



Interested in learning more about security?

**SANS Institute**

**Security Consensus Operational Readiness Evaluation**

This checklist is from the SCORE Checklist Project. Reposting is not permitted without express, written permission.

# SCORE Security Checklist

# HARDENING THE LINUX SYSTEM

Step-by-Step Guide

Release 2

**Paul Loftness**

Baylor University | Undergraduate Researcher UAB

**Simeon Blatchley**

University of Maryland UC | Analyst SAIC

Created July 2012

Updated January 2016

## OVERVIEW

This document is a general checklist for hardening a Linux system. The important thing to remember is that there is no 100% right checklist. There are bound to be variables that must be changed, and all this document is intending on doing, is to allow the Linux user to follow the steps and successfully secure any type of system without needing much knowledge. However, they will still have the ability to further their security with the more advanced checklists. Of course with any of these checklists, there is a chance of breaking something, and thus all steps must be researched for **your** specific distro/system. A single user's security settings will be vastly different from a multi-user system.

### Notes:

1. All commands listed will need to be run as root. You can switch to root by running either 'sudo -l' or 'su.'
2. Where we use "vi" as the command line editor, you can replace it for "gedit" or any other text editor.
3. Where use "apt-get" you can insert your distro version of package management. Or if necessary you can download the binaries and compile them (a somewhat easy process of ./configure, make, make install, etc).
4. Shaded areas are terminal commands, you can cut and paste these, although one should be careful and know what the command actually does.

## MAINTAINANCE

### Update the Operating System

#### Debian, Ubuntu, Kali, etc

```
apt-get update
```

```
apt-get upgrade
```

#### Redhat, YellowDog, CentOS, Scientific Linux, Fedora, etc.

```
yum list updates
```

```
yum update
```

#### Suse/etc

```
zypper ref (Refresh the repos)
```

```
zypper dup (Normal update and install)
```

## HARDEN THE SYSTEM

### Install Bastille

There are a few options around to harden a linux system, but we have tested Bastille in real life scenarios and found it to be the most resilient. It is rather customizable for various types of configurations.

```
apt-get install bastille
```

Choose yes when it asks if you want to continue. Once it is done installing, run:

```
bastille -c
```

This will start the command line interface, to allow you to configure Bastille. From there, you'll accept their terms of agreement, and be on your way. It is safe to say that you can just accept the default values, however you should also read about them. Please see our Bastille Configuration file for a more detailed look at Bastille. It's safe to ignore most errors it throws at the end and beginning of the configuration.

### Install Apparmor

Some packages will install their own enforced profiles. Active profiles for LAM Server:

```
usr.sbin.mysqlld
```

```
usr.sbin.apache2
```

All activity will be logged by auditd and saved to /var/log/audit/audit.log.

```
apt-get install apparmor-profiles
```

```
apparmor_status (to see current profiles and associated modes)
```

```
man apparmor (for more details of what to do with that information)
```

## Configure and Use SELinux

As this is the more complicated and advanced alternative to Apparmor, there is a detailed checklist that can be found here: [https://docs.oracle.com/cd/E37670\\_01/E36387/html/ol\\_selinux\\_sec.html](https://docs.oracle.com/cd/E37670_01/E36387/html/ol_selinux_sec.html) and a troubleshooting guide can be found here <https://wiki.centos.org/HowTos/SELinux>. In this release of our checklist, we have excluded our previous steps and notes on SELinux as we believe making changing to SELinux requires greater detail and knowledge [gained by fully referencing the above documents], in an effort to prevent people from “cutting and pasting” or falling into “script kiddie mode”.

## Configure and use PAM authentication daemon

The instructions below are assuming that you do not have SELinux installed. These configurations may change with the installation of SELinux. They will be covered in the SELinux documents listed above.

Also for further PAM info, refer to the PAM Configurations here:

<http://doc.mapr.com/display/MapR/PAM+Configuration>.

```
vi /etc/pam.d/common-password
```

```
change:
```

```
password requisite pam_unix.so nullok obscure sha512
```

to:

```
password requisite pam_unix.so nullok obscure sha512 min=8
```

Change `min=8` with whatever password length you desire to be enforced.

### Shadow File Password Policy

Change minimum and maximum password ages (most likely set to 0:99999 in the file). We suggest changing those to 1:60 for all entries. . Here is a good example of [changing password aging from the the shadow file](http://www.cyberciti.biz/faq/understanding-etcshadow-file/) <http://www.cyberciti.biz/faq/understanding-etcshadow-file/>

### Shutdown unnecessary services

```
netstat -anp | grep LISTEN | grep -v STREAM
```

Analyze the services and the process id/process name. Determine which services to terminate. Take caution and ensure you research any unknown services (some can break your host).

```
cd /etc
```

```
find . -print | grep XXX (where XXX is part of the name of the program)
```

For those entries in the `/etc/rc#.d` directory, delete them (rm). Some suggestions to disable:

**a. Remove or disable the "r" commands.** This includes rlogind, rshd, rcmd, rexecd, rbootd, rquotad, rstatd, rusersd, rwalld and rexd. These services are inadequately authenticated. It is better to remove these and use SSH and scp instead.

**b. Remove or disable fingerd.** Remove or disable fingerd if present. Apart from the possibility of a software vulnerability, fingerd allows an attacker to enumerate usernames on the system and to determine the timing and frequency of system administrator logins.

**c. Remove or disable tftpd.** Tftpd is unauthenticated and not protected against brute-force attacks seeking to enumerate and download files. Do not use tftpd (trivial file transfer protocol) unless unavoidable.

**d. Remove or disable telnet.** Telnet sends commands unencrypted over the wire. This enables the sniffing of passwords and other information as well as man-in-the-middle attacks. Replace with SSH.

**e. Disable SNMP daemon.** If present by default, disable any SNMP daemon unless this is really required for the role of the computer. Research this service for further info.

### Examine INIT files for discrepancies

```
cd /etc/init or /etc/xinit (should match /etc/init.d)
```

```
cd /etc/init.d (examine the two to make sure they match)
```

```
cd /etc
```

```
find rc*.d | xargs ls -l
```

All entries should be links to the ../init.d directory. Investigate those that aren't.



Examine startup scripts (00755 is the norm, but 00700 is ok here as well)

rc.\* (as rc.1-6 or rc1-6.d):

```
chmod 0700 /etc/rc*  
chmod 0700 /etc/init.d*
```

Here's a good article about services and runlevels:

<https://www.linux.com/news/enterprise/systems-management/8116-an-introduction-to-services-runlevels-and-rcd-scripts/>

## LOCK DOWN THE USER SESSION

### Secure terminals

The relevant configuration file may be called `/etc/ttys`, `/etc/default/login`, `/etc/security` or `/etc/securetty` depending on the system. See the manual pages for file format and usage information. Check that the `secure` option is removed from any local entries that don't need root login capabilities. The `secure` option should be removed from console if you do not want users to be able to reboot in single user mode. [Note: This does not affect usability of the `su` command.] If it is not already the default, consider using a special group (such as the `wheelgroup` on BSD systems) to restrict which users can use `su` to become root.

### PATH advice

Check that the current directory `."` is not in the `PATH`. Note that an empty string is interpreted to mean the same as `."` so also make sure the `PATH` does not contain any empty strings. For example, the following `PATH` is insecure:

```
/sbin:/bin:/usr/sbin::/usr/bin
```

This PATH advice is especially important for the root account. Including “.” in the PATH variable can be used by an attacker to fool a root user into running a malicious binary by substituting `./ls` instead of `/bin/l`s for example.

### **Configure user login sessions to time out automatically.**

After a certain period of inactivity, in particular for the root user. To do this, set the appropriate variable in your shell's startup files.

```
typeset -r TMOU=900 (15 minutes = 900 seconds)
```

### **Securing History**

```
chattr +a .bash_history (append)
```

```
chattr +l .bash_history
```

Users history is being locked and they will have to agree before they use your services.

## **LOCK DOWN config-FILES CONTENT**

### **Analyze DNS**

Looking for rogue entries in the resolv.conf file.

```
vi /etc/resolv.conf
```

Here you should just see the DNS server that the router/modem passed on to your computer, and whatever you have added. Other entries can be considered to be rouge (remember to

scroll down). However, before you go and delete your whole file, be sure and lookup the listed server and do your research. Here is a good link for some basic DNS finding info:

<http://www.cyberciti.biz/faq/how-to-find-out-dns-for-router/>

### **Analyze host files**

Ensure there is nothing redirecting incorrectly here.

```
vi /etc/hosts
```

### **Analyze contents of permission files**

If you are running , root should have \* as the password. If you are running su, it will have a password. Nobody else aside from you and known users should have a password (the big long hash). If they do, make sure they shouldn't be there, and delete that line. Make sure system users have /bin/null set as their shell. Check for rogue users.

```
vi /etc/passwd
```

```
vi /etc/shadow
```

## **SET PERMISSIONS ON SENSITIVE SYSTEM FILES**

Sensitive system files need to have the proper permissions set on them to prevent unauthorized changes (see “Integrity” in the CIA triad).

### **Configuration Files**

#### **a. Firewall**

```
chmod 0700 /etc/profile
```

```
chmod 0700 /etc/hosts.allow
```

```
chmod 0700 /etc/mtab,  
chmod 0700 /etc/utmp  
chmod 0700 /var/adm/wtmp (or /var/log/wtmp),  
chmod 0700 /etc/syslog.pid (or /var/run/syslog.pid)
```

### **b. Kernel**

```
chmod 0700 /etc/sysctl.conf  
chmod 0700 /etc/inittab
```

### **c. Users**

Make sure the owner & group are set to root.root and the permissions are set to 0644 (except on the /etc/shadow file which should be 400). Here is a good link for permission changing in Linux:

<http://articles.slicehost.com/2010/7/17/checking-linux-file-permissions-with-ls>

```
ls -la /etc/fstab
```

Verify: root.root and -rw-r--r-- (644)

```
ls -la /etc/passwd
```

Verify: root.root and -rw-r--r-- (644)

```
ls -la /etc/shadow
```

Verify: root.root and -rw-r----- (400)

```
ls -la /etc/group
```

Verify: root.root and -rw-r—r-- (644)

```
ls -la /etc/sudoers
```

Verify: root.root and -rw-r—r-- (644)

### Log Files

Ensure that these files (usually located in `/var/log/`, `/var/adm`, or `var/tmp`) are only writable by root.

### Any World-Writable Files

Ensure that there are no unexpected world writable files or directories on your system. Use the find command to locate these:

```
find / -type d -perm +2 -ls
```

```
chmod 750
```

```
rm
```

### SET PERMISSIONS ON SENSITIVE BINARIES

Another good security practice is to set the permissions on certain commands. However, it is very important to remember that what you change here depends on what system you're using. Also, the location of binaries will differ based upon the system (for instance `/bin` and `/usr/bin` and `/usr/sbin`). For instance a server used for development would need the "make" command to be able to be run by any user. Whereas, on a production server it would not be needed. Some examples (you'll need to run

these as root):

### Set uid:

```
-i / su  
find / \( -perm -2000 \  
chown root:admin /bin/example  
chmod 02750 /bin/example  
find / \( -perm -4000 \  
chown root:admin /bin/example  
chmod 04750 /bin/su
```

## FURTHER SUGGESTIONS

### Privilege Escalation

```
chmod 02750 /bin/su  
chmod 02750 /bin/sudo
```

### Network settings:

```
chmod 02750 /bin/ping  
chmod 02750 /sbin/ifconfig
```

### Users On:

```
chmod 02750 /usr/bin/w  
chmod 02750 /usr/bin/who
```

### System Configuration

```
chmod 02750 /usr/bin/locate  
chmod 02750 /usr/bin/whereis
```

## Kernel Modules

Ensure that the files holding the kernel and any kernel modules are owned by root, have group ownership set to group id 0 and permissions that prevent them being written to by any non-root users.

### To list current module directory:

```
echo "Modules dir: /lib/modules/$(uname -r) for kernel version $(uname -r)"
```

### To list contents/permissions of that directory:

```
ls -l /lib/modules/$(uname -r)
```

If you would like further information or assistance please contact us and we are always interested in job opportunities.

## PAUL LOFTNESS

[pabloloft@gmail.com](mailto:pabloloft@gmail.com)

<https://www.linkedin.com/in/paulloftness>

## SIMEON BLATCHLEY

[simeon@linkxrdp.com](mailto:simeon@linkxrdp.com)

<https://www.linkedin.com/in/sblatchley>



# Upcoming SANS Training

[Click Here for a full list of all Upcoming SANS Events by Location](#)

SANS San Francisco Winter 2017	San Francisco, CAUS	Nov 27, 2017 - Dec 02, 2017	Live Event
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZUS	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Khobar 2017	Khobar, SA	Dec 02, 2017 - Dec 07, 2017	Live Event
SANS Munich December 2017	Munich, DE	Dec 04, 2017 - Dec 09, 2017	Live Event
European Security Awareness Summit & Training 2017	London, GB	Dec 04, 2017 - Dec 07, 2017	Live Event
SANS Austin Winter 2017	Austin, TXUS	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Frankfurt 2017	Frankfurt, DE	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Bangalore 2017	Bangalore, IN	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DCUS	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LAUS	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS SEC460: Enterprise Threat Beta	San Diego, CAUS	Jan 08, 2018 - Jan 13, 2018	Live Event
SEC599: Defeat Advanced Adversaries	San Francisco, CAUS	Jan 15, 2018 - Jan 20, 2018	Live Event
SANS Amsterdam January 2018	Amsterdam, NL	Jan 15, 2018 - Jan 20, 2018	Live Event
Northern VA Winter - Reston 2018	Reston, VAUS	Jan 15, 2018 - Jan 20, 2018	Live Event
SANS Dubai 2018	Dubai, AE	Jan 27, 2018 - Feb 01, 2018	Live Event
SANS Las Vegas 2018	Las Vegas, NVUS	Jan 28, 2018 - Feb 02, 2018	Live Event
Cyber Threat Intelligence Summit & Training 2018	Bethesda, MDUS	Jan 29, 2018 - Feb 05, 2018	Live Event
SANS Miami 2018	Miami, FLUS	Jan 29, 2018 - Feb 03, 2018	Live Event
SANS London February 2018	London, GB	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS Scottsdale 2018	Scottsdale, AZUS	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS Secure India 2018	Bangalore, IN	Feb 12, 2018 - Feb 17, 2018	Live Event
SANS Southern California- Anaheim 2018	Anaheim, CAUS	Feb 12, 2018 - Feb 17, 2018	Live Event
SANS Dallas 2018	Dallas, TXUS	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS Secure Japan 2018	Tokyo, JP	Feb 19, 2018 - Mar 03, 2018	Live Event
Cloud Security Summit & Training 2018	San Diego, CAUS	Feb 19, 2018 - Feb 26, 2018	Live Event
SANS Brussels February 2018	Brussels, BE	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS London November 2017	OnlineGB	Nov 27, 2017 - Dec 02, 2017	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced