Linux - Sysadmin Security

# Secure OS

## Disable Default User

If there is a ubunut or pi user on the system, it is common for bruteforce login attempts on the user, since it is not always disabled. Remove it doing the following:

Create new user, and give sudo privilidges:

sudo adduser <username>

sudo adduser <username> sudo

Check ssh login access etc

Delete ubunut/pi user:

sudo deluser -remove-home <username>

## Password on Sudo

Make sudo require password:

sudo visudo /etc/sudoers

add to the end of the file:

<username> ALL=(ALL) PASSWD:ALL

## Disable Root Account

Lock down root user. First change the password to something secure:

sudo passwd root

Then lock root account using:

sudo passwd -l root

## Disable Root SSH Access

Check ssh server config file:

sudo vim /etc/ssh/sshd\_config

Ensure the following configuration to:

PermitRootLogin no

ChallengeResponseAuthentication no

UsePAM no

restart ssh server:

sudo service ssh restart

## Change SSH default port

Check ssh daemon server config file:

sudo vim /etc/ssh/sshd\_config

Change line with new port number, uncomment to change from default:

#Port 22

Port <new-ssh-port-number>

restart ssh server:

sudo service ssh restart

## Use SSH Keys to access

Generate personal ssh key (secure with password if local machine unsecure):

ssh-keygen -b 4096 -t rsa -f <key-name> -C “<comment>”

copy public key to server:

scp -P <ssh-port> ~/.ssh/<key-name>.pub pi@IP:/home/<username>/<key-name>.pub

Log into server via ssh

Add keys to allowed keys in server ssh daemon:

cat /home/<username>/<key-name>.pub >> ~/.ssh/authorized\_keys

Add key to local machine keyring with password (-K mac only):

ssh-add -K <keyname>

Exit and connect again, no password should be required.

Check ssh server config file:

sudo vim /etc/ssh/sshd\_config

Change line to remove password auth:

PasswordAuthentication no

ChallengeResponseAuthentication no

AuthenticationMethods publickey

Restart ssh daemon:

sudo service ssh restart

## Block users who don’t require SSH login

Any users who don't require ssh login, such as limited users who run webservers, should be blocked from loggin in via SSH. The easiest way to SSH login is setting either a 'AllowUsers' or 'DenyUsers' list in the SSH daemon configuration.

Set AllowUsers list in the SSH config:

sudo vim /etc/ssh/sshd\_config

AllowUsers <username1> <username2>

restart ssh server:

sudo service ssh restart

## Unattended Upgrades

Keeping software upto date is critical for system security, therefore add unattended upgrades to the server to keep it upto date automatically.

Install unattended-upgrades:

sudo apt-get install unattended-upgrades

Edit the configuration:

sudo vim /etc/apt/apt.conf.d/50unattended-upgrades

Uncomment the updates line so that it looks like this:

"${distro\_id}:${distro\_codename}-updates";

It is recommended to enable the following by uncommenting, but if the system is critical do not allow automatic rebooting:

Unattended-Upgrade::Remove-Unused-Kernel-Packages "true";

Unattended-Upgrade::Remove-New-Unused-Dependencies "true";

Unattended-Upgrade::Remove-Unused-Dependencies "true";

Unattended-Upgrade::Automatic-Reboot "false";

Unattended-Upgrade::Automatic-Reboot-Time "02:55";

Enable update intervals by adding the following lines of code fo the auto-upgrades file:

sudo vim /etc/apt/apt.conf.d/20auto-upgrades

APT::Periodic::Update-Package-Lists "1";

APT::Periodic::Unattended-Upgrade "1";

APT::Periodic::Download-Upgradeable-Packages "1";

APT::Periodic::AutocleanInterval "7";

Test with dry-run:

sudo unattended-upgrades --dry-run --debug

The server should now update once a day.

Firewall - Install

While having a firewall on the WAN to LAN router is common to stop unauthorized packets entering the network, it is also important to have a firewall on the server itself to control what packets enter the system. Therefore, make sure iptables is install to start creating a firewall in ubuntu:

sudo apt-get install iptables

Then install iptables persistent so that the firewall rules are reloaded on system reboot, which will keep saved rules in /etc/iptables/rules.v4 (.v6):

sudo apt-get install iptables-persistent

Reboot server to update kernel:

sudo reboot

There are 3 different categories of connections in iptables:

* Input - incoming connections, e.g. ssh, should be filtered for only what is required for server to perform
* Forward - incoming connections which are to be forwarded onto another device, i.e. what a router does. Since RPi wont be doing any routing, this can be disabled
* Output - Outgoing connections, i.e. what ports/protocols are packets are allowed to be sent from

There are three different rules for connections:

* Accept - Allow the connection
* Drop - Drop connection and act like it never happened
* Reject - Block connection and send back error, allows user to know that firewall has blocked them

Rules can be viewed using:

sudo iptables -L

Rules can be added using the following command, or they can be loaded in directly:

sudo iptables -A <chain> -i <interface> -p <protocol> -s <source> --dport <port no.> -j <target>

* + -A: chain - input, forward, output
  + -i: interface - eth0, lo, etc
  + -p: protocol - tcp, udp, etc
  + -s: source - hostname or ipaddress
  + --dport: destination port
  + -j: target - ACCEPT, DROP, REJECT

check current rules:

sudo sbin/iptables -L

The rules currently applied to the firewall can be added to the persistent rules:

sudo su

iptables-save > /etc/iptables/rules.v4

Or rules can be saved in a file in the same location as the iptables, then loaded using the iptables-restore command, then saved similarly:

add firewall tables ip4 with file, to current-rules.v4

add firewall tables ip6 with file, to current-rules.v6

save files to persistence:

sudo iptables-restore current-rules.v4

sudo ip6tables-restore current-rules.v6

sudo su

iptables-save > /etc/iptables/rules.v4

ip6tables-save > /etc/iptables/rules.v6

Reboot and then check rules with:

sudo iptables -L -v

sudo ip6tables -L -v

## Firewall - Webserver Setup

A websever would likely require port 80 and 443 open for receiving requests on ipv4, other ports and ipv6 can be blocked. An example set of files might be:

sudo vim /etc/iptables/current-rules.v4

\*filter

:INPUT DROP [0:0]

:FORWARD DROP [0:0]

:OUTPUT ACCEPT [0:0]

# Inbound Rules - Drop unless rule allows

# ======================================

# Allows new SSH connections, local only

-A INPUT -i eth0 -s 192.168.2.0/24 -p tcp -m state --state NEW --dport <ssh-port> -j ACCEPT

# Allows new HTTP(S) connections

-A INPUT -i eth0 -p tcp -m state --state NEW --dport 80 -j ACCEPT

-A INPUT -i eth0 -p tcp -m state --state NEW --dport 443 -j ACCEPT

# Allows new ICMP ping requests

-A INPUT -p icmp --icmp-type 8 -s 0/0 -m state --state NEW -j ACCEPT

# Accepts all established and related inbound connections

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

# Allows all loopback, drop all traffic to 127/8 that doesn't use lo0

-A INPUT -i lo -j ACCEPT

-A INPUT ! -i lo -d 127.0.0.0/8 -j DROP

# Block traffic from firewall/router

-A INPUT -s 192.168.3.1/32 -i tcp -p tcp -m tcp --dport 22 -j DROP

# Forward Rules - Drop all

# ======================================

# Outbound Rules - Allow all

# ======================================

COMMIT

sudo vim /etc/iptables/current-rules.v6

\*filter

:INPUT DROP [0:0]

:FORWARD DROP [0:0]

:OUTPUT DROP [0:0]

COMMIT

Restore the newly created rules using:

sudo iptables-restore /etc/iptables/current-rules.v4

sudo ip6tables-restore /etc/iptables/current-rules.v6

Check rules have been applied using:

sudo iptables -L -v

sudo ip6tables -L -v

Save the rules to persistence by loggin into root and using:

sudo su

iptables-save > /etc/iptables/rules.v4

ip6tables-save > /etc/iptables/rules.v6

Reboot the system, ssh in again, and check whether the rules are still applied:

sudo reboot

ssh <username>@<server-ip-address> -p <ssh-port>

sudo iptables -L -v

sudo ip6tables -L -v

If everything is fine and rules are still applied, open up the ports on router to forward to open ports in the firewall:

* 7 - ping requests
* 80 - http
* 443 - https

Checking Logs

All logs are in the /var/log folder, but the main log files to check are:

* /var/log/syslog: main log file for all services
* /var/log/message: whole systems log file
* /var/log/auth.log: all authentication attempts are logged here
* /var/log/mail.log: if you have a mail server, you’ll find a trace of recent emails sent here
* Any critical application log file, for example /var/log/apache2/error.log or /var/log/mysql/error.log

## Add SSH banner

edit sshd server config file and add a banner to appear on ssh login, this is commonly stored in /etc/issue:

sudo vim /etc/ssh/sshd\_config

Banner <path-to-banner>