**CentOS 7 Server Hardening Guide**

Posted on [17/09/2017](https://www.lisenet.com/2017/centos-7-server-hardening-guide/) by [Tomas](https://www.lisenet.com/author/admin/)

This guide is based on a minimal CentOS 7 install following the idea that you only install software that you require.

For those familiar with [OpenSCAP](https://www.open-scap.org/), you will notice the guide divided into two major sections: System Settings and Services. The first part contains rules that check system settings, where the second part is aimed towards hardening services.

General disclaimer applies: do not implement changes to production systems unless you understand what they do.

**1. System Settings – Disk Partitioning and Post installation**

**1.1 Disk Encryption with Kickstart**

The easiest way to encrypt a partition is during Kickstart installation.

This can be achieved by adding the **–encrypted** and **–passphrase=** options to the definition of a physical LVM volume.

Our Kickstart template is provided below. Note that the template requires a 32GB disk.

***#version=CentOS7.5***

***# System authorisation information***

auth --enableshadow --passalgo=sha512

***# Use CDROM installation media***

cdrom

ignoredisk --only-use=sda

***# Keyboard layouts***

keyboard --vckeymap=gb --xlayouts='gb'

***# System language***

lang en\_GB.UTF-8

***# SELinux***

selinux --enforcing

***# Network information***

network --bootproto=dhcp --device=eth0 --onboot=on --activate

network --hostname=ks-c7.example.com

***# Plaintext root password: PleaseChangeMe***

rootpw --iscrypted $6$nS0mBJyS$q/QgCof5unWrT9W3qngTISueSDhDHVNntDqd8sOcgmHp2lq4f/niUbjCmoEzaf3EWQ2x3z/k0eIZaOXkfNtJw/

***# System timezone***

timezone Europe/London --isUtc

***# System bootloader configuration***

bootloader --location=mbr --boot-drive=sda --timeout=3

***# Partition clearing information***

clearpart --all --drives=sda

zerombr

***# Disk partitioning information***

***#***

***# Your disk should be 32GB: 1GB for /boot and 30GB for the physical volume***

***#***

***# Journal for boot is not required therefore ext2***

part /boot --fstype="ext2" --ondisk=sda --size=1024 --label=boot --mkfsoptions="-m 0" --fsoptions="rw,nodev,noexec,nosuid"

***# The line below will create a 30GB physical volume***

part pv.01 --fstype="lvmpv" --ondisk=sda --size=30720 --encrypted --passphrase=PleaseChangeMeToSomethingElse

volgroup vg\_crypto pv.01

***# Need the ability to shrink filesystems therefore ext4 over default xfs***

logvol / --fstype="ext4" --size=6144 --vgname=vg\_crypto --name=lv\_root --mkfsoptions="-m 1"

logvol /home --fstype="ext4" --size=2048 --vgname=vg\_crypto --name=lv\_home --mkfsoptions="-m 0" --fsoptions="rw,nodev,nosuid"

logvol /tmp --fstype="ext4" --size=1024 --vgname=vg\_crypto --name=lv\_tmp --mkfsoptions="-m 1" --fsoptions="rw,nodev,noexec,nosuid"

logvol /var --fstype="ext4" --size=4096 --vgname=vg\_crypto --name=lv\_var --mkfsoptions="-m 1" --fsoptions="rw,nosuid"

logvol /var/log --fstype="ext4" --size=1024 --vgname=vg\_crypto --name=lv\_var-log --mkfsoptions="-m 0" --fsoptions="rw,nodev,noexec,nosuid"

logvol /var/log/audit --fstype="ext4" --size=512 --vgname=vg\_crypto --name=lv\_var-aud --mkfsoptions="-m 0" --fsoptions="rw,nodev,noexec,nosuid"

logvol /var/tmp --fstype="ext4" --size=1024 --vgname=vg\_crypto --name=lv\_var-tmp --mkfsoptions="-m 1" --fsoptions="rw,nodev,noexec,nosuid"

logvol /var/www --fstype="ext4" --size=1024 --vgname=vg\_crypto --name=lv\_var-www --mkfsoptions="-m 0" --fsoptions="rw,nodev,nosuid"

logvol swap --fstype="swap" --size=512 --vgname=vg\_crypto --name=lv\_swap --fsoptions="swap"

%packages

@core

%end

**1.2 Partition Scheme**

Keep the following partitions separate: /boot, /home, /tmp, /var, /var/log, /var/tmp, /var/log/audit, /var/www.

Placing these in their own partitions gives more control over mount options. It also ensures that the system cannot be halted because of some partition running out of disk space.

Splitting off /opt depends on a setup and is generally not useful, but not harmful either.

**1.3 Post installation**

Backup a LUKS header, where /dev/sda2 is the LUKS encrypted partition:

# cryptsetup luksHeaderBackup /dev/sda2 --header-backup-file /root/luks-header.backup

Ensure that the backup file is stored off-site and then removed from the server.

Make sure that the system is up to date:

# yum update

Remove packages which you don’t require on a server, e.g. firmware of sound cards, firmware of WinTV, wireless drivers etc.

# yum remove alsa-\* ivtv-\* iwl\*firmware aic94xx-firmware

**2. System Settings – File Permissions and Masks**

**2.1 Restrict Partition Mount Options**

Partitions should have hardened mount options:

* /boot – rw,nodev,noexec,nosuid
* /home – rw,nodev,nosuid
* /tmp – rw,nodev,noexec,nosuid
* /var – rw,nosuid
* /var/log – rw,nodev,noexec,nosuid
* /var/log/audit – rw,nodev,noexec,nosuid
* /var/www – rw,nodev,nosuid

As a rule of thumb, malicious applications usually write to /tmp and then attempt to run whatever was written. A way to prevent this is to mount /tmp on a separate partition with the options **noexec**, **nodev** and **nosuid** enabled.

This will deny binary execution from /tmp, disable any binary to be suid root, and disable any block devices from being created.

The storage location /var/tmp should be bind mounted to /tmp, as having multiple locations for temporary storage is not required:

/tmp /var/tmp none rw,nodev,noexec,nosuid,bind 0 0

The same applies to shared memory /dev/shm:

tmpfs /dev/shm tmpfs rw,nodev,noexec,nosuid 0 0

The proc pseudo-filesystem /proc should be mounted with **hidepid**. When setting hidepid to 2, directories entries in /proc will hidden.

proc /proc proc rw,hidepid=2 0 0

Harden removeable media mounts by adding **nodev**,**noexec**and**nosuid**, e.g.:

/dev/cdrom /mnt/cdrom iso9660 ro,noexec,nosuid,nodev,noauto 0 0

**2.2 Restrict Dynamic Mounting and Unmounting of Filesystems**

Add the following to /etc/modprobe.d/hardening.conf to disable uncommon filesystems:

install cramfs /bin/true

install freevxfs /bin/true

install jffs2 /bin/true

install hfs /bin/true

install hfsplus /bin/true

install squashfs /bin/true

install udf /bin/true

Depending on a setup (if you don’t run clusters, NFS, CIFS etc), you may consider disabling the following too:

install fat /bin/true

install vfat /bin/true

install cifs /bin/true

install nfs /bin/true

install nfsv3 /bin/true

install nfsv4 /bin/true

install gfs2 /bin/true

It is wise to leave ext4, xfs and btrfs enabled at all times.

**2.3 Prevent Users Mounting USB Storage**

Add the following to /etc/modprobe.d/hardening.conf to disable modprobe loading of USB and FireWire storage drivers:

blacklist usb-storage

blacklist firewire-core

install usb-storage /bin/true

Disable USB authorisation. Create a file /opt/usb-auth.sh with the following content:

#!/bin/bash

echo 0 > /sys/bus/usb/devices/usb1/authorized

echo 0 > /sys/bus/usb/devices/usb1/authorized\_default

If more than one USB device is available, then add them all. Create a service file /etc/systemd/system/usb-auth.service with the following content:

[Unit]

Description=Disable USB auth

DefaultDependencies=no

[Service]

Type=oneshot

ExecStart=/bin/bash /opt/usb-auth.sh

[Install]

WantedBy=multi-user.target

Set permissions, enable and start the service:

# chmod 0700 /opt/usb-auth.sh

# systemctl enable usb-auth.service

# systemctl start usb-auth.service

If required, disable kernel support for USB via bootloader configuration. To do so, append **nousb** to the kernel line GRUB\_CMDLINE\_LINUX in /etc/default/grub and generate the Grub2 configuration file:

# grub2-mkconfig -o /boot/grub2/grub.cfg

Note that disabling all kernel support for USB will likely cause problems for systems with USB-based keyboards etc.

**2.4 Restrict Programs from Dangerous Execution Patterns**

Configure /etc/sysctl.conf with the following:

***# Disable core dumps***

fs.suid\_dumpable = 0

***# Disable System Request debugging functionality***

kernel.sysrq = 0

***# Restrict access to kernel logs***

kernel.dmesg\_restrict = 1

***# Enable ExecShield protection - not available on CentOS 7***

# kernel.exec-shield = 1

***# Randomise memory space***

kernel.randomize\_va\_space = 2

***# Hide kernel pointers***

kernel.kptr\_restrict = 2

Load sysctl settings:

# sysctp -p

**2.5 Set UMASK 027**

The following files require umask hardening: /etc/bashrc, /etc/csh.cshrc, /etc/init.d/functions and /etc/profile.

Sed one-liner:

# sed -i -e 's/umask 022/umask 027/g' -e 's/umask 002/umask 027/g' /etc/bashrc

# sed -i -e 's/umask 022/umask 027/g' -e 's/umask 002/umask 027/g' /etc/csh.cshrc

# sed -i -e 's/umask 022/umask 027/g' -e 's/umask 002/umask 027/g' /etc/profile

# sed -i -e 's/umask 022/umask 027/g' -e 's/umask 002/umask 027/g' /etc/init.d/functions

**2.6 Disable Core Dumps**

Open /etc/security/limits.conf and set the following:

\* hard core 0

**2.7 Set Security Limits to Prevent DoS**

Add the following to /etc/security/limits.conf to enforce sensible security limits:

***# 4096 is a good starting point***

\* soft nofile 4096

\* hard nofile 65536

\* soft nproc 4096

\* hard nproc 4096

\* soft locks 4096

\* hard locks 4096

\* soft stack 10240

\* hard stack 32768

\* soft memlock 64

\* hard memlock 64

\* hard maxlogins 10

***# Soft limit 32GB, hard 64GB***

\* soft fsize 33554432

\* hard fsize 67108864

***# Limits for root***

root soft nofile 4096

root hard nofile 65536

root soft nproc 4096

root hard nproc 4096

root soft stack 10240

root hard stack 32768

root soft fsize 33554432

**2.8 Verify Permissions of Files**

Ensure that all files are owned by a user:

# find / -ignore\_readdir\_race -nouser -print -exec chown root {} \;

Ensure that all files are owned by a group:

# find / -ignore\_readdir\_race -nogroup -print -exec chgrp root {} \;

If required, a specific path can be excluded from the search, e.g.:

# find / -ignore\_readdir\_race -not -path "/proc/\*" -nouser -print -exec chown root {} \;

Automate the process by creating a cron file /etc/cron.daily/unowned\_files with the following content:

#!/bin/bash

find / -ignore\_readdir\_race \( -nouser -print -exec chown root {} \; \) , \( -nogroup -print -exec chgrp root {} \; \)

Set ownership and permissions:

# chown root:root /etc/cron.daily/unowned\_files

# chmod 0700 /etc/cron.daily/unowned\_files

**2.9 Monitor SUID/GUID Files**

Search for setuid/setgid files and identify if all are required:

# find / -xdev -type f -perm -4000 -o -perm -2000

**3. System Settings – Firewall and Network Configuration**

**3.1 Firewall**

Setting the default firewalld zone to drop makes any packets which are not explicitly permitted to be rejected.

# sed -i "s/DefaultZone=.\*/DefaultZone=drop/g" /etc/firewalld/firewalld.conf

Unless firewalld is required, mask it and replace with iptables:

# systemctl stop firewalld.service

# systemctl mask firewalld.service

# systemctl daemon-reload

# yum install iptables-services

# systemctl enable iptables.service ip6tables.service

Add the following to /etc/sysconfig/iptables to allow only minimal outgoing traffic (DNS, NTP, HTTP/S and SMTPS):

\*filter

-F INPUT

-F OUTPUT

-F FORWARD

-P INPUT ACCEPT

-P FORWARD DROP

-P OUTPUT ACCEPT

-A INPUT -i lo -m comment --comment local -j ACCEPT

-A INPUT -d 127.0.0.0/8 ! -i lo -j REJECT --reject-with icmp-port-unreachable

-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

-A INPUT -p tcp -m tcp -m conntrack --ctstate NEW --dport 22 -s 10.0.0.0/8 -j ACCEPT

-A INPUT -p tcp -m tcp -m conntrack --ctstate NEW --dport 22 -s 172.16.0.0/12 -j ACCEPT

-A INPUT -p tcp -m tcp -m conntrack --ctstate NEW --dport 22 -s 192.168.0.0/16 -j ACCEPT

-A INPUT -p tcp -m tcp -m conntrack --ctstate NEW --dport 22 -j ACCEPT

-A INPUT -j DROP

-A OUTPUT -d 127.0.0.0/8 -o lo -m comment --comment local -j ACCEPT

-A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

-A OUTPUT -p icmp -m icmp --icmp-type any -j ACCEPT

-A OUTPUT -p udp -m udp -m conntrack --ctstate NEW --dport 53 -j ACCEPT

-A OUTPUT -p tcp -m tcp -m conntrack --ctstate NEW --dport 53 -j ACCEPT

-A OUTPUT -p udp -m udp -m conntrack --ctstate NEW --dport 123 -j ACCEPT

-A OUTPUT -p tcp -m tcp -m conntrack --ctstate NEW --dport 80 -j ACCEPT

-A OUTPUT -p tcp -m tcp -m conntrack --ctstate NEW --dport 443 -j ACCEPT

-A OUTPUT -p tcp -m tcp -m conntrack --ctstate NEW --dport 587 -j ACCEPT

-A OUTPUT -j LOG --log-prefix "iptables\_output "

-A OUTPUT -j REJECT --reject-with icmp-port-unreachable

COMMIT

Note that the rule allowing all incoming SSH traffic should be removed restricting access to an IP whitelist only, or hiding SSH behind a VPN.

Ideally, outgoing rules should be hardened by restricting access to local DNS, NTP and SMTP servers only. If a local patching system is used (e.g. Red Hat Satellite), then HTTP/S traffic can also be further hardened, depending on a set up.

Add the following to /etc/sysconfig/ip6tables to deny all IPv6:

\*filter

-F INPUT

-F OUTPUT

-F FORWARD

-P INPUT DROP

-P FORWARD DROP

-P OUTPUT DROP

COMMIT

Apply configurations:

# iptables-restore < /etc/sysconfig/iptables

# ip6tables-restore < /etc/sysconfig/ip6tables

**3.2 TCP Wrappers**

Open /etc/hosts.allow and allow localhost traffic and SSH:

ALL: 127.0.0.1

sshd: ALL

The file /etc/hosts.deny should be configured to deny all by default:

ALL: ALL

**3.3 Kernel Parameters Which Affect Networking**

Open /etc/sysctl.conf and add the following:

***# Disable packet forwarding***

net.ipv4.ip\_forward = 0

***# Disable redirects, not a router***

net.ipv4.conf.all.accept\_redirects = 0

net.ipv4.conf.default.accept\_redirects = 0

net.ipv4.conf.all.send\_redirects = 0

net.ipv4.conf.default.send\_redirects = 0

net.ipv4.conf.all.secure\_redirects = 0

net.ipv4.conf.default.secure\_redirects = 0

net.ipv6.conf.all.accept\_redirects = 0

net.ipv6.conf.default.accept\_redirects = 0

***# Disable source routing***

net.ipv4.conf.all.accept\_source\_route = 0

net.ipv4.conf.default.accept\_source\_route = 0

net.ipv6.conf.all.accept\_source\_route = 0

***# Enable source validation by reversed path***

net.ipv4.conf.all.rp\_filter = 1

net.ipv4.conf.default.rp\_filter = 1

***# Log packets with impossible addresses to kernel log***

net.ipv4.conf.all.log\_martians = 1

net.ipv4.conf.default.log\_martians = 1

***# Disable ICMP broadcasts***

net.ipv4.icmp\_echo\_ignore\_broadcasts = 1

***# Ignore bogus ICMP errors***

net.ipv4.icmp\_ignore\_bogus\_error\_responses = 1

***# Against SYN flood attacks***

net.ipv4.tcp\_syncookies = 1

***# Turning off timestamps could improve security but degrade performance.***

***# TCP timestamps are used to improve performance as well as protect against***

***# late packets messing up your data flow. A side effect of this feature is***

***# that the uptime of the host can sometimes be computed.***

***# If you disable TCP timestamps, you should expect worse performance***

***# and less reliable connections.***

net.ipv4.tcp\_timestamps = 1

***# Disable IPv6 unless required***

net.ipv6.conf.lo.disable\_ipv6 = 1

net.ipv6.conf.all.disable\_ipv6 = 1

net.ipv6.conf.default.disable\_ipv6 = 1

***# Do not accept router advertisements***

net.ipv6.conf.all.accept\_ra = 0

net.ipv6.conf.default.accept\_ra = 0

**3.4 Kernel Modules Which Affect Networking**

Open /etc/modprobe.d/hardening.conf and disable Bluetooth kernel modules:

install bnep /bin/true

install bluetooth /bin/true

install btusb /bin/true

install net-pf-31 /bin/true

Also disable AppleTalk:

install appletalk /bin/true

Unless required, disable support for IPv6:

options ipv6 disable=1

Disable (uncommon) protocols:

install dccp /bin/true

install sctp /bin/true

install rds /bin/true

install tipc /bin/true

Since we’re looking at server security, wireless shouldn’t be an issue, therefore we can disable all the wireless drivers.

# for i in $(find /lib/modules/$(uname -r)/kernel/drivers/net/wireless -name "\*.ko" -type f);do \

echo blacklist "$i" >>/etc/modprobe.d/hardening-wireless.conf;done

**3.5 Disable Radios**

Disable radios (wifi and wwan):

# nmcli radio all off

**3.6 Disable Zeroconf Networking**

Open /etc/sysconfig/network and add the following:

NOZEROCONF=yes

**3.7 Disable Interface Usage of IPv6**

Open /etc/sysconfig/network and add the following:

NETWORKING\_IPV6=no

IPV6INIT=no

**3.8 Network Sniffer**

The server should not be acting as a network sniffer and capturing packages. Run the following to determine if any interface is running in promiscuous mode:

# ip link | grep PROMISC

**3.9 Secure VPN Connection**

Install the libreswan package if implementation of IPsec and IKE is required.

# yum install libreswan

**3.10 Disable DHCP Client**

Manual assignment of IP addresses provides a greater degree of management.

For each network interface that is available on the server, open a corresponding file /etc/sysconfig/network-scripts/ifcfg-*interface* and configure the following parameters:

BOOTPROTO=none

IPADDR=

NETMASK=

GATEWAY=

**4. System Settings – SELinux**

Ensure that SELinux is not disabled in /etc/default/grub, and verify that the state is enforcing:

# sestatus

**5. System Settings – Account and Access Control**

**5.1 Delete Unused Accounts and Groups**

Remove any account which is not required, e.g.:

# userdel -r adm

# userdel -r ftp

# userdel -r games

# userdel -r lp

Remove any group which is not required, e.g.:

# groupdel games

**5.2 Disable Direct root Login**

# echo > /etc/securetty

**5.3 Enable Secure (high quality) Password Policy**

Note that running authconfig will **overwrite** the PAM configuration files destroying any manually made changes. Make sure that you have a **backup**.

Secure password policy rules are outlined below.

* Minimum length of a password – 16.
* Minimum number of character classes in a password – 4.
* Maximum number of same consecutive characters in a password – 2.
* Maximum number of consecutive characters of same class in a password – 2.
* Require at least one lowercase and one uppercase characters in a password.
* Require at least one digit in a password.
* Require at least one other character in a password.

The following command will enable SHA512 as well as set the above password requirements:

# authconfig --passalgo=sha512 \

--passminlen=16 \

--passminclass=4 \

--passmaxrepeat=2 \

--passmaxclassrepeat=2 \

--enablereqlower \

--enablerequpper \

--enablereqdigit \

--enablereqother \

--update

Open /etc/security/pwquality.conf and add the following:

difok = 8

gecoscheck = 1

These will ensure that 8 characters in the new password must not be present in the old password, and will check for the words from the passwd entry GECOS string of the user.

**5.4 Prevent Log In to Accounts With Empty Password**

Remove any instances of **nullok** from /etc/pam.d/system-auth and /etc/pam.d/password-auth to prevent logins with empty passwords.

Sed one-liner:

# sed -i 's/\<nullok\>//g' /etc/pam.d/system-auth /etc/pam.d/system-auth-ac

# sed -i 's/\<nullok\>//g' /etc/pam.d/password-auth /etc/pam.d/password-auth-ac

**5.5 Set Account Expiration Following Inactivity**

Disable accounts as soon as the password has expired.

Open /etc/default/useradd and set the following:

INACTIVE=0

Sed one-liner:

# sed -i 's/^INACTIVE.\*/INACTIVE=0/' /etc/default/useradd

**5.6 Secure Pasword Policy**

Open /etc/login.defs and set the following:

PASS\_MAX\_DAYS 60

PASS\_MIN\_DAYS 1

PASS\_MIN\_LEN 14

PASS\_WARN\_AGE 14

Sed one-liner:

# sed -i -e 's/^PASS\_MAX\_DAYS.\*/PASS\_MAX\_DAYS 60/' \

-e 's/^PASS\_MIN\_DAYS.\*/PASS\_MIN\_DAYS 1/' \

-e 's/^PASS\_MIN\_LEN.\*/PASS\_MIN\_LEN 14/' \

-e 's/^PASS\_WARN\_AGE.\*/PASS\_WARN\_AGE 14/' /etc/login.defs

**5.7 Log Failed Login Attemps**

Open /etc/login.defs and enable logging:

FAILLOG\_ENAB yes

Also add a delay in seconds before being allowed another attempt after a login failure:

FAIL\_DELAY 4

**5.8 Ensure Home Directories are Created for New Users**

Open /etc/login.defs and configure:

CREATE\_HOME yes

**5.9 Verify All Account Password Hashes are Shadowed**

The command below should return “x”:

# cut -d: -f2 /etc/passwd|uniq

**5.10 Set Deny and Lockout Time for Failed Password Attempts**

Add the following line immediately before the **pam\_unix.so** statement in the **AUTH** section of /etc/pam.d/system-auth and /etc/pam.d/password-auth:

auth required pam\_faillock.so preauth silent deny=3 unlock\_time=900 fail\_interval=900

Add the following line immediately after the **pam\_unix.so** statement in the **AUTH** section of /etc/pam.d/system-auth and /etc/pam.d/password-auth:

auth [default=die] pam\_faillock.so authfail deny=3 unlock\_time=900 fail\_interval=900

Add the following line immediately before the**pam\_unix.so** statement in the **ACCOUNT** section of /etc/pam.d/system-auth and /etc/pam.d/password-auth:

account required pam\_faillock.so

The content of the file /etc/pam.d/system-auth can be seen below.

***#%PAM-1.0***

auth required pam\_env.so

auth required pam\_faillock.so preauth silent deny=3 unlock\_time=900 fail\_interval=900

auth sufficient pam\_unix.so try\_first\_pass

auth [default=die] pam\_faillock.so authfail deny=3 unlock\_time=900 fail\_interval=900

auth requisite pam\_succeed\_if.so uid >= 1000 quiet\_success

auth required pam\_deny.so

account required pam\_faillock.so

account required pam\_unix.so

account sufficient pam\_localuser.so

account sufficient pam\_succeed\_if.so uid < 1000 quiet

account required pam\_permit.so

password requisite pam\_pwquality.so try\_first\_pass local\_users\_only retry=3 authtok\_type=

password sufficient pam\_unix.so sha512 shadow try\_first\_pass use\_authtok remember=5

password required pam\_deny.so

session optional pam\_keyinit.so revoke

session required pam\_limits.so

-session optional pam\_systemd.so

session [success=1 default=ignore] pam\_succeed\_if.so service in crond quiet use\_uid

session required pam\_unix.so

Also, do not allow users to reuse recent passwords by adding the **remember** option.

Make /etc/pam.d/system-auth and /etc/pam.d/password-auth configurations immutable so that they don’t get overwritten when authconfig is run:

# chattr +i /etc/pam.d/system-auth /etc/pam.d/password-auth

Accounts will get locked after 3 failed login attemtps:

*login[]: pam\_faillock(login:auth): Consecutive login failures for user tomas account temporarily locked*

Use the following to clear user’s fail count:

# faillock --user tomas --reset

**5.11 Set Boot Loader Password**

Prevent users from entering the grub command line and edit menu entries:

# grub2-setpassword

# grub2-mkconfig -o /boot/grub2/grub.cfg

This will create the file /boot/grub2/user.cfg if one is not already present, which will contain the hashed Grub2 bootloader password.

Verify permissions of /boot/grub2/grub.cfg:

# chmod 0600 /boot/grub2/grub.cfg

**5.12 Password-protect Single User Mode**

CentOS 7 single user mode is password protected by the root password by default as part of the design of Grub2 and systemd.

**5.13 Ensure Users Re-Authenticate for Privilege Escalation**

The NOPASSWD tag allows a user to execute commands using sudo without having to provide a password. While this may sometimes be useful it is also dangerious.

Ensure that the NOPASSWD tag does not exist in /etc/sudoers configuration file or /etc/sudoers.d/.

**5.14 Multiple Console Screens and Console Locking**

Install the screen package to be able to emulate multiple console windows:

# yum install screen

Install the vlock package to enable console screen locking:

# yum install vlock

**5.15 Disable Ctrl-Alt-Del Reboot Activation**

Prevent a locally logged-in console user from rebooting the system when Ctrl-Alt-Del is pressed:

# systemctl mask ctrl-alt-del.target

**5.16 Warning Banners for System Access**

Add the following line to the files /etc/issue and /etc/issue.net:

Unauthorised access prohibited. Logs are recorded and monitored.

**5.17 Set Interactive Session Timeout**

Open /etc/profile and set:

readonly TMOUT=900

**5.18 Two Factor Authentication**

The recent version of OpenSSH server allows to chain several authentication methods, meaning that all of them have to be satisfied in order for a user to log in successfully.

Adding the following line to /etc/ssh/sshd\_config would require a user to authenticate with a key first, and then also provide a password.

AuthenticationMethods publickey,password

This is by definition a two factor authentication: the key file is something that a user has, and the account password is something that a user knows.

Alternatively, two factor authentication for SSH can be set up by using Google Authenticator.

**5.19 Configure History File Size**

Open /etc/profile and set the number of commands to remember in the command history to 5000:

HISTSIZE=5000

Sed one-liner:

# sed -i 's/HISTSIZE=.\*/HISTSIZE=5000/g' /etc/profile

**6. System Settings – System Accounting with auditd**

**6.1 Auditd Configuration**

Open /etc/audit/auditd.conf and configure the following:

local\_events = yes

write\_logs = yes

log\_file = /var/log/audit/audit.log

max\_log\_file = 25

num\_logs = 10

max\_log\_file\_action = rotate

space\_left = 30

space\_left\_action = email

admin\_space\_left = 10

admin\_space\_left\_action = email

disk\_full\_action = suspend

disk\_error\_action = suspend

action\_mail\_acct = root@example.com

flush = data

The above auditd configuration should never use more than 250MB of disk space (10x25MB=250MB) on /var/log/audit.

Set**admin\_space\_left\_action=single** if you want to cause the system to switch to single user mode for corrective action rather than send an email.

Automatically rotating logs (**max\_log\_file\_action=rotate**) minimises the chances of the system unexpectedly running out of disk space by being filled up with log data.

We need to ensure that audit event data is fully synchronised (**flush=data**) with the log files on the disk .

**6.2 Auditd Rules**

System audit rules must have mode 0640 or less permissive and owned by the root user:

# chown root:root /etc/audit/rules.d/audit.rules

# chmod 0640 /etc/audit/rules.d/audit.rules

Open /etc/audit/rules.d/audit.rules and add the following:

***# Delete all currently loaded rules***

-D

***# Set kernel buffer size***

-b 8192

***# Set the action that is performed when a critical error is detected.***

***# Failure modes: 0=silent 1=printk 2=panic***

-f 1

***# Record attempts to alter the localtime file***

-w /etc/localtime -p wa -k audit\_time\_rules

***# Record events that modify user/group information***

-w /etc/group -p wa -k audit\_rules\_usergroup\_modification

-w /etc/passwd -p wa -k audit\_rules\_usergroup\_modification

-w /etc/gshadow -p wa -k audit\_rules\_usergroup\_modification

-w /etc/shadow -p wa -k audit\_rules\_usergroup\_modification

-w /etc/security/opasswd -p wa -k audit\_rules\_usergroup\_modification

***# Record events that modify the system's network environment***

-w /etc/issue.net -p wa -k audit\_rules\_networkconfig\_modification

-w /etc/issue -p wa -k audit\_rules\_networkconfig\_modification

-w /etc/hosts -p wa -k audit\_rules\_networkconfig\_modification

-w /etc/sysconfig/network -p wa -k audit\_rules\_networkconfig\_modification

-a always,exit -F arch=b32 -S sethostname -S setdomainname -k audit\_rules\_networkconfig\_modification

-a always,exit -F arch=b64 -S sethostname -S setdomainname -k audit\_rules\_networkconfig\_modification

***# Record events that modify the system's mandatory access controls***

-w /etc/selinux/ -p wa -k MAC-policy

***# Record attempts to alter logon and logout events***

-w /var/log/tallylog -p wa -k logins

-w /var/log/lastlog -p wa -k logins

-w /var/run/faillock/ -p wa -k logins

***# Record attempts to alter process and session initiation information***

-w /var/log/btmp -p wa -k session

-w /var/log/wtmp -p wa -k session

-w /var/run/utmp -p wa -k session

***# Ensure auditd collects information on kernel module loading and unloading***

-w /usr/sbin/insmod -p x -k modules

-w /usr/sbin/modprobe -p x -k modules

-w /usr/sbin/rmmod -p x -k modules

-a always,exit -F arch=b64 -S init\_module -S delete\_module -k modules

***# Ensure auditd collects system administrator actions***

-w /etc/sudoers -p wa -k actions

-w /etc/sudoers.d/ -p wa -k actions

***# Record attempts to alter time through adjtimex***

-a always,exit -F arch=b32 -S adjtimex -S settimeofday -S stime -k audit\_time\_rules

***# Record attempts to alter time through settimeofday***

-a always,exit -F arch=b64 -S adjtimex -S settimeofday -k audit\_time\_rules

***# Record attempts to alter time through clock\_settime***

-a always,exit -F arch=b32 -S clock\_settime -F a0=0x0 -k time-change

***# Record attempts to alter time through clock\_settime***

-a always,exit -F arch=b64 -S clock\_settime -F a0=0x0 -k time-change

***# Record events that modify the system's discretionary access controls***

-a always,exit -F arch=b32 -S chmod -S fchmod -S fchmodat -F auid>=1000 -F auid!=4294967295 -k perm\_mod

-a always,exit -F arch=b32 -S chown -S fchown -S fchownat -S lchown -F auid>=1000 -F auid!=4294967295 -k perm\_mod

-a always,exit -F arch=b64 -S chmod -S fchmod -S fchmodat -F auid>=1000 -F auid!=4294967295 -k perm\_mod

-a always,exit -F arch=b64 -S chown -S fchown -S fchownat -S lchown -F auid>=1000 -F auid!=4294967295 -k perm\_mod

-a always,exit -F arch=b32 -S setxattr -S lsetxattr -S fsetxattr -S removexattr -S lremovexattr -S fremovexattr -F auid>=1000 -F auid!=4294967295 -k perm\_mod

-a always,exit -F arch=b64 -S setxattr -S lsetxattr -S fsetxattr -S removexattr -S lremovexattr -S fremovexattr -F auid>=1000 -F auid!=4294967295 -k perm\_mod

***# Ensure auditd collects unauthorised access attempts to files (unsuccessful)***

-a always,exit -F arch=b32 -S creat -S open -S openat -S open\_by\_handle\_at -S truncate -S ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=4294967295 -k access

-a always,exit -F arch=b32 -S creat -S open -S openat -S open\_by\_handle\_at -S truncate -S ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=4294967295 -k access

-a always,exit -F arch=b64 -S creat -S open -S openat -S open\_by\_handle\_at -S truncate -S ftruncate -F exit=-EACCES -F auid>=1000 -F auid!=4294967295 -k access

-a always,exit -F arch=b64 -S creat -S open -S openat -S open\_by\_handle\_at -S truncate -S ftruncate -F exit=-EPERM -F auid>=1000 -F auid!=4294967295 -k access

***# Ensure auditd collects information on exporting to media (successful)***

-a always,exit -F arch=b32 -S mount -F auid>=1000 -F auid!=4294967295 -k export

-a always,exit -F arch=b64 -S mount -F auid>=1000 -F auid!=4294967295 -k export

***# Ensure auditd collects file deletion events by user***

-a always,exit -F arch=b32 -S rmdir -S unlink -S unlinkat -S rename -S renameat -F auid>=1000 -F auid!=4294967295 -k delete

-a always,exit -F arch=b64 -S rmdir -S unlink -S unlinkat -S rename -S renameat -F auid>=1000 -F auid!=4294967295 -k delete

***# Ensure auditd collects information on the use of privileged commands***

-a always,exit -F path=/usr/bin/chage -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/bin/chcon -F perm=x -F auid>=1000 -F auid!=4294967295 -F key=privileged-priv\_change

-a always,exit -F path=/usr/bin/chfn -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/bin/chsh -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/bin/crontab -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/bin/gpasswd -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/bin/mount -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/bin/newgrp -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/bin/passwd -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/bin/pkexec -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/bin/screen -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/bin/ssh-agent -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/bin/sudo -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/bin/sudoedit -F perm=x -F auid>=1000 -F auid!=4294967295 -F key=privileged

-a always,exit -F path=/usr/bin/su -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/bin/umount -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/bin/wall -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/bin/write -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/lib64/dbus-1/dbus-daemon-launch-helper -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/libexec/openssh/ssh-keysign -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/libexec/utempter/utempter -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/lib/polkit-1/polkit-agent-helper-1 -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/sbin/netreport -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/sbin/pam\_timestamp\_check -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/sbin/postdrop -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/sbin/postqueue -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/sbin/restorecon -F perm=x -F auid>=1000 -F auid!=4294967295 -F key=privileged-priv\_change

-a always,exit -F path=/usr/sbin/semanage -F perm=x -F auid>=1000 -F auid!=4294967295 -F key=privileged-priv\_change

-a always,exit -F path=/usr/sbin/setsebool -F perm=x -F auid>=1000 -F auid!=4294967295 -F key=privileged-priv\_change

-a always,exit -F path=/usr/sbin/unix\_chkpwd -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/sbin/userhelper -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

-a always,exit -F path=/usr/sbin/usernetctl -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged

***# Make the auditd configuration immutable.***

***# The configuration can only be changed by rebooting the machine.***

-e 2

The auditd service does not include the ability to send audit records to a centralised server for management directly.

It does, however, include a plug-in for audit event multiplexor to pass audit records to the local syslog server.

To do so, open the file /etc/audisp/plugins.d/syslog.conf and set:

active = yes

Enable and start the service:

# systemctl enable auditd.service

# systemctl start auditd.service

**6.3. Enable Kernel Auditing**

Open /etc/default/grub and append the following parameter to the kernel boot line GRUB\_CMDLINE\_LINUX:

audit=1

Update Grub2 configuration to reflect changes:

# grub2-mkconfig -o /boot/grub2/grub.cfg

**7. System Settings – Software Integrity Checking**

**7.1 Advanced Intrusion Detection Environment (AIDE)**

Install AIDE:

# yum install aide

Build AIDE database:

# /usr/sbin/aide --init

By default, the database will be written to the file /var/lib/aide/aide.db.new.gz.

# cp /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz

Storing the database and the configuration file /etc/aide.conf (or SHA2 hashes of the files) in a secure location provides additional assurance about their integrity.

Check AIDE database:

# /usr/sbin/aide --check

By default, AIDE does not install itself for periodic execution. Configure periodic execution of AIDE by adding to cron:

# echo "30 4 \* \* \* root /usr/sbin/aide --check|mail -s 'AIDE' root@example.com" >> /etc/crontab

Periodically running AIDE is necessary in order to reveal system changes.

**7.2 Tripwire**

Open Source Tripwire is an **alternative** to AIDE. It is recommended to use one or another, but not both.

Install Tripwire from the EPEL repository:

# yum install epel-release

# yum install tripwire

# /usr/sbin/tripwire-setup-keyfiles

The Tripwire configuration file is /etc/tripwire/twcfg.txt and the policy file is /etc/tripwire/twpol.txt. These can be edited and configured to match the system Tripwire is installed on, see [this blog post](https://www.lisenet.com/2017/configure-tripwire-on-centos-7/) for more details.

Initialise the database to implement the policy:

# tripwire --init

Check for policy violations:

# tripwire --check

Tripwire adds itself to /etc/cron.daily/ for daily execution therefore no extra configuration is required.

**7.3 Prelink**

Prelinking is done by the prelink package, which is not installed by default.

# yum install prelink

To disable prelinking, open the file /etc/sysconfig/prelink and set the following:

PRELINKING=no

Sed one-liner:

# sed -i 's/PRELINKING.\*/PRELINKING=no/g' /etc/sysconfig/prelink

Disable existing prelinking on all system files:

# prelink -ua

**8. System Settings – Logging and Message Forwarding**

**8.1 Configure Persistent Journald Storage**

By default, journal stores log files only in memory or a small ring-buffer in the directory /run/log/journal. This is sufficient to show recent log history with journalctl, but logs aren’t saved permanently. Enabling persistent journal storage ensures that comprehensive data is available after system reboot.

Open the file /etc/systemd/journald.conf and put the following:

[Journal]

Storage=persistent

***# How much disk space the journal may use up at most***

SystemMaxUse=256M

***# How much disk space systemd-journald shall leave free for other uses***

SystemKeepFree=512M

***# How large individual journal files may grow at most***

SystemMaxFileSize=32M

Restart the service:

# systemctl daemon-reload

# systemctl restart systemd-journald

**8.2 Configure Message Forwarding to Remote Server**

Depending on your setup, open /etc/rsyslog.conf and add the following to forward messages to a some remote server:

\*.\* @graylog.example.com:514

Here**\*.\*** stands for **facility.severity**. Note that a single @ sends logs over UDP, where a double @ sends logs using TCP.

**8.3 Logwatch**

Logwatch is a customisable log-monitoring system.

# yum install logwatch

Logwatch adds itself to /etc/cron.daily/ for daily execution therefore no configuration is mandatory.

**9. System Settings – Security Software**

**9.1 Malware Scanners**

Install Rkhunter and ClamAV:

# yum install epel-release

# yum install rkhunter clamav clamav-update

# rkhunter --update

# rkhunter --propupd

# freshclam -v

Rkhunter adds itself to /etc/cron.daily/ for daily execution therefore no configuration is required. ClamAV scans should be tailored to individual needs.

**9.2 Arpwatch**

Arpwatch is a tool used to monitor ARP activity of a **local network** (ARP spoofing detection), therefore it is unlikely one will use it in the cloud, however, it is still worth mentioning that the tools exist.

Be aware of the configuration file /etc/sysconfig/arpwatch which you use to set the email address where to send the reports.

**9.3 Commercial AV**

Consider installing a commercial AV product that provides real-time on-access scanning capabilities.

**9.4 Grsecurity**

Grsecurity is an extensive security enhancement to the Linux kernel. Although it isn’t free nowadays, the software is still worth mentioning.

The company behind Grsecurity stopped publicly distributing stable patches back in 2015, with an exception of the test series continuing to be available to the public in order to avoid impact to the Gentoo Hardened and Arch Linux communities.

Two years later, the company decided to cease free distribution of the test patches as well, therefore as of 2017, Grsecurity software is available to paying customers only.

**10. System Settings – OS Update Installation**

Install the package yum-utils for better consistency checking of the package database.

# yum install yum-utils

Configure automatic package updates via yum-cron.

# yum install yum-cron

Add the following to /etc/yum/yum-cron.conf to get notified via email when new updates are available:

update\_cmd = default

update\_messages = yes

download\_updates = no

apply\_updates = no

emit\_via = email

email\_from = root@example.com

email\_to = user@example.com

email\_host = localhost

Add the following to /etc/yum/yum-cron-hourly.conf to check for bugfix-related updates every hour and automatically download and install them:

update\_cmd = minimal *# yum --bugfix update-minimal*

update\_messages = yes

download\_updates = yes

apply\_updates = yes

emit\_via = stdio

**Note:** security information is provided by RedHat only. When you query a repository that is provided by CentOS it does not supply security metadata (however the EPEL repository does have security metadata).

Therefore if you decide to use update\_cmd = security, yum will always tell you that nothing from CentOS needs a security update.

Enable and start the service:

# systemctl enable yum-cron.service

# systemctl start yum-cron.service

**11. System Settings – Process Accounting**

The package psacct contain utilities for monitoring process activities:

* **ac** – displays statistics about how long users have been logged on.
* **lastcomm** – displays information about previously executed commands.
* **accton** – turns process accounting on or off.
* **sa** – summarises information about previously executed commands.

Install and enable the service:

# yum install psacct

# systemctl enable psacct.service

# systemctl start psacct.service

**1. Services – SSH Server**

Create a group for SSH access as well as some regular user account who will be a member of the group:

# groupadd ssh-users

# useradd -m -s /bin/bash -G ssh-users tomas

Generate SSH keys for the user:

# su - tomas

$ mkdir --mode=0700 ~/.ssh

$ ssh-keygen -b 4096 -t rsa -C "tomas" -f ~/.ssh/id\_rsa

Generate SSH host keys:

# ssh-keygen -b 4096 -t rsa -N "" -f /etc/ssh/ssh\_host\_rsa\_key

# ssh-keygen -b 1024 -t dsa -N "" -f /etc/ssh/ssh\_host\_dsa\_key

# ssh-keygen -b 521 -t ecdsa -N "" -f /etc/ssh/ssh\_host\_ecdsa\_key

# ssh-keygen -t ed25519 -N "" -f /etc/ssh/ssh\_host\_ed25519\_key

For **RSA** keys, 2048 bits is considered sufficient. **DSA** keys must be exactly 1024 bits as specified by FIPS 186-2.

For **ECDSA** keys, the -b flag determines the key length by selecting from one of three elliptic curve sizes: 256, 384 or 521 bits. **ED25519** keys have a fixed length and the -b flag is ignored.

The host can be impersonated if an unauthorised user obtains the private SSH host key file, therefore ensure that permissions of /etc/ssh/\*\_key are properly set:

# chmod 0600 /etc/ssh/\*\_key

Configure /etc/ssh/sshd\_config with the following:

***# SSH port.***

Port 22

***# Listen on IPv4 only.***

ListenAddress 0.0.0.0

***# Protocol version 1 has been exposed.***

Protocol 2

***#***

***# OpenSSH cipher-related release notes.***

***# OpenSSH 6.2: added support for AES-GCM authenticated encryption.***

***# The cipher is available as aes128-gcm@openssh.com and aes256-gcm@openssh.com.***

***# OpenSSH 6.5: added new cipher chacha20-poly1305@openssh.com.***

***# OpenSSH 6.7: removed unsafe algorithms. CBC ciphers are disabled by default:***

***# aes128-cbc, aes192-cbc, aes256-cbc, 3des-cbc, blowfish-cbc, cast128-cbc.***

***# OpenSSH 6.9: promoted chacha20-poly1305@openssh.com to be the default cipher.***

***#***

Ciphers chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-gcm@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr

***#***

***# OpenSSH 6.2: added support for the UMAC-128 MAC as umac-128@openssh.com***

***# and umac-128-etm@openssh.com. The latter being an encrypt-then-mac mode.***

***# Do not use umac-64 or umac-64-etm because of a small 64 bit tag size.***

***# Do not use any SHA1 (e.g. hmac-sha1, hmac-sha1-etm@openssh.com) MACs***

***# because of a weak hashing algorithm.***

***# Do not use hmac-sha2-256, hmac-sha2-512 or umac-128@openssh.com***

***# because of an encrypt-and-MAC mode. See the link below:***

***# https://crypto.stackexchange.com/questions/202/should-we-mac-then-encrypt-or-encrypt-then-mac***

***#***

MACs hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,umac-128-etm@openssh.com

***#***

***# OpenSSH 6.5: added support for ssh-ed25519. It offers better security***

***# than ECDSA and DSA.***

***# OpenSSH 7.0: disabled support for ssh-dss.***

***# OpenSSH 7.2: added support for rsa-sha2-512 and rsa-sha2-256.***

***#***

HostKeyAlgorithms ssh-ed25519,ssh-ed25519-cert-v01@openssh.com,ssh-rsa,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ssh-rsa-cert-v01@openssh.com,ssh-dss-cert-v01@openssh.com,ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com

***#***

***# OpenSSH 6.5: added support for key exchange using elliptic-curve***

***# Diffie Hellman in Daniel Bernstein's Curve25519.***

***# OpenSSH 7.3: added support for diffie-hellman-group14-sha256,***

***# diffie-hellman-group16-sha512 and diffie-hellman-group18-sha512.***

***#***

KexAlgorithms curve25519-sha256@libssh.org,diffie-hellman-group18-sha512,diffie-hellman-group16-sha512,diffie-hellman-group14-sha256

***# HostKeys for protocol version 2.***

HostKey /etc/ssh/ssh\_host\_rsa\_key

HostKey /etc/ssh/ssh\_host\_ed25519\_key

***# Disabled because uses a small 1024 bit key.***

#HostKey /etc/ssh/ssh\_host\_dsa\_key

***# Disabled because uses weak elliptic curves.***

***# See: https://safecurves.cr.yp.to/***

#HostKey /etc/ssh/ssh\_host\_ecdsa\_key

***# INFO is a basic logging level that will capture user login/logout activity.***

***# DEBUG logging level is not recommended for production servers.***

LogLevel INFO

***# Disconnect if no successful login is made in 60 seconds.***

LoginGraceTime 60

***# Do not permit root logins via SSH.***

PermitRootLogin no

***# Check file modes and ownership of the user's files before login.***

StrictModes yes

***# Close TCP socket after 2 invalid login attempts.***

MaxAuthTries 2

***# The maximum number of sessions per network connection.***

MaxSessions 3

***# User/group permissions.***

AllowUsers

AllowGroups ssh-users

DenyUsers root

DenyGroups root

***# Password and public key authentications.***

PasswordAuthentication no

PermitEmptyPasswords no

PubkeyAuthentication yes

AuthorizedKeysFile .ssh/authorized\_keys

***# Disable unused authentications mechanisms.***

RSAAuthentication no ***# DEPRECATED***

RhostsRSAAuthentication no ***# DEPRECATED***

ChallengeResponseAuthentication no

KerberosAuthentication no

GSSAPIAuthentication no

HostbasedAuthentication no

IgnoreUserKnownHosts yes

***# Disable insecure access via rhosts files.***

IgnoreRhosts yes

AllowAgentForwarding no

AllowTcpForwarding no

***# Disable X Forwarding.***

X11Forwarding no

***# Disable message of the day but print last log.***

PrintMotd no

PrintLastLog yes

***# Show banner.***

Banner /etc/issue

***# Do not send TCP keepalive messages.***

TCPKeepAlive no

***# Default for new installations.***

UsePrivilegeSeparation sandbox

***# Prevent users from potentially bypassing some access restrictions.***

PermitUserEnvironment no

***# Disable compression.***

Compression no

***# Disconnect the client if no activity has been detected for 900 seconds.***

ClientAliveInterval 900

ClientAliveCountMax 0

***# Do not look up the remote hostname.***

UseDNS no

UsePAM yes

You can use the [ssh-audit](https://github.com/arthepsy/ssh-audit) tool to test your SSH server configuration.

In case you want to change the default SSH port to something else, you will need to tell SELinux about it.

# yum install policycoreutils-python

For example, to allow SSH server to listen on TCP 2222, do the following:

# semanage port -a -t ssh\_port\_t 2222 -p tcp

Ensure that the firewall allows incoming traffic on the new SSH port and restart the sshd service.

**2. Service – Network Time Protocol**

CentOS 7 should come with Chrony, make sure that the service is enabled:

# systemctl enable chronyd.service

**3. Services – Mail Server**

**3.1 Postfix**

Postfix should be installed and enabled already. In case it isn’t, the do the following:

# yum install postfix

# systemctl enable postfix.service

Open /etc/postfix/main.cf and configure the following to act as a null client:

smtpd\_banner = $myhostname ESMTP

inet\_interfaces = loopback-only

inet\_protocols = ipv4

mydestination =

local\_transport = error: local delivery disabled

unknown\_local\_recipient\_reject\_code = 550

mynetworks = 127.0.0.0/8

relayhost = [mail.example.com]:587

Optionally (depending on your setup), you can configure Postfix to use authentication:

# yum install cyrus-sasl-plain

Open /etc/postfix/main.cf and add the following:

smtp\_sasl\_auth\_enable = yes

smtp\_sasl\_password\_maps = hash:/etc/postfix/sasl\_passwd

smtp\_sasl\_security\_options = noanonymous

smtp\_tls\_CApath = /etc/ssl/certs

smtp\_use\_tls = yes

Open /etc/postfix/sasl\_passwd and put authentication credentials in a format of:

[mail.example.com]:587 user@example.com:password

Set permissions and create a database file:

# chmod 0600 /etc/postfix/sasl\_passwd

# postmap /etc/postfix/sasl\_passwd

Restart the service and ensure that firewall allows outgoing traffic to the SMTP relay server.

**3.2 Mail Distribution to Active Mail Accounts**

Configure the file /etc/aliases to have a forward rule for the root user.

**4. Services – Remove Obsolete Services**

None of these should be installed on CentOS 7 minimal:

# yum remove xinetd telnet-server rsh-server \

telnet rsh ypbind ypserv tfsp-server bind \

vsfptd dovecot squid net-snmpd talk-server talk

Check all enabled services:

# systemctl list-unit-files --type=service|grep enabled

Disable kernel dump service:

# systemctl disable kdump.service

# systemctl mask kdump.service

Disable everything that is not required, e.g.:

# systemctl disable tuned.service

**5. Services – Restrict at and cron to Authorised Users**

If the file cron.allow exists, then only users listed in the file are allowed to use cron, and the cron.deny file is ignored.

# echo root > /etc/cron.allow

# echo root > /etc/at.allow

# rm -f /etc/at.deny /etc/cron.deny

Note that the root user can always use cron, regardless of the usernames listed in the access control files.

**6. Services – Disable X Windows Startup**

This can be achieved by setting a default target:

# systemctl set-default multi-user.target

**7. Services – Fail2ban**

Install Fail2ban from the EPEL repository:

# yum install epel-release

# yum install fail2ban

If using iptables rather than firewalld, open the file /etc/fail2ban/jail.d/00-firewalld.conf and comment out the following line:

#banaction = firewallcmd-ipset

Fail2Ban uses /etc/fail2ban/jail.conf. Configuration snippet for SSH is provided below:

[sshd]

port = ssh

enabled = true

ignoreip = 10.8.8.61

bantime = 600

maxretry = 5

If you run SSH on a non-default port, you can change the port value to any positive integer and then enable the jail.

# systemctl enable fail2ban.service

# systemctl start fail2ban.service

**8. Services – Sysstat to Collect Performance Activity**

Sysstat may provide useful insight into system usage and performance, however, unless used, the service should be disabled, or not installed at all.

# yum install sysstat

# systemctl enable sysstat.service

# systemctl start sysstat.service