Securing Ubuntu 16.04LTS Installations

Dimitrios Politis
Athens, Greece

August 10, 2016

Contents

1 Introduction to Computer Security **2**
1.1 What is Computer Security 2
1.2 Security Controls 2

2 Security Tips for Installation **3**
2.1 Bios Settings 3
2.2 Partitioning the Disk 3
2.3 Securing the Boot Loader 4
2.4 General security tips 5
2.4.1 Maintaining the minimum required base 6
2.4.2 Turn on AppArmor 6
2.4.3 Remove Desktop Managers 7
2.4.4 Disable unneeded kernel modules 7
2.4.5 Disable uncommon filesystems 8
2.4.6 Disable potentially dangerous protocols 8
2.4.7 Check if Apport is disabled 8
2.4.8 Disable Coredumps 8
2.4.9 Lock-down Cronjobs 9
2.4.10 Configure User Security Limits 9
2.4.11 Remove suid from certain executables 9
2.4.12 Impose stricter file and folder permissions 9
2.4.13 Configure DNS resolvers 10
2.4.14 Remove host overrides for users and services 10
2.4.15 Configure access with TCP Wrappers 10
2.4.16 Configure Banners 10
2.4.17 Remove unneeded users 10
2.4.18 Create new users with non default shell 11
2.4.19 Secure NTP 11
2.4.20 Configure logrotate 11
2.4.21 Disable Ctrl+Alt+Delete 11
2.5 Account Security 12
2.5.1 Creating Strong Passwords 12
2.5.2 Forcing Strong Passwords 12
2.5.3 Account Expiration Policy 13
2.5.4 Account Locking 13
2.5.5 Session Locking – Automatic Logout 13
2.5.6 Disallowing Root Access 14
2.6 Risks to Services 14
2.6.1 Securing NFS 15
2.6.2 Securing the Apache HTTP Server 18
2.6.2.1 Apache Security 19
2.6.2.2 Configure settings 19
2.6.2.3 Disable Apache2 Unneeded Modules 21

2.6.2.4 Use the Apache mod_security module . . . 22
2.6.2.5 Use the Apache mod_evasive module . . . 22
2.6.2.6 Encryption Protocol Usage Recommendations . . . 23
2.6.2.7 Securing with fail2ban . . . 25
2.6.3 Securing SSH . . . 25
2.6.4 Securing SAMBA . . . 27
2.7 Securing Network Access . . . 28
2.7.1 Securing Various Network Settings . . . 28
2.7.2 Securing DNS Traffic with DNSSEC . . . 30
2.8 Securing System further . . . 31
2.8.1 Send logs to a centralized logging system . . . 31
2.8.2 Install an Antivirus solution . . . 31
2.8.3 Ensure File Integrity . . . 32
2.8.4 Install Rootkit detection Software . . . 32
2.8.5 Install Intrusion Detection Software . . . 33
2.8.6 System audit . . . 33
2.8.6.1 Installing the audit Packages . . . 34
2.8.6.2 Preconfigured Rules Files . . . 34

3 Use System Hardening Utilities 35
3.1 BastilleLinux - The Bastille Linux Security Hardening Tool . . . 35
3.2 Auditing System Settings with SCAP Security Guide . . . 36

4 System Hardening Bash Script 37
4.1 System Evaluation and Assessment . . . 38

5 Conclusions 39

Appendices 40

A Bash Script source code 41

Abstract

In today's ever changing operating environment, system administrators face new challenges when trying to secure their networks. The same applies with more emphasis in interconnected computer networks, where more valuable info is distributed amongst many devices. In this paper we outline some guidelines to secure our Linux server infrastructure. The aforementioned guidelines apply mainly on Ubuntu 16.04LTS Server and Desktop installations.

1 Introduction to Computer Security

1.1 What is Computer Security

Computer security is a general term that covers a wide area of computing and information processing. Many security consultants and vendors agree upon the standard security model known as CIA, or Confidentiality, Integrity, and Availability. This three-tiered model is a generally accepted component to assessing risks of sensitive information and establishing security policy. The following describes the CIA model in further detail [4]:

Confidentiality Sensitive information must be available only to a set of predefined individuals. Unauthorized transmission and usage of information should be restricted.

Integrity Information should not be altered in ways that render it incomplete or incorrect. Unauthorized users should be restricted from the ability to modify or destroy sensitive information.

Availability Information should be accessible to authorized users any time that it is needed.

1.2 Security Controls

Computer security is often divided into three distinct master categories, commonly referred to as controls:

Physical It incorporates all security measures regarding physical security.

Technical Secure deployment of machines (physical and VM).

Administrative Corporate security policy, personnel access to assets.

2 Security Tips for Installation

The following part is about strengthening the default basic installation of Ubuntu 16.04LTS (either server or desktop variant). Basic command-line knowledge and Linux installation experience is required. Hardening follows a bottom-up sequence, beginning with BIOS settings and finishing by configuring advanced auditing on the system.

2.1 Bios Settings

The two primary reasons for password protecting the BIOS of a computer are:

Preventing Changes to BIOS Settings If an intruder has access to the BIOS, they can set it to boot from a CD-ROM or a flash drive. This makes it possible for them to enter rescue mode or single user mode, which in turn allows them to start arbitrary processes on the system or copy sensitive data.

Preventing System Booting Some BIOSes allow password protection of the boot process. When activated, an attacker is forced to enter a password before the BIOS launches the boot loader.

2.2 Partitioning the Disk

It is a good practice to always create separate partitions for the /boot/, /, /home/, /tmp/, and /var/tmp/ directories. The reasons for each are different, and we will address each one below.

/boot This partition is the first partition that is read by the system during boot up. The boot loader and kernel images that are used to boot your system into Linux are stored in this partition. This partition should not be encrypted. If this partition is included in / and that partition is encrypted or otherwise becomes unavailable then your system will not be able to boot.

It is also desirable to keep `/boot` as read-only. Linux kernel and its related files are in `/boot` directory which is by default as read-write. Changing it to read-only reduces the risk of unauthorized modification of critical boot files. To do this, open `/etc/fstab` file.

```
~]# vi /etc/fstab
```

Edit the corresponding line like this [8]:

```
/boot     /boot     ext2     defaults,ro     1 2
```

Please note that you need to reset the change to read-write if you need to upgrade the kernel in future.

/home When user data (`/home`) is stored in `/` instead of in a separate partition, the partition can fill up causing the operating system to become unstable. Also, when upgrading your system to the next version of Linux distro, it is a lot easier when you can keep your data in the `/home` partition as it will not be overwritten during installation. If the root partition (`/`) becomes corrupt your data could be lost forever. By using a separate partition there is slightly more protection against data loss. You can also target this partition for frequent backups.

/tmp and /var/tmp Both the `/tmp` and `/var/tmp` directories are used to store data that does not need to be stored for a long period of time. However, if a lot of data floods one of these directories it can consume all of your storage space. If this happens and these directories are stored within `/` then your system could become unstable and crash. For this reason, moving these directories into their own partitions is a good idea. Also it is a good idea to **disable exec rights on `/tmp`**.

/dev/shm It is also a good practice to disable exec rights on shared memory, because a lot of exploits use it to commit attacks. To do this, create `/etc/systemd/system/dev-shm.mount` file.

```
~]# cat > /etc/systemd/system/tmp.mount << EOF
# /etc/systemd/system/default.target.wants/tmp.mount -> ../tmp.mount

[Unit]
Description=Temporary Directory
Documentation=man:hier(7)
Before=local-fs.target

[Mount]
What=tmpfs
Where=/tmp
Type=tmpfs
Options=mode=1777,strictatime,nosuid,noexec
EOF
```

Create a symlink like this [6]:

```
~]# ln -s /etc/systemd/system/dev-shm.mount /etc/systemd/system/
default.target.wants/dev-shm.mount
```

Then execute

```
~]# systemctl daemon-reload
```

Both the `/tmp` and `/var/tmp` are secured using the same way.

2.3 Securing the Boot Loader

Editing the `GRUB_TIMEOUT=0` and `GRUB_HIDDEN_TIMEOUT=0` parameter in file `/etc/default/grub` is not always safe and it is recommended to password-protect the GRUB 2 boot loader. The primary reasons for password protecting a Linux boot loader are as follows:

Preventing Access to Single User Mode If attackers can boot the system into single user mode, they are logged in automatically as root without being prompted for the root password.

Preventing Access to the GRUB 2 Console If the machine uses GRUB 2 as its boot loader, an attacker can use the GRUB 2 editor interface to change its configuration or to gather information using the `cat` command.

Preventing Access to Insecure Operating Systems If it is a dual-boot system, an attacker can select an operating system at boot time, for example DOS, which ignores access controls and file permissions.

To enable the use of passwords, specify a superuser who can reach the protected entries. It is recommended that the superuser is **different from system users**. Other users can be specified to access these entries as well. To create a grub password for that user, issue:

```
~]# grub-mkpasswd-pbkdf2
```

Now insert the created hash in the appropriate file:

```
~]# cat >> /etc/grub.d/40_custom <<EOF
set superusers="myuser"
password_pbkdf2 myuser grub.pbkdf2.sha512.10000.19074739ED80F115963D984
BDCB35AA671C24325755377C3E9B014D862DA6ACC77BC110EED41822800A87FD3700C03
7320E51E9326188D53247EC0722DDF15FC.C56EC0738911AD86CEA55546139FEBC366A3
93DF9785A8F44D3E51BF09DB980BAFEF85281CBBC56778D8B19DC94833EA8342F7D73E3
A1AA30B205091F1015A85
export superusers
EOF
```

Now change the following line in `/etc/default/grub`:

```
GRUB_CMDLINE_DEFAULT=--users myuser
```

and finally update grub configuration:

```
~]# update-grub
```

To disable password-less single-user mode make the following changes [1]:

```
~]# vi /lib/systemd/system/emergency.service
~]# vi /lib/systemd/system/rescue.service
```

Find line: `ExecStart` and change `sushell` to `sulogin`.

2.4 General security tips

Apart from securing mounts and bootloader, the following security tips can be employed:

2.4.1 Maintaining the minimum required base

It is desirable to begin by deploying a minimal installation and keep the minimum amount of packages required, disable unnecessary services and enable system firewall. Also keep your system updated frequently by executing security updates via cronjob.

```
~]# systemctl list-units | grep service
~]# systemctl disable rpcbind

~]# ufw enable
```

By default ufw is configured to deny any incoming connections and allow all outgoing ones. So the system is secured by default.

To create a cronjob for auto-install security updates do the following:

```
~]# cat > /etc/cron.weekly/apt-security-updates << EOF
echo "**************" >> /var/log/apt-security-updates
date >> /var/log/apt-security-updates
aptitude update >> /var/log/apt-security-updates
aptitude safe-upgrade -o Aptitude::Delete-Unused=false --assume-yes \
--target-release `lsb_release -cs`-security \
>> /var/log/apt-security-updates
echo "Security updates (if any) installed"
EOF

~]# chmod +x /etc/cron.weekly/apt-security-updates

~]# cat > /etc/logrotate.d/apt-security-updates << EOF
/var/log/apt-security-updates {
        rotate 2
        weekly
        size 250k
        compress
        notifempty
}
EOF
```

This will rotate the log file every week (weekly), or if it's over 250kB in size (see Section 2.4.20), compressing old versions (compress). The previous two log files will be kept (rotate 2), and no rotation will occur if the file is empty (notifempty).

2.4.2 Turn on AppArmor

Ubuntu Security Enhancements. AppArmor is extremely useful for reducing the number of software that can be exploited, as web browser, server software etc [11]. The protected processes are confined by using profiles. We can use additional profiles by installing the apparmor-profiles package.

```
~]# apt-get install apparmor-profiles
```

Profiles are stored in `/etc/apparmor.d`

AppArmor has 2 modes of operation:

- Enforce - All profile rules are enforced, any illegal actions are not permitted and logged in syslog.
- Complain – All actions are only logged without being restricted.

We can check the AppArmor mode of operation by issuing:

```
~]# apparmor_status
```

To change mode of operation (eg. mysqld):

```
~]# apt-get install apparmor-utils
~]# aa-enforce /usr/sbin/mysqld
```

or to make AppArmor only complain about issues:

```
~]# aa-complain /usr/sbin/mysqld
```

To load a profile to the kernel:

```
~]# cat /etc/apparmor.d/profile.name | sudo apparmor_parser -a
```

To enable AppArmor for an app (eg. firefox):

```
~]# rm /etc/apparmor.d/disable/usr.bin.firefox
~]# cat /etc/apparmor.d/usr.bin.firefox | sudo apparmor_parser -a
```

To disable the firefox profile if it's causing problems:

```
~]# ln -s /etc/apparmor.d/usr.bin.firefox /etc/apparmor.d/disable/
~]# apparmor_parser -R /etc/apparmor.d/usr.bin.firefox
```

2.4.3 Remove Desktop Managers

It is advisable to remove X window system, Desktop managers and accompanying programs **on production servers**, because they clutter a server installation and pose various security concerns.

```
~]# apt-get remove x-window-system-core
~]# apt-get autoremove --purge
```

2.4.4 Disable unneeded kernel modules

Kernel device modules that must be disabled for security reasons [1].

```
~]# echo "install bluetooth /bin/true" > /etc/modprobe.d/disablemod.conf
~]# echo "install firewire-core /bin/true" >> /etc/modprobe.d/
disablemod.conf
~]# echo "install net-pf-31 /bin/true" >> /etc/modprobe.d/disablemod.conf
~]# echo "install soundcore /bin/true" >> /etc/modprobe.d/disablemod.conf
~]# echo "install thunderbolt /bin/true" >> /etc/modprobe.d/
disablemod.conf
~]# echo "install usb-midi /bin/true" >> /etc/modprobe.d/disablemod.conf
~]# echo "install usb-storage /bin/true" >> /etc/modprobe.d/
disablemod.conf
```

2.4.5 Disable uncommon filesystems

File-systems that are not used by the kernel must be disabled for security reasons [1].

```
~]# echo "install cramfs /bin/true" > /etc/modprobe.d/disablemnt.conf
~]# echo "install freevxfs /bin/true" >> /etc/modprobe.d/disablemnt.conf
~]# echo "install jffs2 /bin/true" >> /etc/modprobe.d/disablemnt.conf
~]# echo "install hfs /bin/true" >> /etc/modprobe.d/disablemnt.conf
~]# echo "install hfsplus /bin/true" >> /etc/modprobe.d/disablemnt.conf
~]# echo "install squashfs /bin/true" >> /etc/modprobe.d/disablemnt.conf
~]# echo "install udf /bin/true" >> /etc/modprobe.d/disablemnt.conf
~]# echo "install vfat /bin/true" >> /etc/modprobe.d/disablemnt.conf
```

2.4.6 Disable potentially dangerous protocols

Uncommon network protocols that must be disabled for security reasons [1].

```
~]# echo "install dccp /bin/true" >> /etc/modprobe.d/disablenet.conf
~]# echo "install sctp /bin/true" >> /etc/modprobe.d/disablenet.conf
~]# echo "install rds /bin/true" >> /etc/modprobe.d/disablenet.conf
~]# echo "install tipc /bin/true" >> /etc/modprobe.d/disablenet.conf
```

2.4.7 Check if Apport is disabled

Apport is a system which intercepts crashes right when they happen the first time, gathers potentially useful information about the crash and the OS environment and is able to file non-crash bug reports about software, so that developers still get information about package versions, OS version etc. Apport is not enabled by default in stable releases, even if it is installed. The automatic crash interception component of apport is disabled by default in stable releases for a number of reasons:

- Apport collects potentially sensitive data, such as core dumps, stack traces, and log files. They can contain passwords, credit card numbers, serial numbers, and other private material.
- Data collection from apport takes a nontrivial amount of CPU and I/O resources, which slow down the computer and don't allow you to restart the crashed program for several seconds.

To disable Apport issue the following commands:

```
~]# sed -i 's/enabled=.*/enabled=0/' /etc/default/apport
~]# systemctl mask apport.service
```

2.4.8 Disable Coredumps

Coredumps can be a serious information leak, as they may contain passwords and other sensitive data. It is therefore advised to disable this feature in production servers, if debugging is not needed. To disable coredumps issue the following:

```
~]# sed -i 's/^#Storage=.*/Storage=none/' /etc/systemd/coredump.conf
~]# systemctl restart systemd-journald
```

2.4.9 Lock-down Cronjobs

Cron has it's own built in feature, where it allows to specify who may, and who may not want to run jobs. This is controlled by the use of files called `/etc/cron.allow` and `/etc/cron.deny`. To lock a user using cron, simply add user names in `cron.deny` and to allow a user to run cron add in `cron.allow` file. If you would like to disable all users from using cron, add the 'ALL' line to `cron.deny` file [8].

```
~]# echo ALL > /etc/cron.deny
```

2.4.10 Configure User Security Limits

It is a good practice to set user limits in a production server. Security limits include max-logins, max running-processes, coredump file-size etc. To set the aforementioned user security limits, do the following:

```
~]# sed -i 's/^# End of file*//' /etc/security/limits.conf
~]# echo "* hard maxlogins 10" >> /etc/security/limits.conf
~]# echo "* hard core 0" >> /etc/security/limits.conf
~]# echo "* soft nproc 100" >> /etc/security/limits.conf
~]# echo "* hard nproc 150" >> /etc/security/limits.conf
~]# echo "# End of file" >> /etc/security/limits.conf
```

2.4.11 Remove suid from certain executables

It a good security measure to remove the suid bit from certain executable files in `/bin` and `/usr/bin`, as these files pose a risk of exploitation due to them running as user 'root' or as group 'root' (or some other group or user). A good place to start would be to find what SUID files you have. This can be done in a number of ways, like so (example for `/bin/su`):

```
~]# find / -perm -4000 -print
~]# chmod -s /bin/su
```

2.4.12 Impose stricter file and folder permissions

To set stricter permissions on newly created files and folders, set the system global umask to 027. This sets permissions 640 (`rw-r---`) for new files and 750 (`rwxr-x---`) for new folders:

```
~]# sed -i 's/umask 022/umask 027/g' /etc/init.d/rc
~]# echo "umask 027" >> /etc/profile
~]# echo "umask 027" >> /etc/bash.bashrc
```

You might want to enable umask 077 on file servers, as it's absence can cause issues on systems where users share files (SAMBA or NFS shares).

2.4.13 Configure DNS resolvers

It is essential that DNS responses are not tampered with in any way to avoid MITM and other kind of attacks and security risks. To secure DNS edit `/etc/systemd/resolved.conf` as below:

```
[Resolve]
DNS="YOUR PRIMARY DNS"
FallbackDNS=8.8.8.8 8.8.4.4
#Domains=
#LLMNR=yes
DNSSEC=allow-downgrade
```

It is also a good practice to enable DNSSEC as in section 2.7.2.

2.4.14 Remove host overrides for users and services

Removing `.rhosts` and `hosts.equiv` from plain and services users' folders helps improve security of a Linux system. To achieve this:

```
~]# for dir in $(awk -F ":" '{print $6}' /etc/passwd); do
    find "$dir" \( -name "hosts.equiv" -o -name ".rhosts" \) \
    -exec rm -f {} \; 2> /dev/null
done

~]# if [[ -f /etc/hosts.equiv ]]; then
    rm /etc/hosts.equiv
fi
```

2.4.15 Configure access with TCP Wrappers

Files `/etc/hosts.allow` and `/etc/hosts.deny` are used to check whether a host has access to certain service on the system. If you are running a desktop system it is likely that you need to set 'ALL' in `/etc/hosts.deny` and 'localhost' in `/etc/hosts.allow`. For a server installation you may need to put additional lines in `/etc/hosts.allow`. To configure access to your system issue the following commands:

```
~]# echo "ALL: LOCAL, 127.0.0.1" >> /etc/hosts.allow
~]# echo "ALL: PARANOID" > /etc/hosts.deny
```

2.4.16 Configure Banners

Configuring Banners is rather more of a cosmetic than a security feature, but it warns the intruder that this machine is monitored and secure. To configure your banner, put your message in the following files: `/etc/issue`, `/etc/motd`, `/etc/issue.net`.

2.4.17 Remove unneeded users

Removing unneeded users reduces the attack surface by eliminating potential rogue logins. To do so issue the following commands:

```
~]# userdel -r username
```

2.4.18 Create new users with non default shell

It is also useful to create new users with `/bin/false` as default shell initially, to prevent rogue users from logging in with automated tools. To achieve this edit `/etc/adduser.conf` and `/etc/useradd.conf` as below:

```
DSHELL=/bin/false
```

2.4.19 Secure NTP

It is essential that the system time remains accurate. Critical services that require accurate date and time include: logging services, login mechanism, cronjobs, cryptography (verification of certificates etc). To maintain time and date, an accurate source is required. For production servers this can be a GPS-powered one or a trusted NTP server. To configure the system NTP client insert into `/etc/systemd/timesyncd.conf` the following:

```
[Time]
NTP=3.ubuntu.pool.ntp.org pool.ntp.org
FallbackNTP=0.ubuntu.pool.ntp.org 1.ubuntu.pool.ntp.org
```

2.4.20 Configure logrotate

If you have extended logging enabled in your system, log files will keep pilling up to the point that `/var/log` gets full and your system becomes unstable. In case `/var` is not in a separate partition, your system may become unresponsive. To remedy that situation one may outsource logging to a central logging system (as splunk in 2.8.1) and enable logrotate. Logrotate prevents log files from filling up all available space by keeping the most recent events in log, according to given rules. To configure logrotate edit the file `/etc/logrotate.conf`. In the script accompanying this paper (see Chapter 4), logrotate is configured to rotate log files every day.

2.4.21 Disable Ctrl+Alt+Delete

In most Linux distributions, pressing CTRL-ALT-DEL takes your system to reboot process. So, it's not a good idea to have this option enabled at least on production servers, if someone by mistake does this.

This is defined in `/etc/inittab file`, if you look closely in that file you will see a line similar to below. By default line is not commented out. This particular key sequence signaling will shut-down the system. To remove this feature execute the following [11]:

```
~]# sudo systemctl mask ctrl-alt-del.target
~]# sudo systemctl daemon-reload
```

If we want the system to catch this specific key-press and just log the event, we must open `/etc/init/control-alt-delete.conf` and modify the corresponding line [1]:

```
exec /sbin/shutdown -r now "Control-Alt-Delete pressed"
```

to

```
exec /usr/bin/logger -p security.info "Control-Alt-Delete pressed"
```

2.5 Account Security

2.5.1 Creating Strong Passwords

For security purposes, the installation program configures the system to use Secure Hash Algorithm 512 (SHA512) and shadow passwords. It is highly recommended that you do not alter these settings. When creating a secure password, the user must remember that long passwords are stronger than short and complex ones. It is not a good idea to create a password of just eight characters, even if it contains digits, special characters and uppercase letters. Password cracking tools, such as John The Ripper, are optimized for breaking such passwords, which are also hard to remember by a person.

The pwmake is a command-line tool for generating random passwords that consist of all four groups of characters – uppercase, lowercase, digits and special characters.

```
~]# apt-get install libpwquality-tools
~]# pwmake --help
```

2.5.2 Forcing Strong Passwords

The `pam_cracklib` module is used to check a password's strength, against a set of rules [4]. To enable using `pam_cracklib` and restrict users from using short or simple passwords, add the following line in the file `/etc/pam.d/common-password`:

```
password required pam_cracklib.so retry=3 maxrepeat=3 minlen=15
dcredit=-1 ucredit=-1 ocredit=-1 lcredit=-1 difok=8
```

Checking accounts for empty passwords is also a good practice [9]. Any account having an empty password means its opened for unauthorized access to anyone on the web and it's a security threat for a Linux server. So, you must make sure all accounts have strong passwords and no one has any authorized access. Empty password accounts are security risks and that can be easily hackable. To check if there were any accounts with empty password, use the following command.

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

Administrator must also restrict users to use old passwords. This is very useful if you want to disallow users to use same old passwords. Open `/etc/pam.d/common-password`

```
~]# vi /etc/pam.d/common-password
```

Add the following line to disallow a user from re-using last 5 passwords.

```
password [success=1 default=ignore] pam_unix.so obscure use_authtok
try_first_pass sha512 remember=5
```

Only last 5 passwords are remember by server. If you tried to use any of last 5 old passwords, you will get an error like.

```
Password has been already used. Choose another.
```

2.5.3 Account Expiration Policy

It very useful for an administrator to manage expiration of user accounts and passwords in the system. To configure account and password auto-expiration settings edit the `/etc/login.defs` as below:

```
LOG_OK_LOGINS yes
PASS_MIN_DAYS 7
PASS_MAX_DAYS 30
```

The lines above configure the system to log the successful logins, deny two password changes in 7 days period and expire the password every 30 days. To disable account in 35 days of inactivity after a password expiration add the following line to `/etc/default/useradd`:

```
INACTIVE=35
```

2.5.4 Account Locking

In Ubuntu 16.04LTS, the `pam_tally` PAM module allows system administrators to lock out user accounts after a specified number of failed attempts. Limiting user login attempts serves mainly as a security measure that aims to prevent possible brute force attacks targeted to obtain a user's account password [4]. With the `pam_tally` module, failed login attempts are stored in `/var/log/faillog`.

To lock out any user after five unsuccessful attempts and unlock that user after 15 minutes, add the following line to `/etc/pam.d/common-auth` file:

```
auth required pam_tally.so file=/var/log/faillog deny=5 unlock_time=900
```

and this line to `etc/pam.d/common-account`

```
account required pam_tally.so reset
```

To enforce a minimal delay of 4 seconds in case of login-failure add the following line to `/etc/pam.d/login`:

```
auth optional pam_faildelay.so delay=4000000
```

2.5.5 Session Locking – Automatic Logout

When the user is logged in as root, an unattended login session may pose a significant security risk. To reduce this risk, you can configure the system to automatically log out idle users after a fixed period of time. To achieve this edit `/etc/systemd/logind.conf` and edit the following lines as bellow:

```
KillUserProcesses=1
KillExcludeUsers=root
IdleAction=lock
IdleActionSec=15min
RemoveIPC=yes
```

The lines above lock session after 15 minutes idle time (the ssh sessions logout) for every user and kills all running processes for that user. The user root is excluded from the process-killing.

Users may need to leave their workstation unattended for a number of reasons during everyday operation. They may also need to lock a virtual console. This can be done using a utility called vlock. To install this utility, execute the following command as root:

```
~]# apt-get install vlock
```

2.5.6 Disallowing Root Access

If an administrator is uncomfortable allowing users to log in as root, the root password should be kept secret, and access to runlevel one or single user mode should be disallowed through boot loader password protection (see section 2.3).

To prevent users from logging in directly as root, the system administrator can set the root account's shell to `/bin/true` in the `/etc/passwd` file. Programs that do not require a shell, such as FTP clients, mail clients, and many setuid programs, are not prevented from accessing the root account: **sudo, FTP clients, Email clients** are such programs.

To further limit access to the root account, administrators can disable root logins at the console by editing the `/etc/securetty` file. This file lists all devices the root user is allowed to log into [4]. If the file does not exist at all, the root user can log in through any communication device on the system, whether via the console or a raw network interface. This is dangerous, because a user can log in to their machine as root via telnet, which transmits the password in plain text over the network. To remedy this situation do:

```
~]# echo '' > /etc/securetty
```

It is also a good measure to lock-out the root account. In that way, a root-clone user with UID 0 cannot gain access to the system.

```
~]# usermod -L root
```

2.6 Risks to Services

Network services can pose many risks for Linux systems. Below is a list of some of the primary issues:

Denial of Service Attacks (DoS) By flooding a service with requests, a denial of service attack can render a system unusable as it tries to log and answer each request.

Distributed Denial of Service Attack (DDoS) A type of DoS attack which uses multiple compromised machines (often numbering in the thousands or more) to direct a coordinated attack on a service, flooding it with requests and making it unusable.

Script Vulnerability Attacks If a server is using scripts to execute server-side actions, as Web servers commonly do, an attacker can target improperly written scripts. These script vulnerability attacks can lead to a buffer overflow condition or allow the attacker to alter files on the system.

Buffer Overflow Attacks Services that want to listen on ports 1 through 1023 must start either with administrative privileges or the CAP_NET_BIND_SERVICE capability needs to be set for them. Once a process is bound to a port and is listening on it, the privileges or the capability are often dropped. If the privileges or the capability are not dropped, and the application has an exploitable buffer overflow, an attacker could gain access to the system as the user running the daemon. Because exploitable buffer overflows exist, crackers use automated tools to identify systems with vulnerabilities, and once they have gained access, they use automated root-kits to maintain their access to the system.

Examples of inherently insecure services include rlogin, rsh, telnet, and vsftpd. All remote login and shell programs (rlogin, rsh, and telnet) should be avoided in favor of SSH. FTP is not as inherently dangerous to the security of the system as remote shells, but **FTP servers must be carefully configured and monitored** to avoid problems. Services that should be carefully implemented and behind a firewall include:

- auth
- nfs-server
- smb and nbm
- yppasswdd
- ypserv
- ypxfrd

2.6.1 Securing NFS

Protect NFS and rpcbind With TCP Wrappers It is important to use TCP Wrappers (see section 2.4.15) to limit which networks or hosts have access to the rpcbind service, since it has no built-in form of authentication. Securing rpcbind only affects NFSv2 and NFSv3 implementations, since NFSv4 no longer requires it. If you plan to implement an NFSv2 or NFSv3 server, then rpcbind is required, and the following section applies. Furthermore, use only IP addresses when limiting access to the service. Avoid using hostnames, as they can be forged by DNS poisoning and other methods.

Protect rpcbind and rpc.mountd with ufw To further restrict access to the rpcbind service, it is a good idea to add ufw rules to the server and restrict access to specific networks. Below are two example ufw commands. The first allows TCP connections to the port 111 (used by the rpcbind service) from the 192.168.1.0/24 network. The second allows TCP connections to the same port from the localhost. All other packets are dropped.

```
~]# ufw allow proto tcp from 192.168.1.0/24 to any port 111
~]# ufw allow proto tcp from 127.0.0.1 to any port 111
```

To similarly limit UDP traffic, use the following command:

```
~]# ufw allow proto udp from 192.168.1.0/24 to any port 111
```

To further restrict access to the `rpc.mountd service`, add ufw ules to the server and restrict access to specific networks. Below are two example ufw commands. The first allows mountd connections from the 192.168.1.0/24 network. The second allows mountd connections from the local host. All other packets are dropped.

```
~]# ufw allow from 192.168.1.0/24 to any port 32767
~]# ufw allow proto tcp from 192.168.1.0/24 to any port 32767
```

The above lines are valid only if mountd runs on port 32767. Change if necessary.

Use Kerberos and configure permissions Under NFSv4 all operations can use Kerberos; under v2 or v3, file locking and mounting still do not use it [4]. **Only export entire file systems**. Exporting a subdirectory of a file system can be a security issue. It is possible in some cases for a client to "break out" of the exported part of the file system and get to unexported parts (see the section on subtree checking in the `exports(5)` man page. Use the `ro` option to export the file system as read-only whenever possible to reduce the number of users able to write to the mounted file system. To enable Kerberos for NFS, issue the following commands:

```
~]# apt-get install krb5-user libpam-krb5
```

In `/etc/default/nfs-kernel-server` we set:

```
NEED_SVCGSSD=yes
```

To export folder `/export/users` to a local network 192.198.1.0/24 we add the following line to `/etc/exports`:

```
/export/users 192.168.1.0/24(rw,sync,no_subtree_check,sec=krb5,anonuid=
65534,anongid=65534)
```

There are three different modes that nfs can operate in with Kerberos, which should be specified in the mount/export options:

- `krb5`: Use Kerberos for authentication only.
- `krb5i`: Use Kerberos for authentication, and include a hash with each transaction to ensure integrity. Traffic can still be intercepted and examined, but modifications to the traffic will be apparent.
- `krb5p`: Use Kerberos for authentication, and encrypt all traffic between the client and server. This is the most secure, but also incurs the most load.

Lastly, your /etc/krb5.keytab file should only be readable by root.

Do not use the no_root_squash option and review existing installations to make sure it is not used. By default, NFS shares change the root user to the nfsnobody user, an unprivileged user account. This changes the owner of all root-created files to nfsnobody, which prevents uploading of programs with the setuid bit set. If no_root_squash is used, remote root users are able to change any file on the shared file system and leave applications infected by Trojans for other users to inadvertently execute.

The secure option is the server-side export option used to restrict exports to "reserved" ports [3]. By default, the server allows client communication only from "reserved" ports (ports numbered less than 1024), because traditionally clients have only allowed "trusted" code (such as in-kernel NFS clients) to use those ports. However, on many networks it is not difficult for anyone to become root on some client, so it is rarely safe for the server to assume that communication from a reserved port is privileged. Therefore the restriction to reserved ports is of limited value; it is better to rely on Kerberos, firewalls, and restriction of exports to particular clients.

It is good practice not to allow users to login to a server. While reviewing the above settings on an NFS server conduct a review of who and what can access the server. Use the nosuid option to disallow the use of a setuid program. The nosuid option disables the set-user-identifier or set-group-identifier bits. This prevents remote users from gaining higher privileges by running a setuid program. Use this option on the client and the server side.

The noexec option disables all executable files on the client. Use this to prevent users from inadvertently executing files placed in the file system being shared. The nosuid and noexec options are standard options for most, if not all, file systems. Use the nodev option to prevent "device-files" from being processed as a hardware device by the client.

The resvport option is a client-side mount option and secure is the corresponding server-side export option. It restricts communication to a "reserved port". The reserved or "well known" ports are reserved for privileged users and processes such as the root user. Setting this option causes the client to use a reserved source port to communicate with the server.

All versions of NFS now support mounting with Kerberos authentication. The mount option to enable this is: sec=krb5. NFSv4 supports mounting with Kerberos using krb5i for integrity and krb5p for privacy protection. These are used when mounting withsec=krb5, but need to be configured on the NFS server. See the man page on exports (5) for more information.

Be careful not to add extraneous spaces when editing this file. For instance, the following line in the /etc/exports file shares the directory /tmp/nfs/ to the host bob.example.com with read/write permissions.

```
/tmp/nfs/ bob.example.com(rw)
```

The following line in the /etc/exports file, on the other hand, shares the same directory to the host bob.example.com with read-only permissions and shares it to the world with read/write permissions due to a single space character after the hostname.

```
/tmp/nfs/ bob.example.com (rw)
```

It is good practice to check any configured NFS shares by using the showmount command to verify what is being shared:

```
~]# showmount -e <hostname>
```

NFSv4 is the default version of NFS for Ubuntu Server 16.04LTS and it only requires port 2049 to be open for TCP. If using NFSv3 then four additional ports are required as explained below.

Configuring Ports for NFSv3 The ports used for NFS are assigned dynamically by `rpcbind`, which can cause problems when creating firewall rules. To simplify this process, use the `/etc/default/nfs-common` and `/etc/default/nfs-kernel-server` files to specify which ports are to be used:

- In file `/etc/default/nfs-kernel-server` input the following line to configure rpc.mountd port:

  ```
  RPCMOUNTDOPTS="-p 32767"
  ```

- In file `/etc/default/nfs-common` input the following line to configure rpc.statd port:

  ```
  STATDOPTS="--port 32765 --outgoing-port 32766"
  ```

- The rpc.lockd port can be assigned by creating the file `/etc/modprobe.d/nfs_local.conf` and put the following lines:

  ```
  options lockd nlm_udpport=32768 nlm_tcpport=32768
  options nfs callback_tcpport=32764
  ```

  The above lines set rpc.lockd port to 32768 TCP/UDP and rpc.nfs-cb port to 32764.

- Finally, to configure rpc.quotad port, create file `/etc/default/quota` and add the following line:

  ```
  RPCRQUOTADOPTS="-p 32769"
  ```

Port numbers specified must not be used by any other service. Configure your firewall to allow the port numbers specified, as well as TCP and UDP port 2049 (NFS).

2.6.2 Securing the Apache HTTP Server

The Apache HTTP Server is one of the most stable and secure services that ships with Ubuntu Server 16.4LTS. A large number of options and techniques are available to secure the Apache HTTP Server — too numerous to delve into deeply here. The following section briefly explains good practices when running the Apache HTTP Server.

2.6.2.1 Apache Security

There are entire books dedicated to apache security. We will hit some of the high-level suggestions here. Detailed help can be found at `http://httpd.apache.org/`. First, verify that your apache subdirectories are all owned by root and have a mod of 755:

```
~]# ls -lah /etc/apache2
total 88K
drwxr-xr-x   8 root root 4.0K Aug  5 18:26 .
drwxr-xr-x 102 root root 4.0K Aug  6 01:20 ..
-rw-r--r--   1 root root 7.0K Mar 19 11:48 apache2.conf
drwxr-xr-x   2 root root 4.0K Aug  5 18:26 conf-available
drwxr-xr-x   2 root root 4.0K Aug  5 18:26 conf-enabled
-rw-r--r--   1 root root 1.8K Mar 19 11:48 envvars
-rw-r--r--   1 root root  31K Mar 19 11:48 magic
drwxr-xr-x   2 root root  12K Aug  5 18:26 mods-available
drwxr-xr-x   2 root root 4.0K Aug  5 18:26 mods-enabled
-rw-r--r--   1 root root  320 Mar 19 11:48 ports.conf
drwxr-xr-x   2 root root 4.0K Aug  5 18:26 sites-available
drwxr-xr-x   2 root root 4.0K Aug  5 18:26 sites-enabled

~]# ls -lah /usr/sbin/*apache*
-rwxr-xr-x 1 root root 631K Jul 15 18:33 /usr/sbin/apache2
-rwxr-xr-x 1 root root 6.3K Mar 19 11:48 /usr/sbin/apache2ctl
lrwxrwxrwx 1 root root   10 Jul 15 18:33 /usr/sbin/apachectl ->
apache2ctl
```

Likewise, your `apache2` binary should be owned by root, with a mod of 511. You can create a web documents subdirectory outside the normal Apache file tree as your DocumentRoot (`/var/www/html`, which is modifiable by other users – since root never executes any files out of there, and shouldn't be creating files in there.

2.6.2.2 Configure settings

Always verify that any scripts running on the system work as intended before putting them into production. Also, ensure that only the root user has write permissions to any directory containing scripts or CGIs. To do this, run the following command as the root user:

```
~]# chown root <directory_name>
~]# chmod 755 <directory_name>
```

Edit the `/etc/apache2/apache2.conf` file as root. Configure the "Listen" option so it lists a port other than 80 or 443. In this example, apache is configured to listen on port 12345:

```
Listen 127.0.0.1:12345
```

System administrators should be careful when using the following configuration options (configured in `/etc/apache2/apache2.conf`):

FollowSymLinks This directive is enabled by default, so be sure to use caution when creating symbolic links to the document root of the web server. For instance, it is a bad idea to provide a symbolic link to `/`.

Indexes This directive is enabled by default, but may not be desirable. To prevent visitors from browsing files on the server, remove this directive.

UserDir The UserDir directive is disabled by default because it can confirm the presence of a user account on the system. However, if you wish to enable user directory browsing on the server, use the following directives:

```
UserDir enabled
UserDir disabled root
```

These directives activate user directory browsing for all user directories other than `/root/`. To add users to the list of disabled accounts, add a space-delimited list of users on the UserDir disabled line.

ServerTokens The ServerTokens directive controls the server response header field which is sent back to clients [9]. It includes various information which can be customized using the following parameters:

- `ServerTokens Full` (default option) — provides all available information (OS type and used modules), for example:

  ```
  Apache/2.0.41 (Unix) PHP/4.2.2 MyMod/1.2
  ```

- `ServerTokens Prod` or `ServerTokens ProductOnly` — provides the following information:

  ```
  Apache
  ```

- `ServerTokens Major` — provides the following information:

  ```
  Apache/2
  ```

- `ServerTokens Minor` — provides the following information:

  ```
  Apache/2.0
  ```

- `ServerTokens Min` or `ServerTokens Minimal` — provides the following information:

  ```
  Apache/2.0.41
  ```

- `ServerTokens OS` — provides the following information:

  ```
  Apache/2.0.41 (Unix)
  ```

It is recommended to use the ServerTokens Prod option so that a possible attacker does not gain any valuable information about your system.

htaccess To prevent users from setting up .htaccess files that can override security features, change the server configuration file to include:

```
<Directory />
AllowOverride None
</Directory>
```

To prevent users from accessing the entire filesystem (starting with the root directory), add the following to your server configuration file:

```
<Directory />
     Order Deny,Allow
     Deny from all
 </Directory>
```

To provide access into individual directories, add the following:

```
<Directory /usr/users/*/public\_html>
     Order Deny,Allow
     Allow from all
 </Directory>
 <Directory /usr/local/httpd>
     Order Deny,Allow
     Allow from all
 </Directory>
```

IncludesNoExec Do not remove the `IncludesNoExec` directive. By default, the Server-Side Includes (SSI) module cannot execute commands. It is recommended that you do not change this setting unless absolutely necessary, as it could, potentially, enable an attacker to execute commands on the system. Server side includes (SSI) create additional risks, since SSI-enabled files can execute any CGI script or program under the permissions of the user and group apache runs as (as configured in `apache2.conf`). To disable the ability to run scripts and programs from SSI pages, make sure that `IncludesNOEXEC` and not `Includes` is enabled in the options directive. Users may still use `<--#include virtual="..."-->` to execute of CGI scripts if these scripts are in directories designated by a `ScriptAlias` directive. Script Aliased CGI is recommended over non-script aliased CGI. Limiting CGI to special directories gives the administrator control over which scripts can be run.

2.6.2.3 Disable Apache2 Unneeded Modules

In certain scenarios, it is beneficial to remove certain apache2 modules to limit the functionality of the HTTP Server. To do so, simply comment out the entire line which loads the module you want to remove in the `/etc/apache2/apache2.conf` file. For example, to remove the proxy module, comment out the following line by prepending it with a hash sign:

```
#LoadModule proxy_module modules/mod_proxy.so
```

Note that the `/etc/apache2/conf-available` directory contains configuration files which are used to load modules as well.

2.6.2.4 Use the Apache mod_security module

The mod_security module runs on most versions of Apache, mod_security allows you to enhance the overall security of your apache web server by providing additional configuration settings within your `apache2.conf` file. These settings allow you to filter/inspect all traffic, or filter/inspect non-static traffic only (DynamicOnly). You can then set the default action for matching requests – for example, displaying a standard error page. In addition, you can specify allowable ASCII values and set restrictions for file uploads. Mod_security also provides much more logging than the default for apache. More information can be found at `http://www.modsecurity.org`.

To install the module issue the following commands:

```
~]# apt-get install libapache2-mod-security2
~]# mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/
modsecurity.conf
```

We need to download and install the latest OWASP ModSecurity Core Rule Set from the project website. We will also activate the default CRS config file modsecurity_crs_10_setup.conf.example.

```
~]# wget -O SpiderLabs-owasp-modsecurity-crs.tar.gz \
https://github.com/SpiderLabs/owasp-modsecurity-crs/tarball/master
~]# mv /etc/modsecurity/modsecurity_crs_10_setup.conf.example \
/etc/modsecurity/modsecurity_crs_10_setup.conf
~]# cp -R SpiderLabs-owasp-modsecurity-crs-*/* /etc/modsecurity/

~]# cd /etc/modsecurity/base_rules
~]# for f in * ; do sudo ln -s /etc/modsecurity/base_rules/$f /etc/
modsecurity/activated_rules/$f ; done
~]# cd /etc/modsecurity/optional_rules
~]# for f in * ; do sudo ln -s /etc/modsecurity/optional_rules/$f /etc
/modsecurity/activated_rules/$f ; done
```

Now enable the module and restart Apache

```
~]# a2enmod security2
~]# service apache2 restart
```

2.6.2.5 Use the Apache mod_evasive module

Mod_evasive is an evasive maneuvers module for Apache that provides evasive action in the event of an HTTP DoS attack or brute force attack. It is also designed to be a detection and network management tool, and can be easily configured to talk to ipchains, firewalls, routers, and more. mod_evasive presently reports abuse via email and syslog facilities. To install the module issue the following commands:

```
~]# apt-get install libapache2-mod-evasive
~]# mkdir /var/log/mod_evasive
~]# chown www-data:www-data /var/log/mod_evasive/
```

Configure the module

```
~]# cat > /etc/apache2/mods-available/mod-evasive.conf << EOF
<ifmodule mod_evasive20.c>
   DOSHashTableSize 3097
   DOSPageCount  2
   DOSSiteCount  50
   DOSPageInterval 1
   DOSSiteInterval  1
   DOSBlockingPeriod  10
   DOSLogDir   /var/log/mod_evasive
   DOSEmailNotify  root@localhost
   DOSWhitelist   127.0.0.1
</ifmodule>
EOF
```

Now enable the module and restart Apache

```
~]# a2enmod evasive
~]# service apache2 restart
```

2.6.2.6 Encryption Protocol Usage Recommendations

You should use HTTPS instead of plain HTTP when secure info, as login credentials are transfered between the client and the server. Below are some guidelines regarding HTTPS encryption protocols.

- SSL v2 Do not use it. Has serious security vulnerabilities.
- SSL v3 Do not use it. Has serious security vulnerabilities.
- TLS v1.0 Use only for interoperability purposes where needed. Has known issues that cannot be mitigated in a way that guarantees interoperability, and thus mitigations are not enabled by default. Does not support modern cipher suites.
- TLS v1.1 Use for interoperability purposes where needed. Has no known issues but relies on protocol fixes that are included in all the TLS implementations in modern Linux distributions. Does not support modern cipher suites.
- TLS v1.2 Recommended version. Supports the modern AEAD cipher suites.

Modern, more secure cipher suites should be preferred to old, insecure ones. Always disable the use of eNULL and aNULL cipher suites, which do not offer any encryption or authentication at all. If at all possible, ciphers suites based on RC4 or HMAC-MD5, which have serious shortcomings, should also be disabled. The same applies to the so-called export cipher suites, which have been intentionally made weaker, and thus are easy to break.

While not immediately insecure, cipher suites that offer less than 128 bits of security should not be considered for their short useful life. Algorithms that use 128 bit of security or more can be expected to be unbreakable for at least several years, and are thus strongly recommended. Note that while 3DES ciphers advertise the use of 168 bits, they actually offer 112 bits of security.

Always give preference to cipher suites that support perfect forward secrecy (PFS), which ensures the confidentiality of encrypted data even in case the server key is compromised. This rules out the fast RSA key exchange, but allows for the use of ECDHE and DHE. Of the two, ECDHE is the faster and therefore the preferred choice. You should also give preference to AEAD ciphers, such as AES-GCM, before CBC-mode ciphers as they are not vulnerable to padding oracle attacks. Additionally, in many cases, AES-GCM is faster than AES in CBC mode, especially when the hardware has cryptographic accelerators for AES.

Note also that when using the ECDHE key exchange with ECDSA certificates, the transaction is even faster than pure RSA key exchange. To provide support for legacy clients, you can install two pairs of certificates and keys on a server: one with ECDSA keys (for new clients) and one with RSA keys (for legacy ones).

When using RSA keys, always prefer key lengths of at least 3072 bits signed by at least SHA-256, which is sufficiently large for true 128 bits of security.

The Apache HTTP Server can use both OpenSSL and NSS libraries for its TLS needs. Depending on your choice of the TLS library, you need to install either the mod_ssl or the mod_nss module. For example, to install the package that provides the OpenSSL mod_nss module, issue the following command as root:

```
~]# apt-get install libapache2-mod-nss
```

The mod_ssl package installs the `/etc/apache2/mods-available/ssl.conf` configuration file, which can be used to modify the TLS-related settings of the Apache HTTP Server. Similarly, the mod_nss package installs the `/etc/apache2/mods-available/nss.conf` configuration file.

When modifying the settings in the `etc/apache2/mods-available/ssl.conf` configuration file, be sure to consider the following three directives at the minimum:

- SSLProtocol Use this directive to specify the version of TLS (or SSL) you want to allow.
- SSLCipherSuite Use this directive to specify your preferred cipher suite or disable the ones you want to disallow.
- SSLHonorCipherOrder Uncomment and set this directive to on to ensure that the connecting clients adhere to the order of ciphers you specified. For example:

  ```
  SSLProtocol all -SSLv2 -SSLv3 SSLCipherSuite HIGH:!aNULL:!MD5

  SSLHonorCipherOrder on
  ```

  Note that the above configuration is the bare minimum.

To configure and use the mod_nss module, modify the following configuration file: `etc/apache2/mods-available/nss.conf`. The mod_nss module is derived from mod_ssl, and as such it shares many features with it, not least the structure of the configuration file, and the directives that are available. Note that the mod_nss directives have a prefix of NSS instead of SSL.

2.6.2.7 Securing with fail2ban

You can further secure ssh logins by employing fail2ban. fail2ban scans log files and bans IPs that show the malicious signs – too many password failures, seeking for exploits, etc. Generally Fail2Ban is used to update firewall rules to reject the IP addresses for a specified amount of time, although any arbitrary other action could also be configured. Out of the box Fail2Ban comes with filters for various services (apache, courier, ftp, ssh, etc). To install fail2ban issue:

```
~]# apt-get install fail2ban
```

After installation edit the configuration file `/etc/fail2ban/jail.local` and create the filter rules as required. To edit the settings open a terminal window and enter:

```
~]# cat > /etc/fail2ban/jail.local << EOF
[apache]

enabled  = true
port     = 12345
filter   = apache2
logpath  = /var/log/auth.log
maxretry = 3
EOF
```

and enable the configuration with:

```
~]# service fail2ban restart
```

2.6.3 Securing SSH

Secure Shell (SSH) is a powerful network protocol used to communicate with another system over a secure channel. In order to enable the use of cryptographic keys for authentication, the `PubkeyAuthentication` configuration directive in the configuration file `/etc/ssh/sshd_config` needs to be set to "yes" [7]. Note that this is the default setting. Set the `PasswordAuthentication` directive to no to disable the possibility of using passwords for logging in.

```
~]$ ssh-keygen -t rsa
```

Using multiple authentication methods, or multi-factor authentication, increases the level of protection against unauthorized access, and as such should be considered when hardening a system to prevent it from being compromised. Users attempting to log in to a system that uses multi-factor authentication must successfully complete all specified authentication methods in order to be granted access.

```
AuthenticationMethods publickey,gssapi-with-mic publickey,keyboard-
interactive
```

An sshd daemon configured using the above `AuthenticationMethods` directive in `/etc/ssh/sshd_config`, only grants access if the user attempting to log in successfully completes either publickey authentication followed by gssapi-with-mic or by keyboard-interactive authentication [4]. Users are encouraged to make use of SSH-2 in order to

maximize the extent to which the SSH protocol protects the authentication and communication for which it is used. Also, the ECDSA (Elliptic Curve Digital Signature Algorithm) offers better performance at the same equivalent symmetric key length. It also generates shorter keys.

It is also advisable to change the default port that the service is listening to, by changing the line in `/etc/ssh/sshd_config`:

```
Listen :12345
```

Simply disabling-locking a user account will not prevent a user from logging into your server remotely if they have previously set up RSA public key authentication. They will still be able to gain shell access to the server, without the need for any password. Remember to check the users home directory for files that will allow for this type of authenticated SSH access, e.g. `.ssh/authorized_keys`.

Restrict SSH access to only user accounts that should have it. For example, you may create a group called "sshlogin" and add the group name as the value associated with the `AllowGroups` variable located in the file `/etc/ssh/sshd_config`. Or you may only let sudoers ssh into the machine.

```
AllowGroups sudo,sshlogin
```

Then add your permitted SSH users to the group "sshlogin", and restart the SSH service.

```
~]# sudo adduser username sshlogin
~]# sudo systemctl restart sshd
```

To prevent root logins via the SSH protocol, edit the SSH daemon's configuration file, `/etc/ssh/sshd_config`, and change the line that reads:

```
PermitRootLogin no
```

To further enhance security you can run ssh on sandbox:

```
UsePrivilegeSeparation sandbox
```

Disable X Forwarding through ssh. That way a rogue user cannot run X applications even if it gains access to the machine. This is mainly required for desktop installations:

```
X11Forwarding no
```

The script accompanying this paper (see Chapter 4) sets numerous other attributes, such as enabled ciphers, max failed logins, max concurrent sessions etc.

You can further secure ssh logins by employing fail2ban. Edit the configuration file `/etc/fail2ban/jail.local` and create the filter rules as required. To edit the settings open a terminal window and enter:

```
~]# cat >> /etc/fail2ban/jail.local << EOF
[ssh]

enabled  = true
port     = ssh
filter   = sshd
logpath  = /var/log/auth.log
maxretry = 3
EOF
```

and enable the configuration with:

```
~]# service fail2ban restart
```

2.6.4 Securing SAMBA

Samba is an important component to seamlessly integrate Linux Servers and Desktops into Active Directory (AD) environments. It can function both as a domain controller (NT4-style) or as a regular domain member (AD or NT4-style) [5]. Samba is comprised of three daemons (smbd, nmbd, and winbindd). Three services (smb, nmb, and winbind) control how the daemons are started, stopped, and other service-related features. These services act as different init scripts.

There are only two types of security modes for Samba, share-level and user-level, which are collectively known as security levels. Share-level security is deprecated and has been removed from Samba. User-level security is the default and recommended setting for Samba. Even if the 'security=user' directive is not listed in the samba configuration file `/etc/samba/smb.conf`, it is used by default.

In domain security mode (user-level), the Samba server has a machine account (domain security trust account) and causes all authentication requests to be passed through to the domain controllers. The Samba server is made into a domain member server by using the following directives in the `/etc/samba/smb.conf` file:

```
[GLOBAL]
security = domain
workgroup = MARKETING
```

If you have an Active Directory environment, it is desirable to join the domain as a native Active Directory member.

```
[GLOBAL]
security = ADS
realm = EXAMPLE.COM
password server = kerberos.example.com
```

With share-level security, the server accepts only a password without an explicit user name from the client. The server expects a password for each share, independent of the user name. If you must use share-level security do not use security = share parameter, but edit the `/etc/samba/smb.conf`, as following [5]:

```
[GLOBAL]
security = user
map to guest = Bad User
username map = /etc/samba/smbusers

[SHARE]
guest ok = yes
```

You must also edit `/etc/samba/smbusers` as below:

```
nobody = guest.
```

2.7 Securing Network Access

2.7.1 Securing Various Network Settings

Disabling Source Routing Source routing is an Internet Protocol mechanism that allows an IP packet to carry information, a list of addresses, that tells a router the path the packet must take. There is also an option to record the hops as the route is traversed. The list of hops taken, the "route record", provides the destination with a return path to the source. This allows the source (the sending host) to specify the route, loosely or strictly, ignoring the routing tables of some or all of the routers. It can allow a user to redirect network traffic for malicious purposes. Therefore, source-based routing should be disabled.

The `accept_source_route` option causes network interfaces to accept packets with the Strict Source Route (SSR) or Loose Source Routing (LSR) option set. The acceptance of source routed packets is controlled by sysctl settings. Issue the following command as root to drop packets with the SSR or LSR option set:

```
~]# /sbin/sysctl -w net.ipv4.conf.all.accept_source_route=0
```

Disabling the forwarding of packets should also be done in conjunction with the above when possible (disabling forwarding may interfere with virtualization). Issue the commands listed below as root, these commands disable forwarding of IPv4 and IPv6 packets on all interfaces:

```
~]# /sbin/sysctl -w net.ipv4.conf.all.forwarding=0
~]# /sbin/sysctl -w net.ipv6.conf.all.forwarding=0
```

These commands disable forwarding of all multicast packets on all interfaces.

```
~]# /sbin/sysctl -w net.ipv4.conf.all.mc_forwarding=0
~]# /sbin/sysctl -w net.ipv6.conf.all.mc_forwarding=0
```

Accepting ICMP redirects has few legitimate uses. Disable the acceptance and sending of ICMP redirected packets unless specifically required. These commands disable acceptance of all ICMP redirected packets on all interfaces.

```
~]# /sbin/sysctl -w net.ipv4.conf.all.accept_redirects=0
~]# /sbin/sysctl -w net.ipv6.conf.all.accept_redirects=0
```

This command disables acceptance of secure ICMP redirected packets on all interfaces.

```
~]# /sbin/sysctl -w net.ipv4.conf.all.secure_redirects=0
```

This command disables acceptance of all IPv4 ICMP redirected packets on all interfaces.

```
~]# /sbin/sysctl -w net.ipv4.conf.all.send_redirects=0
```

Reverse Path Forwarding Reverse Path Forwarding is used to prevent packets that arrived via one interface from leaving via a different interface. When outgoing routes and incoming routes are different, it is sometimes referred to as asymmetric routing. Routers often route packets this way, but most hosts should not need to do this. Exceptions are such applications that involve sending traffic out over one link and receiving traffic over another link from a different service provider. For example, using leased lines in combination with xDSL or satellite links with 3G modems. If such a scenario is applicable to you, then turning off reverse path forwarding on the incoming interface is necessary. In short, unless you know that it is required, it is best enabled as it prevents users spoofing IP addresses from local subnets and reduces the opportunity for DDoS attacks.

Reverse Path Forwarding is enabled by means of the `rp_filter` directive. The sysctl utility can be used to make changes to the running system, and permanent changes can be made by adding lines to the `/etc/sysctl.conf` file. The `rp_filter` option is used to direct the kernel to select from one of three modes. To make a temporary global change, enter the following commands as root:

```
~]# sysctl -w net.ipv4.conf.default.rp_filter=integer
~]# sysctl -w net.ipv4.conf.all.rp_filter=integer
```

where integer is one of the following:

- 0 — No source validation.
- 1 — Strict mode as defined in RFC 3704.
- 2 — Loose mode as defined in RFC 3704.

The setting can be overridden per network interface using the command as follows:

```
~]# sysctl -w net.ipv4.conf.interface.rp_filter=integer
```

To make these settings persistent across reboots, modify the system configuration file `/etc/sysctl.conf`. For example, to change the mode for all interfaces, open the `/etc/sysctl.conf` file with an editor running as the root user and add a line as follows :

```
net.ipv4.conf.all.rp_filter=1
```

Disable Zeroconf Networking Zeroconf network typically occurs when you fail to get an address via DHCP, the interface will be assigned a 169.254.0.0 address. To prevent this [1]:

```
~]# apt-get purge avahi-autoipd
```

Ignore ICMP or Broadcast Request Add following line in `/etc/sysctl.conf` file to ignore ping or broadcast request.

- Ignore ICMP request:

  ```
  net.ipv4.icmp_echo_ignore_all = 1
  ```

- Ignore Broadcast request:

  ```
  net.ipv4.icmp_echo_ignore_broadcasts = 1
  ```

- Log Suspicious Martian Packets / Un-routable Source Addresses – Add following line in `/etc/sysctl.conf` file

  ```
  net.ipv4.conf.all.log_martians = 1
  ```

Turn Off IPv6 If you're not using a IPv6 protocol, then you should disable it because most of the applications or policies do not require IPv6 protocol. Open `/etc/modprobe.d/disablenet.conf` and add the option:

```
ipv6 disable=1
```

Disable IPv6 via sysctl for all interfaces by adding these lines to `/etc/sysctl.conf` [6].

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

Lastly disable IPv6 kernel wide. Edit `/etc/default/grub` to add `ipv6.disable=1` on the line configuring default kernel settings. `GRUB_CMDLINE_LINUX_DEFAULT`. Then update grub configuration by issuing the following command:

```
~]# update-grub
```

Disable Support for RPC IPv6 RPC services like NFS attempt to start using IPv6 even if it's disabled in `/etc/modprobe.d` [1]. To prevent this behaviour open `/etc/netconfig` and comment the following lines:

```
#udp6       tpi_clts      v     inet6    udp     -       -
#tcp6       tpi_cots_ord  v     inet6    tcp     -       -
```

2.7.2 Securing DNS Traffic with DNSSEC

DNSSEC is a set of Domain Name System Security Extensions (DNSSEC) that enables a DNS client to authenticate and check the integrity of responses from a DNS nameserver in order to verify their origin and to determine if they have been tampered with in transit. To enable DNSSEC run the following commands:

```
~]# apt-get install unbound
~]# systemctl enable unbound
~]# systemctl start unbound
```

The unbound daemon allows configuration of local data or overrides using the following directories: The `/etc/unbound/conf.d` directory is used to add configurations for a specific domain name. This is used to redirect queries for a domain name to a specific DNS server. This is often used for sub-domains that only exist within a corporate WAN. The `/etc/unbound/keys.d` directory is used to add trust anchors for a specific domain name. This is required when an internal-only name is DNSSEC signed, but

there is no publicly existing DS record to build a path of trust. Another use case is when an internal version of a domain is signed using a different DNSKEY than the publicly available name outside the corporate WAN. The `/etc/unbound/local.d` directory is used to add specific DNS data as a local override. This can be used to build blacklists or create manual overrides. This data will be returned to clients by unbound, but it will not be marked as DNSSEC signed.

To see whether DNSSEC is working, one can use various command line tools. The best tool to use is the dig command from the bind-utils package. Other tools that are useful are drill from the ldns package and unbound-host from the unbound package. The old DNS utilities nslookup and host are obsolete and should not be used. To send a query requesting DNSSEC data using dig, the option +dnssec is added to the command, for example:

```
~]$ dig +dnssec uom.gr
```

2.8 Securing System further

To protect your System Integrity the following measures must be taken:

2.8.1 Send logs to a centralized logging system

To install splunk forwarder, execute the following as root:

```
~]# curl -RO https://download.splunk.com/products/
universalforwarder/releases/6.4.2/linux/
splunkforwarder-6.4.2-00f5bb3fa822-linux-2.6-amd64.deb

~]# dpkg -i ./splunkforwarder-6.4.2-00f5bb3fa822-linux-2.6-amd64.deb

~]# /opt/splunkforwarder/bin/splunk enable boot-start
~]# /opt/splunkforwarder/bin/splunk add forward-server hostname.domain:
9997
~]# service splunk start
```

where 'hostname.domain' is the fully qualified address or IP of the splunk index server (like indexer.splunk.com). All logs can be analyzed and viewed through the splunk index server web interface.

2.8.2 Install an Antivirus solution

Common Antivirus solutions include [10]:

- BitDefender (commercial)
- ClamAV

To install clamav antivirus. execute the following as root:

```
~]# apt-get install clamav clamdscan clamav-daemon
~]# service clamav-daemon start
~]# freshclam
~]# service clamav-freshclam start
```

To make clamav scan a specific folder daily, create the following file:

```
~]# echo > /etc/cron.daily/manual_clamscan << EOF
#!/bin/bash
SCAN_DIR="/home"
LOG_FILE="/var/log/clamav/manual_clamscan.log"
/usr/bin/clamscan -i -r $SCAN_DIR >> $LOG_FILE
EOF

~]# chmod +x /etc/cron.daily/manual_clamscan
```

2.8.3 Ensure File Integrity

To ensure file integrity of a system, after installation of all required packages, install one of the following [9]:

- Tripwire (last update 2013)
- AIDE (Advanced Intrusion Detection Environment)
- OSSEC
- Samhain

To install AIDE issue the following commands:

```
~]# apt-get install aide-common
~]# sed -i 's/^Checksums =.*/Checksums = sha512/' /etc/aide/aide.conf
~]# aideinit --yes
```

Occasionally you may want to use the package manager abilities to check for changes in installed files:

```
~]# debsums | grep -v OK
```

2.8.4 Install Rootkit detection Software

Available choices as follows [10]:

- Rkhunter (last update 2013)
- OSSEC

To install Rkhunder enter the following commands:

```
~]# apt-get install rkhunter
~]# vi /etc/default/rkhunter
```

Change the following lines to enable rkhunter

```
CRON_DAILY_RUN="yes"
APT_AUTOGEN="yes"
```

Finally run:

```
~]# rkhunter --propupd
```

2.8.5 Install Intrusion Detection Software

Available choices as follows:

- Snort with acidbase
- OSSEC
- fail2ban

As a one-stop solution, OSSEC can be installed to ensure System Integrity. It is advisable to use OSSEC in conjunction with a centralized security manager, as AlienVault OSSIM. To install OSSEC agent execute the following as root user:

```
~]# apt-key adv --fetch-keys http://ossec.wazuh.com/repos/apt/conf/
ossec-key.gpg.key
~]# echo "deb http://ossec.wazuh.com/repos/apt/ubuntu xenial main" >>
/etc/apt/sources.list
~]# apt-get update
~]# apt-get install ossec-hids ossec-hids-agent
```

Export a key for the client in OSSIM and write it down along with the OSSIM server IP address. Edit `/var/ossec/etc/ossec.conf` with the server's IP address. Again in the client machine, execute as root:

```
~]# /var/ossec/bin/ossec-configure
```

follow the instructions and input the client key when requested. In the end start the client.

```
~]# /var/ossec/bin/ossec-control start
```

2.8.6 System audit

The Linux Audit system provides a way to track security-relevant information on your system, based on pre-configured rules, Audit generates log entries to record as much information about the events that are happening on your system as possible. Use cases include the following[4]:

Watching file access Audit can track whether a file or a directory has been accessed, modified, executed, or the file's attributes have been changed. This is useful, for example, to detect access to important files and have an Audit trail available in case one of these files is corrupted.

Monitoring system calls Audit can be configured to generate a log entry every time a particular system call is used. This can be used, for example, to track changes to the system time by monitoring the `settimeofday`, `clock_adjtime`, and other time-related system calls .

Recording commands run by a user Because Audit can track whether a file has been executed, a number of rules can be defined to record every execution of a particular command. For example, a rule can be defined for every executable in the `/bin` directory. The resulting log entries can then be searched by user ID to generate an audit trail of executed commands per user.

Recording security events Audit can be set up to record failed login attempts as well, and provides additional information about the user who attempted to log in.

Searching for events Audit provides the `ausearch` utility, which can be used to filter the log entries and provide a complete audit trail based on a number of conditions.

Running summary report The `aureport` utility can be used to generate, among other things, daily reports of recorded events. A system administrator can then analyze these reports and investigate suspicious activity furthermore.

Monitoring network access - The `iptables` and `ebtables` utilities can be configured to trigger Audit events, allowing system administrators to monitor network access.

2.8.6.1 Installing the audit Packages

In order to use the Audit system, you must have the audit packages installed on your system. If you do not have these packages installed, execute the following command as the root user to install them:

```
~]# apt-get install auditd
```

2.8.6.2 Preconfigured Rules Files

In the `/usr/share/doc/auditd/examples/` directory, the audit package provides a set of pre-configured rules files according to various certification standards:

To use these config files, create a backup of your original `/etc/audit/audit.rules` file and copy the configuration file of your choice over the `/etc/audit/audit.rules` file and enable the service:

```
~]# cp /etc/audit/audit.rules /etc/audit/audit.rules_backup
~]# cp /usr/share/doc/auditd/examples/stig.rules.gz /etc/audit/
audit.rules.gz
~]# gunzip /etc/audit/audit.rules.gz
~]# systemctl enable auditd
```

3 Use System Hardening Utilities

To assess the applied security settings or to harden the Linux system even more, the following utilities can be used.

3.1 BastilleLinux - The Bastille Linux Security Hardening Tool

Bastille Linux is able to check and report the level of security in the system. It is also able to log security issues and optionally correct unsafe configurations.

The package is not included in any repository since Ubuntu 12.04LTS. One way to get it is to install prerequisites and get the deb package from `http://packages.ubuntu.com`.

```
~]# apt-get install libcurses-perl
~]# curl -RO http://gr.archive.ubuntu.com/ubuntu/pool/universe/b/
bastille/bastille_3.0.9-13ubuntu1_all.deb
~]# dpkg -i ./bastille_3.0.9-13ubuntu1_all.deb
```

The package includes a user interface, and configuration engine. The primary user interface is an X interface using the Perl/Tk system, and there is also a Curses-based text interface as well. You may use Bastille Linux in two primary modes:

- Interactively: Allows Bastille Linux to ask you a series of questions, with explanations of the concept involved and hardens your system according to your answers to those questions.
- Non-Interactively: You may also edit a configuration file which may then be used with Bastille Linux to enforce the security hardening measures. This is a good way to automate the hardening of several servers, for example.

The changes Bastille Linux can make the system can potentially render parts it inoperative, or have other adverse affects. The administrator should have a very good understanding of what will occur for every change allows Bastille Linux to make, and understand any potential ramifications which may arise later from those changes.

Apart from that, active development on bastille-linux (bastille-unix since 2007 [2]) ceased back in 2008. So bastille-linux can mainly be used as a reporting tool by running it like this:

```
~]# bastille --assess
```

or

```
~]# bastille --assessnobrowser
```

The second invocation runs Assessment mode without report display. Bastille will create three versions of the report, which places in `/var/log/Bastille/Assessment`:

- audit-report.html Full HTML version with javascript
- audit-report.txt Text-only version
- audit-log.txt Machine-parseable text version)

This report will include details and a score.

3.2 Auditing System Settings with SCAP Security Guide

The SCAP Security Guide (SSG) project's package, scap-security-guide, contains the latest set of security polices and known vulnerabilities for Linux systems. To install the SCAP Security Guide package on your system, run the following command as root:

```
~]# apt-get install libopenscap8
```

To inspect the security content available with scap-security-guide, use the oscap info module. Ubuntu does not have yet ready-to-deploy openscap definitions, so the default debian ones can be used.

```
~]$ curl -RO https://www.debian.org/security/oval/oval-definitions-2016.
xml
~]$ oscap info ./oval-definitions-2016.xml
```

The output of this command is an outline of the known vulnerability definitions. To audit your system settings, run the appropriate evaluation command. For example, the following command is used to assess the given system against a draft SCAP profile for Debian 8:

```
~]$ oscap oval eval \
--results ssg-debian8-oval-result.xml \
--report ssg-debian8-report.html \
./oval-definitions-2016.xml
```

After applying the script accompanying this paper (see Chapter 4) and running the assessment, resulting report shows no vulnerabilities.

4 System Hardening Bash Script

To enforce the aforementioned security settings and policies, a bash script was developed. The script makes the following changes to the system:

- Installs needed packages
- Sets Auto-installation of security updates via cronjob
- Enforces AppArmor
- Secures the bootloader
- Disables AppPort
- Disables unwanted services
- Disables unneeded kernel modules
- Disables unneeded file systems
- Secures Mounts
- Disables unwanded and potentially dangerous protocols
- Disables creation of user and system core dumps
- Configures sysctl parameters
- Configures user security limits
- Removes suid bits from certain executables
- Sets global system umask to 027
- Locks up CTRL+ALT+DEL
- Disables root logins - Locks out root account
- Secures user and services hosts files
- Configures Banners
- Configures TCP Wrappers
- Applies account password policy
- Removes unneeded users
- Secures Apache Server
- Secure NFS Server
- Secures SSHD server

- Locks up cronjobs for users other than root
- Configures UFW
- Disables IPV6
- Configures secure DNS resolvers
- Secures NTP Client
- Configures logrotate
- Enforces auditd rules
- Enables RKHUNTER
- Enables CLAMAV
- Sets AIDE

Tasks that required interactive input from the user, such as keys from OSSIM Server or setting Splunk forwarder to an indexing server, where skipped intentionally, but the user can easily set them up by copy-pasting the commands from the appropriate sections, to the system console.

4.1 System Evaluation and Assessment

Finally after employing the scripts, the system was assessed using openscap profiles for debian8, as outlined in Section 3.2, with no vulnerabilities found.

The system was also scanned using Nessus and Greenbone Security Assistant and scored 85-90% and had minor info remarks. The developed script can be found in Appendix A of the current paper.

5 Conclusions

In this paper we tried to outline the most important security recommendations for a basic installation of an Ubuntu Linux system. The aforementioned list is not exhaustive and should be considered as the bare minimum. A script was also developed to enforce the security settings, discussed in his paper, in an automated way. Other security related settings may be employed in accordance to specific environment and needs of the networking infrastructure.

Appendices

A Bash Script source code

```
#!/usr/bin/env bash
################################################################################
################################################################################
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.

# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <http://www.gnu.org/licenses/>.
################################################################################
################################################################################
# Script tries to harden a default setup of Ubuntu Server 16.04LTS
# All configuration parameters lie in the begining of the file,
# in terms of global variables (Capitalized). Fell free to change
# the configuration according to your needs. Only change if you know
# what you're doing..You have been warned!
################################################################################
# CONFIGURATION STARTS HERE
################################################################################
ADDUSER='/etc/adduser.conf'
APACHE2DFILE='/etc/apache2/conf-available/custom_secure.conf'
AUDITDCONF='/etc/audit/auditd.conf'
AUDITRULES='/etc/audit/rules.d/hardening.rules'
COMMONPASSWD='/etc/pam.d/common-password'
COMMONACCOUNT='/etc/pam.d/common-account'
COMMONAUTH='/etc/pam.d/common-auth'
COREDUMPCONF='/etc/systemd/coredump.conf'
DEBIAN_FRONTEND='noninteractive'
DEFAULTGRUB='/etc/default/grub'
DISABLEFS='/etc/modprobe.d/disablemnt.conf'
DISABLEMOD='/etc/modprobe.d/disablemod.conf'
DISABLENET='/etc/modprobe.d/disablenet.conf'
EXPECT='/usr/bin/expect'
FW_LOCAL='127.0.0.1'
GRUB_PASSPHRASE='password'
GRUB_SUPERUSER='myuser'
JOURNALDCONF='/etc/systemd/journald.conf'
LIMITSCONF='/etc/security/limits.conf'
LOGINDCONF='/etc/systemd/logind.conf'
LOGINDEFS='/etc/login.defs'
LOGROTATE='/etc/logrotate.conf'
MKPASSWD='/usr/bin/grub-mkpasswd-pbkdf2'
MOD='bluetooth firewire-core net-pf-31 soundcore thunderbolt usb-midi'
MODSEC='/etc/modsecurity/modsecurity.conf'
PACKAGES="acct aide-common apache2 apparmor-profiles apparmor-utils auditd \
clamav clamdscan clamav-daemon debsums expect fail2ban git haveged \
libapache2-mod-security2 libapache2-mod-evasive libpam-cracklib \
libpam-tmpdir nfs-kernel-server openssh-server rkhunter samba $VM"
PAMLOGIN='/etc/pam.d/login'
RESOLVEDCONF='/etc/systemd/resolved.conf'
RKHUNTERCONF='/etc/default/rkhunter'
SECURITYACCESS='/etc/security/access.conf'
SERVER='Y'
SSHDFILE='/etc/ssh/sshd_config'
SSH_GROUPS='sudo'
SYSCTL='/etc/sysctl.conf'
SYSTEMCONF='/etc/systemd/system.conf'
TERM='linux'
TIMESYNCD='/etc/systemd/timesyncd.conf'
UFWDEFAULT='/etc/default/ufw'
USERADD='/etc/default/useradd'
```

```
USERCONF='/etc/systemd/user.conf'
UNW_PROT='dccp sctp rds tipc'
UNW_SERVICES='rpcbind'
UNW_FS='cramfs freevxfs jffs2 hfs hfsplus squashfs udf vfat'
VERBOSE='Y'
################################################################################
# CONFIGURATION ENDS HERE
# Do not change anything below this line!
################################################################################
# Prepare ENV (OK)
export TERM
export DEBIAN_FRONTEND
################################################################################
# Check that we have bare minimum..(OK)
if [ $EUID -ne 0 ]; then
    echo "This script must be run with root privileges."
    echo
    exit 1
fi

if ! lsb_release -i | grep 'Ubuntu'; then
    echo "Unsupported Linux distribution. Only Ubuntu Supported"
    echo
    exit 1
fi

if ! ps -p $$ | grep -i bash; then
    echo "Please install bash to continue.."
    echo
    exit 1
fi

if ! [ -x "$(which systemctl)" ]; then
echo "systemctl required. Unsupported setup.."
    echo
exit 1
fi

if ! test -f "$UFWDEFAULT"; then
    echo "$UFWDEFAULT firewall config file not found."

    if ! dpkg -l | grep ufw 2> /dev/null 1>&2; then
     echo 'Please install ufw package to continue.'
    fi
    exit 1
fi

echo "End of Pre-Flight checks.."
# End of Pre-Flight checks..
################################################################################
# Set paths(OK)
echo "Setting paths..."

sed -i 's/PATH=.*/PATH=\"\/usr\/local\/bin:\/usr\/bin:\/bin"/' /etc/environment

cat > /etc/profile.d/initpath.sh <<EOF
#!/bin/bash

if [[ $EUID -eq 0 ]];
  then
    export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
  else
    export PATH=/usr/local/bin:/usr/bin:/bin
fi
EOF

chown root:root /etc/profile.d/initpath.sh
chmod 0644 /etc/profile.d/initpath.sh
################################################################################
# Install needed packages(OK)
if [[ $VERBOSE == "Y" ]]; then
    APT_ENV='-y'
else
```

```
    APT_ENV='-qq -y'
fi

APT="apt-get $APT_ENV"

echo "Updating the package index files..."
$APT update

echo "Upgrading installed packages..."
$APT upgrade

echo "Installing base packages..."

# Are we running a VM?
if dmidecode -q --type system | grep -i vmware; then
    VM="open-vm-tools"
fi

if dmidecode -q --type system | grep -i virtualbox; then
    VM="virtualbox-guest-dkms virtualbox-guest-utils"
fi

for deb in $PACKAGES; do
    $APT install --no-install-recommends "$deb"
done
################################################################################
# Install security updates via cronjob(OK)
cat > /etc/cron.weekly/apt-security-updates <<EOF
echo "**************" >> /var/log/apt-security-updates
date >> /var/log/apt-security-updates
aptitude update >> /var/log/apt-security-updates
aptitude safe-upgrade -o Aptitude::Delete-Unused=false --assume-yes --target-release \
`lsb_release -cs`-security >> /var/log/apt-security-updates
echo "Security updates (if any) installed"
EOF

chmod +x /etc/cron.weekly/apt-security-updates

cat > /etc/logrotate.d/apt-security-updates <<EOF
/var/log/apt-security-updates {
        rotate 2
        weekly
        size 250k
        compress
        notifempty
}
EOF
################################################################################
# Enforce AppArmor(OK)
echo "Enforcing apparmor profiles..."

find /etc/apparmor.d/ -maxdepth 1 -type f -exec aa-enforce {} \;
aa-complain /etc/apparmor.d/usr.sbin.rsyslogd
################################################################################
# Secure bootloader(OK)
echo "Securing bootloader..."

expect_script(){
    cat <<EOF
    log_user 0
    spawn  ${MKPASSWD}
    sleep 0.33
    expect  "Enter password: " {
        send "$GRUB_PASSPHRASE"
        send "\n"
    }
    sleep 0.33
    expect "Reenter password: " {
        send "$GRUB_PASSPHRASE"
        send "\n"
    }
    sleep 0.33
    expect eof {
```

```
        puts "\$expect_out(buffer)"
    }
    exit 0
EOF
}

if [ -n "$GRUB_PASSPHRASE" ]; then
    sed -i 's/^GRUB_CMDLINE_LINUX=.*/GRUB_CMDLINE_LINUX="--users $GRUB_SUPERUSER"/' "$DEFAULTGRUB"
    echo "set superusers=$GRUB_SUPERUSER" >> /etc/grub.d/40_custom
    GRUB_PASS=$(expect_script "$1" | $EXPECT | sed -e "/^\r$/d" -e "/^$/d" -e "s/.* \(.*\)/\1/")
    echo "password_pbkdf2 $GRUB_SUPERUSER $GRUB_PASS" >> /etc/grub.d/40_custom
    echo 'export superusers' >> /etc/grub.d/40_custom
fi
################################################################################
# Disable AppPort(OK)
echo "Disabling apport"

sed -i 's/enabled=.*/enabled=0/' /etc/default/apport
systemctl mask apport.service

if [[ $VERBOSE == "Y" ]]; then
    systemctl status apport.service --no-pager
    echo
fi
################################################################################
# Disable unwanted services(OK)
echo "Disabling unwanted services"

for disable in $UNW_SERVICES; do
    systemctl disable $disable
done
################################################################################
# Disable unneeded kernel modules(OK)
echo "Disabling unwanted kernel modules"

for disable in $MOD; do
    if ! grep -q "$disable" "$DISABLEMOD" 2> /dev/null; then
        echo "install $disable /bin/true" >> "$DISABLEMOD"
    fi
done

if [[ $SERVER == "Y" ]]; then
    echo "install usb-storage /bin/true" >> "$DISABLEMOD"
fi
################################################################################
# Disable unneeded file systems(OK)
echo "Disabling unneeded file systems"
for disable in $UNW_FS; do
    if ! grep -q "$disable" "$DISABLEFS" 2> /dev/null; then
        echo "install $disable /bin/true" >> "$DISABLEFS"
    fi
done
################################################################################
# Securing Mounts(OK)
echo "Securing mounts"

cat > /etc/systemd/system/tmp.mount <<EOF
# /etc/systemd/system/default.target.wants/tmp.mount -> ../tmp.mount

[Unit]
Description=Temporary Directory
Documentation=man:hier(7)
Before=local-fs.target

[Mount]
What=tmpfs
Where=/tmp
Type=tmpfs
Options=mode=1777,strictatime,nosuid,nodev
EOF

sed -i '/floppy/d' /etc/fstab
```

```
if [ -e /etc/systemd/system/tmp.mount ]; then
    sed -i '/^\/tmp/d' /etc/fstab

    for t in $(mount | grep -e "[[:space:]]/tmp[[:space:]]" -e \
    "[[:space:]]/var/tmp[[:space:]]" -e "[[:space:]]/dev/shm[[:space:]]" \ | awk '{print $3}'); do
        umount "$t"
    done

    sed -i '/[[:space:]]\/tmp[[:space:]]/d' /etc/fstab

    ln -s /etc/systemd/system/tmp.mount /etc/systemd/system/default.target.wants/tmp.mount
    sed -i 's/Options=.*/Options=mode=1777,strictatime,nodev,nosuid/' /etc/systemd/system/tmp.mount

    cp /etc/systemd/system/tmp.mount /etc/systemd/system/var-tmp.mount
    sed -i 's/\/tmp/\/var\/tmp/g' /etc/systemd/system/var-tmp.mount
    ln -s /etc/systemd/system/var-tmp.mount /etc/systemd/system/default.target.wants/var-tmp.mount

    cp /etc/systemd/system/tmp.mount /etc/systemd/system/dev-shm.mount
    sed -i 's/\/tmp/\/dev\/shm/g' /etc/systemd/system/dev-shm.mount
    ln -s /etc/systemd/system/dev-shm.mount /etc/systemd/system/default.target.wants/dev-shm.mount
    sed -i 's/Options=.*/Options=mode=1777,strictatime,noexec,nosuid/' /etc/systemd/system/dev-shm.mount

    chmod 0644 /etc/systemd/system/tmp.mount
    chmod 0644 /etc/systemd/system/var-tmp.mount
    chmod 0644 /etc/systemd/system/dev-shm.mount

    systemctl daemon-reload
else
    echo '/etc/systemd/system/tmp.mount was not found.'
fi
################################################################################
# Disable unwanded and potentially dangerous protocols(OK)
echo "Disabling unwanded protocols.."
for disable in $UNW_PROT; do
    if ! grep -q "$disable" "$DISABLENET" 2> /dev/null; then
        echo "install $disable /bin/true" >> "$DISABLENET"
    fi
done
################################################################################
# Disable core dumps(OK)
echo "Disabling coredump"
sed -i 's/^#DumpCore=.*/DumpCore=no/' "$SYSTEMCONF"
sed -i 's/^#CrashShell=.*/CrashShell=no/' "$SYSTEMCONF"
sed -i 's/^#DefaultLimitCORE=.*/DefaultLimitCORE=0/' "$SYSTEMCONF"
sed -i 's/^#DefaultLimitNOFILE=.*/DefaultLimitNOFILE=100/' "$SYSTEMCONF"
sed -i 's/^#DefaultLimitNPROC=.*/DefaultLimitNPROC=100/' "$SYSTEMCONF"

sed -i 's/^#DefaultLimitCORE=.*/DefaultLimitCORE=0/' "$USERCONF"
sed -i 's/^#DefaultLimitNOFILE=.*/DefaultLimitNOFILE=100/' "$USERCONF"
sed -i 's/^#DefaultLimitNPROC=.*/DefaultLimitNPROC=100/' "$USERCONF"

systemctl daemon-reload

if test -f "$COREDUMPCONF"; then
    echo "Fixing Systemd/coredump.conf"
    sed -i 's/^#Storage=.*/Storage=none/' "$COREDUMPCONF"

    systemctl restart systemd-journald

    if [[ $VERBOSE == "Y" ]]; then
        systemctl status systemd-journald --no-pager
        echo
    fi
fi
################################################################################
# Configure sysctl parameters(OK)
echo "Configuring sysctl parameters..."

cat > $SYSCTL <<EOF
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
```

```
#
# Documentation:
# "Draft Red Hat 7 STIG Version 1, Release 0.1"
# "Guide to the Secure Configuration of Red Hat Enterprise Linux 5"
# "CIS Ubuntu 12.04 LTS Server Benchmark v1.0.0"
# https://wiki.ubuntu.com/Security/Features
#

fs.protected_hardlinks = 1
fs.protected_symlinks = 1
fs.suid_dumpable = 0
kernel.core_uses_pid = 1
kernel.kptr_restrict = 2
kernel.panic = 60
kernel.panic_on_oops = 60
kernel.perf_event_paranoid = 2
kernel.randomize_va_space = 2
kernel.sysrq = 0
kernel.yama.ptrace_scope = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.conf.default.log_martians = 1
net.ipv4.conf.default.rp_filter= 1
net.ipv4.conf.default.secure_redirects = 0
net.ipv4.conf.default.send_redirects = 0
net.ipv4.icmp_echo_ignore_broadcasts = 1
net.ipv4.icmp_ignore_bogus_error_responses = 1
net.ipv4.ip_forward = 0
net.ipv4.tcp_max_syn_backlog = 2048
net.ipv4.tcp_rfc1337 = 1
net.ipv4.tcp_synack_retries = 2
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_syn_retries = 5
net.ipv4.tcp_timestamps = 0
net.ipv4.conf.all.forwarding = 0
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
net.ipv6.conf.all.use_tempaddr = 2
net.ipv6.conf.all.accept_ra = 0
net.ipv6.conf.all.accept_redirects = 0
net.ipv6.conf.default.accept_ra = 0
net.ipv6.conf.default.accept_ra_defrtr = 0
net.ipv6.conf.default.accept_ra_pinfo = 0
net.ipv6.conf.default.accept_redirects = 0
net.ipv6.conf.default.autoconf = 0
net.ipv6.conf.default.dad_transmits = 0
net.ipv6.conf.default.max_addresses = 1
net.ipv6.conf.default.router_solicitations = 0
net.ipv6.conf.default.use_tempaddr = 2
net.ipv6.conf.eth0.accept_ra_rtr_pref = 0
net.ipv6.conf.all.forwarding = 0
net.netfilter.nf_conntrack_max = 2000000
net.netfilter.nf_conntrack_tcp_loose = 0
EOF

sed -i '/net.ipv6.conf.eth0.accept_ra_rtr_pref/d' "$SYSCTL"

for i in $(arp -n -a | awk '{print $NF}' | sort | uniq); do
    echo "net.ipv6.conf.$i.accept_ra_rtr_pref = 0" >> "$SYSCTL"
done

echo 1048576 > /sys/module/nf_conntrack/parameters/hashsize

chmod 0600 "$SYSCTL"
systemctl restart systemd-sysctl
```

```
if [[ $VERBOSE == "Y" ]]; then
    systemctl status systemd-sysctl --no-pager
    echo
fi
################################################################################
# Configure user security limits(OK)
echo "Setting limits..."

sed -i 's/^# End of file*//' "$LIMITSCONF"
echo "* hard maxlogins 10" >> "$LIMITSCONF"
echo "* hard core 0" >> "$LIMITSCONF"
echo "* soft nproc 100" >> "$LIMITSCONF"
echo "* hard nproc 150" >> "$LIMITSCONF"
echo "# End of file" >> "$LIMITSCONF"
################################################################################
# Remove suid bits(OK)
echo "Removing suid bits"

for p in /bin/fusermount /bin/mount /bin/ping /bin/ping6 /bin/su /bin/umount \
         /usr/bin/bsd-write /usr/bin/chage /usr/bin/chfn /usr/bin/chsh \
         /usr/bin/mlocate /usr/bin/mtr /usr/bin/newgrp /usr/bin/pkexec \
         /usr/bin/traceroute6.iputils /usr/bin/wall /usr/sbin/pppd;
do
    if [ -e "$p" ]; then
        oct=$(stat -c "%a" $p |sed 's/^4/0/')
        ug=$(stat -c "%U %G" $p)
        dpkg-statoverride --remove $p 2> /dev/null
        dpkg-statoverride --add "$ug" "$oct" $p 2> /dev/null
        chmod -s $p
    fi
done

for SHELL in $(cat /etc/shells); do
    if [ -x "$SHELL" ]; then
        chmod -s "$SHELL"
    fi
done
################################################################################
# Set umask(OK)
echo "Setting umask..."
sed -i 's/umask 022/umask 027/g' /etc/init.d/rc

if ! grep -q -i "umask" "/etc/profile" 2> /dev/null; then
    echo "umask 027" >> /etc/profile
fi

if ! grep -q -i "umask" "/etc/bash.bashrc" 2> /dev/null; then
    echo "umask 027" >> /etc/bash.bashrc
fi
################################################################################
# Lock up CTRL+ALT+DEL(OK)
echo "Lockup Ctrl-alt-delete"

systemctl mask ctrl-alt-del.target

if [[ $VERBOSE == "Y" ]]; then
    systemctl status ctrl-alt-del.target --no-pager
    echo
fi
################################################################################
# Disable root logins(OK)
echo "Disabling root logins..."

sed -i 's/^#+ : root : 127.0.0.1/+ : root : 127.0.0.1/' "$SECURITYACCESS"
echo '' > /etc/securetty
################################################################################
# Secure user and services host files(OK)
echo "Securing .rhosts and hosts.equiv"

for dir in $(awk -F ":" '{print $6}' /etc/passwd); do
    find "$dir" \( -name "hosts.equiv" -o -name ".rhosts" \) -exec rm -f {} \; 2> /dev/null
done
```

```
if [[ -f /etc/hosts.equiv ]]; then
    rm /etc/hosts.equiv
fi
################################################################################
# Configure Banners(OK)
echo "Configuring Banners..."

for f in /etc/issue /etc/issue.net /etc/motd; do
    TEXT="\nAuthorized users only. All activity may be monitored and reported.\n"
    echo -e "$TEXT" > $f
done
################################################################################
# Configure TCP Wrappers(OK)
echo "Configuring TCP Wrappers"

if [[ $SERVER == "Y" ]]; then
    echo "sshd : ALL : ALLOW" > /etc/hosts.allow
fi
echo "ALL: LOCAL, 127.0.0.1" >> /etc/hosts.allow
echo "ALL: PARANOID" > /etc/hosts.deny
chmod 644 /etc/hosts.allow
chmod 644 /etc/hosts.deny
################################################################################
# Configure logindefs(OK)
echo "Configuring logindefs..."

sed -i 's/^.*LOG_OK_LOGINS.*/LOG_OK_LOGINS\t\tyes/' "$LOGINDEFS"
sed -i 's/^UMASK.*/UMASK\t\t077/' "$LOGINDEFS"
sed -i 's/^PASS_MIN_DAYS.*/PASS_MIN_DAYS\t\t7/' "$LOGINDEFS"
sed -i 's/^PASS_MAX_DAYS.*/PASS_MAX_DAYS\t\t30/' "$LOGINDEFS"
sed -i 's/DEFAULT_HOME.*/DEFAULT_HOME no/' "$LOGINDEFS"
sed -i 's/USERGROUPS_ENAB.*/USERGROUPS_ENAB no/' "$LOGINDEFS"
sed -i 's/^# SHA_CRYPT_MAX_ROUNDS.*/SHA_CRYPT_MAX_ROUNDS\t\t10000/' "$LOGINDEFS"
################################################################################
# Configure loginconf(OK)
echo "Configuring logind..."

sed -i 's/^#KillUserProcesses=no/KillUserProcesses=1/' "$LOGINDCONF"
sed -i 's/^#KillExcludeUsers=root/KillExcludeUsers=root/' "$LOGINDCONF"
sed -i 's/^#IdleAction=ignore/IdleAction=lock/' "$LOGINDCONF"
sed -i 's/^#IdleActionSec=30min/IdleActionSec=15min/' "$LOGINDCONF"
sed -i 's/^#RemoveIPC=yes/RemoveIPC=yes/' "$LOGINDCONF"

systemctl daemon-reload
################################################################################
# Locking new user shell by default(OK)
echo "Setting new user settings..."

sed -i 's/DSHELL=.*/DSHELL=\/bin\/false/' "$ADDUSER"
sed -i 's/SHELL=.*/SHELL=\/bin\/false/' "$USERADD"
sed -i 's/^# INACTIVE=.*/INACTIVE=35/' "$USERADD"
################################################################################
# Apply account password policy(OK)
echo "Applying Account password Policy..."

sed -i 's/^password[\t].*.pam_cracklib.*/password\trequired\t\t\tpam_cracklib.so \
retry=3 maxrepeat=3 minlen=15 dcredit=-1 ucredit=-1 ocredit=-1 lcredit=-1 difok=8/' "$COMMONPASSWD"
sed -i 's/try_first_pass sha512.*/try_first_pass sha512 remember=5/' "$COMMONPASSWD"
sed -i 's/nullok_secure//' "$COMMONAUTH"

if ! grep tally "$COMMONAUTH"; then
    sed -i '/^$/a auth required pam_tally.so file=/var/log/faillog deny=5 unlock_time=900' "$COMMONAUTH"
    sed -i '/pam_tally.so/d' "$COMMONACCOUNT"
    echo 'account required pam_tally.so reset' >> "$COMMONACCOUNT"
fi

sed -i 's/pam_lastlog.so.*/pam_lastlog.so showfailed/' "$PAMLOGIN"
sed -i 's/delay=.*/delay=4000000/' "$PAMLOGIN"
################################################################################
# Lock out root account(OK)
echo "Locking out root account..."

usermod -L root
```

```
if [[ $VERBOSE == "Y" ]]; then
    passwd -S root
    echo
fi
################################################################################
# Remove unneeded users(OK)
echo "Removing unwanted users"

for users in games gnats irc list news uucp; do
    userdel -r "$users" 2> /dev/null
done
################################################################################
# Secure Apache(OK)
chmod 511 /usr/sbin/apache2
chown 0:0 /usr/sbin/apache2
chattr +i /etc/apache2/apache2.conf

a2dismod autoindex

cat > "$APACHE2DFILE" <<EOF
<Directory />
  Order Deny,Allow
  Deny from all
  Options None
  AllowOverride None
</Directory>

<Directory /var/www/>
    Order Allow,Deny
    Allow from all
    Options +FollowSymLinks -Indexes +IncludesNoExec
    AllowOverride None
    Require all granted
</Directory>

ServerSignature Off
ServerTokens Prod
TraceEnable Off
EOF

a2enconf custom_secure

# Enable mod_security
mv /etc/modsecurity/modsecurity.conf-recommended $MODSEC
sed -i 's/.*SecRuleEngine.*/SecRuleEngine On/' "$MODSEC"
sed -i 's/.*SecRequestBodyLimit.*/SecRequestBodyLimit 16384000/' "$MODSEC"
sed -i 's/.*SecRequestBodyInMemoryLimit.*/SecRequestBodyInMemoryLimit 16384000/' "$MODSEC"

wget -O /tmp/SpiderLabs-owasp-modsecurity-crs.tar.gz \
https://github.com/SpiderLabs/owasp-modsecurity-crs/tarball/master
cd /tmp
tar -zxvf ./SpiderLabs-owasp-modsecurity-crs.tar.gz
cp -R SpiderLabs-owasp-modsecurity-crs-*/* /etc/modsecurity/
rm -R SpiderLabs-owasp-modsecurity-crs-*
mv /etc/modsecurity/modsecurity_crs_10_setup.conf.example /etc/modsecurity/modsecurity_crs_10_setup.conf

cd /etc/modsecurity/base_rules
for f in * ; do sudo ln -s /etc/modsecurity/base_rules/$f /etc/modsecurity/activated_rules/$f ; done
cd /etc/modsecurity/optional_rules
for f in * ; do sudo ln -s /etc/modsecurity/optional_rules/$f /etc/modsecurity/activated_rules/$f ; done

cat > /etc/apache2/mods-available/security2.conf <<EOF
<IfModule security2_module>
        # Default Debian dir for modsecurity's persistent data
        SecDataDir /var/cache/modsecurity

        # Include all the *.conf files in /etc/modsecurity.
        # Keeping your local configuration in that directory
        # will allow for an easy upgrade of THIS file and
        # make your life easier
        IncludeOptional /etc/modsecurity/*.conf
        IncludeOptional /etc/modsecurity/activated_rules/*.conf
```

```
</IfModule>
EOF

# Enable mod_evasive
mkdir /var/log/mod_evasive
chown www-data:www-data /var/log/mod_evasive/

cat > /etc/apache2/mods-available/evasive.conf <<EOF
<ifmodule mod_evasive20.c>
   DOSHashTableSize 3097
   DOSPageCount  2
   DOSSiteCount  50
   DOSPageInterval 1
   DOSSiteInterval  1
   DOSBlockingPeriod  10
   DOSLogDir   /var/log/mod_evasive
   DOSEmailNotify  root@localhost
   DOSWhitelist   127.0.0.1
</ifmodule>
EOF

a2enmod ssl evasive security2 headers
service apache2 restart

# Enable fail2ban
cat >> /etc/fail2ban/jail.d/defaults-debian.conf <<EOF

[apache-modsecurity]
enabled = true

[apache-shellshock]
enabled = true
EOF

service fail2ban restart
################################################################################
# Secure NFS(OK)
echo "Enabling Kerberos Authentication fos NFS4"
sed -i 's/.*NEED_SVCGSSD=.*/NEED_SVCGSSD=yes/' /etc/default/nfs-kernel-server
################################################################################
# Configure sshd server(OK)
echo "Configuring sshd..."

cp "$SSHDFILE" "$SSHDFILE-$(date +%s)"

sed -i '/HostKey.*ssh_host_dsa_key.*/d' "$SSHDFILE"
sed -i 's/.*AuthenticationMethods.*/AuthenticationMethods publickey,gssapi-with-mic \
publickey,keyboard-interactive/' "$SSHDFILE"
sed -i 's/.*X11Forwarding.*/X11Forwarding no/' "$SSHDFILE"
sed -i 's/.*Port.*/Port 1027/' "$SSHDFILE"
sed -i 's/.*LoginGraceTime.*/LoginGraceTime 20/' "$SSHDFILE"
sed -i 's/.*PermitRootLogin.*/PermitRootLogin no/' "$SSHDFILE"
sed -i 's/.*KeyRegenerationInterval.*/KeyRegenerationInterval 1800/' "$SSHDFILE"
sed -i 's/.*UsePrivilegeSeparation.*/UsePrivilegeSeparation sandbox/' "$SSHDFILE"
sed -i 's/.*LogLevel.*/LogLevel VERBOSE/' "$SSHDFILE"
sed -i 's/.*UseLogin.*/UseLogin no/' "$SSHDFILE"
sed -i 's/.*Banner.*/Banner \/etc\/issue.net/' "$SSHDFILE"
sed -i 's/.*Subsystem sftp.*/Subsystem sftp \/usr\/lib\/ssh\/sftp-server -f AUTHPRIV -l INFO/' "$SSHDFILE"

if ! grep -q "AllowGroups" "$SSHDFILE" 2> /dev/null; then
    echo "AllowGroups $SSH_GROUPS" >> "$SSHDFILE"
fi

if ! grep -q "MaxAuthTries" "$SSHDFILE" 2> /dev/null; then
    echo "MaxAuthTries 4" >> "$SSHDFILE"
fi

if ! grep -q "ClientAliveInterval" "$SSHDFILE" 2> /dev/null; then
    echo "ClientAliveInterval 300" >> "$SSHDFILE"
fi

if ! grep -q "ClientAliveCountMax" "$SSHDFILE" 2> /dev/null; then
    echo "ClientAliveCountMax 0" >> "$SSHDFILE"
```

```
fi

if ! grep -q "PermitUserEnvironment" "$SSHDFILE" 2> /dev/null; then
    echo "PermitUserEnvironment no" >> "$SSHDFILE"
fi

if ! grep -q "KexAlgorithms" "$SSHDFILE" 2> /dev/null; then
    echo 'KexAlgorithms curve25519-sha256@libssh.org,ecdh-sha2-nistp521,\
    ecdh-sha2-nistp384,ecdh-sha2-nistp256,diffie-hellman-group-exchange-sha256' >> "$SSHDFILE"
fi

if ! grep -q "Ciphers" "$SSHDFILE" 2> /dev/null; then
    echo 'Ciphers chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes256-ctr' >> "$SSHDFILE"
fi

if ! grep -q "Macs" "$SSHDFILE" 2> /dev/null; then
    echo 'Macs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,\
    hmac-sha2-512,hmac-sha2-256' >> "$SSHDFILE"
fi

if ! grep -q "MaxSessions" "$SSHDFILE" 2> /dev/null; then
    echo "MaxSessions 2" >> "$SSHDFILE"
fi

if ! grep -q "UseDNS" "$SSHDFILE" 2> /dev/null; then
    echo "UseDNS yes" >> "$SSHDFILE"
fi

# Fail2Ban is already enabled by default for sshd
# Restarting Service
systemctl restart sshd.service

if [[ $VERBOSE == "Y" ]]; then
    systemctl status sshd.service --no-pager
    echo
fi
################################################################################
# Lock up cronjobs(OK)
echo "Locking up cronjobs..."

rm /etc/cron.deny 2> /dev/null
rm /etc/at.deny 2> /dev/null

echo 'root' > /etc/cron.allow
echo 'root' > /etc/at.allow

chown root:root /etc/cron*
chmod og-rwx /etc/cron*

chown root:root /etc/at*
chmod og-rwx /etc/at*

systemctl mask atd.service
systemctl stop atd.service
systemctl daemon-reload

sed -i 's/^#cron./cron./' /etc/rsyslog.d/50-default.conf

if [[ $VERBOSE == "Y" ]]; then
    systemctl status atd.service --no-pager
    echo
fi
################################################################################
# Configure UFW(OK)
echo "Configuring Firewall.."
sed -i 's/IPT_SYSCTL=.*/IPT_SYSCTL=\/etc\/sysctl\.conf/' "$UFWDEFAULT"
ufw --force enable

for ip in $FW_LOCAL; do
    ufw allow log from "$ip" to any port 1027 proto tcp # SSH
done

if [[ $SERVER == "Y" ]]; then
```

```
    ufw allow proto tcp from any to any port 1027 #SSH
    ufw allow http
    ufw allow samba
    ufw allow nfs
fi

if [[ $VERBOSE == "Y" ]]; then
    systemctl status ufw.service --no-pager
    ufw status verbose
    echo
fi
################################################################################
# Disable IPV6(OK)
sed -i 's/^GRUB_CMDLINE_LINUX=.*/GRUB_CMDLINE_LINUX="ipv6.disable=1"/' "$DEFAULTGRUB"
update-grub

sed '/udp6/d' /etc/netconfig
sed '/tcp6/d' /etc/netconfig
################################################################################
# Configure DNS resolvers(OK)
echo "Confuguring DNS..."

dnsarray=( $(grep nameserver /etc/resolv.conf | sed 's/nameserver//g') )
dnslist=${dnsarray[@]}

sed -i "s/^#DNS=.*/DNS=$dnslist/" "$RESOLVEDCONF"
sed -i "s/^#FallbackDNS=.*/FallbackDNS=8.8.8.8 8.8.4.4/" "$RESOLVEDCONF"
sed -i "s/^#DNSSEC=.*/DNSSEC=allow-downgrade/" "$RESOLVEDCONF"
sed -i '/^hosts:/ s/files dns/files resolve dns/' /etc/nsswitch.conf

systemctl daemon-reload

if [[ $VERBOSE == "Y" ]]; then
    systemctl status resolvconf.service --no-pager
    echo
fi
################################################################################
# Securing NTP(OK)
echo "Securing NTP..."

LATENCY="50"
SERVERS="4"
APPLY="YES"
CONF="$TIMESYNCD"
SERVERARRAY=()
FALLBACKARRAY=()
TMPCONF=$(mktemp --tmpdir ntpconf.XXXXX)

if [[ -z "$NTPSERVERPOOL" ]]; then
    NTPSERVERPOOL="0.ubuntu.pool.ntp.org 1.ubuntu.pool.ntp.org \
    2.ubuntu.pool.ntp.org 3.ubuntu.pool.ntp.org pool.ntp.org"
fi

echo "[Time]" > "$TMPCONF"

PONG="ping -c2"

for s in $(dig +noall +answer +nocomments $NTPSERVERPOOL | awk '{print $5}'); do
    if [[ $NUMSERV -ge $SERVERS ]]; then
        break
    fi

    PINGSERV=$($PONG "$s" | grep 'rtt min/avg/max/mdev' | awk -F "/" '{printf "%.0f\n",$6}')
    if [[ $PINGSERV -gt "1" && $PINGSERV -lt "$LATENCY" ]]; then
        OKSERV=$(nslookup "$s"|grep "name = " | awk '{print $4}'|sed 's/.$//')
        if [[ $OKSERV && $NUMSERV -lt $SERVERS && ! (( $(grep "$OKSERV" "$TMPCONF") )) ]]; then
            echo "$OKSERV has latency < $LATENCY"
            SERVERARRAY+=("$OKSERV")
            ((NUMSERV++))
        fi
    fi
done
```

```
for l in $NTPSERVERPOOL; do
    if [[ $FALLBACKSERV -le "2" ]]; then
        FALLBACKARRAY+=("$l")
        ((FALLBACKSERV++))
    else
        break
    fi
done

    if [[ ${#SERVERARRAY[@]} -le "2" ]]; then
        for s in $(echo "$NTPSERVERPOOL" | awk '{print $(NF-1),$NF}'); do
            SERVERARRAY+=("$s")
        done
    fi

    echo "NTP=${SERVERARRAY[@]}" >> "$TMPCONF"
    echo "FallbackNTP=${FALLBACKARRAY[@]}" >> "$TMPCONF"

    if [[ $APPLY = "YES" ]]; then
        cat "$TMPCONF" > "$CONF"
        systemctl restart systemd-timesyncd
        rm "$TMPCONF"
    else
        echo "Configuration saved to $TMPCONF."
    fi

    if [[ $VERBOSE == "Y" ]]; then
        systemctl status systemd-timesyncd --no-pager
        echo
    fi
################################################################################
# Configure logrotate(OK)
echo "Configuring logrotate..."

cat > "$LOGROTATE" <<EOF
# see "man logrotate" for details
# rotate log files daily
daily

# use the syslog group by default, since this is the owning group
# of /var/log/syslog.
su root syslog

# keep 7 days worth of backlogs
rotate 7

# create new (empty) log files after rotating old ones
create

# use date as a suffix of the rotated file
dateext

# compressed log files
compress

# use xz to compress
compresscmd /usr/bin/xz
uncompresscmd /usr/bin/unxz
compressext .xz

# packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp and btmp -- we'll rotate them here
/var/log/wtmp {
    monthly
    create 0664 root utmp
    minsize 1M
    rotate 1
}

/var/log/btmp {
    missingok
```

```
    monthly
    create 0600 root utmp
    rotate 1
}

# system-specific logs may be also be configured here.
EOF

sed -i 's/^#Storage=.*/Storage=persistent/' "$JOURNALDCONF"
sed -i 's/^#ForwardToSyslog=.*/ForwardToSyslog=yes/' "$JOURNALDCONF"
sed -i 's/^#Compress=.*/Compress=yes/' "$JOURNALDCONF"

systemctl restart systemd-journald

if [[ $VERBOSE == "Y" ]]; then
    systemctl status systemd-journald --no-pager
    echo
fi
################################################################################
# Enforcing auditd rules
echo "Enforcing Auditd rules..."
sed -i 's/^action_mail_acct =.*/action_mail_acct = root/' "$AUDITDCONF"
sed -i 's/^admin_space_left_action = .*/admin_space_left_action = halt/' "$AUDITDCONF"
sed -i 's/^max_log_file_action =.*/max_log_file_action = keep_logs/' "$AUDITDCONF"
sed -i 's/^space_left_action =.*/space_left_action = email/' "$AUDITDCONF"
sed -i 's/^GRUB_CMDLINE_LINUX=.*/GRUB_CMDLINE_LINUX="ipv6.disable=1 audit=1"/' "$DEFAULTGRUB"

cat > /etc/audit/audit.rules <<EOF
## Remove any existing rules
-D

## Buffer Size
-b 8192

## Failure Mode
-f 2

## Audit the audit logs
-w /var/log/audit/ -k auditlog

## Auditd configuration
-w /etc/audit/ -p wa -k auditconfig
-w /etc/libaudit.conf -p wa -k auditconfig
-w /etc/audisp/ -p wa -k audispconfig

## Monitor for use of audit management tools
-w /sbin/auditctl -p x -k audittools
-w /sbin/auditd -p x -k audittools

## Monitor AppArmor configuration changes
-w /etc/apparmor/ -p wa -k apparmor
-w /etc/apparmor.d/ -p wa -k apparmor

## Monitor usage of AppArmor tools
-w /sbin/apparmor_parser -p x -k apparmor_tools
-w /usr/sbin/aa-complain -p x -k apparmor_tools
-w /usr/sbin/aa-disable -p x -k apparmor_tools
-w /usr/sbin/aa-enforce -p x -k apparmor_tools

## Monitor Systemd configuration changes
-w /etc/systemd/ -p wa -k systemd
-w /lib/systemd/ -p wa -k systemd

## Monitor usage of systemd tools
-w /bin/systemctl -p x -k systemd_tools
-w /bin/journalctl -p x -k systemd_tools

## Special files
-a always,exit -F arch=b64 -S mknod -S mknodat -k specialfiles

## Mount operations
-a always,exit -F arch=b64 -S mount -S umount2 -k mount
```

```
## Changes to the time
-a always,exit -F arch=b64 -S adjtimex -S settimeofday -S clock_settime -k time

## Cron configuration & scheduled jobs
-w /etc/cron.allow -p wa -k cron
-w /etc/cron.deny -p wa -k cron
-w /etc/cron.d/ -p wa -k cron
-w /etc/cron.daily/ -p wa -k cron
-w /etc/cron.hourly/ -p wa -k cron
-w /etc/cron.monthly/ -p wa -k cron
-w /etc/cron.weekly/ -p wa -k cron
-w /etc/crontab -p wa -k cron
-w /var/spool/cron/crontabs/ -k cron

## User, group, password databases
-w /etc/group -p wa -k etcgroup
-w /etc/passwd -p wa -k etcpasswd
-w /etc/gshadow -k etcgroup
-w /etc/shadow -k etcpasswd
-w /etc/security/opasswd -k opasswd

## Monitor usage of passwd
-w /usr/bin/passwd -p x -k passwd_modification

## Monitor for use of tools to change group identifiers
-w /usr/sbin/groupadd -p x -k group_modification
-w /usr/sbin/groupmod -p x -k group_modification
-w /usr/sbin/addgroup -p x -k group_modification
-w /usr/sbin/useradd -p x -k user_modification
-w /usr/sbin/usermod -p x -k user_modification
-w /usr/sbin/adduser -p x -k user_modification

## Monitor module tools
-w /sbin/insmod -p x -k modules
-w /sbin/rmmod -p x -k modules
-w /sbin/modprobe -p x -k modules

## Login configuration and information
-w /etc/login.defs -p wa -k login
-w /etc/securetty -p wa -k login
-w /var/log/faillog -p wa -k login
-w /var/log/lastlog -p wa -k login
-w /var/log/tallylog -p wa -k login

## Network configuration
-w /etc/hosts -p wa -k hosts
-w /etc/network/ -p wa -k network

## System startup scripts
-w /etc/inittab -p wa -k init
-w /etc/init.d/ -p wa -k init
-w /etc/init/ -p wa -k init

## Library search paths
-w /etc/ld.so.conf -p wa -k libpath

## Local time zone
-w /etc/localtime -p wa -k localtime

## Time zone configuration
-w /etc/timezone -p wa -k timezone

## Kernel parameters
-w /etc/sysctl.conf -p wa -k sysctl

## Modprobe configuration
-w /etc/modprobe.conf -p wa -k modprobe
-w /etc/modprobe.d/ -p wa -k modprobe
-w /etc/modules -p wa -k modprobe

# Module manipulations.
-a always,exit -F arch=b64 -S init_module -S delete_module -k modules
```

```
## PAM configuration
-w /etc/pam.d/ -p wa -k pam
-w /etc/security/limits.conf -p wa -k pam
-w /etc/security/pam_env.conf -p wa -k pam
-w /etc/security/namespace.conf -p wa -k pam
-w /etc/security/namespace.init -p wa -k pam

## Postfix configuration
-w /etc/aliases -p wa -k mail
-w /etc/postfix/ -p wa -k mail

## SSH configuration
-w /etc/ssh/sshd_config -k sshd

## Changes to hostname
-a exit,always -F arch=b64 -S sethostname -k hostname

## Changes to issue
-w /etc/issue -p wa -k etcissue
-w /etc/issue.net -p wa -k etcissue

## Capture all failures to access on critical elements
-a exit,always -F arch=b64 -S open -F dir=/etc -F success=0 -k unauthedfileaccess
-a exit,always -F arch=b64 -S open -F dir=/bin -F success=0 -k unauthedfileaccess
-a exit,always -F arch=b64 -S open -F dir=/sbin -F success=0 -k unauthedfileaccess
-a exit,always -F arch=b64 -S open -F dir=/usr/bin -F success=0 -k unauthedfileaccess
-a exit,always -F arch=b64 -S open -F dir=/usr/sbin -F success=0 -k unauthedfileaccess
-a exit,always -F arch=b64 -S open -F dir=/var -F success=0 -k unauthedfileaccess
-a exit,always -F arch=b64 -S open -F dir=/home -F success=0 -k unauthedfileaccess
-a exit,always -F arch=b64 -S open -F dir=/root -F success=0 -k unauthedfileaccess
-a exit,always -F arch=b64 -S open -F dir=/srv -F success=0 -k unauthedfileaccess
-a exit,always -F arch=b64 -S open -F dir=/tmp -F success=0 -k unauthedfileaccess

## Monitor for use of process ID change (switching accounts) applications
-w /bin/su -p x -k priv_esc
-w /usr/bin/sudo -p x -k priv_esc
-w /etc/sudoers -p rw -k priv_esc

## Monitor usage of commands to change power state
-w /sbin/shutdown -p x -k power
-w /sbin/poweroff -p x -k power
-w /sbin/reboot -p x -k power
-w /sbin/halt -p x -k power

## Monitor admins accessing user files.
-a always,exit -F dir=/home/ -F uid=0 -C auid!=obj_uid -k admin_user_home

## Monitor changes and executions in /tmp and /var/tmp.
-w /tmp/ -p wxa -k tmp
-w /var/tmp/ -p wxa -k tmp

## Make the configuration immutable
-e 2
EOF

sed -i "s/arch=b64/arch=$(uname -m)/g" /etc/audit/audit.rules
cp /etc/audit/audit.rules "$AUDITRULES"
update-grub 2> /dev/null

systemctl enable auditd
systemctl restart auditd.service

if [[ $VERBOSE == "Y" ]]; then
    systemctl status auditd.service --no-pager
    echo
fi
################################################################################
# Enable RKHUNTER(OK)
echo "Enabling rkhunter..."

sed -i 's/^CRON_DAILY_RUN=.*/CRON_DAILY_RUN="yes"/' "$RKHUNTERCONF"
sed -i 's/^APT_AUTOGEN=.*/APT_AUTOGEN="yes"/' "$RKHUNTERCONF"
```

```
rkhunter --propupd
################################################################################
# Enable CLAMAV(OK)
echo "Enabling CLAMAV..."
service clamav-daemon start
freshclam
service clamav-freshclam start

echo > /etc/cron.daily/user_clamscan <<EOF
#!/bin/bash
SCAN_DIR="/home"
LOG_FILE="/var/log/clamav/user_clamscan.log"
/usr/bin/clamscan -i -r $SCAN_DIR >> $LOG_FILE
EOF

chmod +x /etc/cron.daily/user_clamscan
################################################################################
# Disable Prelinking(OK)
 echo "Disabling Prelink for AIDE..."

if dpkg -l | grep prelink 1> /dev/null; then
    "$(which prelink)" -ua 2> /dev/null
    "$APT" purge prelink
fi
################################################################################
# Secure AIDE Configuration(OK)
echo "Securing Aide..."

sed -i 's/^Checksums =.*/Checksums = sha512/' /etc/aide/aide.conf
################################################################################
# Set AIDE Postinstall(OK)
echo "Building AIDE initial db, this will take a while..."

aideinit --yes
cp /var/lib/aide/aide.db.new /var/lib/aide/aide.db

echo "Enabling AIDE check daily..."

cat > /etc/systemd/system/aidecheck.service <<EOF
[Unit]
Description=Aide Check

[Service]
Type=simple
ExecStart=/usr/bin/aide.wrapper --check

[Install]
WantedBy=multi-user.target
EOF

cat > /etc/systemd/system/aidecheck.timer <<EOF
[Unit]
Description=Aide check every day at midnight

[Timer]
OnCalendar=*-*-* 00:00:00
Unit=aidecheck.service

[Install]
WantedBy=multi-user.target
EOF

chmod 0644 /etc/systemd/system/aidecheck.*

systemctl reenable aidecheck.timer
systemctl start aidecheck.timer
systemctl daemon-reload

if [[ $VERBOSE == "Y" ]]; then
    systemctl status aidecheck.timer --no-pager
    echo
fi
################################################################################
```

```
# Remove unused packages(OK)
echo "Removing unused packages..."

$APT purge expect

if [[ $SERVER == "Y" ]]; then
    echo "Removing X-Window System"
    $APT purge x-window-system-core
    echo
fi

$APT clean
$APT autoclean
$APT autoremove
################################################################################
# Check systemddelta(OK)
if [[ $VERBOSE == "Y" ]]; then
    echo "Checking systemd-delta..."
    systemd-delta --no-pager
    echo
fi
################################################################################
# check if reboot is required(OK)
if [ -f /var/run/reboot-required ]; then
    cat /var/run/reboot-required
fi

echo
```

Bibliography

[1] Arr0way. Security harden centos 7, 2015.

[2] Jay Beale. The bastille hardening program, 2012.

[3] Jason Cannon. Linux security and hardening, the practical security guide, 2016.

[4] Martin Prpič et al. *Red Hat Enterprise Linux 7 Security Guide*. RedHat, US, 2013.

[5] Maxim Svistunov et al. File and print servers (samba), 2016.

[6] Hitesh Jethva. Linux hardening and system auditing, 2015.

[7] Jeffrey Orloff. Hardening the linux server, 2014.

[8] Ravi Saive. 25 hardening security tips for linux servers, 2015.

[9] Sans. Linux security checklist, 2014.

[10] Tom6. Ubuntu security guides, 2015.

[11] Tom6. Ubuntu security server guide, 2016.