250-SIZE 10485760
250-VRFY
250-ETRN
250-AUTH PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250 SMTPUTF8
mail from:<user1>
250 2.1.0 Ok
rcpt to:<user2>
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
Now, bring me that horizon
.
250 2.0.0 Ok: queued as 1BEBE67631
quit
221 2.0.0 Bye
Connection closed by foreign host.
[root@localhost ~]# $$
```

The commands to do this are:

- **mail from:<sendingusername>**
- **rcpt to:<receivingusername>**
- **data**
- *This is where you type the message*, type dot (.) and enter to finish
- **quit**

Email should be received under ***/home/receivingusername/Maildir/new/***

```
[root@localhost ~]# cd /home/user2/Maildir/new
[root@localhost new]# ls
1496444182.Vfd00Ic64d3M441762.localhost.localdomain
[root@localhost new]# cat 1496444182.Vfd00Ic64d3M441762.localhost.localdomain
Return-Path: <user1@linghziexistence.com>
X-Original-To: user2
Delivered-To: user2@linghziexistence.com
Received: from localhost (localhost [127.0.0.1])
        by mail.linghziexistence.com (Postfix) with ESMTP id 1BEBE67631
        for <user2>; Fri,  2 Jun 2017 17:55:37 -0500 (CDT)
Message-Id: <20170602225549.1BEBE67631@mail.linghziexistence.com>
Date: Fri,  2 Jun 2017 17:55:37 -0500 (CDT)
From: user1@linghziexistence.com

Now, bring me that horizon
[root@localhost new]#
```

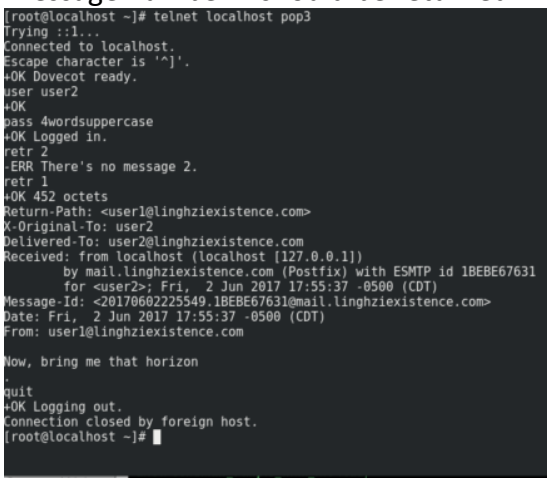
to open, use the **cat** command and the name of the message file.

Testing Dovecot

Access Dovecot through telnet

- telnet localhost pop3
- user username
- pass userpassword
- retr 1

Message number 1 should be returned



- quit

Since working with mail in command form is difficult, we will install a mail client, Squirrelmail, to send and receive messages via a web browser.

Setting up Squirrelmail

Install Squirellmail

- **dnf -y install squirrelmail**

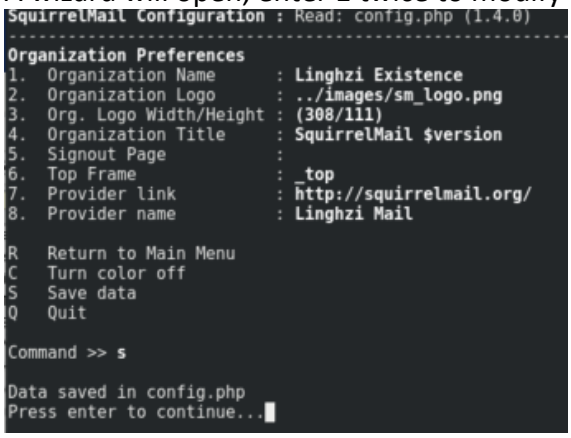
Configure settings, navigate to:

- **cd /usr/share/squirrelmail/config**

run the command conf.pl found in this folder:

- **./conf.pl**

A wizard will open, enter 1 twice to modify organization details



Once you have changed anything that you want, enter **s** to save, and **r** to return to main menu

Select 2 in main menu to edit the Server Settings:
In the same manner as in previous step, select what needs to be changed by number.
Important changes include:
Select 1 to change domain, and select 3 to change from sendmail to SMPT.

```
Server Settings

General
-------
1.  Domain                  : linghziexistence.com
2.  Invert Time             : false
3.  Sendmail or SMTP        : SMTP

A.  Update IMAP Settings    : localhost:143 (uw)
B.  Update SMTP Settings    : localhost:25

R   Return to Main Menu
C   Turn color off
S   Save data
Q   Quit

Command >>
```

Select s to save and q to quit.

Create a Squirrelmail vhost in Apache

This is done in config file httpd.conf

- **vim /etc/httpd/conf/httpd.conf**

Add the following to the end of the file:

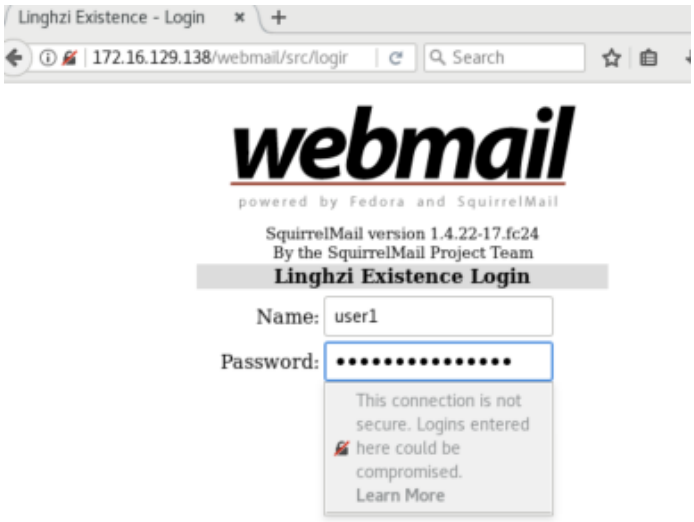
```
Alias /webmail /usr/share/squirrelmail

<Directory /usr/share/squirrelmail>
        Options Indexes FollowSymLinks
        RewriteEngine On
        AllowOverride All
        DirectoryIndex index.php
        Order allow,deny
        Allow from all
</Directory>
```

-Restart Apache service:

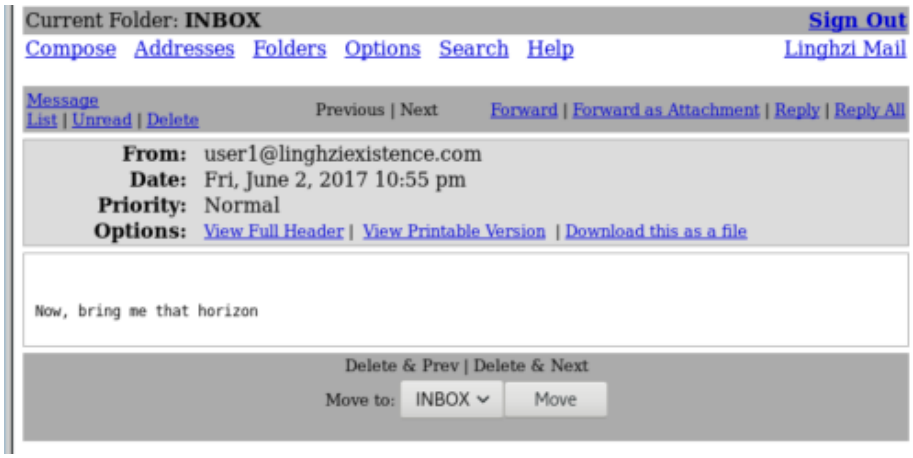
- **systemctl restart httpd**

Accessing Webmail

Navigate to ***http://ip-address/webmail.*** The following should appear:



Now you will be able to sign in to your users to send mail to each other within the local network:



Securing server using SSL

*Using SSL to encrypt SMTP/POP3/IMAP connections requires an SSL certificate. We will create a self-signed SSL certificate, which is free. However, although this type of certificate implements full encryption, a self-signed certificate will prompt security alerts on most web servers because they are not verified by a third party. In the case of mail servers, a self-signed certificate might get other mail servers to refuse to deliver your messages. For this reason, if the mail server is being used for business or other real life purposes you should use a formal certificate from a trusted certificate authority.

Creating self-signed SSL certificates

Generate a private key:

Navigate to

- **/etc/pki/tls/certs**

Generate a key with the following command:

- **make server.key**

You should now be prompted for a passphrase. The longer and more complex the passphrase is, the stronger the key will be. Enter the key, and reenter to confirm. This passphrase will need to be entered every time you start the secure server, so remember it.

Remove passphrase from Key (optional):

*If this step is not done, you will be prompted for password every time server is started. This is not always convenient. It is possible to remove the Triple-DES encryption from the key. If this step is performed, it is critical that this file is only available/readable to the root user.

- **openssl rsa -in server.key -out server.key**
- Enter passphrase. The new created server.key has no passphrase in it.

Generate a Certificate Signing Request:

- **make server.csr**
- In this step, you will be asked to enter information that will be incorporated into the certificate request. To leave a field blank enter dot (.)
  - Country: The 2 letter code for United States is US
  - Under common name, enter the domain name of the server to be protected
  - Can simply enter on the last two ‘extra’ attributes

```
[root@localhost certs]# make server.csr
umask 77 ; \
/usr/bin/openssl req -utf8 -new -key server.key -out server.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:tEXAS
Locality Name (eg, city) [Default City]:San Antonio
Organization Name (eg, company) [Default Company Ltd]:UTSA
Organizational Unit Name (eg, section) []:Linghzi
Common Name (eg, your name or your server's hostname) []:mail.linghziexistence.com
Email Address []:user1@linghziexistence.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
[root@localhost certs]#
```

This is the point at which you would send your newly created CSR to a Certificate Authority. In this example, we will self-sign the CSR. To generate the self-signed certificate:

- **openssl x509 -in server.csr -out server.crt -req -signkey server.key -days 3650**
- the -days flag indicates how many days the certificate is good for, in this case 3650.

Configuring Postfix and Dovecot for SSL

- **vim /etc/postfix/main.cf**

```
#add to end of file:
smtpd_use_tls = yes
smtp_tls_mandatory_protocols = !SSLv2, !SSLv3
smtpd_tls_mandatory_protocols = !SSLv2, !SSLv3
smtpd_tls_cert_file = /etc/pki/tls/certs/server.crt
smtpd_tls_key_file = /etc/pki/tls/certs/server.key
smtpd_tls_session_cache_database = btree:/etc/postfix/smtpd_scache
```

- **vim /etc/postfix/master.cf**

```
#line 28-30: uncomment
smtps     inet  n       -       n       -       -       smtpd
-o syslog_name=postfix/smtps
-o smtpd_tls_wrappermode=yes
```

- **vim /etc/dovecot/conf.d/10.ssl.conf**

```
#line 8: change
ssl = yes
# line 14,15: specify certificates
ssl_cert = </etc/pki/tls/certs/server.crt
ssl_key = </etc/pki/tls/certs/server.key
# line 51: uncomment and add
ssl_protocols = !SSLv2 !SSLv3
```

*Source: https://www.server-world.info/en/note?os=Fedora_25&p=mail&f=4

Restart Postfix/Dovecot:

- **systemctl restart postfix dovecot**

Allow SMTPS/POP3s/IMAPS/HTTPS services on Firewalld:

- **firewall-cmd --add-service={smtps,pop3s,imaps,https) --permanent**
- **firewall-cmd --reload**

Securing Squirrelmail/Apache using SSL

Make sure SSL module is installed

- **dnf -y install mod_ssl**

Navigate to and edit:

- **vim /etc/httpd/conf.d/ssl.conf**

```
#Edit the following lines:
#Lines 59-60
DocumentRoot /usr/share/squirrelmail
ServerName mail.hostname.com
#Line 70
SSLEngine on
#Line 98
SSLCertificateFile: /etc/pki/tls/certs/server.crt
#Line 105
SSLCertificateKeyFile: /etc/pki/tls/certs/server.key
```

*Make sure you use the paths to the certificate/key created previously

Access Squirrelmail with a web browsers using https. Use either https://ip-address/webmail or https://domain/webmail. Note that the web browser will give you a warning about the website not being secure. This is because the certificate we used is self-signed.

Redirecting http to https

Navigate to and edit:

- **vim /etc/httpd/conf/httpd.conf**

Where the squrrelmail vhost is specified, edit/add these lines to redirect http to https:

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
```

- **systemctl restart httpd**

Access Squrrelmail through http on a browser and you should be redirected to the https

```
Alias /webmail /usr/share/squirrelmail

<Directory /usr/share/squirrelmail>
        Options Indexes FollowSymLinks
        RewriteEngine On
        RewriteCond %{HTTPS} off
        RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
        AllowOverride All
        DirectoryIndex index.php
        Order allow,deny
        Allow from all
</Directory>
```

Mail server on Debian 9
Local IP: 10.30.30.9

Postfix:

1. Change **/etc/host.conf** to read host file first, add:
   order hosts,bind
   multi on

2. Set up FQDN to add domain name to **/etc/hosts**
   > hostnamectl set-hostname mail.abra.local
   > echo “10.30.30.9 abra.local mail.abra.local” >> /etc/hosts
   > init 6

3. Install postfix
   >**apt-get install postfix**
   On installation select Internet Site

4. Backup original postfix and edit main config on Postfix (/etc/postfix/main.cf)
   > cp /etc/postfix/main.cf{,.backup}
   > vim /etc/postfix/main.cf
   SEE APPENDIX A

5. Check for errors/dump PostFix main file with command:
   > **postconf -n**

6. Restart postfix and check:
   > systemctl restart postfix
   > systemctl status postfix
   > netstat -tlpn
   Postfix **master** service should be binding on **port 25.**

7. Test postfix:
   > apt-get install mailutils
   > echo “mail body” | mail -s “test mail” root
   > mailq
   > mail
   New mail should be under:
   ls Maildir/new/ or /home/username/Maildir/new

8. Configure Maildir variable for every user:
   > **echo 'export MAIL=$HOME/Maildir' >> /etc/profile**

9. **Log file for mail:**
   > tailf /var/log/mail.log

Dovecot IMAP:

1. Insall:
   > **apt install dovecot-core dovecot-imapd**

2. edit /etc/dovecot/dovecot.conf
   -Uncomment line 30
   -See APPENDIX B

3. edit /etc/dovecot/conf.d/10-auth.conf
   - disable_plaintext_auth = no
   - auth_mechanisms = plain login
   - See APPENDIX C

4. edit /etc/dovecot/conf.d/10-mail.conf
   -mail_location = maildir:~/Maildir
   -See APPENDIX D

5. edit /etc/dovecot/conf.d/10-master.conf
   -Uncomment lines after #Postfix smtp-auth and add:
   unix_listener /var/spool/postfiz/private/auth{
   mode = 0666
   user = postfix
   group = postfix
   }

6. Restart Dovecot and check:
   > systemctl restart dovecot.service
   > systemctl status dovecot.service
   > netstat -tlpn
   *Dovecot should be binding on **port 143**

Extra (OPTIONAL):
Redirect root's email to a different local mail account:
> echo "root: abra" >> /etc/aliases
> newaliases

Test server using nc or telnet:

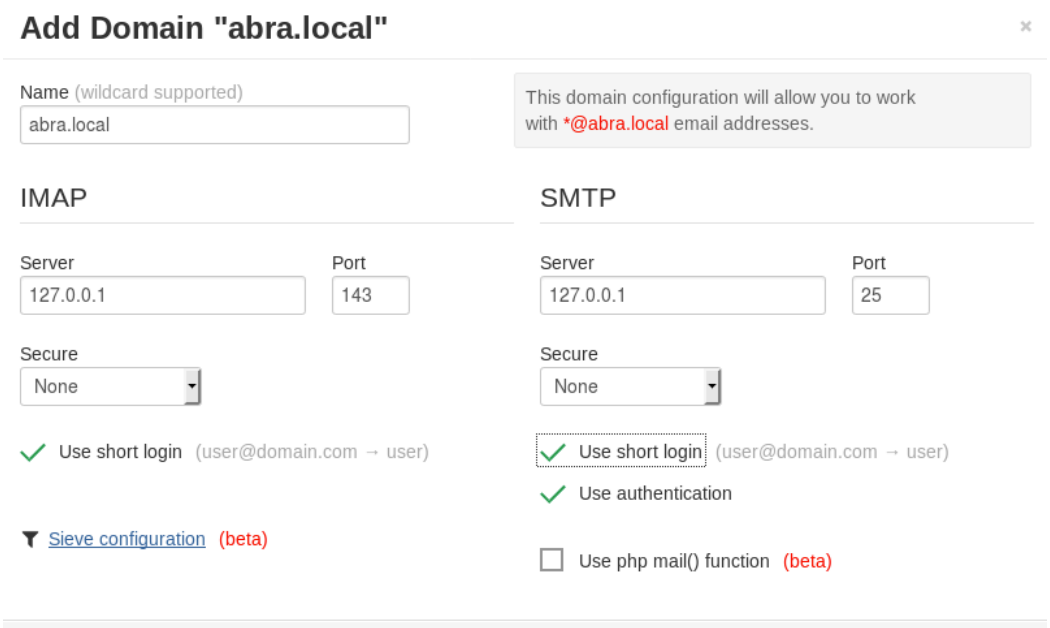
```
root@mail:/home/abra/Maildir/new# nc localhost 25
220 mail.abra.local ESMTP
ehlo localhost
250-mail.abra.local
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250 SMTPUTF8
mail from: root
250 2.1.0 Ok
rcpt to: kadabra
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
subject: test
helllo kadabra
.
250 2.0.0 Ok: queued as C6DF665788
quit
221 2.0.0 Bye
root@mail:/home/abra/Maildir/new#
```

Webmail

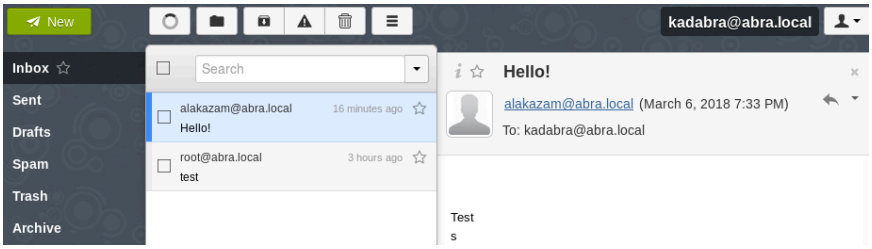
1. Prepare:
   > apt install apache2 php7.0 libapache2-mod-php7.0 php7.0-curl php7.0-xml
   - Go to /var/www/html/ and **rm index.html**

2. Install RainLoop:
   > curl -sL https://repository.rainloop.net/installer.php | php

3. Navigate to: http://10.30.30.9/?admin
   Log in with:
   User - admin
   Pass - 12345
   **Remember to change this default asap.*
4. Navigate to Domains and Add Domain:



5. After, log out of admin and you should be able to log in as such:
   - username@$mydomain
   ie. kadabra@abra.local



Appendix A: **/etc/postfix/main.cf**

```
smtpd_banner = $myhostname ESMTP
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# See http://www.postfix.org/COMPATIBILITY_README.html -- default to 2 on
# fresh installs.
compatibility_level = 2

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
```

```
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_d
estination
myhostname = mail.abra.local
mydomain = abra.local
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = $mydomain
mydestination = $myhostname, $mydomain, localhost.$mydomain, localhost
relayhost =
mynetworks = 127.0.0.0/8 10.30.30.0/24
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = ipv4
home_mailbox = Maildir/

#SMTP-Auth settings
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain = $myhostname
smtpd_recipient_restrictions = permit_mynetworks, permit_auth_destination,permit_sasl
_authenticated, reject
```

Appendix B - **/etc/dovecot/dovecot.conf**

```
# those. These are exceptions to this though: No sections (e.g. namespace {})
# or plugin settings are added by default, they're listed only as examples.
# Paths are also just examples with the real defaults being based on configure
# options. The paths listed here are for configure --prefix=/usr
# --sysconfdir=/etc --localstatedir=/var

# Enable installed protocols
!include_try /usr/share/dovecot/protocols.d/*.protocol

# A comma separated list of IPs or hosts where to listen in for connections.
# "*" listens in all IPv4 interfaces, "::" listens in all IPv6 interfaces.
# If you want to specify non-default ports or anything more complex,
# edit conf.d/master.conf.
listen = *, ::

# Base directory where to store runtime data.
#base_dir = /var/run/dovecot/

# Name of this instance. In multi-instance setup doveadm and other commands
# can use -i <instance_name> to select which instance is used (an alternative
# to -c <config_path>). The instance name is also added to Dovecot processes
# in ps output.
#instance_name = dovecot
# these networks. Typically you'd specify your IMAP proxy servers here.
#login_trusted_networks =

# Space separated list of login access check sockets (e.g. tcpwrap)
#login_access_sockets =

# With proxy_maybe=yes if proxy destination matches any of these IPs, don't do
# proxying. This isn't necessary normally, but may be useful if the destination
# IP is e.g. a load balancer's IP.
#auth_proxy_self =

# Show more verbose process titles (in ps). Currently shows user name and
# IP address. Useful for seeing who are actually using the IMAP processes
# (eg. shared mailboxes or if same uid is used for multiple accounts).
#verbose_proctitle = no

# Should all processes be killed when Dovecot master process shuts down.
# Setting this to "no" means that Dovecot can be upgraded without
# forcing existing client connections to close (although that could also be
# a problem if the upgrade is e.g. because of a security fix).
#shutdown_clients = yes

##
## Dictionary server settings
##

# Dictionary can be used to store key=value lists. This is used by several
# plugins. The dictionary can be accessed either directly or though a
# dictionary server. The following dict block maps dictionary names to URIs
# when the server is used. These can then be referenced using URIs in format
# "proxy::<name>".

dict {
  #quota = mysql:/etc/dovecot/dovecot-dict-sql.conf.ext
  #expire = sqlite:/etc/dovecot/dovecot-dict-sql.conf.ext
}

# Most of the actual configuration gets included below. The filenames are
# first sorted by their ASCII value and parsed in that order. The 00-prefixes
# in filenames are intended to make it easier to understand the ordering.
!include conf.d/*.conf

# A config file can also tried to be included without giving an error if
# it's not found:
!include_try local.conf$
```

Appendix C- **/etc/dovecot/conf.d/10-auth**

```
## Authentication processes
##

# Disable LOGIN command and all other plaintext authentications unless
# SSL/TLS is used (LOGINDISABLED capability). Note that if the remote IP
# matches the local IP (ie. you're connecting from the same computer), the
# connection is considered secure and plaintext authentication is allowed.
# See also ssl=required setting.
disable_plaintext_auth = no

# Authentication cache size (e.g. 10M). 0 means it's disabled. Note that
# bsdauth, PAM and vpopmail require cache_key to be set for caching to be used.
#auth_cache_size = 0
# Time to live for cached data. After TTL expires the cached record is no
# longer used, *except* if the main database lookup returns internal failure.
# We also try to handle password changes automatically: If user's previous
# authentication was successful, but this one wasn't, the cache isn't used.
# For now this works only with plaintext authentication.
#auth_cache_ttl = 1 hour
# TTL for negative hits (user not found, password mismatch).
# 0 disables caching them completely.
#auth_cache_negative_ttl = 1 hour
# Space separated list of wanted authentication mechanisms:
#   plain login digest-md5 cram-md5 ntlm rpa apop anonymous gssapi otp skey
#   gss-spnego
# NOTE: See also disable_plaintext_auth setting.
auth_mechanisms = plain login

##
## Password and user databases
##

#
# Password database is used to verify user's password (and nothing more).
# You can have multiple passdbs and userdbs. This is useful if you want to
# allow both system users (/etc/passwd) and virtual users to login without
# duplicating the system users into virtual database.
#
# <doc/wiki/PasswordDatabase.txt>
#
# You can have multiple passdbs and userdbs. This is useful if you want to
# allow both system users (/etc/passwd) and virtual users to login without
# duplicating the system users into virtual database.
#
# <doc/wiki/PasswordDatabase.txt>
#
# User database specifies where mails are located and what user/group IDs
# own them. For single-UID configuration use "static" userdb.
#
# <doc/wiki/UserDatabase.txt>

#!include auth-deny.conf.ext
#!include auth-master.conf.ext

!include auth-system.conf.ext
#!include auth-sql.conf.ext
#!include auth-ldap.conf.ext
#!include auth-passwdfile.conf.ext
#!include auth-checkpassword.conf.ext
#!include auth-vpopmail.conf.ext
#!include auth-static.conf.ext
~
```

**Assume everything else is commented out*

Appendix D: **/etc/dovecot/conf.d/10-mail.conf**

```
#   mail_location = mbox:~/mail:INBOX=/var/mail/%u
#   mail_location = mbox:/var/mail/%d/%1n/%n:INDEX=/var/indexes/%d/%1n/%n
#
# <doc/wiki/MailLocation.txt>
#
mail_location = maildir:~/Maildir

# If you need to set multiple mailbox locations or want to change default
# namespace settings, you can do it by defining namespace sections.
#
# You can have private, shared and public namespaces. Private namespaces
# are for user's personal mails. Shared namespaces are for accessing other
# users' mailboxes that have been shared. Public namespaces are for shared
# mailboxes that are managed by sysadmin. If you create any shared or public
# namespaces you'll typically want to enable ACL plugin also, otherwise all
# users can access all the shared mailboxes, assuming they have permissions
# on filesystem level to do so.
namespace inbox {
  # Namespace type: private, shared or public
  #type = private

  # Hierarchy separator to use. You should use the same separator for all
  # namespaces or some clients get confused. '/' is usually a good one.

  # There can be only one INBOX, and this setting defines which namespace
  # has it.
  inbox = yes

  # If namespace is hidden, it's not advertised to clients via NAMESPACE
  # extension. You'll most likely also want to set list=no. This is mostly
  # useful when converting from another server with different namespaces which
  # you want to deprecate but still keep working. For example you can create
  # hidden namespaces with prefixes "~/mail/", "~%u/mail/" and "mail/".
  #hidden = no

  # Show the mailboxes under this namespace with LIST command. This makes the
  # namespace visible for clients that don't support NAMESPACE extension.
  # "children" value lists child mailboxes, but hides the namespace prefix.
  #list = yes

  # Namespace handles its own subscriptions. If set to "no", the parent
  # namespace handles them (empty prefix should always have this as "yes")
  #subscriptions = yes
```

**Assume everything else is commented out*

Appendix E: **/etc/dovecot/conf.d/10-master.conf**

```
# login user, so that login processes can't disturb other processes.
#default_internal_user = dovecot

service imap-login {
  inet_listener imap {
    #port = 143
  }
  inet_listener imaps {
    #port = 993
    #ssl = yes
  }

  # Number of connections to handle before starting a new process. Typically
  # the only useful values are 0 (unlimited) or 1. 1 is more secure, but 0
  # is faster. <doc/wiki/LoginProcess.txt>
  #service_count = 1

  # Number of processes to always keep waiting for more connections.
  #process_min_avail = 0

  # If you set service_count=0, you probably need to grow this.
  #vsz_limit = $default_vsz_limit
}

service pop3-login {
  inet_listener pop3 {
    #port = 110
  }
  inet_listener pop3s {
    #port = 995
    #ssl = yes
  }
}

service lmtp {
  unix_listener lmtp {
    #mode = 0666
  }

  # Create inet listener only if you can't use the above UNIX socket
  #inet_listener lmtp {
    # Avoid making LMTP visible for the entire internet
    #address =
    #port =
  #}
}
```

Appendix E: cont

```
service imap {
  # Most of the memory goes to mmap()ing files. You may need to increase this
  # limit if you have huge mailboxes.
  #vsz_limit = $default_vsz_limit

  # Max. number of IMAP processes (connections)
  #process_limit = 1024
}

service pop3 {
  # Max. number of POP3 processes (connections)
  #process_limit = 1024
}

service auth {
  # auth_socket_path points to this userdb socket by default. It's typically
  # used by dovecot-lda, doveadm, possibly imap process, etc. Users that have
  # full permissions to this socket are able to get a list of all usernames and
  # get the results of everyone's userdb lookups.
  #
  # The default 0666 mode allows anyone to connect to the socket, but the
  # userdb lookups will succeed only if the userdb returns an "uid" field that
  # matches the caller process's UID. Also if caller's uid or gid matches the
```

CHANGE BELOW (#Postfix smtp-auth)

```
  # To give the caller full permissions to lookup all users, set the mode to
  # something else than 0666 and Dovecot lets the kernel enforce the
  # permissions (e.g. 0777 allows everyone full permissions).
  unix_listener auth-userdb {
    #mode = 0666
    #user =
    #group =
  }

 # Postfix smtp-auth
  unix_listener /var/spool/postfix/private/auth {
    mode = 0666
    user = postfix
    group = postfix
  }

  # Auth process is run as this user.
  #user = $default_internal_user
}

service auth-worker {
  # Auth worker process is run as root by default, so that it can access
  # /etc/shadow. If this isn't necessary, the user should be changed to


service dict {
  # If dict proxy is used, mail processes should have access to its socket.
  # For example: mode=0660, group=vmail and global mail_access_groups=vmail
  unix_listener dict {
    #mode = 0600
    #user =
    #group =
  }
}
```

DNS Config Files

/etc/bind/named.conf.local

```
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";


zone "alix.local"{
        type master;
        file "/etc/bind/db.alix.local";
};

zone "30.30.10.in-addr.arpa"{
        type master;
        notify no;
        file "/etc/bind/db.alix.local.rev";

};
```

/etc/bind/named.conf.options

```
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

         forwarders {
               8.8.8.8;
               8.8.4.4;
         };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation auto;

        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };
};
```

Forward Lookup file (specified in first block of named.conf.local) :
**Note: A default/empty version of this file can be copied from /etc/bind/db.local*

```
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     dns.alix.local. root.alix.local. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      dns.alix.local.
@       IN      A       127.0.0.1
@       IN      AAAA    ::1

alix.local.     IN      MX      10      mail.alix.local.
dns     IN      A       10.30.30.94
sense   IN      A       10.30.30.1
mail    IN      A       10.30.30.56
```

Reverse Lookup file (specified in second block of named.conf.local) :
**Note: A default/empty version of this file can be copied from /etc/bind/db.127*

```
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     dns.alix.local. root.alix.local. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      dns.
;1.0.0  IN      PTR     dns.
94      IN      PTR     dns.alix.local.
1       IN      PTR     sense.alix.local.
56      IN      PTR     mail.alix.local.
```

Testing

>nslookup sense.alix.local
>dig sense.abra.local +short
>dig -x 10.30.30.1 +short

Find DNS:
>traceroute 10.30.30.1
In case no DNS, this command might work to find hostames:
>smbclient 10.30.30.7

Other useful commands:
>host 10.30.30.80
>arp -a
>nmap -A

Installing Arch Linux and VMware guest tools in VMware Workstation Pro

1. Download Arch Linux ISO from https://www.archlinux.org/download/
2. Create Virtual Machine for Arch Linux
    a. File > New VM > Typical > Browse ISO > Other Linux 3.x Kernel
    b. Maximum disk size ‘n’ GB, Store as a single file
    c. Customize Hardware
        i. Memory (Adjust as preferred)
        ii. Processors > Number of processors 1, Number of cores per processor 2
        iii. Network Adapter > Bridged
        iv. Remove USB Controller and Printer
3. Power on VM and begin installation
    a. Boot 32-bit Arch Linux (i686)
4. Partitioning Disk
    a. View disk layout with `fdisk -l`
    b. Create partitions with `cfdisk /dev/sda`
    c. Select label type `dos`
    d. First partition: New > 1G > Primary > Bootable
    e. Second partition: New > 2G > Primary > Type > Linux swap
    f. Third partition: New > Remaining amount > Primary
    g. Write changes to disk > `yes` > quit
    h. Reconfirm disk layout with `fdisk -l`. There should be three partitions
    i. Format partitions with `mkfs.ext4 /dev/sda1` starting with the boot partition

j. `mkswap /dev/sda2` for the swap partition

k. `mkfs.ext4 /dev/sda3` for the Linux file system partition

l. Mount Linux file system partition with `mount /dev/sda3 /mnt`

m. `mkdir /mnt/boot /mnt/var /mnt/home`

n. Mount boot partition with `mount /dev/sda1 /mnt/boot/`

5. Install base operating system with `pacstrap /mnt base`

    a. Install base development with `pacstrap /mnt base-devel`

    b. Update the system with `pacman –Syu`

    c. Verify the updates again with `pacman –Syu`

6. Customize Installation

    a. Create fstab file: `genfstab –p /mnt >> /mnt/etc/fstab`

    b. Access the file system with `arch-chroot /mnt`

    c. Set the clock with `hwclock –systohc –utc`

    d. Generate image: `mkinitcpio –p linux`

    e. Set password for root with `passwd root`

    f. Add user ‘sysadmin’: “`useradd –m –g users –G wheel –s /bin/bash sysadmin`”

    g. Install grub with `pacman –S grub-bios`

    h. Install grub configuration to /dev/sda: `grub-install /dev/sda`

    i. Continue to configure grub: `grub-mkconfig –o /boot/grub/grub.cfg`

    j. Enable dhcp client service: `systemctl enable dhcpcd@.service`

    k. `exit` > `exit` > `reboot`

    l. Disconnect CD-ROM device. Meanwhile, on reboot the grub menu should appear

m. Add host name into `/etc/hostname`

n. Check network configuration with `ip addr`

o. Enable dhcp for specific NIC: `systemctl enable dhcpcd@ens33.service`

p. Install sudo package: `pacman -S sudo`

q. Add 'sysadmin' user to sudoers file at `/etc/sudoers`

7. Install and configure X

a. Install X server packages (required by LXDM and LXDE): `pacman -S xorg`

b. Install xterm: `pacman -S xterm`

c. `pacman -S xorg-xclock`

d. Continue installing Xorg related packages: `pacman -S xorg-twm`

e. `pacman -S xorg-xinit`

f. `pacman -S xorg-server-utils`

g. `pacman -S mesa`

8. Install and configure LXDM and LXDE

a. Install LXDM desktop manager: `pacman -S lxdm`

b. Install LXDE desktop environment: `pacman -S lxde`

c. Enable LXDM service: `systemctl enable lxdm.service`

d. Reboot

9. Perform post-installation tasks

a. Install VMware tools

i. Open a terminal and change to root

ii. `pacman -S net-tools`

iii. `pacman -S open-vm-tools`

iv. `cat /proc/version > /etc/arch-release`

v. Change configuration for open-vm-tools with a text-editor such as

`nano /usr/lib/systemd/system/vmtoolsd.service`

vi. Under [Service] add `KillSignal=SIGKILL`

vii. `systemctl enable vmtoolsd.service`

viii. `reboot`

ix. View > Fit Guest Now

References:

https://www.youtube.com/watch?v=JguJMRu1riA

Hardening
Change Passwords for Services
Apache Hardening
Nginx Hardening
ModSecurity – Apache
PHP Hardening

Change passwords for services

1. Find configuration files:
   `find. -type f -exec grep -H <keyword> {} \;`
   <keywords>: DB_PASS, dbname, PASS, password
2. Change mysql user passwords:
   a. Check mysql (DB login) users: `mysql > select user from mysql.user;`
   b. Change mysql.user passwords: `mysql > use mysql; set password for \ 'user'@'localhost'=password('newpassword'); flush privileges;`

   Ex: `mysql > set password for 'phpbb3'@'localhost'=password('P@ssw0rd');`

   c. Check web application users: `mysql > show databases;`

   Ex:
   ```
   mysql > use wordpress;
         > show tables;
         > show columns from wp_users;
         > select user_pass from wp_users;
   ```

   d. Change web application user passwords:

   Ex:
   ```
   mysql > use phpbb3;
   > update phpbb_users set user_password = md5('newpassword') where \
     username='admin';
   > flush privileges;
   ```
3. Common web applications and default configuration file locations after web root (/var/www/):
   osCommerce: `/catalog/includes/configure.php`
   `/catalog/admin/includes/configure.php`
   Magento: `/magento/app/etc/env.php`
   Wordpress: `/wordpress/wp-config.php`
   PHPBB: `/etc/dbconfig-common/phpbb3.conf`
   phpMyAdmin: `/etc/phpMyAdmin/config.inc.php`
4. Match the configuration file to the mysql database user password by editing the config file:
   Ex: `vi /wordpress/wp-config.php`
5. Change PHPBB admin password (dpkg-reconfigure method):
   Debian: `dpkg-reconfigure phpbb3`
6. Change Webmin password (perl script method):
   RedHat: `/usr/libexec/webmin/changepass.pl /etc/webmin admin newpassword`
   Debian: `/usr/share/webmin/changepass.pl /etc/webmin admin newpassword`
   FreeBSD: `/usr/local/lib/webmin/changepass.pl /usr/local/etc/webmin \ admin newpassword`
7. Change Magento password (magento CLI method):
   Navigate to magento installation: `cd /var/www/magento/`

Edit mysql.user information: `echo "select email,firstname,lastname from admin_user where username='[admin]'" | mysql -u [db_user] [db_name] -p`

Reset the password with magento CLI: `php bin/magento admin:user:create --admin-user="[admin]" --admin-password="[newpassword]" --admin-email="[test@example.com]" --admin-firstname="[Test]" --admin-lastname="[Test]`

8. Other service discovery:
    a. Open Turnkey Linux confconsole: `/usr/bin/confconsole`
9. Additional pitfalls:
    a. Three root users are generated by default when you first create a database. They are all created without passwords. At that time, the installation also recommends you set a password by running **`mysqladmin -u root password`**, which will change the password for 'root'@'%

    Solution: `DELETE FROM mysql.user WHERE Password=''; FLUSH PRIVILEGES`
    b. Other services to look out for:
        - FTP **21**
        - SMTP **25**
        - Openldap **389**
        - HTTPS **443**
        - Samba **445**
        - Cups **631**
        - Stunnel4
        - AJAX webshell-shellinabox **12320**
        - Webmin **10000** > **12321**
        - Adminer-mysql **12322**

10. Additional Reference

WordPress Documentation:
https://codex.wordpress.org/Resetting_Your_Password#Through_MySQL.2FMariaDB_Command_Line

#First step is to remove banner (version and OS)

#add the following to apache2.conf/httpd.conf (newer versions of Apache will see these in #/etc/apache2/conf-available/security.conf

ServerTokens Prod
ServerSignature Off

#then restart Apache

#to remove dir listing:

#edit httpd.conf/apache2.conf directory configs

Options -Indexes
FileETag None
#PCI compliance requires setting FileETag None in .conf

#Etag allows remote attackers to obtain sensitive information like Inode number, multipart MIME #boundary, and child process through Etag header.

#At this point you'll want to make sure you password protect any administration directorys on the #server such as install/ admin/ etc..

#to do this we will use htpasswd and an .htaccess file

#first you'll want to create a htpasswd file somewhere the server can access but NOT in the webroot.

Htpasswd -Bbc /etc/apache2/.htpasswd user password

#^^ this looks dirty but really it's saying to use bcrypt to hash the password, use the password I just #gave you in the command and create a new htpasswd file in /etc/apache2

#then, to make sure this works you'll have to make sure the AllowOverride directive in httpd.conf is set #to all NOT None

#you'll want the following in the .htaccess file

AuthUserFile /etc/apache2/.htpasswd
AuthName "Under new management, Blue team only, sorry."
AuthType Basic
Require valid-user

#To make sure apache isn't running as root,

groupadd apache

useradd -G apache apache
chown -R apache:apache /apache/binary (the binary that is apache (httpd/apache2)

#modify your .conf

User apache
Group apache

#restart apache

#you'll most likely want to chmod -R 750 bin conf your website files to ensure that no one can view or #edit them except apache

#.htaccess files can be added on a dir by dir basis UNLESS
#edit .conf

<Directory />
	Options -Indexes -Includes
	AllowOverride None
</Directory>

#You're also going to want to limit http methods to get post and head in any Directory directive you #find:

<LimitExcept GET POST HEAD>
	deny from all
</LimitExcept>

#COOKIES
#Trace enables theft of cookies, disable in httpd.conf(2.2) or security.conf(2.4)
TraceEnable off

#mitigate most CSS attacks by using HttpOnly and Secure flag in a cookie:
#mitigate clickjacking:
#make sure mod_headers is enabled

a2query -m headers

#add the following to httpd.conf (2.2) or security.conf(2.4)

Header set Set-Cookie HttpOnly;Secure
Header always append X-Frame-Options SAMEORIGIN
Header set X-XSS-Protection "1;mode=block"

#block http 1.0 (requires mod_rewrite)
a2query -m rewrite

#a2enmod rewrite
#or append:
LoadModule rewrite_module modules/mod_rewrite.so (if it's not already enabled)

RewriteEngine On
RewriteCond %{THE_REQUEST} !HTTP/1.1$
RewriteRule .* - [F]

^^^^^ in .conf

Timeout 60 (sans the 300 second timeout thing.. DoS)

#little side note on mysql users:

you'll want to add

bind = 127.0.0.1

to /etc/my.cnf to make it only listen locally

#to add a new mysql user for the website to use:

create user 'username'@'localhost' identified by 'password';

grant create,insert,select,update,delete on oscommerce.* to 'username'@'localhost';

#alternatively (and possibly easier) just do the following to make it impossible #for the current mysql user to drop or delete tables:

show grants for 'oscadmin'@localhost;

will show all privileges which that user has on what databases/tables. Keep in mind that what is shown is a list of commands that you would have to type to replicate those privileges. GRANT USAGE ON… is another way of saying the
user has no privileges other than to login to the server.

revoke drop on *.* from 'oscadmin'@localhost;

will only work if they were granted those permissions explicitly, usually you're going to have to run a revoke all privileges command and then grant only creationary privileges, ie the grant command seen above

SSL NOT FINISHED!!!!

#openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout localhost.key -out localhost.crt

#openssl req -out localhost.csr -new -newkey rsa:2048 -nodes -keyout localhost.key

httpd-ssl.conf:

SSLCertificateFile localhost.crt
SSLCertificateKeyFile localhost.key
SSLCACertificateFile localhost.csr

Hardening Nginx

1. Upgrade Nginx and necessary dependencies to the latest version
```
# apt-get upgrade nginx
```

2. Check if Nginx name and version shows in HTTP headers
```
# curl -I http://localhost
```
If output looks like:
HTTP/1.1 200 OK
Server: nginx/1.6.2
X-Powered-By: PHP/5.5.9-lubuntu4.14

Hide this information by editing /etc/nginx/nginx.conf: `# nano /etc/nginx/nginx.conf`
http {
server_tokens off;
...

Reload Nginx configuration: `# service nginx reload`

Hide PHP information by editing php.ini
expose_php = Off

3. Disable client-side error pages: `# nano /etc/nginx/sites-enabled/default`
server {
...
error_page 401 403 404 /404.html;
...

Reload Nginx configuration: `# service nginx reload`

4. Configure SSL
```
# mkdir /etc/nginx/ssl
# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout \
    /etc/nginx/ssl/nginx.key -out /etc/nginx/ssl/nginx.crt
```
// Most important line is Common Name (server FQDN). Enter domain name or IP address

Example entries:

Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:New York
Locality Name (eg, city) []:New York City
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Bouncy Castles, Inc.
Organizational Unit Name (eg, section) []:Ministry of Water Slides
Common Name (e.g. server FQDN or YOUR name) []:your_domain.com
Email Address []:admin@your_domain.com

Edit server block:

default	SSL

server {
listen 80 default_server;
listen [::]:80 default_server ipv6only=on;
 	
root /usr/share/nginx/html;
index index.html index.htm;
 	
server_name your_domain.com;
 	
location / {
try_files $uri $uri/ =404;
}
}
	server {
	listen 80 default_server;
	listen [::]:80 default_server ipv6only=on;
	
	listen 443 ssl;
	
	root /usr/share/nginx/html;
	index index.html index.htm;
	
	server_name your_domain.com;
	ssl_certificate /etc/nginx/ssl/nginx.crt;
	ssl_certificate_key /etc/nginx/ssl/nginx.key;
	
	location / {
	try_files $uri $uri/ =404;
	}
	}

Restart nginx

```
# service nginx restart
```

5. Restrict Access by IP

```
# nano /etc/nginx/sites-enabled/default
```

```
server {
...
        location /wp-admin/ {
        allow 192.168.1.1/24;
        allow 10.0.0.1/24;
        deny all;
}
...
```

7. Install additional tools:

Naxsi- third party Nginx module providing web application firewall
fail2ban- issue temporary bans on offending IP addresses
Monit- service monitoring application that can perform event-based actions

6. Perform security audit with 'wapiti' web vulnerability scanner

```
# apt-get install wapiti
# wapiti http://website.com -n 10 -b folder
```

```
#!/bin/bash

# Hide nginx version
sed -i "s/# server_tokens off;/server_tokens off;/g" /etc/nginx/nginx.conf

# Remove ETags
sed -i 's/server_tokens off;/server_tokens off;\netag off;/'
/etc/nginx/nginx.conf

# Remove default page
echo "" > /var/www/html/index.html

# Enable HttpOnly and Secure flags
sed -i "s|^\s*try_files \\\$uri \\\$uri/ =404;|try_files \\\$uri \\\$uri/
=404;\nproxy_cookie_path / \"/; secure; HttpOnly\";|" /etc/nginx/sites-
available/default

# Clickjacking Attack Protection
sed -i "s|root /var/www/html;|root /var/www/html;\nadd_header X-Frame-Options
DENY;|" /etc/nginx/sites-available/default

# XSS Protection
sed -i "s|root /var/www/html;|root /var/www/html;\nadd_header X-XSS-
Protection \"1; mode=block\";|" /etc/nginx/sites-available/default

# MIME sniffing Protection
sed -i "s|root /var/www/html;|root /var/www/html;\nadd_header X-Content-Type-
Options nosniff;|" /etc/nginx/sites-available/default

# Prevent Cross-site scripting and injections
sed -i "s|root /var/www/html;|root /var/www/html;\nadd_header Content-
Security-Policy \"default-src 'self';\";|" /etc/nginx/sites-available/default

# Set X-Robots-Tag
sed -i "s|root /var/www/html;|root /var/www/html;\nadd_header X-Robots-Tag
none;|" /etc/nginx/sites-available/default

service nginx restart


# Shamelessly ripped from http://securityblog.gr/4474/nginx-hardening-
security-script/
```

Install modsecurity [apache2] from repository

Tested on Ubuntu 16 x64

```
sudo apt-get install libapache2-mod-security2

cp /etc/modsecurity/modsecurity.conf-recommended
/etc/modsecurity/modsecurity.conf

# Line 7
sed -i ‘s/SecRuleEngine DetectionOnly/SecRuleEngine On/’
/etc/modsecurity/modsecurity.conf
# Lines 192 - 196
sed -i ‘s/SecAuditLogType Serial/SecAuditLogType Concurrent/’
/etc/modsecurity/modsecurity.conf
sed -i ‘s/SecAuditLog /#SecAuditLog ’ !*
sed -i ‘s/#SecAuditLogStorageDir /SecAuditLogStorageDir /’ !*

mkdir -p /opt/modsecurity/var/audit/
chown -R www-data:www- data /opt/modsecurity/var/audit

# git clone https://github.com/SpiderLabs/owasp-modsecurity-crs.git
wget https://github.com/SpiderLabs/owasp-modsecurity-
crs/archive/v3.0.2.tar.gz
tar -xzf v3.0.2.tar.gz
cd owasp*
cp -R rules/ /etc/modsecurity/
cp crs-setup.conf.example /etc/modsecurity/crs-setup.conf

sed -i ‘/IncludeOptional \/etc\/mod/a \ \ \ \ \ \ \ Include
\/etc\/modsecurity\/rules\/\*.conf’ /etc/apache2/mods-
available/security2.conf

systemctl restart apache2
```

Resource Link:
https://www.trustwave.com/Resources/SpiderLabs-Blog/Web-Application-Defender-s-Cookbook--CCDC-Blue-Team-Cheatsheet/
https://linode.com/docs/web-servers/apache-tips-and-tricks/configure-modsecurity-on-apache/

Hardening PHP

Open php.ini file - if location is unknown, you can find it using:

```
➢ php -i | grep “php.ini”
```

Disable Functions

Find `disable_functions` and set new list as:

```
disable_functions=exec,passthru,shell_exec,system,proc_open,popen,curl
_exec,curl_multi_exec,parse_ini_file,show_source
```

Full list of PHP functions which could be insecure:

apache_child_terminate, apache_setenv, define_syslog_variables, escapeshellarg, escapeshellcmd, eval, exec, fp, fput, ftp_connect, ftp_exec, ftp_get, ftp_login, ftp_nb_fput, ftp_put, ftp_raw, ftp_rawlist, highlight_file, ini_alter, ini_get_all, ini_restore, inject_code, mysql_pconnect, openlog, passthru, php_uname, phpAds_remoteInfo, phpAds_XmlRpc, phpAds_xmlrpcDecode, phpAds_xmlrpcEncode, popen, posix_getpwuid, posix_kill, posix_mkfifo, posix_setpgid, posix_setsid, posix_setuid, posix_setuid, posix_uname, proc_close, proc_get_status, proc_nice, proc_open, proc_terminate, shell_exec, syslog, system, xmlrpc_entity_decode

Disable Remote File Includes

```
allow_url_fopen=Off
allow_url_include=Off
```

Restrict Includes

This limits PHP operations to listed directory and below.

```
open_basedir = /path/to/web/root
```

ie. *open_basedir ="/var/www/html/:/usr/local/php/"*

Limit Max Size of Post Requests

```
post_max_size = 256K
```

Restrict or Turn Off File Uploads

If not using upload functionality turn off:

```
file_uploads = Off
```

If you are using it, restrict:

```
upload_max_filesize = 128KB
```

Save & close and restart the httpd/apache2 server.

Another method to find php.ini:

1. Create a simple PHP script (ie. phpinfo.php) which displays PHP info containing:
   `<?php phpinfo(); ?>`
2. Place in web root directory (/var/www/html)
3. With browser, navigate to *http://localhost/phpinfo.php* , and the location should appear under "*Loaded Configuration File*"

Hunting
Setting up Graylog with docker
Linux Hunting Notes

Tested on Ubuntu 16 amd64:

Installing docker:
Add the gpg key for official docker repository to the system
```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Add docker repository to apt sources
```
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"

sudo apt-get update
sudo apt-get install -y docker-ce
```

To avoid typing sudo for docker commands, add your user to the docker group
```
sudo usermod -aG docker ${USER}
```

Installing docker-compose:
```
sudo curl -L
https://github.com/docker/compose/releases/download/1.18.0/docker-compose-
`uname -s`-`uname -m` -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose
```

Create docker-compose.yml with the following:
```
=======================================
version: '2'
services:
  some-mongo:
    image: "mongo:3"
  some-elasticsearch:
    image: "elasticsearch:2"
    command: "elasticsearch -Des.cluster.name='graylog'"
  graylog:
    image: graylog2/server:2.1.1-1
    environment:
      GRAYLOG_PASSWORD_SECRET: somepasswordpepper
      GRAYLOG_ROOT_PASSWORD_SHA2:
8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918
      GRAYLOG_WEB_ENDPOINT_URI: http://127.0.0.1:9000/api
    links:
      - some-mongo:mongo
      - some-elasticsearch:elasticsearch
    ports:
```

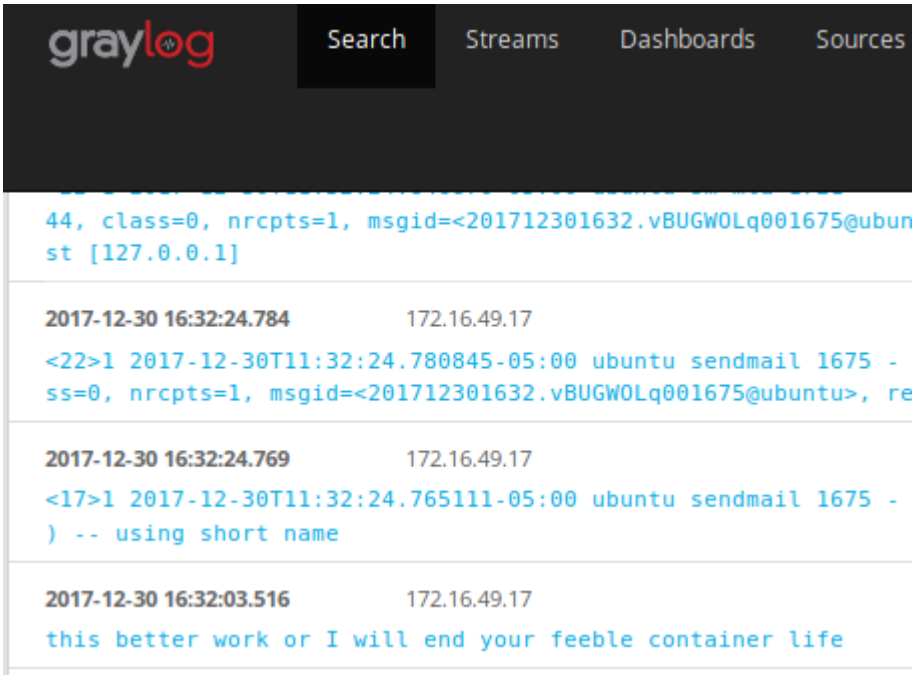
```
      - "9000:9000"
      - "5555:5555"
      - "12201:12201/udp"
      - "514:514/udp"
========================================
```

Generate graylog web admin password hash	`echo -n P@ssw0rd | shasum -a 256`

Start up the three containers with `docker-compose up`, then navigate to http://127.0.0.1:9000 and login with admin:admin

In the web interface, go to System > Inputs and launch a new input to match to the corresponding ports and types of data to be captured. Afterwards, "Show received messages" in the local inputs section to watch the logs begin to roll in.

Forward rsyslog to graylog server, edit /etc/rsyslog.conf with the following:	
Send a test message to graylog	`echo 'test' | nc localhost 5555`
Check if client is forwarding logs to graylog server	`sudo tcpdump -i ens33 host 172.16.49.17 and udp port 514`
UDP	`*.* @172.16.49.16:514;RSYSLOG_SyslogProtocol23Format`
TCP	`*.* @@172.16.49.16:5555;RSYSLOG_SyslogProtocol23Format`



References:

https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-16-04
https://hub.docker.com/r/graylog2/server/
https://docs.docker.com/compose/install/#install-compose
https://github.com/Graylog2/graylog2-server/issues/2996
https://www.graylog.org/blog/28-centralized-docker-container-logging-with-native-graylog-integration
https://stackoverflow.com/questions/38736062/how-to-send-logs-to-graylog-docker

Hunt - Finding the bad guys

Goal:

- Find weakness in the security posture of the network or system
- FInd malicious binaries, services, backdoors, user accounts, processes
- Educate users and sys admins which services are exposed and how to secure them
- Kick out bad actors
- Clean up the system
- Repeat

“Security is not a solution-- it is a process”

Where to start?

Everywhere

What is mission critical to company?

What is being scored/brings in revenue?

What is worst to management/scoring [prioritization]

Downtime or being hacked

Adjust priorities

Compromise, malicious user, or rogue admin?

Top level-

- Get an information snapshot of the system
  - Kernel version: uname
  - Get list of users
    - How many are admin
    - Who is a valid user
    - Who is currently logged in
      - Is SSH root login allowed?
  - Look at current services and processes running
    - Is anything out of place or blatantly bad
      - nc?
      - .ru website
  - Network
    - Listening ports
    - Any active connections?
      - Mysql connection to rogue IP instead of localhost
    - Firewall enabled?

Find abnormalities [time to dig deeper]

Network connections

```
lsof -ni
ip a
ifconfig -a
netstat -tlpn
arp
sockstat -l
nmap -sT -sU -PN -O localhost
ss -lntu
route
netstat -route
```

etc.

Finding users

```
cut -d: -f1 /etc/passwd
awk -F':' '{ print $1}' /etc/passwd
getent passwd | cut -d: -f1
```

Find admins

```
getent group wheel
getent group sudo
grep "^UID_MIN" /etc/login.defs
```

Minimum number a user will have, the rest are system users

```
getent passwd | cut -d : -f 1 | xargs groups
```

List all user and groups

```
getent passwd | cut -f1 -d: | sudo xargs -L1 sudo -l -U | grep \
    -v 'not allowed'
who
w
lastlog
last
```

Look for suspicious processes

Look for running processes

```
pstree
```

Look for strangely named binaries, shell scripts

Look for processes that are branched from weird locations

```
lsof -ni
ps aux | grep <pid>
```

Cronjobs

```
for user in $(cut -f1 -d: /etc/passwd); do crontab -u $user -l; \
    done
```

Logs

/var/log/messages

/var/log/auth.log
/var/log/syslog
/var/log/secure
/var/log/ufw.log
/var/log/apache2/{access.log,**error.log**,other*}
error.log will show shell shock attempts
/var/log/fail2ban.log
/var/log/kern.log
/var/log/dpkg.log
/var/log/alternatives.log
/var/log/dmesg
/var/log/faillog
/var/log/apport.log
/var/log/lynis.log
baseline
/var/log/mail.log
/var/log/psad
Intrusion detection with iptables
/var/log/mysql/error.log

Find recently modified,created files

```
find . -type f -ctime -7
find . -type f -mtime -7
```

Look at /etc/{cron.d,cron.daily,cron.hourly,cron.monthly,crontab,cron.weekly}

```
history
```

Cheatsheet:
goo.gl/mv17hh

Notes:
telnet port 23 [mirai, reaper]

Miscellaneous
Host Info Sheet
Repository Guide
FreeBSD Quick Guide
Networking on Ubuntu 17

This will be identified by box number…. _____ on date:

_____

This info sheet is to be filled out and sent to networking team ASAP

What OS is it? `cat /etc/issue` (Linux) `sysinfo` (Windows)	
What ports are listening? `lsof -Pi` (Linux) or `netstat -tulpn` `netstat -ano` (Windows)	
What is the IP address? `ifconfig` (Linux) `ipconfig /all` (Windows)	
What is the gateway? `cat /etc/network/interfaces` (Linux) or `route -n`	
What is the dns server? `cat /etc/resolv.conf` (Linux)	
Can you ping 8.8.8.8? (Is it networked?)	
Can you ping espn.com? (Is it resolving?)	
Other notes:	

Ubuntu Repository Guide

`no add-apt-repository`

Solution:

```
apt-get install python-software-properties
apt-get install software-properties-common
```

To add repositories:

Edit /etc/apt/sources.list

Format will be: `deb http://mirrorlist release_name main restricted`

Official mirrorlist @ https://launchpad.net/ubuntu/+archivemirrors

http://archive.ubuntu.com/ubuntu/

Old Releases (end of life) @ http://old-releases.ubuntu.com/

Find the release_name with `lsb_release -sc`

Example:

```
deb http://ftp.utexas.edu/ubuntu/ release_name main restricted
deb http://us.archive.ubuntu.com/ubuntu/ yakkety main restricted
universe multiverse
```

Example for old releases:

```
deb http://old-releases.ubuntu.com/ubuntu/ release_name main
restricted universe multiverse
deb http://old-releases.ubuntu.com/ubuntu/ release_name-updates main
restricted universe multiverse
deb http://old-releases.ubuntu.com/ubuntu/ release_name-security main
restricted universe multiverse
```

+https://superuser.com/questions/339537/where-can-i-get-the-repositories-for-old-ubuntu-versions

+https://askubuntu.com/questions/148932/how-can-i-get-a-list-of-all-repositories-and-ppas-from-the-command-line-into-an

Ubuntu Sources List Generator @ https://repogen.simplylinux.ch/

Debian Repository Guide

Official mirrorlist @: https://www.debian.org/mirror/list
US mirror @: http://ftp.us.debian.org/debian/

Edit /etc/apt/sources.list

Format:
```
deb http://site.example.com/debian distribution component1 component2
component3
```

Example entry:
```
deb http://deb.debian.org/debian stretch main
```

+https://wiki.debian.org/SourcesList#Repository_URL

To install from source:
+https://wiki.debian.org/Packaging/SourcePackage?action=show&redirect=SourcePackage

In the case of using an End-of-Life distribution, do the following:
+https://www.howtoforge.com/using-old-debian-versions-in-your-sources.list
+https://www.debian.org/distrib/archive
Edit /etc/apt/sources.list with http://archive.debian.org/:
```
deb http://archive.debian.org/debian/ etch main non-free
contrib

apt-get install debian-archive-keyring

apt-get update
```

FreeBSD Repository Guide

Source: https://www.freebsd.org/doc/handbook/ports-finding-applications.html

FreeBSD mirrors @ https://pkg.freebsd.org/

https://svnweb.freebsd.org/base/

FIX REPOSITORIES

FEDORA

- 32 bits:

```
yum-config-manager --add-repo=http://download.fedoraproject.org/pub/fedora/linux/updates/testing/20/i386/
```

- 64 bits:

```
yum-config-manager --add-repo=http://download.fedoraproject.org/pub/fedora/linux/updates/testing/20/x86_64/
```

Otherwise, you could try to copy paste my own `/etc/yum.repos.d/fedora-updates-testing.repo` file:

- Type as root : `gedit /etc/yum.repos.d/fedora-updates-testing.repo`
- Paste the content of my *.repo* file below and save.
- Update the database and see if it's working: `yum check-update`

My `fedora-updates-testing.repo` file (fedora 20):

```
[updates-testing]
name=Fedora $releasever - $basearch - Test Updates
failovermethod=priority
#baseurl=http://download.fedoraproject.org/pub/fedora/linux/updates/testing/$releasever/$basearch/
metalink=https://mirrors.fedoraproject.org/metalink?repo=updates-testing-f$releasever&arch=$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$releasever-$basearch

[updates-testing-debuginfo]
name=Fedora $releasever - $basearch - Test Updates Debug
failovermethod=priority
#baseurl=http://download.fedoraproject.org/pub/fedora/linux/updates/testing/$releasever/$basearch/debug/
metalink=https://mirrors.fedoraproject.org/metalink?repo=updates-testing-debug-f$releasever&arch=$basearch
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$releasever-$basearch

[updates-testing-source]
name=Fedora $releasever - Test Updates Source
failovermethod=priority
#baseurl=http://download.fedoraproject.org/pub/fedora/linux/updates/testing/$releasever/SRPMS/
metalink=https://mirrors.fedoraproject.org/metalink?repo=updates-testing-source-f$releasever&arch=$basearch
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$releasever-$basearch
```

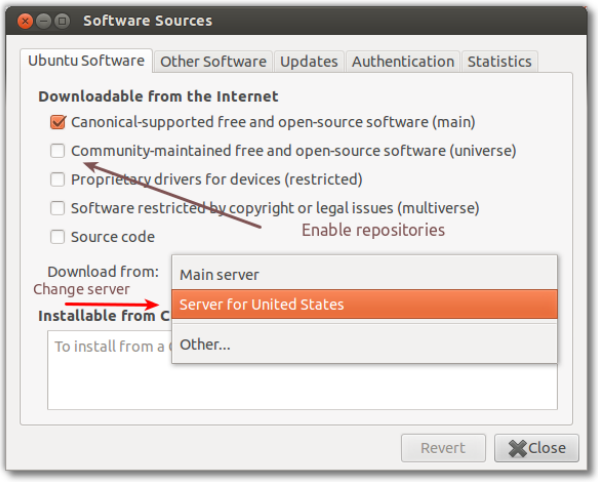
UBUNTU

UBUNTU SOURCES LIST GENERATOR:
https://repogen.simplylinux.ch/

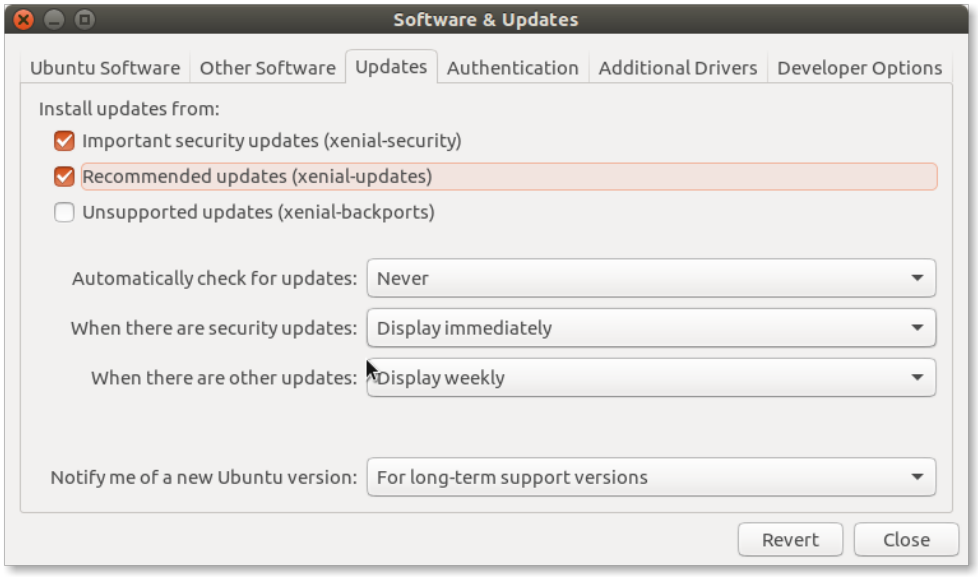
You can use this trick. Open a terminal (Pressing **Ctrl+Alt+T**) and do these

- Remove the corrupted one
- `sudo rm /etc/apt/sources.list`
- Open software-properties-gtk

- `sudo -i software-properties-gtk`
  This will open `software-properties-gtk` with no repository selected.
  Then change the server to US or to any other server of your choice. You must enable some repositories from the new window in order to create new `sources.list` file in `/etc/apt/`.



- After enabling some sources from **Ubuntu software** tab, you can enable the updates. To do so, switch to **Updates** tab and select one or more updates channel. I recommend selecting **security** and **updates** channel at least. (This image is later added from Ubuntu xenial, so there can be some differences)



Updated with inline content

This is the `sources.list` file for 12.04 Precise Pangolin. If you're using other release, you need to replace the `precise` word with your ubuntu release name. You can see the which name you should use with this command

```
lsb_release -c -s
```

And to replace the word you can use this `sed` command (assuming you copied the sources content in `/etc/apt/sources.list`

```
sudo sed -i "s/precise/$(lsb_release -c -s)/" /etc/apt/sources.list
```

Now, content of `sources.list` with main ubuntu mirror:

```
###### Ubuntu Main Repos
deb http://archive.ubuntu.com/ubuntu/ precise main restricted universe multiverse
deb-src http://archive.ubuntu.com/ubuntu/ precise main restricted universe multiverse

###### Ubuntu Update Repos
deb http://archive.ubuntu.com/ubuntu/ precise-security main restricted universe multiverse
deb http://archive.ubuntu.com/ubuntu/ precise-updates main restricted universe multiverse
deb http://archive.ubuntu.com/ubuntu/ precise-proposed main restricted universe multiverse
deb http://archive.ubuntu.com/ubuntu/ precise-backports main restricted universe multiverse
deb-src http://archive.ubuntu.com/ubuntu/ precise-security main restricted universe multiverse
deb-src http://archive.ubuntu.com/ubuntu/ precise-updates main restricted universe multiverse
deb-src http://archive.ubuntu.com/ubuntu/ precise-proposed main restricted universe multiverse
deb-src http://archive.ubuntu.com/ubuntu/ precise-backports main restricted universe
multiverse

###### Ubuntu Partner Repo
deb http://archive.canonical.com/ubuntu precise partner
deb-src http://archive.canonical.com/ubuntu precise partner

###### Ubuntu Extras Repo
deb http://extras.ubuntu.com/ubuntu precise main
deb-src http://extras.ubuntu.com/ubuntu precise main
```

Note 1: the word `deb` and `deb-src` refers to the repository format. `deb is for binary package and`deb-src` is for source package.
Note 2: Using `#` at the start of the line make that line a comment. Apt will ignore it so all repositories mentioned on that line will be disabled.
Note 3: There are repository lines which includes all four components `main`, `universe`, `restricted`, `multiverse`. You can disable one or more of them by removing the word.

UBUNTU - Fix Broken Packages

After trying

sudo apt-get update –fix-missing

and

sudo dpkg –configure -a
and

sudo apt-get install -f
the problem of a broken package still exist the solution is to **edit the dpkg status** file manually.

1. $ **sudo vim /var/lib/dpkg/**
2. Locate the corrupt package, and remove the whole block of information about it and save the file.

_____

Unlock the dpkg – (message /var/lib/dpkg/lock)
sudo fuser -vki /var/lib/dpkg/lock
sudo dpkg –configure -a

For 12.04 and newer:

You can **delete the lock file** with the following command:
sudo rm /var/lib/apt/lists/lock
You may also need to **delete the lock file in the cache** directory
sudo rm /var/cache/apt/archives/lock

What is FreeBSD?

FreeBSD is a UNIX-like computer operating system descended from Research Unix via the Berkeley Software Distribution (BSD).

What's the difference between FreeBSD and OpenBSD?

FreeBSD and OpenBSD share a common ancestry in the BSD family of operating systems. The differences in the operating systems can be understood by recognizing the design goals of its developers. FreeBSD's goal is to make a free general-purpose operating system whereas OpenBSD's goal is to cultivate operating system security above all else.

What is FreeBSD's port collection?

The FreeBSD ports and packages collection is a package management system containing necessary patches to allow compilation of the original application's source code. Precompiled packages also exist and can be downloaded as well.

What is the FreeBSD startup system?

Whereas many Linux distributions use the SysV init system, FreeBSD uses the traditional BSD-style init. Thus, there are no run-levels or /etc/inittab. Startup is controlled by rc scripts. Upon system boot, /etc/rc.conf is read to determine which services to start. Then, the initialization scripts in /etc/rc.d and /urs/local/etc/rc.d are run.

What is a jail?

In FreeBSD, a jail is a method of operating system virtualization where FreeBSD is partitioned into several independent mini-systems called jails.

FreeBSD Directory Structure	
/	root directory
/bin	user utilities
/boot	boot programs and configuration files
/etc	configuration files of core system utilities
/etc/rc.conf	essential system configuration
/proc	process file system
/usr	user utilities and applications
/usr/local	default location of package installations
/usr/local/etc/rc.d	startup scripts
/var	variable files (logs)
/home/<username>	home directory

- The root directory is equivalent to the C:\ drive of the Windows file system

FreeBSD User Management	
adduser	command-line utility for adding users
rmuser	command-line utility for removing users
chpass	tool for changing user database information
passwd	change user passwords
pw	modify all aspects of user accounts
pw groupmod wheel -m <username>	add user to wheel group
pw lock <username>	disable user account
/usr/local/etc/sudoers	location of sudoers configuration file

- Disabling or "locking" a user account can also be performed by manually editing /etc/master.passwd by prepending "*LOCKED*" to the user entry

FreeBSD Package Management	
pkg search <packagename>	Search package repository for an application
pkg install <packagename>	Install package
pkg delete <packagename>	Delete package
pkg upgrade	Upgrade installed packages
pkg audit -F	Check for software vulnerabilities

Service Management	
service <servicename> restart	Restart a service
service <servicename> stop	Stop a service
service <servicename> start	Start a service
vi /etc/rc.conf <servicename>_enable="YES"	Enable service to start on boot

- FreeBSD uses the run-commands (rc) system of startup scripts during system initialization and for managing services listed in /etc/rc.d
- rc primarily starts and stops services at system startup and shutdown, to start/stop/restart a service regardless of rcvar in /etc/rc.conf, execute: service <servicename> onerestart

Network Management	
vi /etc/rc.conf ifconfig_dc0="DHCP"	Setup DHCP
vi /etc/rc.conf ifconfig_dc0="inet <ipaddress> netmask <netmask>"	Setup manual IP address
echo 'defaultrouter="<defaultgateway>"' >> /etc/rc.conf echo 'nameserver <DNSserver>' >> /etc/rc.conf	Setup manual default gateway and DNS server
service netif restart	Restart networking
service routing restart	Restart default gateway

FreeBSD Additional Features:

ZFS

Ports collection

FreeBSD Startup system

Jails

Linux emulation

DTrace

Network Virtualization

Sources:
https://www.freebsd.org/doc/faq/introduction.html
http://www.freebsdworld.gr/freebsd/bsd-family-tree.html
https://www.freebsd.org/doc/handbook/dirstructure.html
http://www.infoworld.com/article/2858288/unix/intro-to-freebsd-for-linux-users.html
https://www.freebsd.org/doc/en/articles/linux-users/startup.html

Revision History

Date of Change	Summary of Change
January 16, 2017	Initialized document
January 17, 2017	Added FreeBSD Operating System basics

Possible Revisions
Formatting
Additional Frequently Asked Questions
Complete additional features

NetPlan is a new network configuration tool introduced in Ubuntu 17.10 replacing `/etc/network/interfaces`. Configuration is now in `/etc/netplan/01-netcfg.yaml`

DHCP:

```
network:
 version: 2
 renderer: networkd
 ethernets:
   ens33:
     dhcp4: yes
     dhcp6: yes
```

Static:

```
network:
 version: 2
 renderer: networkd
 ethernets:
   ens33:
     dhcp4: no
     dhcp6: no
     addresses: [192.168.1.2/24]
     gateway4: 192.168.1.1
     nameservers:
       addresses: [8.8.8.8,8.8.4.4]
```

To run changes: `sudo netplan apply`

DNS:
Edit `/etc/systemd/resolved.conf`

```
[Resolve]
DNS=192.168.1.152
```

Restart the service: `service systemd-resolved restart`

Last resort:

```
sudo rm -f /etc/resolv.conf
sudo ln -s /run/systemd/resolve/resolv.conf /etc/resolv.conf
reboot
```

Resources:
Set static and DHCP: https://websiteforstudents.com/configuring-static-ips-ubuntu-17-10-servers/
Set DNS with resolv-conf: https://askubuntu.com/questions/973017/confusion-about-dns-settings-in-ubuntu-17-10

Scripts
Check website defacement
Install wordpress, ftp, ssh

```
#!/bin/bash
# Usage: watch -n 300 bash defaced.sh 172.16.49.17
# Dependencies: apt install wdiff curl
# Methodology of checking if website is defaced: 1. Checksum of static web pages, 2. Diff and count number of characters that are different given a threshold, 3. Check for sensitive key-words like [hacked]

# Get base of web page that will be checked and compared to
if [ ! -f “index.html” ]; then
    curl --connect-timeout 5 -o “index.html” “$1”
fi

# Compare current status of website with baseline
threshold=20
time=`date +%F_%R`
curl --connect-timeout 5 -o “index.html.$time” “$1”
echo “=============================================================”
split=($(wdiff -s “index.html” “index.html.$time” | grep “% changed”))
change=${split[10]}
echo “Change: $change”
empt=“”
result=${change//%/$empt}
if [ $result -gt $threshold ]; then
    echo “Diff is gt 20% Site may have been defaced”
else
    echo “Site seems to be fine”
fi
echo “=============================================================”
rm index.html.$time

# Resources: https://www.silverf0x00.com/website-deface-detection-script/
```

```
#!/bin/bash
# Usage [run as root]: bash ftp_wordpress.sh
# Tested on Ubuntu 16, thus package names may differ from other distros / versions
# deb http://security.ubuntu.com/ubuntu xenial-security main restricted
# deb http://security.ubuntu.com/ubuntu xenial-security universe
# deb http://security.ubuntu.com/ubuntu xenial-security multiverse
# Script to set up and [mostly] configure an anonymous ftp, wordpress, and ssh server
# Anywhere there are colors are values that should be changed accordingly, other than links
# Probably better off setting up vagrant / docker
# You will be prompted to set up mysql-server root:P@ssw0rd

# Download all packages and dependencies
apt-get install -y vsftpd apache2 mysql-server php7.0 php7.0-fpm php7.0-mysql
libapache2-mod-php7.0 php-curl php-gd php-mbstring php-mcrypt php-xml php-
xmlrpc openssh-server
wget https://wordpress.org/latest.tar.gz

# Configure vsftpd for anonymous access only
cp /etc/vsftpd.conf /etc/vsftpd.conf.orig
mkdir -p /var/ftp/pub
chown nobody:nogroup /var/ftp/pub
echo "vsftpd test file" | sudo tee /var/ftp/pub/test.txt
sed -i '-e s/anonymous_enable=.*/anonymous_enable=YES/' '-e
s/local_enable=.*/local_enable=NO/' /etc/vsftpd.conf
echo "anon_root=/var/ftp/" >> /etc/vsftpd.conf
echo "no_anon_password=YES" >> /etc/vsftpd.conf
echo "hide_ids=YES" >> /etc/vsftpd.conf
echo "pasv_min_port=40000\npasv_max_port=50000" >> /etc/vsftpd.conf
systemctl restart vsftpd

# Set up mysql database with one-liner to run queries from command line, the lack of space after -p is
not a typo
mysql -u root -pP@ssw0rd -e "CREATE DATABASE wordpress;CREATE USER
'myuser'@'localhost' IDENTIFIED BY 'P@ssw0rd';GRANT ALL ON wordpress.* TO
'myuser'@'localhost' IDENTIFIED BY 'P@ssw0rd';FLUSH PRIVILEGES;"

# Configure wordpress
tar -xzf latest.tar.gz
cp wordpress/wp-config-sample.php wordpress/wp-config.php
cp -r wordpress/* /var/www/html/
mv /var/www/html/index.html /var/www/
chown -R www-data:www-data /var/www/html/
chmod -R 777 /var/www/html/
```

```
sed -i '-e s/database_name_here/wordpress/' '-e s/username_here/myuser/' '-e
s/password_here/P@ssw0rd/' /var/www/html/wp-config.php

# At this point go to web browser and navigate to http://localhost/wp-admin to continue the installation
in the web portal. When finished, press space to continue and finish the rest of the installation
systemctl restart apache2
systemctl reload apache2
echo "============================================================"
echo "============================================================"
echo "Open your web browser and go to http://localhost/wp-admin to continue
the installation. When you are finished there, then"
read -n1 -r -p "Press space to continue..." key

if [ "$key" = '' ]; then
    echo "Continuing"
fi

# Configure wordpress localhost redirect
mysql -u root -pP@ssw0rd <<MY_QUERY
USE wordpress
UPDATE wp_options set option_value='http://172.16.49.16' where
option_name='siteurl';
UPDATE wp_options set option_value='http://172.16.49.16' where
option_name='home';
MY_QUERY




: '
Resources:
https://www.shellhacks.com/mysql-run-query-bash-script-linux-command-line/
http://linux-sys-adm.com/how-to-install-and-configure-ssh-on-ubuntu-server-14.04-lts-step-by-step/
https://unix.stackexchange.com/questions/134437/press-space-to-continue
https://www.digitalocean.com/community/tutorials/how-to-set-up-vsftpd-for-anonymous-downloads-
on-ubuntu-16-04
'
```