*service --status-all* -all programs

*uname -a* -the os version

*ls -a* -list hidden files

*pwd* -print working directory

*rm -r <folder name>* -remove directory

*sudo -I*  -to become root

*sudo -s*  -to become root

*exit* -to exit out of root

*less* -writes content of file to screen (q to quit, space-bar for next page)

*head* -first ten lines of file ( head -5 for just first 5)

*tail* -last ten lines

*grep* -searches file for word or pattern (-I to ignore case sensitivity)

*wc* -word count(-l to get how many lines)

*ps* -list current processes

*kill <pid>* -kill process (kill -9 <pid> if a process refuses to be killed)

*whereis <option>* -bianary search

*which <option>*  -locates path of executable in path (ex which ls)

*whatis <option>* -brief information from man page

*find . <option>* -searches files for attributes given (ex find . -name “\*.txt” -print or find . -size +1M -ls)

*history* -history for command used (!! For last, !-3 for third most recent, !grep for commands starting with grep, or set history=100 to set size of buffer)

*tar xzf* -extract .gz file to a file of same name

*gunzip* -extract .gz file

*tar -xvf* -extract .tar file

*make* -build packages (make check to see if everything compiled successfully)

*make install* -installs package (run from directory with ./)

*wget <link>* -download from link

/etc/init.d/<program> -the executables (can start, stop, restart, reload, status)

----------------------------------------------------------------------------------------------------------------------------------------

change access rights to a file

*chmod* ( u : user, g : group, o : other, a : all, r : read, w : write and delete, x : execute, + : add premession, - : takeaway permissions)

ex: chmod go-rwx biglist (removes read,write, and exectute permission on file for group and others)

----------------------------------------------------------------------------------------------------------------------------------------

Change hostname

/etc/hostname

/etc/host

----------------------------------------------------------------------------------------------------------------------------------------

network config

renew ip:

*sudo dhclient -r*

*sudo dhclient*

for eth0:

*sudo dhclient -r eth0*

*sudo dhclient eth0*

for wireless:

*sudo dhclient -v -r eth0*

*sudo dhclient -v eth0*

or

*# ifdown eth0*

*# ifup eth0*

*### RHEL/CentOS/Fedora specific command ###*

*# /etc/init.d/network restart*

Or

*### Debian / Ubuntu Linux specific command ###*

*# /etc/init.d/networking restart*

/etc/network/interfaces

For static ip address:

*auto enp0s8*

*iface enp0s8 inet static*

*address 10.0.0.41*

*netmask 255.255.255.0*

*network 10.0.0.0*

*broadcast 10.0.0.255*

*gateway 10.0.0.1*

*dns-nameservers 10.0.0.1 8.8.8.8*

*dns-domain acme.com*

*dns-search acme.com*

for dhcp enabled interface:

*auto enp0s3*

*iface enp0s3 inet dhcp*

on the fly:

*Ifconfig <interface name> <ip address> netmask <netmask>*

Ex. Ifconfig enp0s8 10.0.0.1 netmask 255.255.255.0

to reset

*sudo ifdown*

*sudo ifup*

----------------------------------------------------------------------------------------------------------------------------------------

nano /etc/sudoers -checking sudoers file

nano ~/.bash\_history -bash commands that have been run

sudo apt-get --purge remove <program> -removes program, --purge removes configs

sudo visudo -to add user to sudoers (need to add <name> ALL=ALL(ALL:ALL) ALL

sudo userdel -r <name> remove a user from system

useradd [option] <name> -create user account

useradd -e {yyyy-mm-dd} {username} -create account with disable date

useradd -f {days} {username} -set defauly passowrd expiry (0 to diable immediately after the password expires, -1 for account is not be diable after password expires)

----------------------------------------------------------------------------------------------------------------------------------------

Adding a group commands

groupadd <groupname>

usermod -a -G <groupname> username -adding a user to a group

usermod -a -G admins <username> -add a user to group admins

usermod -g <groupname> username -change a users primary group

id <username> -check permissions of a user

groups -list of groups

useradd -g <groupname> username -add a new user and assign a group

Nagios

Sudo /etc/init.d/Nagios restart -restart Nagios

Add to commands config

Add to services config

Add to host config

----------------------------------------------------------------------------------------------------------------------------------------

#!/bin/sh — Execute the file using sh, the Bourne shell, or a compatible shell

#!/bin/csh — Execute the file using csh, the C shell, or a compatible shell

#!/usr/bin/perl -T — Execute using Perl with the option for taint checks

#!/usr/bin/php — Execute the file using the PHP command line interpreter

#!/usr/bin/python -O — Execute using Python with optimizations to code

#!/usr/bin/ruby — Execute using Ruby

Checking permissions of files/users

The suid bit:

find / -perm -4000 -o -perm –2000

Check for users with blank passwords:

awk -F '($2 == "") {print $1}' /etc/shadow

Baselining:

Which processes are running?

Ps aux > out.txt

What users have been created recently?

sort –nk3 -t: /etc/passwd

Who has root? (make sure "ALL" is in caps)

Sudo grep ALL /etc/sudoers

What ports are active?

Netstat –nlp

What services are there?

systemctl | grep "^.\*\.service"

Look at bash history files/conf files:

~/.bash\_history

~/.bashrc

~/.profile

Look at crontab:

/var/spool/cron/crontabs

Find what sudo commands have been run:

journalctl | grep sudo

Look at stuff in /tmp

Check $PATH variable

Reboot

Defeating persist

Disable Public Key Authentication:

Change:

PubkeyAuthentication yes

To:

PubkeyAuthentication yes

In:

/etc/ssh/sshd\_config

Disable ssh:

Systemctl disable ssh

Disable Root ssh login:

Change:

General steps:

Review bash\_history and logs. Do general discovery

Lock down services (simple and fast)

Reboot

Lock down services (time consuming processes)

Record bash\_history

Cat /home/\*/.bash\_history > history.txt

Grep mysql history.txt

Grep apt history.txt

Grep yum history.txt

Files to check:

/etc/passwd

/etc/shadow

/etc/sudoers

~/.bash\_history

~/.bashrc

~/.shrc

~/.profile

/var/spool/crontabs

Make sure /boot/grub/grub.cfg is owned by root:root

Checking permissions of files/users

The suid bit:

find / -perm -4000 -o -perm –2000

Check for users with blank passwords:

awk -F '($2 == "") {print $1}' /etc/shadow

Baselining:

Which processes are running?

Ps aux > out.txt

What users have been created recently?

sort –nk3 -t: /etc/passwd

Who has root? (make sure "ALL" is in caps)

Sudo grep ALL /etc/sudoers

What ports are active?

Netstat –nlp

What services are there?

systemctl | grep "^.\*\.service"

Find what sudo commands have been run:

journalctl | grep sudo

Look at stuff in /tmp

Check $PATH variable

SSH

Disable root login

Disable private key login

Fail2ban (low priority)

Server Specific Instructions:

LAMP Stack (generic):

Remove unused MySQL users

Change all MySQL passwords

Install mod\_security apache plugin (libapache2-modsecurity ubuntu package) use owasp rule set

Change php settings (ex: display\_errors = Off in /etc/php.ini) see http://www.madirish.net/199

Set firewall (see Firewall section)

WordPress:

See LAMP Stack (above)

Check for unnecessary plugins

Check for unnecessary users

FTP Server

Disable anonymous users ( chang eanonymous\_enable=YES to anonymous\_enable=NO in vsftp conf file.

KAA Server

Check /etc/kaa-node/conf/sql-dao.properties for password, username, and provider.

Check /etc/kaa-node/conf/nosql-dao.properties for password, username, and provider.

Disable unused sql and nosql databases.

Check the permissions of those files, since they contain passwords.

Remove excess users

Nagios

-To add services

Add to commands config

Add to services config

Add to host config

Samba/nfs

Disable/stop if not used

Databases:

Check if necessary, and disable/stop if unnecessary

Firewalls:

Web Server (Allows http, https, and ssh traffic)

#!/bin/bash

iptables -P INPUT DROP

iptables -P FORWARD DROP

iptables -P OUTPUT ACCEPT

iptables -A INPUT -i lo -j ACCEPT

iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# allow icmp

Iptables –A INPUT –p icmp –j ACCEPT

# allow ssh traffic

iptables -A INPUT -p tcp --dport ssh -j ACCEPT

# allow http and https traffic

iptables -A INPUT -p tcp --dport http -j ACCEPT

iptables -A INPUT -p tcp --dport https -j ACCEPT

