Cuong Nguyen

Linux OS security mechanisms and how to implement them

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor'sThesis

Date 5 Sep 2016

Helsinki Metropolia of Applied Sciences **Abstract**

Author Title Number of Pages Date	Cuong Nguyen Linux OS security mechanisms and how to implement them 36 pages 5 Sep 2016
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Supervisor	Markku Nuutinen, Senior Lecturer

The purpose of this final year project was to study Linux security mechanism. The aim was to learn the basic concepts of Linux file structure, how security is implemented and propose more methods to enhance system security.

The first part of the study describes the overall information about Linux operating system. The second one deals with protection mechanism for file system: file structure, journals, privileges. The third one lists some common hacking methods to Linux system. The final one proposes some strategies to improve Linux security. The method used in this project is combining theories with current software to make a secure Linux system.

Some suggestions for improvement for this project is finding more security methods to make Linux OS more secure and complete. Methods should be implemented in a real machine and examine the results.

Keywords	Linux ,network security, attacks, exploits, prevention

Contents

Abbreviations

ISP	Internet Service Provider
JFS	Journaled File System
JBD	Journaling Block Device
VFS	Virtual File System
IDS	Intrusion Detection System
HTTP	The Hypertext Transfer Protocol
FTP	File Transfer Protocol
SMTP	Simple Mail Transfer Protocol
IP	Internet Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
ICMP	Internet Control Message Protocol
TTL	Time To Live
XSS	Cross-Site Scripting
XML	Extensible Markup Language
SQL	Structured Query Language
ARP	Address Resolution Protocol
DNS	Domain Name System
URL	Uniform Resource Locator
LAN	Local Area Network
SSL	Secure Sockets Layer
CA	Certificate Authority
TTY	teletypewriter
LIDS	Linux Intrusion Detection System
SCP	Secure copy

SFTP	Secure File Transfer Protocol
SSHD	SSH Daemon
NAT	Network Address Translation
DMZ	Demilitarized Zone
CGI	Common Gateway Interface
SMB	Server Message Block
NIDS	Network Intrusion Detection Systems
LFM	Last File Manager

1. Introduction

In today’s society, the need of exchanging information via the Internet has become extremely important. We are living in the era of computer and technology. Information technology is the basis to make new technologies that change our lives. Technology affects every aspect of life: work management, entertainment, communication, commerce. Along with the development of computer network, the risk of facing security attacks such as Trojan, virus, worm is very high. Ensuring information safety and security is the main priority of organizations, companies and service providers. Hence, choosing a suitable and reliable operating system is very important.

Linux operating system (Linux OS) then was invented with many safety features such as ab open source system, high security mechanism. Many companies and organizations use Linux as a platform for their products. Some ISPs use Linux servers as Internet gateways. Linux is free, so it helps people to access to technology it easily. It is developed continuously by some communities. However, operating system has been a target of hackers to deploy and attack because of its increasing popularity. Linux has great defense but many novice users do not know to use them to prevent attacks.

Therefore, I chose the topic “Linux OS security mechanisms and how to implement them” as my thesis paper. The goal of this paper is to improve knowledge for Linux users how the security mechanisms work, and to know what to do to enhance system security.

2. Overall about Linux OS

2.1. History

Linux was invented by Linus Torvalds, a student in University of Helsinki. First, he used Minix, a non-free Unix-like system, in order to create a Unix operating system that can run on PC with Intel 80386 processor in 1991. And by September of that year, he released the first version 0.01. In January 1992, version 0.02 was released with shell and C compiler. Linux did not need Minux to operate anymore and Linus named his operating system Linux. In 1994, the official version 1.0 was released. After that Linux became a phenomenon, with a large community of programmers who continue to debug, develop, and improve the source code until today.

2.2. Advantages and disadvantages of Linux

2.2.1. Advantages

Linux is an open source OS, so it cheaper than other traditional operating systems and easier to be modified and enhanced by developers. It is possible to have your own Linux based on your interest, which is impossible to do with Windows without Microsoft permission and support. Because of a large Linux community, there are many custom versions of Linux and it is easy to find a suitable one according to your own desire.

Linux is compatible to many environments and platforms. Linux can run on server, PC, smart phones, and game console. Linux kernel is used in control devices such as palm, and robot. Application range of Linux is very enormous.

Linux has a high security system. It has a very strict role-based access control, and only root user and install and modify the system, which helps the whole system be stable and avoid failures. In addition, because of its open source feature, any vulnerability will be fixed soon by the whole community together. Everything in Linux is public, which means high security performance.

2.2.2. Disadvantages

Linux usability is still a problem for users. Every work is executed in command line and it requires some knowledge to run the system. Therefore, using Linux is still performed primarily by expert and technical people. Moreover, there are currently many Linux versions from an original kernel: RedHat, SuSE, Knoppix, and so on. Hence, it is difficult for novice user to choose a suitable version.

Linux does not support many hardwares as Windows does because hardware manufactures does not release drivers for Linux. Linux also has similar application softwares, but the quality is not good as in Windows.

There is no company who is responsible for Linux OS. While there is a large community that can ask for help but sometimes the question is unique and difficult. There is no immediate technical support form help center.

3. Data protection mechanism for file system on Linux

3.1. Linux file structure

3.3.1. File system layout

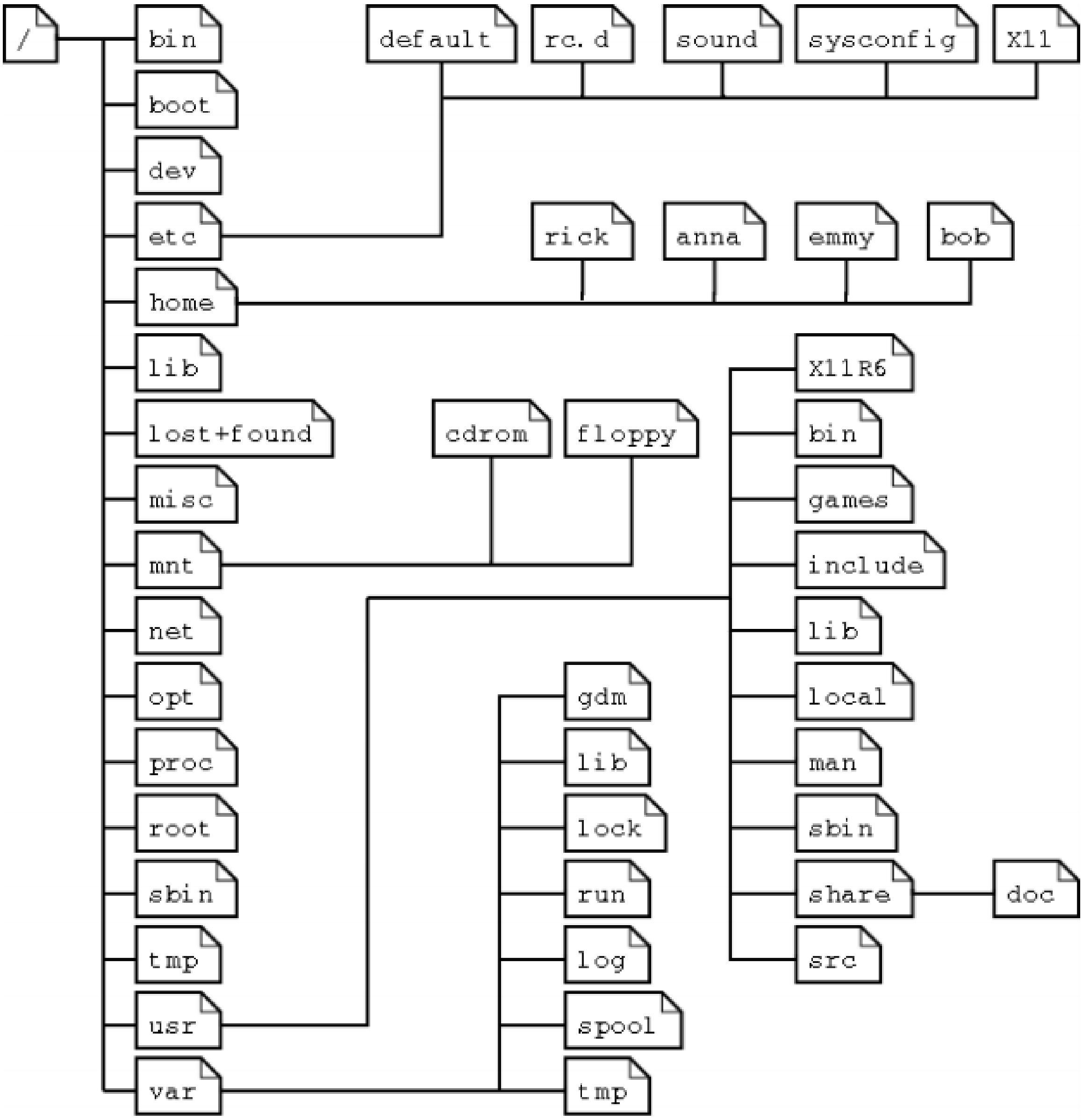


Figure 1. Linux file system layout. [13]

Linux layout is often presented by a tree structure as shown in figure 1. Depend on administrator’s intention, the structure may vary by adding or removing some files. The forward slash (/) in the figure shows the root directory, which include all underlying directories and files. When showing the directory that is right below the root, a slash is used before the directory name to tell its position in the system. [8, 456]

3.3.2. System files

In table 1, some system files are showed with their meanings:

Directory	Meaning
/bin	Common programs, shared by the system, the system administrator and the users.
/boot	The startup files and the kernel, vmlinuz. In some recent distributions also grub data. Grub is the GRand Unified Boot loader and is an attempt to get rid of the many different boot-loaders we know today.
/dev	Contains references to all the CPU peripheral hardware, which are represented as files with special properties.
/etc	Most important system configuration files are in /etc, this directory contains data similar to those in the Control Panel in Windows
/home	Home directories of the common users.
/initrd	(on some distributions) Information for booting. Do not remove!
/lib	Library files, includes files for all kinds of programs needed by the system and the users.
/lost+found	Every partition has a lost+found in its upper directory. Files that were saved during failures are here.
/misc	For miscellaneous purposes.
/mnt	Standard mount point for external file systems, e.g. a CD-ROM or a digital camera.
/net	Standard mount point for entire remote file systems
/opt	Typically contains extra and third party software.
/proc	A virtual file system containing information about system resources. More information about the meaning of the files in proc is obtained by entering the command **man *proc*** in a terminal window. The fileproc.txt discusses the virtual file system in detail.
/root	The administrative user's home directory. Mind the difference between /, the root directory and /root, the home directory of the *root* user.
/sbin	Programs for use by the system and the system administrator.
/tmp	Temporary space for use by the system, cleaned upon reboot, so don't use this for saving any work!
/usr	Programs, libraries, documentation etc. for all user-related programs.

Directory	Meaning
/var	Storage for all variable files and temporary files created by users, such as log files, the mail queue, the print spooler area, space for temporary storage of files downloaded from the Internet, or to keep an image of a CD before burning it.

Table 2. Subdirectories of the root directory. Copied from Brian 2014 [10, 38]

3.2. Management programs

In table 2, some programs are showed with their functions:

Programs	Functions
badblocks	Find bad blocks on a disk’s partion
cfdisk	Repair or make disk’s partion
fdisk	Repair or make disk’s partion
debugfs	Allow to access directly data system
df	Check how much disk space is using by file systems
dosfsck	Check or repair MS-DOS system
du	Show disk usage of one directory and all its subdirectories
dump	Back up Ext2 file system
dump2fs	Show information of super block and block information of a file system
e2fsck	Check expansion Linux file system
e2label	Change group in Ext2 file system
exportfs	Export file as nfs format
fdformat	Used for floppy disks

Table 2. Meaning of some management programs in Linux.

Files in etc directory

In table 3, some files in etc directories are showed with their meanings:

Directory	Meaning
profile	Contain scripts running when system starts
/dev/MAKEDEV	Connect file in /dev with drivers in kernal
/etc/aliases	User has a suitable email

Directory	Meaning
/etc/bootptab	Configuration files for BOOTP daemon
/etc/crontab	Scheduled commands
/etc/dhcpd.conf	Configuration files for DHCP daemon
/etc/ethers	Make map from hardware address to IP address
/etc/exports	Files that describe export system for NFS services
/etc/fdprm	Parameter of floppy disk
/etc/fstab	Information of all storage devices
/etc/group	Information all which group a person belongs to
/etc/gshadow	Contain passwords of groups
/etc/passwd	Contain password of user
/etc/shells	List of reliable shells
/etc/terminfo	I/O commands
/etc/services	List of services that the system supports
/etc/issue	Result of fast login

Table 3. Meaning of some files in etc directory. Copied from Brian 2014 [10, 128]

3.3. Journaling file system

3.3.1. Definition

When the system is down suddenly (power loss, software failure…), there are errors in the file system because files are being written and their addresses are not updated yet. If the file system is not a journal file, operating system will recognize the system failure and use the file system check (fsck) to fix the problem after being rebooted. If the hard disk is big, fsck procedure will take a long time to finish. And if the error is serious that fsck cannot fix, the operating system wil run to single user mode for the user to fix himself. Journal file system (JFS), as illustrated in figure 2, helps to avoid file system failure by writing a journal, a file that records every change of file system to a cache memory instead of directly to file system on hard disk. After some scheduled intervals, the changes can be written to file system officially. If the system is down suddenly, journal file will be used to restore unsaved information and to avoid damage to metadata of the file system. Metadata is the information of data on the hard disk: created time, file owner, file size, and so forth. In conclusion, JFS is a self-restored file system by using a journal to save every change before the changes are actually written to the file system. [14, 52]

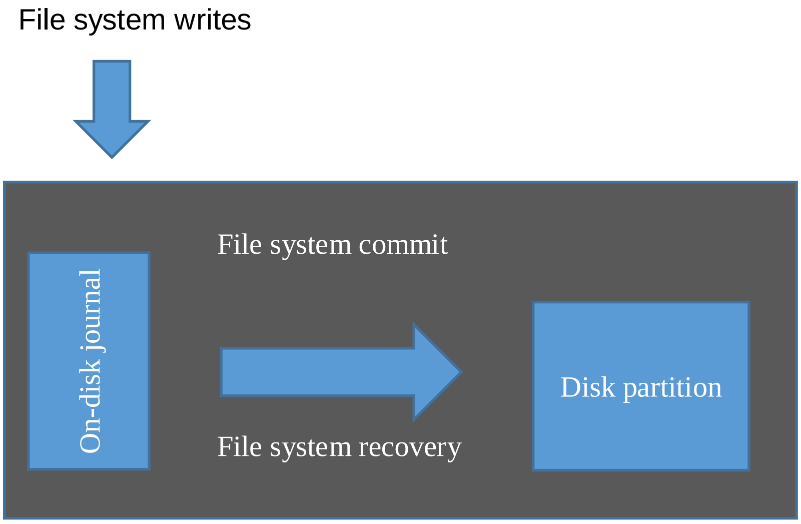


Figure 2. File system diagram.

The fist file system journal is IBM JFS invented in 1990. The most common EFS is being used now is ext3fs (third extended file system, abbreviated ex3). Ext3 (with journal) is the extension of ext2 (no journal). Linux file system inherits characteristics from Ext2. Ext 2 is improved from the original Linux file system Ext, which is developed from Minix. Minix file system has a maximum 64 Mb capacity and file name length is 14 characters. Ext supports 2 Gb capacity but its speed is limited. Ext2 support 4 Tb and remove the disadvantages of previous versions. Linux kernel uses a virtual file system (VFS) to communicate with file system, which helps Linux to support many file system types such as DOS, FAT16, FAT32, and et cetera. [8, 738]

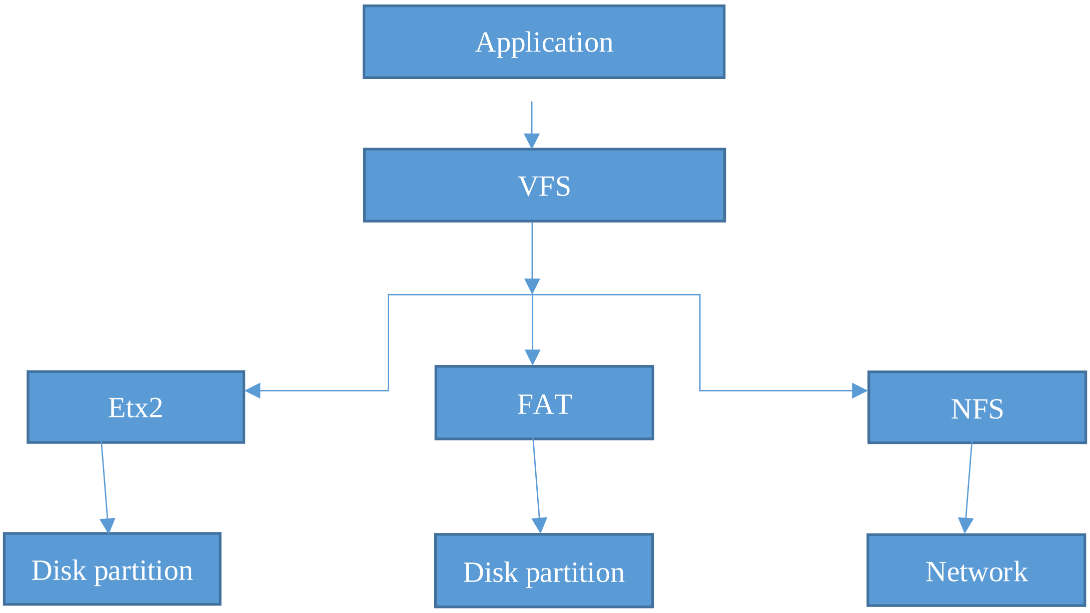


Figure 6. VFS position.

3.3.2. JFS mechanism

When OS is started, it always uses fcsk program to check integrity of file system. If the file system has a failure or has not been unmounted, fcsk will scan the system and try to restore the data. The recovery process duration depends on hard disk capacity. For example, it will take hours to scan 500 Gb hard disk. This method is applied in Unix ufs (Sun or Hp computer) or ext2 that Linux is using. If the file system is able to record the activities that was done or is being done, it can recognize failure files without scanning all the system, which helps recovery process be more effective and reliable. Such file system is called JFS.

JFS mechanism:

- Record file system’s updates as transactions.
- Every transaction is recorded in a log file.
- One transaction is committed when it is recorded fully in log file. (file system may not be updated)
- When the file system is fully updated with every task in transaction, it will be deleted in log file.
- If there is failure in file system, the OS will be fixed and restored based on transactions in log file.

3.3.3. File systems that support JFS

Reiserfs

The idea of building Reiserfs file system originated from the demand of optimizing file storing process and access to these files. Reiserfs uses method ”B * Trees”, developed from ”B + Trees” to organize data. One folder can contain up to 100.000 subfolders. In addition, Reiserfs does not provide a fixed amount of storage, for example 1 KB or 4 KB like what other file systems do. It provides exactly how much necessary capacity, which helps optimize storage usage with small files. Files accessing speed can be increased from 8-15 times and files capacity can be saved over 5% compared to ext2 file system. Reiserfs support journal for meta-data only. To use Reiserfs on Linux, these modules must be moved to kernal and formatted partition by using Reiserfs file system. In reality, the use of Reiserfs on squid proxy server makes internet speed faster when accessing small files (*.html, *.gif, *.class) in cache. [14, 53]

XFS

XFS was developed by Silicon Graphic in 1990 to overcome obstacles of file system in terms of size and number of partitions, directories and files. Nowadays, XFS is used in Linux with powerful functions and applications. XFS is a 64-bit file system. It can manage files with 9 Exabyte size. It has Volume Manager tool to manage 128 volumes, each volume can join up to 100 partitions and support journal fully. Another plus of XFS is the ability to maintain high speed access on hard disk, which is very important to high quality video providing service. Kernel Linux 2.4.17 and over supports very well XFS. To use XFS, it is necessary to patch the kernel with modules of XFS. [14, 54]

JFS

Developed by IBM, JFS is a 64-bit file system for internal file server. JFS helps reboot system faster and has a high capacity for data recording. JFS supports journaling by recording file changes (transaction). In case of system failure, it is not necessary to read the whole log file, but only restore transaction, which makes the recovery faster. JFS's Logical Volumes allow to connect physical partitions into logical partitions with high capacity. To use JFS in Linux, kernel must be patched and partition must be formatted with JFS partition tool.

Ext3

Built based on an ext2 file system that Linus is using, ext3 introduces a very important function: journaling file system, which means more security for data handling. Ext3 also uses JBD mechanism (journaling block device) to protect information on disk. Hence, it is considered more reliable than file system which carries out journaling only for metadata such as Reiserfs, XFS or JFS. With the double protection like that, data processing performance is slower. Hence, XFS is a good choice for an application whose priority is data safety than data processing speed. Because ext3 is based on etx2, it is easy to transfer data from ext2 to ext3. With Linux kernel version 2.4.15 and over, there is no need any patch version. [14, 55]

3.4. User and group management

Each file and directory in Linux belong to one user and one group and has some permissions to be used. Linux file system control file and directory access by using mod file.

Information about a file or directory has the following formula:

drwx-w-r-x cuong developer 4096 Aug 8 13:11 book

Access permission User Group File size Date and time File name

By default, the user who created that file is the owner of that one and he can change access permissions for that file. The first ten characters drwx-w-r-x describes file type and access permissions to that file. The first character denotes what kind of file it is. The following table lists the most common types of file in Linux:

Character	Type of file
d	Directory
b	Block-type special file
c	Character-type special file
l	Symbolic link
p	pipe
s	Socket
-	Regular file

Table 4. Types of file in Linux.

The next 9 characters show access rights of 3 groups: owner, group and others.

3.4.1. Permission groups

One file or directory has three permission groups:

Owners: permissions that applied to the user who owns the file or directory
Groups: permissions that applied to the group who owns the file or directory
Others: permissions to all other users on the system

3.4.2. Permission types

As showed in table 5:

Read permission allows users to view the content of a file but they cannot change, modify or delete anything inside it. However, they can copy that file and modify the copied version.

Write permission allows users to change the content of a file. If the file has both read and write permission, the file can be edited and viewed. If the file has only write permission, it is possible to add information into file, but cannot view the file's content.

Execute permission allows users to run a file if that one is a script or program. Execute permission is independent to other permissions. Some programs have only execute permission, which means users can run the program but cannot see how to make or copy it.

Letter	To file	To directory
r	View the content of the file	Show the directory's content
w	Change content of the file	Modify the directory's content, need execution (x) permission to be set to work
x	Execute or run the file	The directory can be accessed with cd

Table 5. Types of file in Linux.

3.4.3. Access modes

Linux has three special permission modes: suid (set user id), sgid (set group id) and sticky bit.

Imagine that in some circumstances, a user needs to change his own password. It means that he need to be able to modify /etc/passwd file. It is not allowed because the file's write permission is only for root user. That how suid and sgid comes into play. If an executable program has suid mode, it will run as if it is the user who opened it is the owner. In the same way, with assigned sgid mode, the program will be executed as if the user who opened it belongs to the file's group.

/usr/bin/passwd has suid access mode:

```
# ls -l /usr/bin/passwd
-rwsr-xr-x. 1 root root 10000 Aug 8 2016 /usr/bin/passwd
```

There is s letter instead of x for user's permissions. It illustrates that suid and execute modes are set. When passwd is executed, the user will have full right as super user (root user). Now he can modify his own password. The character to denote suid for sgid is s.

When sgid is assigned to a directory, any subdirectories and files created inside will have the same group as the main directory. It is very useful for a group of people who work on the same project.

Sticky bit is commonly used in a shared environment where everyone can modify a file but not remove it. It can be deleted only by root user, file owner or the ones who have write permission. Sticky bit is represented by letter t.

```
# ls -l text.txt
-rwxrwxrwt 1 root root 0 Aug 22 02:06 text.txt
```

3.4.4. Granting privileges

Chown command

To change the owner of a file, chrown command is used in the following format:

```
chown user:group /filepath
```

The owner of “text.txt” is changed to “cuong” user in “dev” group.

```
chown cuong:dev text.txt
```

-R option is added to change the owner of a directory and all subdirectories and files inside.

```
Chown -R cuong:dev text.txt
```

Chgrp command

When logging in, user belongs to one certain default group created by root user. One user can belong to many groups, but only one group can be changed to use at a time. To change the group ownership of one or many files, the following syntax will be used:

```
Chgrp [options] group_file
```

For example: chgrp work file.txt -> Change the owning group of the file file.txt to the group named work.

Chmod command

Chmod is used to change permissions of files or directories. The syntax is:

```
chmod [options] mode file
```

There are 2 modes: octal and symbolic modes.

Octal mode accepts up to 4 digits as shown in table 6. The first digit represents suid, sgid, sticky bit. The last 3 letters specify permissions for owner, group and other users. [10, 34]

#	Permission	rwx
7	read, write and execute	rwx
6	read and write	rw-
5	read and execute	r-x
4	read only	r--
3	write and execute	-wx
2	write only	-w-
1	execute only	--x
0	none	---

Table 6: digits for changing user privilege

Example:

# chmod 444 text.txt: allow only to read the file text.txt for everyone

# chmod 777 text.txt: allow to read, write, execute the file text.txt for everyone

Symbolic mode consists of three parts: reference (table 7) + operator (table 8) + mode (table 9)

Reference	Class	Description
u	owner	file's owner
g	group	file's group
o	others	other users
a	all	all ugo

Table 7. References in symbolic mode.

Operator	Description
+	add the specified modes
-	remove the specified modes
=	make file or group’s permissions as exact modes as specified

Table 8. Operators in symbolic mode.

Mode	Name
r	read
w	write
x	execute
s	setuid/gid
t	sticky

Table 9. Modes in symbolic mode.

Example:

chmod a=rw list.txt: assign read and write permissions for list.txt to every one
chmod u+s text.txt: set suid for text.txt.

4. Common hacking techniques on Linux

4.1. Hacking the system

4.1.1. Local access control

Console access

Once a hacker has a chance to gain physical access to a Linux server, if there is no password for BIOS and boot option is enables, he can delete root user's account and password in /etc/shadow file and booting into the system with full access by using a bootable Linux CD. [1, 44]

BIOS Bypassing password

Even if BIOS password is set, it still can be bypassed by some techniques:

- Using boot disk utilities: if the system can boot from USB or optical drive, BIOS password removal tools can be used. For example: KILLCMOS, CMOS password removal…
- CMOS battery removal: If boot option is disabled, the other way is to remove CMOS battery. Password then is erased and BIOS returns to the original setting. [1, 46]

Privilege escalation:

- Hardware, driver, and module exploitation: hardware manufacturers do not tend to develop Linux drivers for their products. Therefore, drivers for Linux are developed by third parties, which causes difficulties in choosing a right driver. Hackers take this opportunity to make drivers with bugs, then exploits turns into remote shells. A remote shell is a kind of physical access, that hackers can gain full access to the system. [1, 57]

- Software vulnerability exploitation: similar to hardware exploitation, any software can contain bugs and exploits. The risk is even greater than hardware exploitation because there is a great number of software from unreliable sources. There is no perfect code and everything can be dangerous. [1, 59]

- Exploiting Deamons Running as Privileged Users: deamon is a particular background process, runs by default as root. Once deamon is compromised, attackers can upload local exploits and gain full access. [1, 61]

File permission

Be default, once a file is created, read permission is enabled for owner, group and other users. Hence, if permissions are not set carefully, hackers can steal sensitive information from other account’s files. Below is a file with default permissions created from an unprivileged account. [1, 63]

```
# touch file1
# ls –l
# total 0
-rw-r--r-- 1 test users 0 Aug 8 11:29 file2
```

Notice that although file1 Is owned by test, everyone can read it. This is not confidential at all.

Local passwords

There are a couple of different authentication methods for Linux, but local password is still the most common method. Local passwords are stored in /etc/passwd directory and encrypted using DES, MD5, Blowfish or some kinds of password salt.

DES is the oldest and least secure method because it only allows 8-byte password. Blowfish is the new one and securer. MD5 usage is causing controversy because it links to brute force attack. Password salt adds more complexity to password hashing, which increases protection against brute force attack. [1, 80]

4.1.2. Data network security

Network visibility holes

The aim of network visibility is to monitor and troubleshoot network traffic. Lack of network visibility does not cause exploits to the system, but if there is any vulnerability, system is likely to be exploited. The system should be supervised by an intrusion detection system (IDS), an application that analyzes a network traffic for malicious activities. [1, 89]

Network and systems profiling

The first stage of network-based attack is to find out target's information: OS, patch, version, application, port. When the information is collected, it is easier to implement attacks. The process is simplified and involved traffic is minimized, which helps to avoid IDS detection. [1, 94]

- Banner grabbing: is a method to collect system information on its open ports. Some common service ports are 80, 21, 25 used by HTTP, FTP, SMTP respectively. Telnet is the popular tool to perform banner grabbing. For example, one can establish a Telnet connection to a target website, then an SMTP request is sent. The response will include information about system service. Once attackers identity system and version, they can hack into the system.

- System fingerprinting: is the technique of identifying a targeted network's node underlying operating system by sending several packets to the system and analyzing the response. Fingerprints can contain TCP/IP stack behaviors (IP, TCP, UDP, and IMCP protocols), error messages, open ports, TTL values. Httpscan, amap and nmap are common tools used for fingerprinting. [7, 210]

Network architecture

While there are a numerous number of security devices and software, it does not guarantee enough security for the whole system. Security must be first considered from network topology

- Weak network architecture: Design a secure network architecture is often underestimated and ignored. Security relies too much on firewalls, but not things behind them. Weak network architecture can lead to the possibility that hosts are hacked, then so are adjacent hosts.

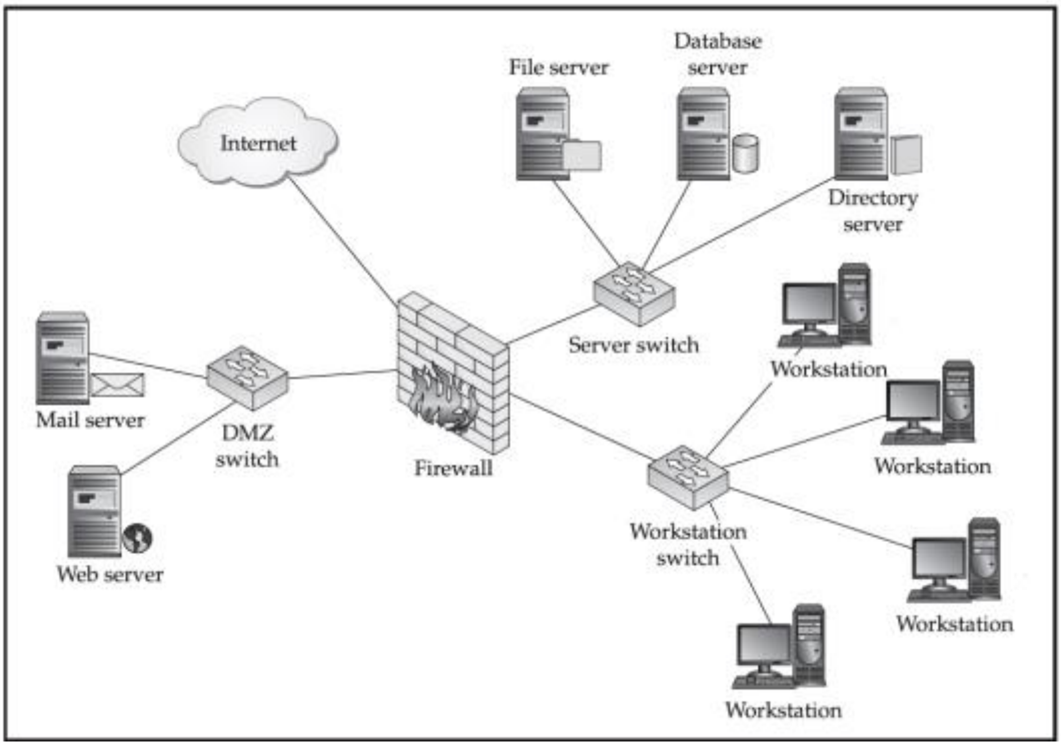


Figure 3. Traditional network topology. [1, 99]

Figure 3 illustrates a network divided into 2 separate small parts: a DMZ and an internal network. Once an attacker can compromise one server or workstation, he can access to neighbor server or workstation.

- Compromising extraneous services: The more services a system has, the easier attackers can file vulnerabilities. Some services can be used to exploit and some can facilitate snooping. Hence, minimize unnecessary that a system has can reduce the chance for hackers to compromise the system. [1, 103]

Clandestine administration

- Backdoors: a backdoor is a mean of accessing the system by bypassing system authentication. Backdoor is often secret and unauthorized, so it is usually dangerous. Backdoor can be a part of a software, a separate application or hardware feature. Backdoor can be installed by administrator in case of troubleshooting or other purposes. However, it is often used by attackers who compromised the system before. It is the way they hack into the system next time. There are instances that some hackers deliberately make backdoors for their own software to steal victim's information. [1, 111]

- Rootkits: a rootkit is even worse than a backdoor. It consists of many malicious applications combined into a kit. Rootkit in general a kind of software package that maintains root privilege. Rootkit installation can be made when intruders get access to the system. Then rootkit hides inside the system and maintain privileged access. It is difficult to recognize rootkit and removing it is a challenge or impossible. [1, 113]

Input/Output devices

Today every computer supports Bluetooth interface, which allows wireless keyboard, headphone, microphone to connect to it. However, that interface is an ideal route for hacker to compromise computer system. [1, 289]

- **Faking devices entities**

Any Bluetooth device has an assigned name for connection. When you want to use your device, you choose the one has the device's name. An intruder can make use of this feature to overcome security by setting an identical name for his faking device.

- **Eavesdropping on wireless communication**

Information when exchanging from one device to another one via Bluetooth is not encrypted by default. The data can be eavesdropped during that process. This is a very serious problem when transferring confidential information.

- **Information gathering**

An attacker can list all Bluetooth devices around by issuing a device inquiry. On Linux, hcitool is used to list all devices that answer the device inquiries.

4.2. Hacking the user

4.2.1. Web application hacking

Enumeration

- Passive profiling and intelligent scouting:

If a hacker wants to carry out an attack to a specific company, first he will collect passive information leaked about the company from public sources: organization, personnel, system. Organization information can be hierarchical, department, location, customers, investors, financial report. collecting from the company's official website or any relating webpage. Personal information relates to employee's information: name, email, salary, home address, position, training, relationship. That information can be searched from Facebook, LinkIn, MySpace. System information associates with website domain, hosts, ports, servers. It can be easily found from WHOIS database such as RIPE, ARIN, and APNIC. [1, 367]

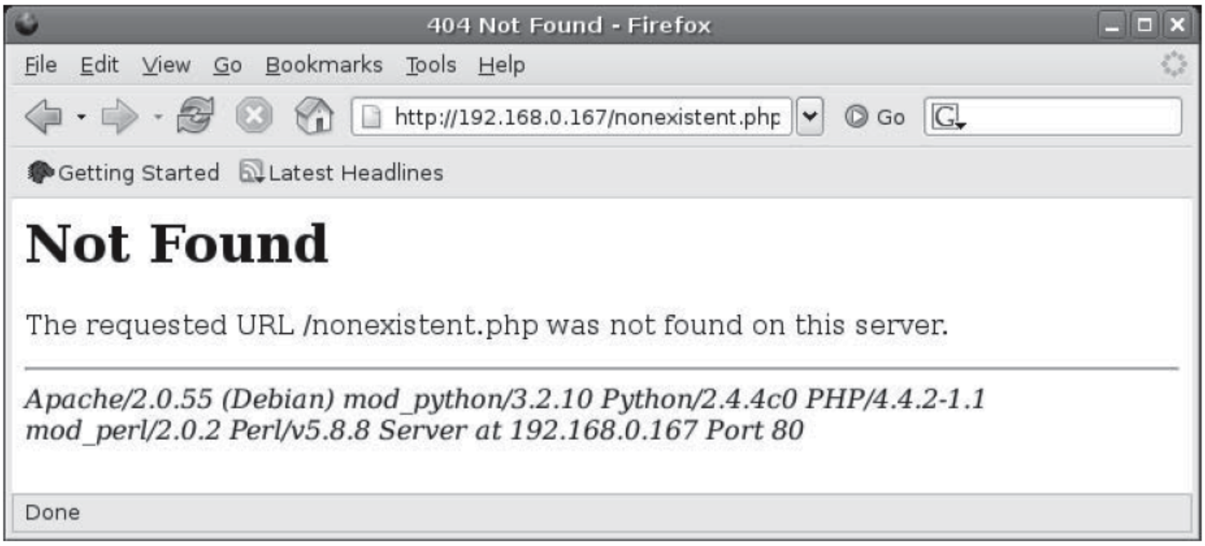
- Active web application enumeration

The next phase will be using active enumeration techniques to find out the vulnerabilities from all existing information. It can be done by scanning ports and services by using a popular port scanner such as Unicorn Scan, Amap, and Hping, or analyzing fingerprint of the target by using mapping application like Nmap. [1, 370]

Access and control exploitation

Many web application reveals sensitive information which is helpful for administrator when troubleshooting the problem, but also useful for hackers who want to find system's exploit.

- Poor error handling: If a webpage shows an error message, that message might contain information about system, version, database. That information is then used to guide attackers to attack more accurately. Figure 4 illustrates a webpage with a typical message:



Not Found

The requested URL /nonexistent.php was not found on this server.

Apache/2.0.55 (Debian) mod_python/3.2.10 Python/2.4.4c0 PHP/4.4.2-1.1 mod_perl/2.0.2 Perl/v5.8.8 Server at 192.168.0.167 Port 80

Figure 4. Default Apache error message reveals web server type and version.

- Comments in code: using comment is a good practice because it helps other team members to understand the code and debug the program. However, when the web code comes into a final product, all comments should be removed to prevent information leakage. Comments in HTML may reveal versions, types, developer information, even usernames and passwords.

- Misconfigured web servers: a web server that is not configured securely can be a vulnerable to various attacks. By default, web server configuration has many unsecure settings. For example, hackers can enumerate directory structure, and browse files and folders. [1, 380]

Insufficient data validation

HTTP headers and parameters are sent to server by web application when user makes a request from web client to web server. After that, client receives response from the web application. If the header is not validated, a number of vulnerabilities may appear such as SQL injection, XML injection, cross-site scripting and HTTP response splitting. The consequences are serious because the database can be compromised and authentication credentials are stolen.

- SQL injection: Currently, SQL injection is one of the most popular techniques to hack a website. It exploits a security weakness in a system and makes unexpected results. For example, by injecting malicious SQL commands into input, hackers can login without a username and password and retrieve information from the website. Any web browsers can be used for that intention, such as Internet Explorer, Firefox, and Google Chrome. Consequences of SQL injection attacks are very serious because attackers have the full rights to use the database. He can execute any commands, including dropping the database, which results in deleting all information of the website. [2, 291]

- XML injection: Similar to SQL injection, XML injection uses invalidated input to modify the XML structure, thus performing unauthorized actions or accessing sensitive data. [2, 384]

- Cross-site scripting: cross-site scripting (XSS) is the most common and serious type of injection attack. Malicious scripts are injected and sent to end users in the form of browser side script. By using XSS, hackers can steal secret information from users such as bank information and user account. [2, 442]

- HTTP response splitting: HTTP response splitting often relates to weak website infrastructure, where data is sent to HTTP header without being validated sufficiently.

Man-in-the-middle

Man-in-the-middle (MITM) attack refers to the attack that hackers alter information between two parties who believe they are communicating with each other. Some programs that enable MITM attacks to be carried out including arpspoof, dnsspoof, webmitm, dsniff, and webspy can be found in dsniff package on Linux. [2, 566]

- DNS spoofing: when a user wants to connect to a website, a DNS request is sent to the configured DNS server. After that, DNS server sends back to the user an IP, which is used to connect to website’s URL. Web browser downloads requested webpage from that address. By making in advance a website that has a similar to that page, the hacker can perform a ARS spoofing attack to obtain DNS requests from the victim and send back an IP that links to his fake webpage. At the same time, a denial of service attack is performed to DNS server to guarantee that the attacker’s reply comes first to the victim than DNS server’s reply. It is very serious in case of banking website with confidential information.

- Unencrypted attacks: HTTP protocol was initially a stateless protocol, which means connections to web server are destroyed when the request is returned. Each request is dependent and not related to the previous ones. In each session, information or status or user is not stored by the server. When web application became more complex, session identifiers and cookies were used to retain the internal state on the server. The session identifiers are sent for each request so that the web server can decide whether a user can access to his request or not. If HTTP traffic is not encrypted on local LAN, then MITM attack is performed to capture the entire sessions. Those sessions may include information about username, password, and any sensitive information about users. Afterwards, hackers can log in to user’s account. Even if the http traffic is encrypted with basic authentication, attackers still decode the data to gain authorized access.

- Insecure cookies: some websites use both HTTP for nonsensitive information and HTTPS for sensitive information to be transferred. Typically, only information via HTTPS is strongly encrypted and it is difficult to hack information from here. However, if a web application uses the same cookie for both HTTP and HTTPS, cookie can be stolen in HTTP and hacker can use that cookie to perform a session in HTTPS. Cookie generated only by web application itself is not considered secure enough.

- Fake SSL certificates: When user request a website via HTTPS, given that the encryption is secure enough, the web server will send a SSL certificate signed by a trusted Certificate Authority (CA). The browser will check this SSL certificated from the CA database to ensure that the requested page is truthful. If an attacker is doing an APR spoofing on the local LAN, and at the same time a user is sending a request via HTTPS, the attacker can intercept the request and make a fake SSL back to user to convince him to trust his fake webpage. In this case, although SLL is not signed by any CA and web browser will display a warning about that, the user sometimes will ignore that warning and continue surfing the fake page.

- Weak Cipher and Encryption Protocols: Cipher suits decide what algorithm is used to perform encryption. For example, the RSA_WITH_RC4_128_MD5 cipher suite uses RSA for key exchange, RC4 with a 128-bit key for bulk encryption, and MD5 for message authentication.

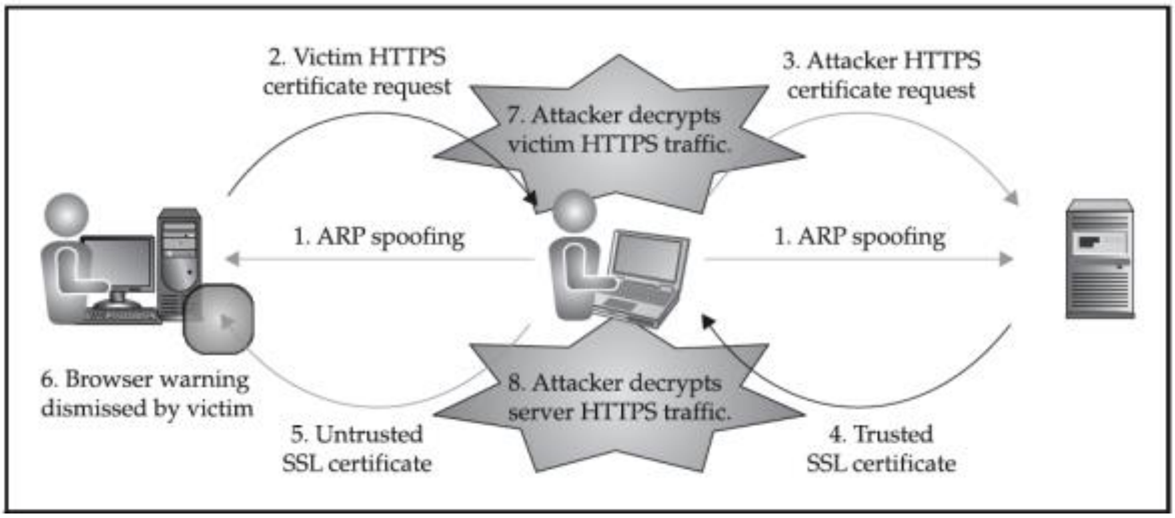

2. Victim HTTPS certificate request
3. Attacker HTTPS certificate request
7. Attacker decrypts victim HTTPS traffic.
1. ARP spoofing
1. ARP spoofing
6. Browser warning dismissed by victim
8. Attacker decrypts server HTTPS traffic.
5. Untrusted SSL certificate
4. Trusted SSL certificate

Figure 5. MITM attack carried out on an HTTPS connection. [1, 420]

However, common flaws appear in the web server using weak cipher suits, for example, in figure 5. A weak cipher suits include keys smaller than 128 bit, which can be attacked by brute-force attack to decrypt the information. Even a cipher suit has every key is more than 128 bits, some old protocol which was considered to be secure now has been compromised. For example, SSLv2 was once preferred encryption protocol, is now vulnerable to downgrade attack.

4.2.2. Mail services

Email is the most common mean of communication in the internet nowadays. Therefore, mail services are the main target from hacking activities. There 3 main types of mail attacks based on their objectives.

Fraud

- SPAM: spam relates to unsolicited messages, especially advertisements. It annoys users, violates user's privacy policy and consumes a lot of storage capacity. It may contain malware or links that lead to dangerous websites.

- Phishing: Phishing is a kind of fraud email that lures customers to go to visit a website, where they will be asked to update personal information. Phishing emails try to steal confidential information such as credit card information, social security number, email log in information.

- Computer viruses and other malware: Malware is a broad category of software designed to infiltrate or damage a computer system. Types of malware include spyware, adware, Trojan horses, Worms, and true viruses. Malware is commonly attached to SPAM. A computer virus is a self-replicating computer program written to change the way a computer operates, without the permission or knowledge of the user.

- Open relays: An open relay is a misconfigured mail server that allows third parties to send or receive emails to any email domain. Spammers always search for open replays, and once they find them, they will abuse them to send SPAM in a large volume.

Alteration of data or information

- Sender validation/impersonation/repudiation: sometimes, it is very important to make sure that the sender is real. There is no great way to confirm the authenticity of that email than call back directly to the sender.

- Root Privileges: permissions and privileges should be carefully considered in emails. Most mail daemons need to be executed as privileged users

Denial of service or availability

- Lack of redundancy: there are many reasons for the unavailability or network for some time: network failure, DoS attack, human error or hardware failure.

- Brute-force logins and password reset questions: brute-force attack is the simplest and oldest method to steal passwords from login information. However, sometimes it is the most effective method. Hackers try to guess passwords from the information they enumerate from users. The weaker the passwords, the more easily hackers can find out them by some combination tools. Sometimes, it is even easier for hackers to gain passwords by answering passwords reset questions. Those questions can be guessed straightforwardly if hackers know some basic personal information of victims. [9, 107]

5. Current techniques to implement security on Linux

5.1. Password security

5.1.1. Set BIOS password

It is recommended to disable boot options from BIOS and set a password for BIOS. This will help to prevent hackers from changing BIOS setting and using hacking tools on removable media or optical disk through boot options.

5.1.2. Figure out a strong password

General rules are: [5, 377]

- The length should be at least 8 characters
- There is at least one capital letter and one special character
- Password must not include easily guessed information such as name, birthday, home address, and phone number

By default, Linux allows a user to set a 6-length digit password. To prevent users from setting a short password, administrators can edit file name “/etc/login.defs” and change the following line:

```
PASS_MIN_LEN 6
```

To

```
PASS_MIN_LEN 8
```

5.2. System configuration

5.2.1. Root account

Administrators sometimes forget to log out from the system after finishing their work. In order to solve this problem, a bash shell can be created to auto log out after a certain time of no activity. A variable named “TMOUT” is added below the text HISTFILESIZE in the file “vi/etc/profile”. TMOUT is counted by seconds. If TMOUT = 7200, it means 7200 seconds, equal to 2 hours. After 2 hours of inactivity, root account will log out itself.

5.2.2. Disable console program access

By default, any user can shut down or restart the system from console. It is advisable to disable console-equivalent such as shutdown, reboot, hall, power off in the server by running the following line:

```
rm -rf /etc/security/console.apps/reboot
rm -rf /etc/security/console.apps/halt
rm -rf /etc/security/console.apps/poweroff
rm -rf /etc/security/console.apps/shutdown
```

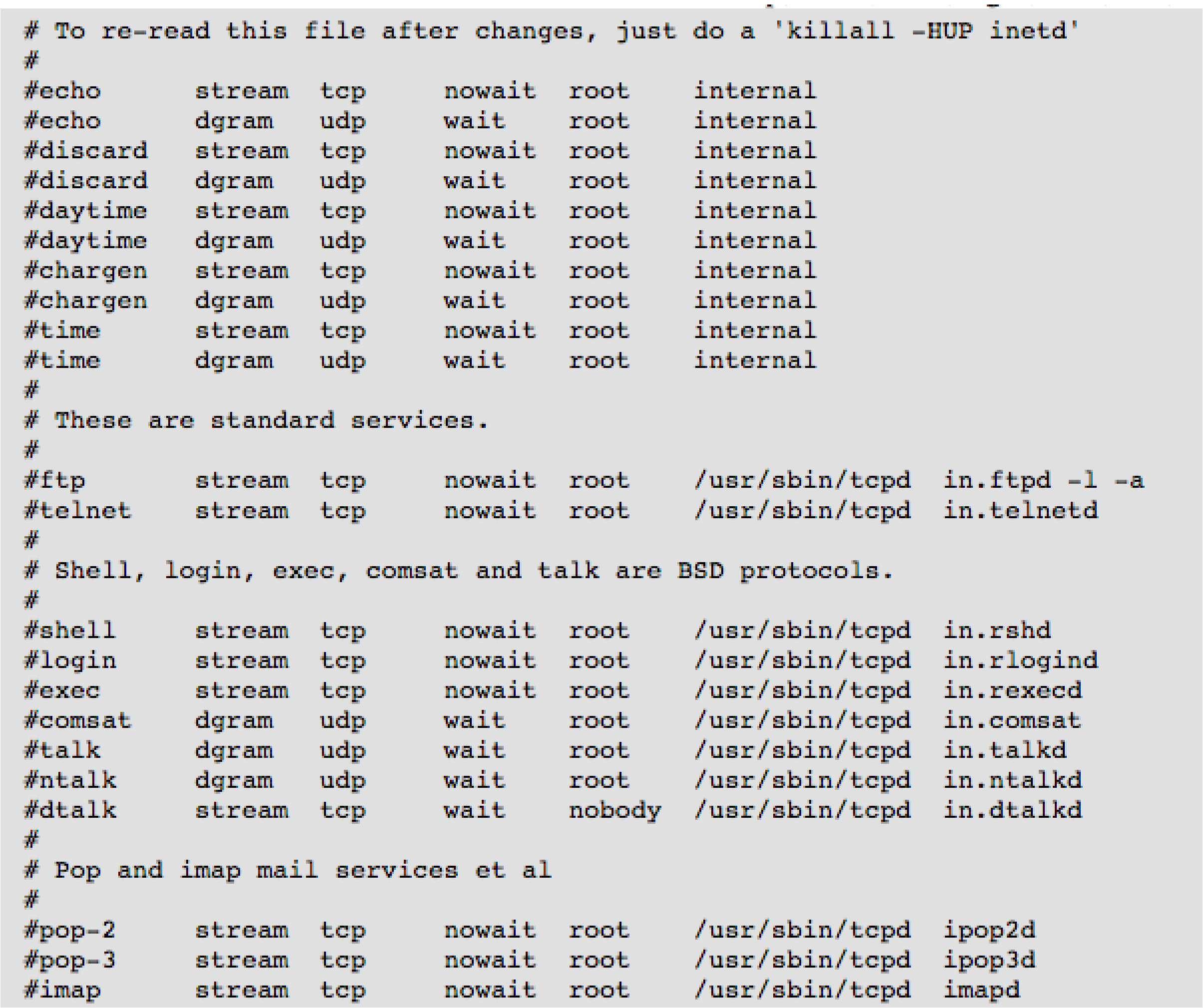
5.2.3. /etc/inetd.conf

Inetd, or supersaver is a service provider that offers programs based on requests from network. Inetd.conf file informs inedt what port to use and what sever to begin for each port. It is necessary to disable what services system does not offer and disable them by adding comments or simply adding "#" character in the beginning of the line. Therefore, attackers have less chance to file vulnerabilities in the system. After disabling unnecessary services, for example, in figure 6, SIGHUP command should be executed to update inetd.conf file. [15]

Step 1: change permission of /etc/inetd.conf file
```
#chmod 600 /etc/inetd.conf
```

Step 2: Make sure that owner is root user
```
#stat /etc/inetd.conf
```

Step 3: Edit /etc/inetd.conf file to stop redundant services



```
# To re-read this file after changes, just do a 'killall -HUP inetd'
#
#echo      stream  tcp     nowait  root    internal
#echo      dgram   udp     wait    root    internal
#discard   stream  tcp     nowait  root    internal
#discard   dgram   udp     wait    root    internal
#daytime   stream  tcp     nowait  root    internal
#daytime   dgram   udp     wait    root    internal
#chargen   stream  tcp     nowait  root    internal
#chargen   dgram   udp     wait    root    internal
#time      stream  tcp     nowait  root    internal
#time      dgram   udp     wait    root    internal
#
# These are standard services.
#
#ftp       stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd -l -a
#telnet    stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
#
# Shell, login, exec, comsat and talk are BSD protocols.
#
#shell     stream  tcp     nowait  root    /usr/sbin/tcpd  in.rshd
#login     stream  tcp     nowait  root    /usr/sbin/tcpd  in.rlogind
#exec      stream  tcp     nowait  root    /usr/sbin/tcpd  in.rexecd
#comsat    dgram   udp     wait    root    /usr/sbin/tcpd  in.comsat
#talk      dgram   udp     wait    root    /usr/sbin/tcpd  in.talkd
#ntalk     dgram   udp     wait    root    /usr/sbin/tcpd  in.ntalkd
#dtalk     stream  tcp     wait    nobody  /usr/sbin/tcpd  in.dtalkd
#
# Pop and imap mail services et al
#
#pop-2     stream  tcp     nowait  root    /usr/sbin/tcpd  ipop2d
#pop-3     stream  tcp     nowait  root    /usr/sbin/tcpd  ipop3d
#imap      stream  tcp     nowait  root    /usr/sbin/tcpd  imapd
```

Figure 6. Unnecessary services in /etc/inetd.conf. [15]

Step 4: Update changes
# killall -HUP inetd

Step 5: Set inetd.conf immutable
# chattr +i /etc/inetd.conf

5.2.4. TCP_WRAPPERS

By default, Linux allows all hosts from outside. To allow only reliable hosts, all hosts must be denied in “etc/hosts.deny” file and reliable hosts must be listed in “etc/hosts.allow” file.

Step 1: Deny all hosts in hosts.deny by editing /etc/hosts.deny file and add the following line
ALL:ALL@ALL, PARANOID

Step 2: Add reliable hosts by adding them into /etc/hosts.allow
sshd: 192.168.1.100 gate.com

Step 3: use tcpdchk to check tcdp wrapper configuration and find errors if there is any

```
# tcpdchk
```

5.2.5. /etc/services

/etc/services file allows the server and client programs to change name services to port numbers, where standard services are offered. Each host keeps the list stored in the file /etc/services. To implement security, the file must not be changed. To make it immutable, the following command is executed: [16]

```
#chattr +i/etc/services
```

This prevents adding or removing services without unauthorized permission.

5.2.6. /etc/securetty

TTY is a device that supports tele typing, for example, terminal. The file /etc/securetty defines what TTY device root users can log in. Any unused tty device can be disabled by adding # character before that device. For example: [17]

```
#tyy1
#tyy2
#tyy3
#tyy4
tyy5
```

In the above system, root user can log in into tyy5. Typically, there should be one tyy device allowed, and use su command to get root permission if more tyys are needed. All tyy devices must be logged in as root.

5.2.7. Default accounts

Another important thing is to delete default accounts from manufacturers after each upgrade or new software installation, which helps the access to system become easier.

Some common usernames that are not necessary:

```
#userdel adm
#userdel lp
#userdel sync
#userdel shutdown
#userdel halt
#userdel news
#userdel operator
#userdel golpher
#userdel ftp
```

Some common groups that is not necessary:

```
#groupdel adm
#groupdel lp
#groupdel news
#groupdel uucp
#groupdel ppusers
#groupdel dip
#groupdel slipusers
```

Prevent user from becoming root user with “su” command.

“Su” commmand allows users to become temperary root users. If nobody is allowed to become root user or only somebody is allowed to do so, the two following lines are added to the beginning of the file ”/etc/pam.d”

```
auth sufficient /lib/security/pam_rootok.so debug
auth required /lib/security/pam_wheel.so group=wheel
```

5.2.8. Protecting directories on web server

Many web servers like Apache uses .htacess file to protect directories on the web server. If a directory has a .htaccess file inside it, when the user wants to open the directory, webserver will pop up a dialog box to ask for a username and a password to log in. Only right username and password can help to see what is inside that folder. Username and password are listed in .htpasswd file.

Step 1: add username and password for .htaccess

```
htpasswd -c ./users test
```

```
New password: pass
Re-type new password: pass
Adding password for user test
```

Username and password will be saved with the following syntax:

```
 <username1>:<encrypted-password1>
<username2>:<encrypted-password2>
...
<usernamen>:<encrypted-passwordn>
```

Password is often encrypted by DES algorithm. Des is commonly used in *nix environment, especially in /etc/passwd or /etc/shadow. DES is very difficult to be hacked.

Step 2: Make a .htaccess file with the following content

```
AuthName "Restriced area"
AuthType Basic
AuthUserFile /somepaths/users
require user test
# add the following lines to prevents users from downloading
#.htaccess and .htpasswd files
<files .htaccess>
Order allow,deny
Deny from all
</files>
<files .htpasswd>
Order allow,deny
Deny from all
</files>
```

Step 3: upload .htaccess file into protected directory and set chmod 644 permissions for .htaccess file and users.

5.2.9. Improve security for kernel

Original kernel should be patched to improve security for Linux. The patch version has the following advantages:

- Protect system files from changing even with root user.
- Protect important processes from “kill” command

- Warn administrator when the server is scanned or any task breaks the defined rules

Some common used patches for kernel are LIDS, Medusa.

5.2.10. Safe transaction

Many traditional unencrypted server-client protocols are used by internet services such as TELNET, FTP, RLOGIN, HTTP, and POP3. Data package when transferred by those protocols are easily stolen somewhere between two ends. Some remote administration tools, for example linuxconf, rlogin are not encrypted, too. To improve security, HTTP/POP3 should be provided via SSL, or telnet, rlogin should be replaced by SSH.

5.2.11. OpenSSH

OpenSSH is an open source program to encrypt communication between hosts by using secure shell (SSH). It is a safe replacement for telnet, rlogin, rsh, because it always encrypts all communications and hides usernames, passwords for remote logins. Data transferred between two hosts are also encrypted. Although it is built on OpenBSD platform, it can be compatible with many OS from Unix family: Linux, Solaris, MacOS X, HP-UX. An OpenSSH package includes: [6, 139]

- OpenSSH Client (ssh): used for remote log in with encryption for each login session.
- Secure Copy Program (scp): supports for copying files from different hosts with username and password
- Secure File Transfer Program (sftp): used for secure FTP usage
- OpenSSH Deadmon (shhd): set OpenSSH to run in daemon mode

Some outstanding features of OpenSSH:

- Strong encryption: OpenSSH uses triple DES (3DES) and Blowfish encryption methods. 3DES uses longer key length than DES, thus reducing the chance for hackers to break the key. Blowfish fastens the encryption process.
- Strong authentication: OpenSSH uses Public Key, One Time Password (OPTs), Keberos mechanism. They help to fight against intrusion techniques such as IP Spoof, DNS Spoof, fake router…
- Encryption for port forwarding: Allowing port forwarding to another system via an encrypted channel. OpenSSH is used on internet protocols that does not encrypt data, such as SMTP, POP, FTP, and Telnet.

5.3. Monitoring and analyzing log file

In a Linux system, the log file is very important. It acts like a “black box” of the system with very detailed information. For example, bash log takes not every command used with shell bash. Some Unix versions keep log files in /usr/adm. In some newer versions, log files are kept in /var/log. Here are some common log files and their functions: [6, 110]

acct or pacct: commands used by all users
aculog: dial-out-modem activities
lastlog: the last successful login and failed logins
login log: bad attacks (attempt to log in but not successful)
sulog: commands of SU
utmp: current user logins, logouts, system events
wtmp: show history of utmp
kern: errors and status updates when OS is booted
cron: cron job log
httpd: apache errors and access log
mysql: MySQL database log file
maillog: mail server log
vold.log: errors of multimedia devices: CD ROM, floppy disk
xferlog: FTP activities

Other log files’ functions are written in /etc/syslog.conf

To ensure security, recognize and prevent attacks soon enough, users should check login time after every login. If the time does not match, it means somebody else is using the account. The user must change his password immediately and inform the administrator. Unfortunately, last log file’s content is replaced with every new login session. Therefore, backing up the file log can help to overcome the problem. It is recommended to save log file every 6 hours.

```
mv /var/adm/lastlog.3 /var/adm/lastlog.4
mv /var/adm/lastlog.2 /var/adm/lastlog.3
mv /var/adm/lastlog.1 /var/adm/lastlog.2
cp /var/adm/lastlog /var/adm/lastlog.1
```

5.4. System upgrade and installation

5.4.1. Linux OpenSSL Server

Almost every software such as Samba, FTP and Apache require user authentication before the user is allowed to use them. However, authentication information is not encrypted and can be read or changed by attackers. SSL encryption will ensure the information security with a huge range of crypto functions. Once OpenSSL is installed on Linux, it can be used as a tool for other software to utilize. [3, 142]

5.4.2. Firewall

Firewall has always been playing an important role in maintaining security for network security. Firewall is a combination of rules, applications and policies to ensure that users are safe from outside attacks. There are two kinds of basic firewall architectures: proxy/application and filtering gateway firewall. The modern firewall system is a hybrid of those two kinds of firewall.

Many companies and service providers use Linux as Internet gateways which are served for mail, web, ftp or dialup work stations. Furthermore, they also operate as a firewall, implement policies between the Internet and company network. Flexibility of Linux makes it become a replacement for commercial operating system.

Standard Linux firewall features provided in Linux kernel are built from two components: ipchains and IP Masquerading.

Linux IP chains is a IP filtering mechanism. IP chains allow Linux configuration to become a filtering gateway easily. IP Masquerading, a network address translation (NAT) feature, allows to hide real IP of internal networks. To use IP chains, rules defining allowed or denied connection must be established: [5, 68]

- Accept: packet is allowed to pass to the appropriate chain
- Deny: packet is not dropped
- Reject: packet is not good and inform the sender via ICMP packet.
- Masq: used for IP NAT
- Redirect: send packet to somebody else for processing
- Return: stop the rule list

It is also possible to use commercial firewall products such as Firewall -1, XSentry Firewall, Raptor, Gateway Guardian or free, open source products such as Dante, TIS Firewall, and Tookkit.

Set up firewall (Iptables)

Iptables is an extremely strong data filtering firewall and free on Linux. There are two components as showed in figure 7: netfilers inside the Linux kernel and Iptables outside the kernel. Iptables is in charge of communication between users and Netfilter. Netfilter directs users’ rules to Iptables to solve. After that, Netfilter carries out filtering data packages. Netfilter works directly and fast inside the kernel without reducing system speed.

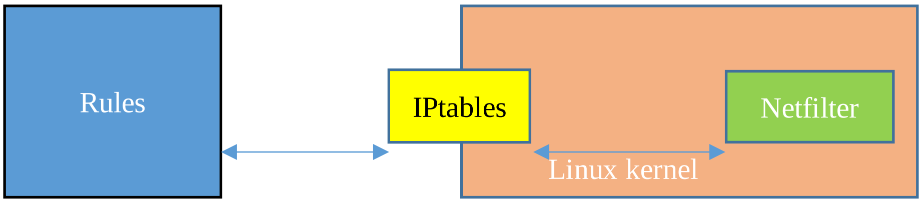


Figure 7. Iptables structure. [4, 11]

Iptables is divided into 4 tables as in figure 8: filter table to filter data package, NAT table to communicate with prerouting and postrouting chains, table to change parameter in IP package, and conntact table to monitor connections. Each table consists of many chains. Each chain comprises rules to communicate with data package. Rules can be accepting, dropping, rejecting, or referencing to another chain. [4, 10]

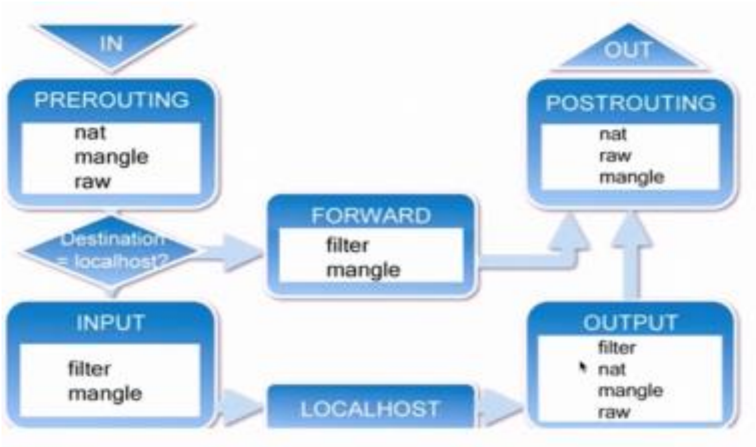


Figure 8. Table structure of Iptables. Copied from Ran 2014 [12]

To reduce the chance of DoS attack and improve host machine speed, load-balancing technique can be used:

Method 1: run host machines in different IP addresses

Method 2: place host machines in a DMZ network

5.4.3. Network Intrusion Detection System (NIDS)

If system is connected to the internet, it can be a target for finding vulnerabilities. Another problem is denial of service attack (DoS). Network intrusion detection system can track system status and recognize an intrusion or DoS attack. For example, NIDS monitors a large number of TCP request to many ports on some machines, thus finding out if somebody is trying to scan TCP ports.

NIDS can run in a specific machine or work dependently to monitor a whole network. Many NIDS programs can be used together to make a large secure system. For example, Tcpwrapper to control, accept registered services, Swatch to analyze system for suspected port scan, Sentinel to identify DoS attack or information stealing. When using NIDS, system performance should be checked by some tools such as ISS scanner, Nessus, CIS.

There are 3 kinds of NIDS:

- Network intrusion detection system: supervise data package on the internet and identify hackers who are penetrating into the system.
- Log file monitors (LFM): monitor log files created by internet services to find out intruders. For example, Swatch analyzes file logs of HTTP server to identify attacker based on famous exploits.
- Deception systems: contain pseudo services containing stimulate identified errors to trap hackers.

The following are some common NIDS programs:

Linux Intrusion Detection System (LIDS)

LIDS includes:

- A source code is integrated into the Linux kernel (patched into kernel’s source code).
- LIDS administration tool (lidasm) and some configuration files in /etc/lids/ to set up or change configuration of LIDS

LIDS features:

- Protect files, folders in any kind of file system (ext2, ext3, or vfat…) from unauthorized access, including root and system programs.

- Protect process from being killed by any users
- Control and prevent raw I/O by unexpected programs.
- Protect hard disk from being changing format

Snort

Snort is a program that identifies system penetration, analyzes real time traffic and logs packet. For example, Snort can recognize buffer over flow, port scanning, CGI attack, SMB exploration. Snort alerts real time for syslog; users' files, UNIX socket, Winpopup messages to clients. Snort has 3 operation modes: sniffer mode to read and decode packets and send them to stdout, packet logger mode to write packet onto disk with ASCII code, intrusion detection mode to identify attacks.

SXid

SUID/SGID files can be a danger for security issue. To avoid this problem, bit 's' is removed from programs owned by root. Nonetheless, other files can be set 's' bit without any notice. SXiD is a program that monitor suid/sgid system and is designed to run from cron. If there is any change in suid/sgid files and directories, sXid will inform the administrator by email or command line.

Portsentry

A firewall protects network from intrusion. Administrators can decide what ports are open and closed. Nobody else can know this information. However, hackers have some programs to scan ports and find out port information. Portsentry is designed to respond to identified scans by one of the following ways:

- Incidents are logged via syslog file.
- Target host is placed into /etc/hosts.deny
- Local host is reconfigured to remove data from the target

Tripwire

A Red Hat Linux server system after the first installation has around 30.000 files. If administrators cannot check the system's integrity and attackers can access to servers, file system can be changed without any notice. This is where Tripwire comes into play. It works at the most basic stage, protects servers and combines them into an integrated network. First, Tripwire scans one server and makes a database of file system, a kind of system snapshot. Administrators can configure Tripwire to monitor only system files or additional important files. Once database is created, administrators can check system integrity at any time. By scanning the current file system and comparing it to the database, Tripwire can identify and report any changes. If the change is accepted, the database is updated with new information.

5.5. Reaction when there is an attack

It is almost impossible to prevent all kinds of attacks, so the following procedures should be followed when there is an attack:

- Gather a team of experienced people to make a plan for protection
- Based on company or organization policies, inform related people about the attack
- Seek for help from Internet service providers or network security companies
- Use other means of communication to ensure that information is not leaked more.
- Record all activities: phone calls, file changes…
- Monitor important system by NIDS programs

6. Conclusion

The goal of this project was to gain knowledge about a Linux security mechanism as well as develop some methods to implement security on Linux. After being done, the paper has mentioned the following points: find out Linux OS, its file system, log in files; Linux security mechanism and how it is applied, especially journals and privileges; investigate attack methods used by hacker and then implement security methods to prevent those attacks.

However, due to the limitation of the time to study and implement the project, the topic implementation was not specific enough and some points were not explained clearly.

Therefore, further study is recommended for the following points:

- Implement security method in a case study
- Do research and develop more security services to make Linux system more stable and complete.

References

1. ISECOM. Hacking exposed Linux 3nd Edition. The McGraw-Hill Companies; 2008.

2. Dafydd Stuttard and Marcus Pinto. The Web Application Hacker’s Handbook: Finding and Exploiting Security Flaws, Second Edition 2nd Edition. John Wiley & Sons, Inc.; 2011.

3. Tom Adelstein and Bill Lubanovic. Linux System Administration. O’Reilly Media, Inc.; 2007.

4. Michael Rash. Linux firewalls. William Pollock; 2007.

5. Brent Chapman and Elizabeth D. Zwicky. Building Internet Firewalls 2nd Edition. O’Reilly Media, Inc.; 2001.

6. Rob Flickenger. Linux server hacks 3nd Edition. O’Reilly Media, Inc.; 2003.

7. Justin Hutchens. Kali Linux Network Scanning Cookbook. Packt Publishing Ltd.; 2014.

8. Daniel P. Bovet and Marco Cesati. Understanding the Linux Kernel, Third Edition. O’Reilly Media, Inc.; 2006.

9. Joseph Muniz and Aamir Lakhani. Web penetration testing with Kali Linux. Packt Publishing Ltd; 2013.

10. 10 Brian Ward. How Linux Works 2nd Edition. No Starch Press; 2014.

11. Jichiang Tsai, Chung-Hsin Feng, and Chuyuan Tsai. A network safety-defense mechanism with the Linux security module. TENCON 2006. 2006 IEEE Region 10 Conference; 2006. http://ieeexplore.ieee.org.ezproxy.metropolia.fi/document/4142570/

12. Ran Xing. Linux IPTABLES Firewall Basics [online]. URL: https://ranxing.wordpress.com/2014/11/11/linux-iptables-firewall-basics/
    Accessed 12 Jul 2016.

13. Introduction to Linux [online]. URL: http://www.tldp.org/LDP/intro-linux/html/sect_03_01.html
    Accessed 12 Jul 2016.

14. Ricardo Galli (2011, Dec). Journal File Systems in Linux *UPGRADE, 2*(6), 50-56.

15. Securing and Optimizing Linux: RedHat Edition -A Hands on Guide [online]. URL: http://www.tldp.org/LDP/solrhe/Securing-Optimizing-Linux-RH-Edition-v1.3/chap5sec36.html
    Accessed 12 Jul 2016.

16. Securing and Optimizing Linux: RedHat Edition -A Hands on Guide [online]. URL: http://www.tldp.org/LDP/solrhe/Securing-Optimizing-Linux-RH-Edition-v1.3/chap5sec40.html

Accessed 12 Jul 2016.

17. Securing and Optimizing Linux: RedHat Edition -A Hands on Guide [online]. URL:
http://www.tldp.org/LDP/solrhe/Securing-Optimizing-Linux-RH-Edition-v1.3/chap5sec41.html
Accessed 12 Jul 2016.