Ubuntu Hardening Guide

Keep System up to Date

An extremely crucial part of hardening any system is to ensure that it is always kept up to date. Doing this will keep any known bugs or vulnerabilities patched. The following commands are ways to update an Ubuntu system:

apt-get update && apt-get upgrade

Accounts

A good place to start when dealing with any operating system's security is to ensure that user accounts are locked down.

- **Ensure Only Root Has UID of 0**

Accounts that have a UID set to 0 have the highest access to a system. In most cases, this should only be the root account. Use the below command to list all accounts with a UID of 0:

awk -F: '($3=="0"){print}' /etc/passwd

- **Check for Accounts With Empty Passwords**

Accounts that have no password essentially have no security. The command below will print all accounts that have an empty password:

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

- **Lock Accounts**

In addition, you can use the command below to lock any accounts (i.e., it prepends a ! to the user's password hash):

passwd -l accountName

- **Adding New User Accounts**

It's a best practice to keep use of the root account to a minimum. To do this, add a new account that will be primarily used with the command below:

adduser accountName

This will automatically create a user with the default configuration defined in '/etc/skel'.

- **Sudo Configuration**

The sudo package allows a regular user to run commands in an elevated context. This means a regular user can run commands normally restricted to the root account. Often, this is the ideal way of making system configurations or running elevated commands – not

by using the root account. The configuration file for sudo is in /etc/sudoers. However, it can only be edited by using the “visudo” command. There are many different configuration options that limit the use of sudo to certain users, groups, IPs, and commands. The general configuration format is below:

%www ALL = (ALL)NOPASSWD:/bin/cat,/bin/ls

%www = All users of the www group.

ALL = From any host/IP.

(ALL) = Can run as any user.

NOPASSWD = No password required (i.e., omit to require a password).

:/bin/cat,/bin/ls = Commands able to run as sudo. In this case, “cat” and “ls”.

To run any elevated command, simply place “sudo” in front of it as a properly configured user.

Iptables

Iptables is essentially your operating system’s firewall. Iptables is extremely powerful in controlling the network traffic going into and out of your server. While basic example configurations are listed below, it’s recommended that anyone looking into hardening their Ubuntu OS do research into Iptables implementation.

Running the commands below will configure your box to allow inbound connections only on ports 80, 22, and the loopback interface, and drop all other packets (i.e., configure to your own server’s needs):

iptables -A INPUT -p tcp -m tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A INPUT -p tcp -m tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A INPUT -I lo -j ACCEPT

iptables -A INPUT -j DROP

(Or use iptables -P INPUT DROP to automatically drop all packets without a rule)

SSH

All services hosted on your server should be adequately configured and locked down. Since SSH is almost always going to be running on your server, it’s essential to lock it down as much as possible. The SSH service configuration file can be found at ‘/etc/ssh/sshd_config’.

- **Disable Root Login**

This configuration will limit SSH to users other than root. Find and ensure the line for "PermitRootLogin" exists and looks like the one below:

PermitRootLogin no

- **Allow Specific Users**

This line will allow you to specify which users can log into the SSH service:

AllowUsers accountName

- **Change Default Port From 22**

This line will specify which port to host the SSH service on. It's recommended to change this to a non-default high port number. Remember to fix your Iptables accordingly!

Port 22222

- **Disable Empty Passwords**

This line ensures that no user can log in with an empty password. This adds a nice layer of security if there is a user without a password set:

PermitEmptyPasswords no

- **Restart Service**

As always, be sure to restart after making changes to a service.
service ssh restart

Display All Current Connections, Listening Services, and Processes

The below command can be an Ubuntu sysadmin's best friend. It will list all current connections and listening services on a system along with the processes and PIDs for each connection:

netstat -tulpn

Display Services and Status

The command below will list all services on the system and their status:

service --status-all

Use grep to specify the running services only:

service --status-all | grep "[+]"

Check for Rootkits

The package “rkhunter” is useful for doing a quick scan of your system for any known rootkits:

apt-get install rkhunter

rkhunter -C

Common Configuration File Locations

Below are configuration file locations for a few common services:

/etc/apache/apache2.conf #Apache 2

/etc/ssh/sshd_config #SSH Server

/etc/mysql/mysql.cnf #MySQL

/var/lib/mysql/ #This entire directory contains all of the database in MySQL

Log Locations

Below are the common default log locations:

/var/log/message — Where whole system logs or current activity logs are available.

/var/log/auth.log — Authentication logs.

/var/log/kern.log — Kernel logs.

/var/log/cron.log — Crond logs (cron job).

/var/log/maillog — Mail server logs.

/var/log/boot.log — System boot log.

/var/log/mysqld.log — MySQL database server log file.

/var/log/secure — Authentication log.

/var/log/utmp or /var/log/wtmp — Login records file.

/var/log/apt — Apt package manager logs.