Improved File System Security through Restrictive Access

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Information Security

Submitted by

Navneet Kaur

Roll no 801433020

Under the supervision of

Dr. Maninder Singh

Associate Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA 147004

July 2016

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, *"Improved File System Security through Restrictive Access"*, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Information Security submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Maninder Singh*, and refers other researchers work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(Navneet Kaur)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Maninder Singh)
(Associate Professor, CSED)

Countersigned by:

(Dr. Maninder Singh)
Head,Computer Science and Engineering Department
Thapar University, Patiala

(Dr. S. S. Bhatia)
Dean (Academic Affairs)
Thapar University, Patiala

ACKNOWLEDGEMENT

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life. It is a great privilege to express my gratitude and admiration towards my respected supervisor Dr. Maninder Singh,(Head and Associate Professor of Computer Science & Engineering Department). I am highly indebted to him for constantly encouraging me by giving his critics on my work. I am grateful to him for giving me the support and confidence that helped me a lot in carrying out the research work in the present form.And for me its an honor to work under him. I am also heartily thankful to Dr.Jhilik Bhattacharya, PG coordinator, for motivation and providing uncanny guidance and support throughout the preparation of the thesis report.

I will be failing in my duty if I do not express my gratitude to Dr. S. S. Bhatia, Dean of Academic Affairs, for making provisions of infrastructure such as library facilities, computer labs equipped with net facilities, immensely useful for the learners to equip themselves with the latest in the field.

I am also thankful to the entire faculty and staff members of Computer Science and Engineering Department for their direct-indirect help, cooperation, which made my stay at Thapar University memorable. Last but not least, I would like to thank my family , friends for their wonderful love and encouragement, without their blessings none of this would have been possible.

Navneet Kaur

(801433020)

ABSTRACT

Security is a prime concern in today's era of technology when dealing with digital data. Information is managed by the file system which is the core layer of security in an Operating System. Due to lack of security at this layer, private information can be accessed by an intruder or in case of theft data can be read via mounting it on to a mount point and accessing the information. Other layer which is of similar importance is the Shell. It is a command language interpreter that takes input from the standard input device and acts as an interface to execute commands. Restricting shell access to minimal commands gives an extra level of security to the Operating System. If a shell is not restricted, intruder can gain access to the system and is able to execute all the commands. Shell is so powerful that intruder may control the whole system to execute the malicious code. Another concern area of security is user authentication. When a user login into the system, it proves its authentication by providing username and password to a server. Then server will validate user-authentication by matching its credentials. If server gets hacked then all the credentials will be stolen by hacker that can be used by hacker to gain access of a system. In this research, we propose a modular framework to enhance the security of file system where one of the module create secure and restrictive shell with minimum privileges given to any user to execute small set of commands, other module includes authentication of user by login into server using ssh login key-pair and another module provides restrictive file system access by encrypting the file system.

Keywords- Kernel Loading Modules, Shell, Chroot jail, System call, Secure socket Shell.

Table of Contents

List of Figures

Chapter 1

Introduction

Security is a prime concern area for protecting digital data. Each bit of data is managed by file system. So, file system is the first and by for most critical layer of securing operating system and user defined information. It's is a major research area for security researchers to provide security to data. The weakening of system's security invites cyber-criminals to breach the security of system. Shell is another critical part of security. Restricting shell gives extra layer of security to Operating system (OS). OS must protect itself from security breach, For example launching programs with excess privileges, access permissions, and stack overflows. Every OS contains these two essential parts such as Shell and kernel. This chapter describes about the existing file systems and security features provided by them. It briefly describes Research Motivation for carrying out research in this area and structure for the rest thesis.

1.1 File System

In this era of technology, everything is stored in the form of data as data contains crucial and sensitive information. Data should be managed in a proper way that is taken care by file system. File system is the way by which Operating system keeps tracks of the files stored on the disk or partition. If there is no file system data would be stored on large piece of body, in which there is no method to tell from where data

is starting and ending. So, file system is used to organize the data. There are different types of file system which have their own structure, logic to organize data, security, speed and other properties. Some file systems are designed for specific purpose. For optical disks ISO 9660 file system was designed. Every file system makes use of some storage device or storage media to store the data such as hard drive, magnetic tapes, floppy disks, flash memory, optical disks.

1.2 Types of Open Source File System

There are different types of open source file system which have its own login, structure, security and features.

1.2.1 EXT2

Ext2 is the second extended file system. It was the default file system in various Linux distributions such as Redhat Linux, Debian. This file system is used for flash drives such as USB drives and SD cards. Here data blocks are used basic unit to store data in files. All the data blocks are of same length. It has some limitations such as journaling is not supported by ext2 file system. Access control List was not included in this file system. No journaling feature was included in ext2 file system. No Access control lists (ACL) in the ext2 file system.

1.2.2 EXT3

Ext3 is the third extended file system. It is a journal based file system which is used by Linux kernel. File size supported by ext3 is from 16 GiB- 2TiB. Ext3 is enhanced version of ext2 in which new feature is added in the file system called as journal. It overcomes the need of checking file system after improper shutdown of system. In ext2 this was time consuming process. Journaling file system maintains a file called as journal or log which keeps all the changes made to the file system and it is use to avoid consistency in data due to improper shutdown of system, system crash or

power failure. Once the log or journal of the system is updated, then the changes are marked to that area of file system and entry is marked in the log to say that data is committed. If the system is crash down then the file system can be brought back to its original state by using journal.

Different Features of Ext3 File System

- **Data Integrity**

  Ext3 file system maintains consistency in data, when an improper shut down of occurs due to power failure or system crash. Ext3 partition are configured in such a way that they maintain data integrity by default.

- **Easy Transition**

  It is easy to switch from ext2 to ext3 file system to earn the benefit of "journal". It is possible because ext3 has identical hard-disk formatting as of ext2.

- **More efficient storage space**

  Those blocks which are not used by filesystem metadata can be "trim"or discarded by it using mkfs. There are advantages as well as disadvantages of Ext3 file system.

  The advantages of Ext3 file system is it supports journaling, it allows in place upgrade from ext2 without having to backup and restore data and the disadvantages is it dont support recovery of deleted files, no checksum in journal and File consistency check (fsck) time can be extremely long.

1.2.3 Reiser FS

ReiserFS is the journaled file system that was included in the Linux Kernel. In various Linux distributions it was the default file system. Maximum file size supported is 1 Exbibyte which is equal to 2^{60} bytes. UNIX permissions and access control list are features that are supported by this file system. Balanced tree is used by this file system. Reiser file system can make the storage of file more efficiently by

allowing the tails of multiple file to be stored on the shared inode. User can put 1, 00,000 files in a directory according to Reiser file system but many other file systems don't allow this.Extremely fast recovery in the event of unplanned machine shutdown. Generally high performance for all files sizes. Reiserfs takes few seconds to check the consistency of file system. The limitation of Reiser File System is that it doesn't support encryption and compression.

1.3 Physical Structure of Open Source File System

The Figure 5.5 shows physical layout of file system which consists of boot block, block group, Group Descriptor, Inode Table, Data block fields.

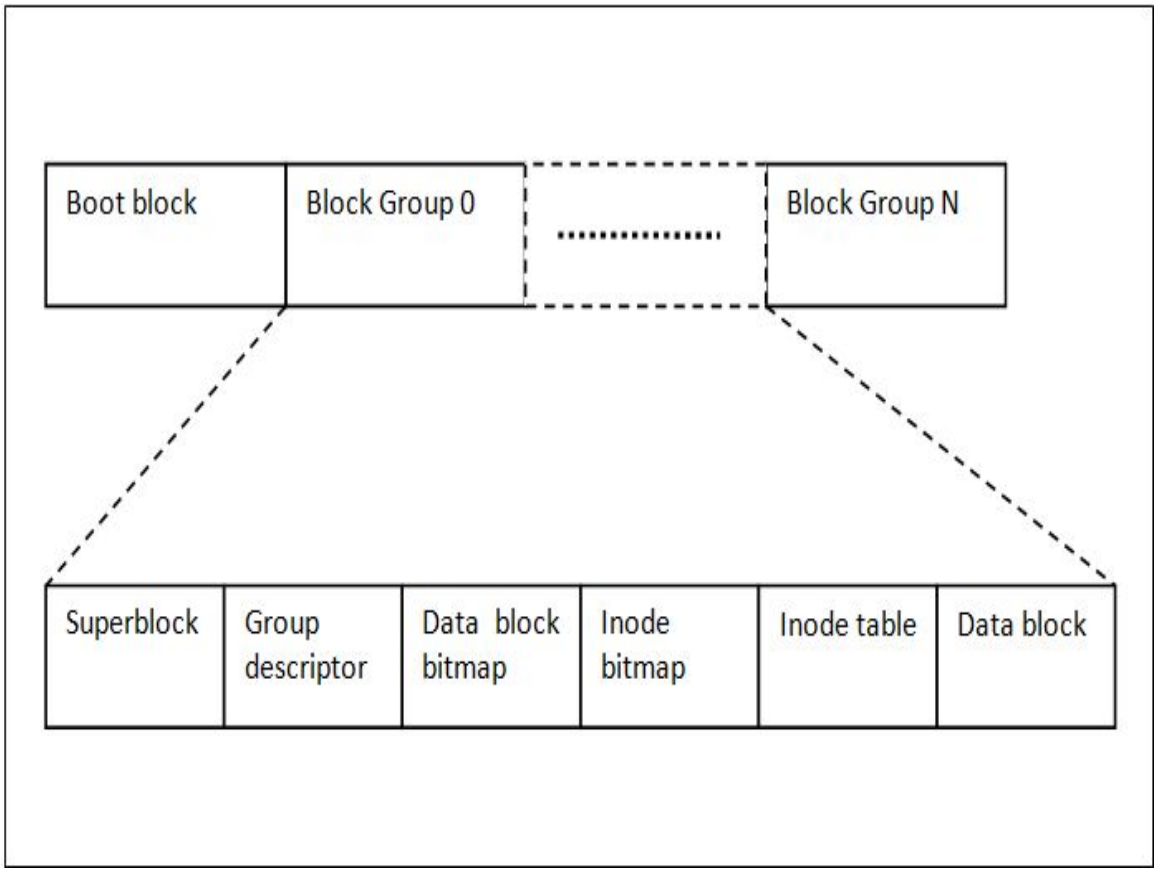


Figure 1.1: Physical Layout of Open Source File System

The blocks are combined together to form block groups. Every block group consists

of superblock, block group “descriptor table ”, “block bitmap ”, inode table, “inode bitmap ”and the “data blocks ”.

- **Boot block**

  Boot block is the first block on the first track located on the storage medium such as hard-disk, floppy disk. It contains Bootstrap program which is used to initialize the Operating System.

- **Super Block**

  Superblock is very crucial for the file system as it contains metadata of file system that's why many redundant copies of Superblock are stored in file system. If the superblock becomes corrupted then Operating system cannot mount the file system. First block group contains primary of “superblock ”. The time you mount the file system, Operating system reads the primary copy of “superblock ”. Therefore, primary superblock is stored in the block group zero. Super Block contains different number of fields such as filesystem type, block in the filesystem, number of free blocks, number of inode in single block group and, so on.

- **Group Descriptor**

  A data structure describes Each Block Group. Similar to Superblock, every Block Group is duplicated in each Block Group for all the group descriptors in case of a corrupted file system.

- **Block Bitmap**

  The Block allocations block number bitmaps for this Block Group. It is used during allocation and de-allocation of block

- **Inode table**

  Inode is a “data-structure”which signifies file system objects. When a file is created within directory, two attributes are assigned to the file such as, file name and inode number. When a user wants to access the file , the user will use the file name to access it but internally, the file name will be first mapped to

its inode number which is stored in inode table. Then the inode will be accessed by inode number. The mapping of inode with inode number is done by inode number. Access Control List, owner, group, number of free blocks, number of blocks allocated are contained within the metadata.

- **Data Block**

  This is the actual area where file's data is stored and data blocks are contained within the inode. If additional space needed then it will be allocated dynamically.

1.4 Logical Structure of Open Source File System

In Open source file system, everything is considered as file whether it a text file, directory, device driver, partition or image. Every file system has a control block which contains information of that file system while some other blocks such as inode contain information of files, and data blocks hold the data stored in it. File system hierarchy standard (FHS) specifies the structure of directory and contents of directory. All the files and directories are under the root directory (/). Figure 4.1 shows about the logical structure of Open source File system which contains following directories such as :

- **/–Root Directory**

  It is the parent of all directories and subdirectories. Under this directory, privileges are only given to the root user.

- **/–bin –User Directory**

  It contains all the binary files that are needed to execute the commands in single-user mode. For example, cat, ls, pwd, cp, ping. It includes the shells such as bash, csh, ksh.

- **/–home –Home Directory**

  It is a users home directory which contains users data, programs and directories. It contains the users configuration files required for applications.

- **/–dev –Device Files**

  This directory contains information about where device files are located. It contains the information about further partitions done on the master drive. Device file acts as interface for device drivers and in windows they are called special files.

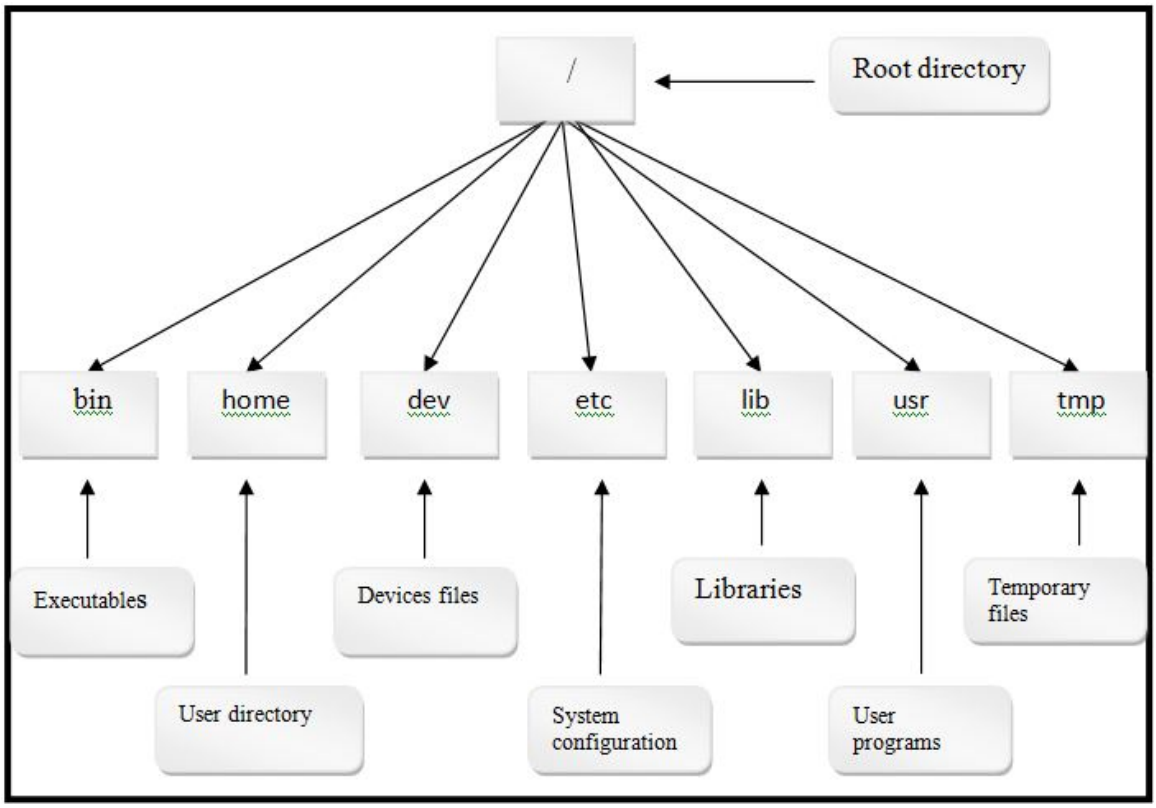


Figure 1.2: Logical layout of Open Source File System

- **/–etc –directory**

  It contains all the configuration files required by the system. All are text files and it doesnt contain any binary. It contains different types of text file such as /etc/passwd, /etc/fstab, /etc/init.d and /etc/shadow and, so on. /etc/passwd file contains all the necessary information of user such as password, registered user etc. /etc/fstab file holds information of all the devices which gets mounted when system boots. Shadow file stores users password in encrypted form.

- **/ –lib –Shared libraries**

  This directory contains all the files required to run the binaries in /bin and /sbin. These libraries are essential to boot the system and to execute the commands within root of system

- **/–usr –User programs**

  This directory is used for files that can be shared among different machines. Following are the subdirectories the /usr directory contains

  |–bin/

  |–etc/

  |–bin/

  |–local/

  |–tmp/

- **/–tmp –Library Files** This directory consists of files which are needed temporarily and used to store data temporary. At boot time most of the Operating systems delete the contents of this directory.

1.5 File System Security

File system assign different permissions for owner and user which inhibits the unauthorized access.

- **Umask Settings**

  When a new file is created by user in Linux Operating system, the files are created with the default set of permissions. For example, if a file has permissions 666 it means everyone is granted to write and read the file.

- **File Permissions**

  In Linux Operating system, when a file is created the system assign the permissions for the owner of file, members of the group and to others users. Permissions can be assigned to read, write and execute the file. The file permission can be changed by using chmod command.

1.6 File Data Security

Information in the computer system is stored in the form of files. File is the basic unit to store the information. In Unix-Operating systems, everything is considered as file whether it is a file, device, directory and executable. Securing data that resides in the file has become a security issue. There can be different methods which can be used to secure the data. Many people use very common utilities such as aescrypt or crypt to secure the files data. These utilities use filename and password to produce encrypted files. When a user wants to encrypt a file, every time he needs to give different password to each file and for the retrieval of file password should be remembered. This is somewhat cumbersome process because if user forgets its password, it cannot be recovered easily. User also needs to erase the plain text file from the hard-disk so that only one copy of the file remains and that will be too in the encrypted form.

Another approach to secure files data is by incorporating encryption engine in the utility or application software. For example, the time text editor is used, it can ask for a key to open a file and can encrypt and decrypt files data. The difficulty with this approach is every application has to use same encryption engine and any modification require changes in all. This approach has difficulties such as incorporation of encryption engine with all utilities or application software and if it requires any up gradation it to integrate again. The third approach is use hard-disk controller with embedded encryption hardware which can be used to encrypt entire disk. This approach is also somewhat cumbersome because of sharing keys among all system users.

So it seen from above described approaches that each of the approach has some disadvantages. These approaches can be inconvenient and cumbersome to the users. So, there is a need of mechanism which can secure the data in an efficient manner.

1.7 Motivation

In this era of technology, with the span of Internet number of users has increased exponentially, user stores their sensitive or confidential data either locally or on the cloud. The flaws in system security invite cybercriminals to breach the security of a system. Compromised security can lead to system's degradation. Computers do not incorporate the threat to its security but, when connected to Internet it evokes the chances of risk to computer system, as it contains crucial and private data of the user. Everything needs to be secure whether it is an Operating System or any other system. Operating system (OS) should protect itself from security breach, For example launching programs with excess privileges, access permissions, and stack overflows.

File system is one of the targets for an attacker to compromise the security. There can be different reasons to attack on file system. Modification in system programs and its database can be done by an intruder to have entry in future. System logs can also be altered or deleted by an attacker to avoid their track . UNIX file systems are exposed to threats in the presence of intruders, unofficial users, worms and viruses. Sometimes it is difficult to detect damage done to files, databases and programs. This gives rise in security risks because hackers or unauthorized users always want to collect data either for their financial benefit or to defame the company. Now a day, users secure their data either by encrypting it on the disk or by encrypting it on the wire. But due to weak security policies, hackers are able to decrypt it and can execute malicious code or shell scripts. To resolve this security issues and prevent the hacker from executing the malicious script we propose a framework to create going to create a secure and restrictive shell with limited file system access

1.8 Thesis Outline

Rest of the chapters in this dissertation organized as follows.

Chapter 2- This chapter explain file system security and other features. It briefly

shows the systematic review in the era of file system.

Chapter 3- This chapter explains problem statement and objective for this research.

Chapter 4- This chapter explains proposed framework to enhance the security of file system.

Chapter 5- This chapter explains results and discussion for the proposed framework that involves restriction user at console level, user authentication using SSH Key Login pair and encrypted file system thus enhancing the security of file system.

Chapter 6 This chapter explains conclusions and Future Work for this research work. It briefly explains future scope to provide protection to various systems by implementing this framework at various levels.

Chapter 2

Literature Review

Securing digital data has become an area of concern for protecting information against attacks. With the span of Internet, an unauthorized access, breaking through security devices, services and network has also increased which results in breach of security. Earlier intrusions in the system were made by the attackers with the motive to just break into the system but these days, attacks are done for financial, political and other objectives. Several survey reports on the computer security shows that breach in the security occurs frequently. This chapter provides systematic review in the area of File System. It briefly explains about various work done to enhance the file system security.

2.1 Systematic Review

File System Security is very important topic over the World Wide Web. The research in this area is unstoppable. Here, the survey has been stated in the area of file system from 1991 to 2012

2.1.1 Operating System Security

One of the prime concern areas of security is Operating System security because Operating System (OS) is responsible for managing all the resources (memory, space)required

to carry out all the operations, managing hardware such as routers, firewalls (network devices), PDAs, tablets, so on. And all the running applications in the system are supervised by the OS too [2].

Some open source and commercial Operating Systems are Linux, Mac, Microsoft windows. Operating System also plays a crucial role to maintain the security. The lack of security in Operating System can affect the security of the running applications on system. If any program is executed by unwanted or unauthorized user then it can cause damage to the security of system. Computer system should employ some security policies for OS.

The challenge in the Operating System is to maintain access for all users from inside of system. Operating System must take care of defining access, providing access to the users, permissions to file objects so that no violation can be made by any user [1].

Authentication is a mean to provide security to Operating System. Authentication is used to identify an authorized user of system. Operating system authenticates user by User-name/Password, Usercard/Key and by user attributes such as fingerprints, eye retina or signature.

2.1.2 Traditonal File System

There are different traditional file systems available such as FAT, NTFS, EXT2, EXT3 which have their own logical structure, speed, security features.

2.1.2.1 FAT File System

It is a File Allocation table(FAT) that is constructed for folder structure and small disks. Fat allocation table always present at the starting of the volume. To start the system properly, root folder and file allocation table should be placed at the same

location. The volume can be protected by keeping two copies of File Allocation Table if one copy gets damaged.

Root folder comprises of all the entries of file and folders on the root. Root folder and other folder differ from each other as root folder is present at the specific location having fixed size. On the objects of file system we use default permission entries, uniquely for the root and System folders. Fat file system was originally designed for floppy disks [3][4]

Security Issues

par Fat file system doesn't provide any security as no security features are deployed on individual files and folders. Permissions can be applied only on shared folder not on files and folders. Network access can be restricted by FAT file system. If any user is logging into system, he can access all the files and folders in FAT partitioning. Changes in the default permissions for root and system folder can reduce security and can evoke access problems.

2.1.2.2 NTFS File System

NTFS is a New Technology Filesystem that is used to store and retrieve files on the hard-disk. Maximum file size supported by NTFS is from 16TiB to 1KiB. NTFS included new features which were not present in FAT Filesystem such as ACL, transparent encryption. NTFS filesystem maintains Master File Table (MFT) which keeps the information about every file and folder stored on NTFS volume. Security Descriptor (SD) maintains information about security and permission.

Security Features

- **Encryption and Decryption**

  In NTFS file system encryption is implemented at both file and directory level.

It uses Encryption File System (EFS) to store encrypted data-files on volume or partition. EFS protect the files from unauthorized access. EFS encrypts file with symmetric key which is also called as File Encryption Key (FEK). EFS make use of symmetric encryption algorithm as it takes minimum amount of time to encipher and decipher the large amount of data as compared to asymmetric key [5].

Symmetric key is used to encrypt a file, that is, FEK is encrypted by a public key that is related with the user who encrypts a file. Encrypted FEK is further stored in the EFS with other data stream of the encrypted file. To decrypt the symmetric FEK, EFS uses private key of user. It is stored in file header and then uses symmetric key to decrypt the file. Encryption is transparent to the user who encrypts the file. The file system automatically decrypts the file when a user accesses it. File encryption is reapplied when the file is saved. An unauthorized user receives an access denied message when he tries to access the encrypted file or folder [5].

Permission in NTFS

NTFS file system store the shared folders on NTFS volume. It is necessary that the files must be stored on NTFS volume if permissions are to be given at file level [6].
ii. When a user is logged on to system, an access token is generated by the logon process to authenticate the user credentials. User's credentials are stored in authentication database. Access token (AT) contains security identifier (SID) of logged on user [7].

If user's authentication is successful, then security identifier is returned by the login process which will contain information about permissions given to user. For example, Windows want to assure that unwanted or untrusted user's cannot access the critical parts of Operating system which can lead to problem in the system. So,

windows manage all this with the help of Access token. AT can be used to recognize security context of a user and a process. There are two types of Access tokens namely; primary access token and impersonation token [8]. In Primary Access Token the security context of a process or thread is described by an object is called an access token. Every process is having a primary token that explains the user account security context related with the process. A system uses a primary token by default, when a process thread interact with a securable object. A primary access token is typically specified to represent a default information security for that process.and Impersonation Token is commonly used for client/server premises. It enables a thread to implement in a security context that varies from the security context of a process having a thread. A thread has both a primary token and an impersonation token if it impersonates a client [9].

NTFS permissions are managed by Access Control Lists (ACLs). ACL indicates which user or a group has the right to access or to modify a specific file. ACL comprises of Access Control Entries (ACE). ACE consists of SIDs, Flags, Access mask[10] To manage ACLs command line tools are available such as Xcacls.exe and Cacl commands. The permissions can be viewed, saved and restored on particular file, folder by using icacls.exe utility which is provided by Microsoft. commands permits to modify or to display Discretionary ACLs where as Xcacls.exe can be used to set security options for filesystem security. Shared and NTFS permissions could be given to the following folder types: Public Folder, Drop Folder, Application Folder and Home Folder. Shared and NTFS permissions are different for different types of folders [6]

2.1.2.3 EXT2

Ext2 is the second extended file system used in Linux kernel. In ext2 file system data-blocks are basic storage units for data-files. ext2 file system data blocks are used basic unit to store data in files. All the data blocks are of same length. It has some limitations such as journaling is not supported by ext2 file system. Access

control was not included in this file system [11].

2.1.2.4 EXT3

Ext3 is the third extended file system that was included in Linux kernel. It is a journal based file system. Ext3 supports maximum file size up to 2 TB. Journaling file system maintains a file called as journal. Any change in the data is first marked onto journal and then changes will be written onto disk. It is used to avoid inconsistency of data in case of system failure or improper shutdown of system.

2.1.3 Advanced File System Security

2.1.3.1 EXT4

Ext4 is the fourth extended file system supported with journaling. Maximum size of journal in ext4 is 2^{32} blocks. In ext4 file system storage space is divided into number of block groups. To improve the performance of ext4, the block allocator keeps each block of file within the same group to reduce seek time. In ext4 maximum file size is 16TiB and is able to store billions of files. Blocks are basic storage unit in ext4 file system [12] [13].

Features of ext4

- **Checksum in journaling**

  If inaccurate or fallacious data find its way to enter into journal, then inaccurate data will corrupt the file system even it supports journaling. To overcome this checksum was introduced in ext4 to assure that only valid changes are done in the file system [12].

- **Online Defragmentation**

  Although ext4 has a feature to lessen the fragmentation but sometimes it is unavoidable due to existence of file system for a long duration. Due to this reason, an online tool exists for defragmentation to defragment the individual file and file system to improve performance of system. An online tool transcripts

data files into new inode. Ext4 marks unused data blocks that exists inside the inode table so that fsck process skips them to fasten the fsck process of file system [12].

- **Extents**

  Ext3 is efficient in handling files of small size but cannot handle files of large size. This results in addition of extents in ext4 file system which is able to handle files of large size. Extents are continuous blocks on disk space that will keep small sized files together to reduce fragmentation. Extents maintains information about where a large list of continuous blocks are stored [12] [13].

- **Forward and backward compatibility**

  Ext4 file system can migrate to ext3 filesystem only if extents are not used by ext4 and similarly ext3 can also migrate to ext4 file system [12]

- **Multiple Mount Protection (MMP)**

  It is a feature that was included in ext4 file system. If multiple hosts try to use it simultaneously then it will provide protection to file system [13]

2.1.3.2 Encryption based File system

Thomas J. Roberts et. al [14] describes about encrypting the superblock data while storage. In this plain text of data is split down into buffer and the buffer is further split down into blocks. By making use of any method to encrypt block, the blocks of each plaint text data is encrypted into intermediate key by using first key. Then bytes of encrypted block are permuted by using second key to produce encrypted buffer. The output of permuted words is stored . Thus the permutation making difficult for unauthorized users to decrypt the data while storage.

As cryptographic techniques play an important role and provides security against unauthorized access still it offers some vulnerability. The first encrypted file system for UNIX Operating system was Cryptographic File system (CFS) given by Blaze et.

al /citep. The Cryptographic file system introduces a layer between the disk and the virtual file system. Users link cryptographic keys with the desired directories to be encrypted. CFS uses DES algorithm to encrypt the data of files. The files in the directories are encrypted and decrypted by using key without the user's intervention and clear text is not stored on the disk. It uses non-volatile storage media to store file and its metadata is in encrypted form only. The protection is provided to use it against some unwanted or unauthorized user who wants physical control of the storage unit.

The design of our Steganographic file system [StegFS] is based on the idea of Ross Anderson et. al [15], but this differs in efficiency and is more practical in use. It doesn't require separate partition table for of hard-disk instead it make use of unused blocks of volume or partition to place the hidden files into it, which also consists of normal files and are managed by standard file system. The researchers used separate allocation table for blocks which contains 128-bit of encrypted entries that cannot be distinguished from random bits. The StegFS is installed along with the Minix, ext2fs drivers between the virtual file system and block buffer cache. It uses unused blocks of ext2 to store the hidden files in blocks. The behavior of StegFS is similar to that of ext2. In this File system, the time the file is deleted its corresponding block will be overwritten by some random bits [16].

The issue of Cryptographic file system is management of key which is addressed by Self-certifying file system (SFS). It introduces the idea of self-certifying pathname which is needed to verify that the user is communicating with the legitimate server as self-certifying pathname has the server's public key which is secured by hash. The client in this file system is authenticated by using server's public key and has a secure communication channel. If the server is verified by the client, secure channel will be established between server and client to access the actual file. /SFS is the mount point used to access SFS remotely. Syntax for SFS pathname is /sfs/location: host-id/real/pathname, where location indicates IP address of server and host id is hash

of the server's public key. It is not concern of SFS that how the user is obtaining pathname; host-id will be obtained by using Public Key Infrastructure (PKI). Once path has been obtained, there is no need to remember public key by the user [17] [18].

Dm-crypt is the encrypting file system of Linux which is the part of standard Kernel. It makes use of device mapper to encrypt the block devices transparently by using Linux 2.6 CryptoAPI. The user can specify any symmetric cipher, a key and an iv generation mode and a new block device can be created in /dev [20] .

Ecryptfs is a enterprise cryptographic file system for Linux. It makes use of key per-file and user specified keys for fine-grain sharing. Cryptographic metadata of each file is stored by the file header format. The kernel implementation makes use of stackable vnode to plug an encryption layer which can fit into any underlying native file system. The metadata into file contents as header reduces transparency and require tools to manage file sharing.

Increasing theft of confidential or sensitive data of individual or organization needs an effective solution to deal with the data storage security issue. Most of the existing systems have been designed for its personal use. It is obvious that data has more importance than the storage device. Theft of USB drive or laptop increases the chances of identity theft, loss of financial plus personal data. Several incidents have drawn an attention to provide a solution to the problem that can protect data by using cryptographic method in the favor of personal or organizational scenario. Transcrypt is an encrypting file system for Linux to address the issues like data recovery, trust models, backup and key management. User-space processes and administrative account having super user privileges have been excluded from trust model by incorporating advance key management. Thus, making Transcrypt a trust model to make the system immune to different types of attacks thus attacks can be launched from inside of system. The basic features of transcrypt are privacy and file integrity and to share the file system objects between multi-users without sharing passphrase [21].

Root is excluded from trust model to implement Transcrypt function in the Linux kernel. For key management, user-space utilities are used. A random file Encryption Key (FEK) is produced when a file is created by user. File Encryption Key is blinded with File System Key (FSK) which is known to the kernel only and is a file system specific. The public key of user is enclosed in the form of X509 certificate. Token is created by encrypting File Encryption Key with user's public key and token is stored with file's metadata. When a file is opened to read or write in a file, then the token is taken and is sent to auth server (transcrypt-auth server) through trancryptd. Transcrypt-auth server acts as an interface for Private Key Store (PKS) of user. By making use of Private Key of user, token is decrypted at Private Key Store and is sent to the Kernel. Then the FEK is decrypted by kernel by using FSK. To read and write the data onto file FEK is used. The unauthorized user cannot decrypt token because user's private key is used to encrypt blinded FEK. File FEK can be obtained only by kernel. Thus, ensuring only authorized users have access to file [22].

2.1.3.3 Compressed based File System

Compressed file system stores data in a compressed form and retrieves data from storage media in decompressed form. Working with compressed file is always useful for saving disk space on system with less storage space like ram disk based systems.

JFFS2 is a journaling Flash File System used for flash memory devices. It is purely logged based and supports compression of metadata and data. It is not the first file system which uses compression; there are other file systems also that support compression such as disk based file system. JFFS2 two algorithms for compression were developed. It also uses Zlib compression library [23].

CramFS is a type of compressed file system that is developed by Linus Torvalds which is included in Linux Kernel. It is simple and it takes less space in disk and also small foot-print. In CramFS file system, each page of data-file is individually

compressed, allowing random page access. The maximum size of file is limited to 16MB and image size of file system is up to 256 MB. Entries like . .. are not allowed in the directories. Despite of all these restrictions, CramFS is most well-known and highly used read-only file system for embedded devices [24].

LeCramFS is a Less CrampFS file system which means less compressed file system. LeCramFS is a proficient and small file system intended for low-end portable devices and NAND flash memory that have limited amount of RAM. Indirection to advancement of file system performance it implements a series of well-organized mechanisms. This result in better file system performance than the other compressed file system as discussed by Seunghwan Hyun et. al [25].

2.1.3.4 Distributed File System

In Distributed file system, strong network-attached disk security mechanism was introduced to enhance the security. Traditionally, in Distributed File system data is stored on the centralized server which is managed by system administrator. System administrator has the privilege to access the whole file system, if anyone prove itself as a legitimate user then he will be the administrator which can make changes in the file system or can make vulnerable for the attack. So, to overcome this security problem centralized server is replaced by network-attached disk which makes the file system more vulnerable to attack as hard-disk attached with the network.

This has been overcome by developing a security mechanism which makes use of cryptography for Network-attached disk to protect data, that is, Secure Network-attached disk (SNAD). Strong cryptography is used by the system to conceal data from unwanted users which means that if any unauthorized person wants to gain access to hard-disk, the user cannot obtain data and it allows backup of unencrypted data without the intervention of super user access. This system is evolved by making use of raw-disk and it combined with native file system. The data which is stored is transferred only in encrypted form. Decryption of data is done at client side only. The

drivers don't have sufficient knowledge to decipher the data, so if data is stolen by attacker, the attacker cannot acquire access to data. Similarly, system administrator who also do backup of file system can access to enciphered copies of the data files only, that is, the user who has the authority or authentication of specific file can only access unencrypted contents of files [26] Client uses cryptographic algorithm at client side such as Blowfish or RC5 to encipher data which can ensure that data is readable only by that user who can decrypt it. To decrypt the file public-key cryptography is used. It also uses hashes such as SHA-1 and MD5 that can compute a number form data-block. I any changes will be done to input data, hash values will also get changed [26].

2.1.4 Tools and techniques to secure File system

There are different tools and techniques which act as safeguards for the files.

2.1.4.1 Ranish Partition Manager

It is a hard-disk partition, boot manager and hard-disk cloning tool. It can copy, create, clear, resize free space of partition and can format the primary and extended partitions. Logical volume can be changed to primary partition with the help of Ranish PM to make them bootable. The advantage of Ranish PM is that it can activate and hide the partitions on the fly. From Advanced Boot Manager you can select from which partition to boot the system, once changes are made in Master Boot Table, it hide rest of the disk space. Thus only the authorized user has the privilege to see the free disk space providing safeguard to files [27].

Partition table is situated in the first sector of the hard-disk that gives information to Operating system about partition location and size on the disk. Some computer systems have multiple Operating Systems and hard-disk is divided into different partitions. Partition table keeps indicating about active partition, that is, active partition contains information about from where the system will boot. When the computer

is turned ON BIOS gets loaded into memory. The first sector of hard disk where Partition table is present also contains Initial Program Loader (IPL) that searches for partition table to see Active partition and gets loaded in Master Boot Record.

Ranish Partition Manager (Ranish PM) can be booted from logical volume. It has two boot managers such as compact and advanced . Compact boot manager gets fitted into Master Boot Record (MBR). Compact boot manager have the option to check the boot viruses before Operating system gets loaded into memory because Some MBR viruses get loaded after the OS which cannot be detected by anti-virus. If IPL found any changes in the interrupt vector then it means the virus is residing in the MBR. Thus, in this case IPL will give warning to user either to stay with the virus or reboot the system [27].

2.1.4.2 TrueCrypt

TrueCrypt is a free utility that is used to encrypt data. On The Fly Encryption (OTFE) means that as you access your data, it automatically encrypts and decrypt data before the file is saved on storage media. Data cannot be read until a keyphrase or encryption key is provided by user to authenticate itself. TrueCrypt encrypts whole of the file system that includes metadata, folder name, file name, contents of file and free space. Computer file can be transcript from mounted TrueCrypt volume like normal disk performs copy operation by drag and drop operation.

When user reads a file from mounted TrueCrypt volume, data files are decrypted On the Fly in RAM or main memory and similarly they are encrypted onto the volume before the data is written onto disk. TrueCrypt enciphers USB, whole hard-disk partition, floppy. Physical partitions can also be encrypted which is called as TrueCrypt device-hosted volume. It supports hidden volumes and no TrueCrypt signatures, by which nobody can prove that a device, partition or a file is a TrueCrypt volume or is in encrypted form. TrueCrypt utility uses various encryption algorithm to form encrypted volume such as AES, TWO fish, Blowfish, serpent etc. System Encryption

offers highest level of security, as all the data files, applications and temporary files on the hard-disk partition are encrypted permanently.

Pre-boot authentication is also involved in the system encryption. It means if anyone wants to get access of file and to use encrypted system or to read and write file onto disk space, have to enter keyphrase. Thus it provides security not to the whole disk plus provides security to the system also from unauthorized access [29].

2.1.4.3 Disk Duplicator (dd):

dd is a Linux utility that is used by kernel to create boot images. Only super user has the privilege to execute dd command. dd is used to copy data from file to volume and from disk to file.To copy the disk, the destination of disk which is to be copied should be disk only not a volume or partition. Some disk duplicator copies hard-disk data sector by sector. Some disk duplicator have FAST COPY option which is used to find out where data is present on disk and it will duplicate only those sectors which have data. This results in fast duplication of data. For example, if you want backup of whole disk [30] . To do backup of whole copy of disk to another disk which is attached to same system dd command is used as: dd if =/dev/sda of =/dev/sdb

Here if presents input-file and of presents output-file. Source disk is /dev/sda and destination disk is /dev/sdb. Therefore /dev/sdb is the copy of /dev/sda . If any file, disk or partition becomes corrupted or gets deleted by mistake data can be recovered easily if dd command is used by admin [31].

2.1.4.4 Fdisk

Fdisk is a command-line utility that is used to create partition and to change the partition table on hard-disk. Hard-disks are split up into partitions and the information about this partition is depicted in the partition table. Firstly, partition table is made when a disk is partitioned. When partitioning is done each of the partition will have device name and device number. There can be four primary partitions in

any file system but minimum one partition is required for root file system. Fdisk also gives information about creation, deletion and changes in the disk partition. Fdisk can also be used to hide partitions which can protect your data from unauthorized or unwanted user. Thus it acts as safeguard to protect the files [30].

Chapter 3

Problem Statement and Objectives

3.1 Problem Statement

In today's era of technology, number of Internet user's are increasing with the growth of technology that has also increased the risks to secure the data. Everything needs to be secure whether it is a computer machine or any other machine. Attackers always tries to find useful information by gaining physical access or remote access to computer system as everything resides in system.

One problem formulated from previous gaps is that in case of theft of hardware-devices like hard-drives and USB drives, an attacker can physically access the system by mounting the hard-drives, USB drives on some other system that will help him to steal sensitive data, credentials from it.This problem has been overcome by encrypted file system as other file system will not be able to read the mount point of encrypted file system.

Another gap from research papers have been found that an attacker can also gain access to system by executing commands or malicious script on the shell, so to overcome this problem minimal shell or console access is given to the user which inhibits the unauthorized user from executing its own code. Attacker always tries to steal password to access any information of user's, as in traditional user authentication,

user proves its authentication to server by storing his password and username on server. Attacker can easily steal passwords and username from servers. To overcome this problem SSH key login pair has been formulated in which user has to provide only username to prove its authentication to server.

3.2 Objectives

The main objective of this framework is to enhance the security of file system which can restrict an unauthorized user to execute its own set of command or malicious script and prevent mounting of file system by any other user. The following are the prime objectives of this work:

- To setup various File system's (FS) on a Linux box and study their security features.
- To layout secure File System on Linux box.
- To implement secure and restrictive user-interaction layer.
- To verify and validate proposed layered mechanism.

Chapter 4

Proposed Framework

This chapter demonstrates design of proposed framework through Improved File system Security through Restrictive Access. It briefly explains how jail environment can be created by using shell script and through manual setup, encrypted file system and key login Pair.

4.1 Framework for Restrictive Shell access

The whole experiment is performed in virtual environment. In this Operating System has been installed on VMware workstation to study the different security features supported by different File systems. Different file system that are laid down is such as FAT, NTFS, ext3, Reiser FS .Every File system has its own features which depend upon its own logical and storage structure. The Jail environment setup comprises of installing Operating system lay down file system, creation of jail environment, copy binaries, copy library files and assigning Shell to user. The Figure 4.1 represents the block diagram of jail environment created to restrict system access.

4.1.1 Installing Operating System

Everything needs to be secure whether it is an Operating System or any other system. Operating System (OS) should protect itself from security breach, For example

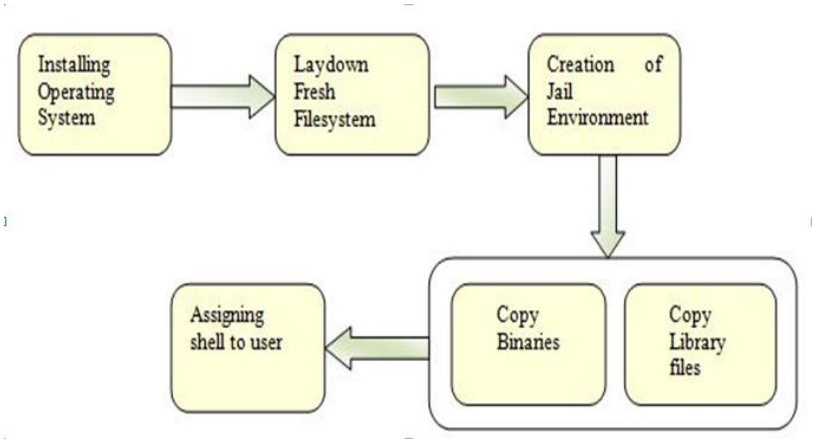


Figure 4.1: Framework for Restrictive Access

launching programs with excess privileges, access permissions, and stack overflows. Operating System (OS) is software that acts as an interface between end-user and hardware and helps in carrying out all the basic tasks. OS provides an environment in which user can execute application programs. OS is used to manage resources and provide services to programs. Every OS has its own interface which should be ease in use and should provide maximum CPU utilization. When the computer is switched on, firstly OS gets loaded into main memory. Once the Operating system is loaded, user can run application programs to accomplish its tasks. To create this setup, latest version of Operating system is installed in virtual environment that is 15.04 version of Ubuntu.

4.1.2 Fresh File system Laydown

In this section, different types of File system have been lay down such as FAT, NTFS, Ext2, ReiserFS. File system. Every computer file contains important data which is stored on volume or partition of hard-disk and file system plays an important role in managing these files. If file system doesn't exist, information would be stored on a large body where there is no manner to tell from where one piece of information is starting and the next will begin. A file system is the methods and data structures that an operating system uses to keep track of files on a disk or partition; that is, the way the files are organized on the disk. Hardening File system security can prevent

any type of attacks and it act as an interface between user and operating system.

4.1.3 Partitioning of hard-drive

In this Figure 4.2 fdisk utility is used that is creating new partition on the hard-drive. Fdisk is a command line utility that is used to make partitions and to modify the partition table on hard-disk. Fdisk can also be used to re-partition the hard-disk. Hard-drive can be divided into number of partition. There can be maximum four primary partitions in file system. This shows a new partition is created which is the

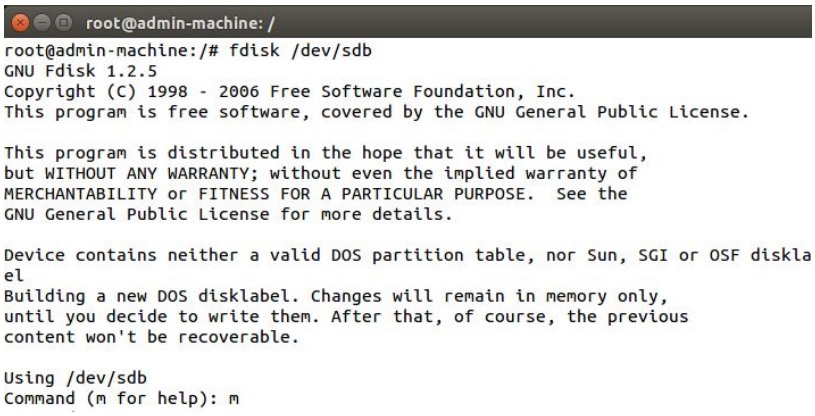


Figure 4.2: Partitioning of hard-drive

primary partition for the hard-disk. There are different options available in fdisk utility such as s option is used to create a new partition on the hard-disk. M is used to print all the available partitions on the hard-disk. A new partition can be added by using n-option. Once a new partition is made it will write on the hard-disk by using w option.

4.1.4 Formatting of new disk

Once a new partition is made, formatting of a new disk is done by using mkfs commands. It is used to lay down a fresh file system on a formatted storage space. Reiser file system is installed by formatting the disk space by using mkfs command. It is a Journaling file system that maintains a file called as journal. Any change in the data is first marked onto journal and then changes will be written onto disk. It is used to

avoid inconsistency of data in case of system failure or improper shutdown of system. This figure shows 4.3 that reiserfs is lay down successfully.

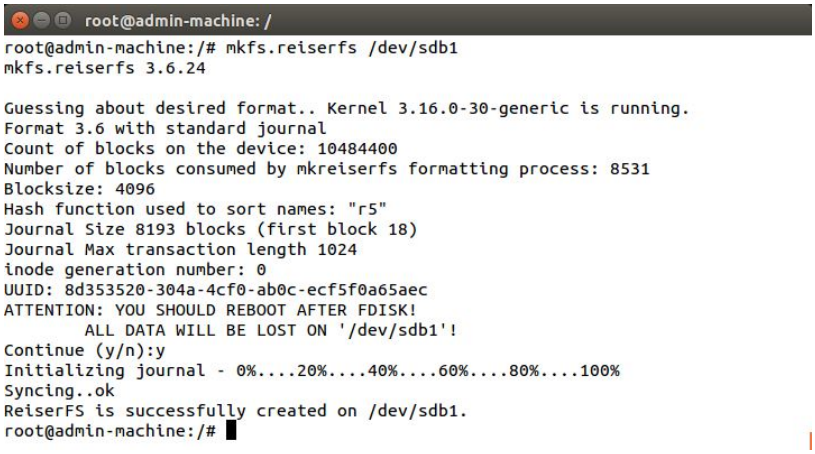


Figure 4.3: Laydown of File system

4.1.5 Mounting of file system

To access a file on Linux Operating system, file system should be mounted by using mount command. All the files, directories, special files and file system are available to user only if it mounted on disk-space. Mounting is the process where a raw (physical) partition is prepared for access and assigned a location on mount point. Mounting involves attaching the disk space with some directory to access the data stored in it. Mount command always need admin or root privileges. File system which mounted can be seen in /etc/fstab.

In this figure 4.4a new directory is made which is named as dell. Here the /dev/sdb1 partition is attached with the dell directory to access disk space. Once the command is executed, we can see the mounted partition by using df f command. Total size of mounted partition /dev/sdb1 is 40GB and is mounted on directory dell.

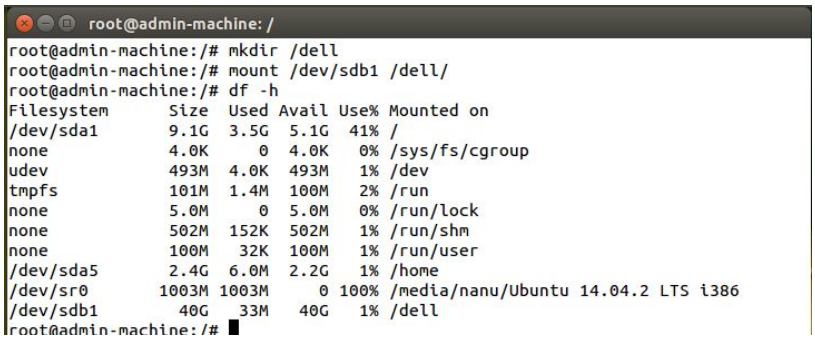


Figure 4.4: Mounting File system

4.1.6 Updating fstab.

It is a configuration file which contains information about where the partitions are mounted.Figure 4.5 shows updation of fstab.

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda1 during installation
UUID=3f8ef1df-7124-4bc6-a36d-0366ecec49d6 /               ext4    errors=remount$
# /boot was on /dev/sda5 during installation
UUID=1d67a91d-4711-495d-aa8f-ecb06c1e5a5d /boot           ext4    defaults      $
# /home was on /dev/sda6 during installation
UUID=9b95db8d-9007-478b-ac41-45bf1c38afd5 /home           ext4    defaults      $
/dev/fd0        /media/floppy0  auto    rw,user,noauto,exec,utf8 0       0
/dev/sdb1       /disk/          reiserfs        default 1,2
```

Figure 4.5: Update fstab

4.2 Jail Environment Creation

Chroot change the root directory of the current running process. Chroot jail is used to create an environment, where a user process is separated from the whole system. In this, users request is limited to this environment only and can execute limited set of commands. The shell user has hardly any permission in the jail environment.

4.2.1 Jail creation through manual setup

Chroot Jail Directory

Figure 4.7 shows while creating jail, we required to create jail directory. In figure

4.6 we have created multiple directory using p option of mkdir command. This will become the root directory for chroot jail directory.

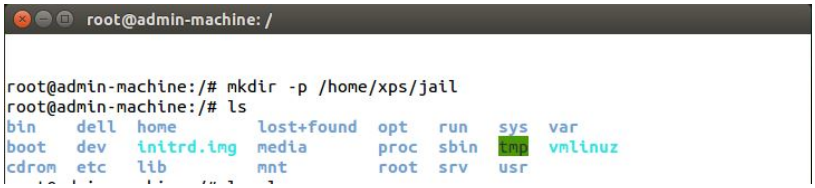


Figure 4.6: Creation of Chroot jail directory

Listing required for Library Files

In jail environment, we want to create a restrictive shell. Ldd is a utility which tell which libraries are needed by the program on the command line. Figure 4.7 depicts which libraries are needed to execute commands in chroot jail.

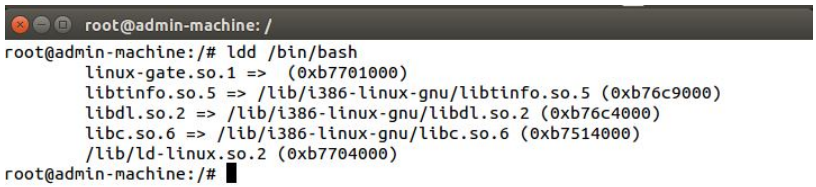


Figure 4.7: Listing required Library files

Subdirectories Creation

Figure 4.8 shows new subdirectories are made under the chroot jail directory where library and binary files will be copied. This Figure depicts new subdirectories are made, that is, bin and lib. In bin directory we will copy binary files needed for the

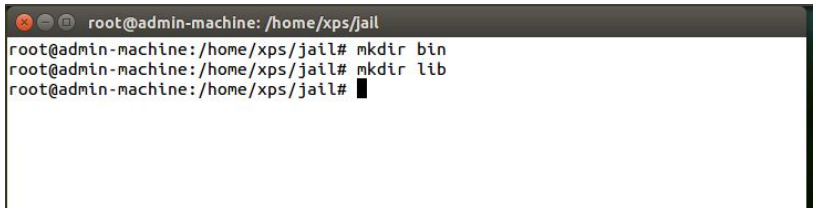


Figure 4.8: Subdirectories for Chroot jail

commands and in lib directory we will copy libraries needed to execute the commands.

Copying shared and binary files

The Figure 4.9 depicts copying of shared and binary libraries under chroot jail directory.This Figure depicts that the shared libraries of commands such as ping, netstat, ifconfig are copied under lib subdirectory of chroot jail directory and binary libraries required for the particular set of commands are also copied under bin directory of chroot jail directory. Both of shared and binary libraries are essential to execute the commands

Figure 4.9: Copying shared and binary files

New Restrictive Shell

The Figure 4.10 shows execution of particular set of commands in chroot jail. The command chroot, with chroot jail directory is executed which will offer a new shell, that is bash shell where user can execute only small set of commands which are defined by an administrator. This Figure shows that following commands such as ls, pwd, ping, iproute, netstat can be executed under chroot jail as the required shared and binary libraries have been copied under this directory. Here the present working directory is root directory.

4.2.2 Jail Creation through Shell Script

Another method to restrict the user from using all the commands is by using Shell script. Shell script is a series of commands formulated in a text file and is stored in a file. Shell takes input as a file and executes all the commands defined in the shell script. Fig 4.11shows the shell script comprises of limited set of commands defined by the admin so that user can execute limited series of commands only. The commands which are defined by the admin are arp, arping, clear cpuinfo, diskuse, exit, ifconfig,

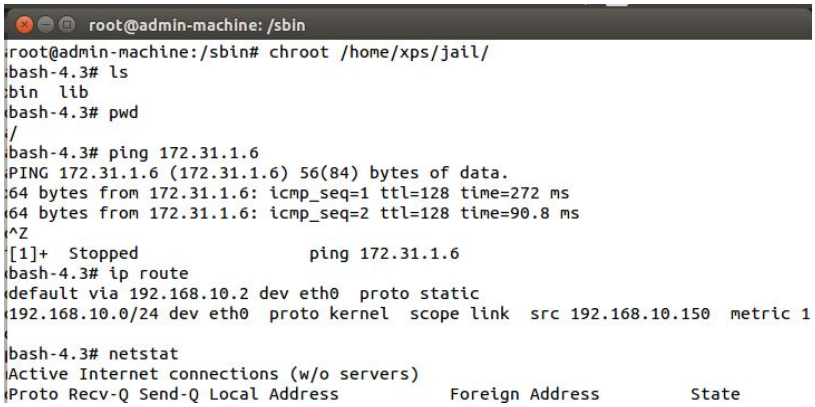


Figure 4.10: New Restrictive Shell

netstat, ping, shutdown and telnet.

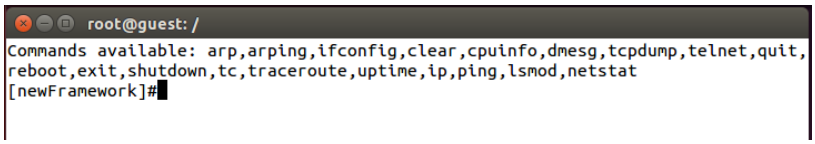


Figure 4.11: Available Commands in Restrictive Shell

Figure 4.12 shows snippet code for shell script which results in creation of restrictive shell.

ARP

ARP stands for Address Resolution Protocol cache that comprises of table in which IP address of the network neighbor is stored. Arp can show entries made in the ARP table. It need root permission to delete the entries from the table.

Arping

It is a software tool for probing and finding hosts on a computer network. Arping inquires hosts on the attached network link by sending frames of Link Layer with the help of Address Resolution Protocol (ARP) request placed to a host which is identified by the network interface MAC address. To resolve an IP address gave by the user, the utility program an ARP.

```
#!/bin/sh
commands()
{
    Echo "Commands available are: arp, arping, ifconfig, clear,
    cpuinfo, tcpdump, telnet, quit, reboot, exit, shutdown, tc,
    traceroute, uptime, ip, ping, lsmod, netstat"
}
shell()
{
    clear
    while :
    do
    echo -n "[New Framework]#"
    read ans
    clear
    #val=`echo $ans | cut -f1 -d " "`
    val=$ans
    if [ -z $val ]
    then
    return
    fi
    echo $ans |grep "/" 2>&1 > /dev/null
    if [ $? -ne 0 ]
    then
    case $val in
    ping)
    $ans -n
    ;;
    exit)
    break
    ;;
    netstat)
    $ans
    ;;
    uptime)
    $ans
    ;;
    cpuinfo)
    cat /proc/cpuinfo
```

Figure 4.12: Snippet Code For Shell Script

Ifconfig

In Unix-like operating system, ifconfig is a system administration utility for network interface configuration. The utility is a tool of command line interface and is also used in the system startup scripts for many operating systems.

Clear Clear command is used to clear the content from the screen.

CpuInfo

Cpuinfo displays the information of CPU like, type of processor, model and vendor of the processor, model name, the speed of processor, number of cores and so on.

Tcpdump Tcpdump is a packet analyzer or command line packet sniffer that captures TCP/IP packets that has been transferred or received from the network. It also provides an option which keeps the captured packets in a file for future use and file is saved in pcap format. It is commonly available in most of the Linux Operating system. The contents of saved file can be seen by using tcpdump command or by using wire-shark that is open source GUI tool. Wireshark understand tcpdump pcap format files.

Telnet Telnet operates on application layer and uses TCP/IP protocol to access the computers remotely. Through telnet an administrator or other user can access some other person's computer. With the help of Telnet client connection is made to TCP port and communication is done using telnet port.

Reboot Reboot can be used to restart or reboot the system. Sometimes rebooting is required to re-initialize the hardware devices or to recover the system from an error.

Exit It is used to exit from the shell and the user will come in normal shell.

Shutdown
it is used to shut down the system from a terminal session.". It might take some moments to terminate all the processes. The computer will reboot itself.
TC Tc is a traffic signal command used to configure the Linux kernel packet scheduler. It is commonly packaged as part of the iproute2 package.

Traceroute The Internet is a huge and complex collection of network hardware, connected by gateways. To track the route of packets follow (or searching the gateway that's removing your packets) can be difficult. Traceroute uses the "time to live" field of IP protocol and tries to elicit an ICMP TIME_EXCEEDED reply with the path to some host from each gateway.

Uptime It gives a single-line display of the following information: - the current time - duration of system running - number of users currently logged on.

Ifconfig Ifconfig stands for Internet Configuration. It displays the information like, ip address of network interface broadcast address and hardware address.

IP For assigning an address to a network interface and to configure the parameters of network interface on Linux OS. This command helps to replace old good and disapprove ifconfig command on latest Linux distributions.

lsmod lsmod displays or "lists" the status of modules in the Linux kernel. It is a simple program that formats the file/proc/modules contents, that consist information of all currently-loaded LKM's status.

Netstat

Netstat stands for network statics that shows information like, routing tables, network connections for Transmission Control protocol (TCP), and gateway.

4.3 Encrypted File System

Encrypted file systems can be used to protect sensitive documents, financial documents, Keys and so on. Loop back device is a virtual device that can be used as a media device. Media devices are partitions of hard drive such as /dev/hda1, /dev/sda1, and /dev/sdb1. These devices are used to keep files and directory structures, and can be formatted with any required file system and are then mounted. The loopback file system creates a file system within file system. A file can be associated on another file system as a device with the help of loopback file system and it can be formatted and mounted like normal devices as listed above. All this can be done with the help of devices such as /dev/loop0 etc which is linked with the file and it is mounted as a new virtual device. This provides extra layer of security to the file

system.

To encrypt the file system different modules can be used such as loop, cryptoloop and aes module. Cryptoloop is used to encrypt the module for Linux. Cryptoloop has an advantage that an encrypted file system can be created on a single partition. Cryptoloop make use of loopback device that can be called with file system request. Cryptoloop module is to encrypt partition and can be mounted within Linux OS. Loop device is a pseudo device that enables regular file to access as block device. The advantage of loop device is, a file system can be created with a regular file if no partition is currently available.

Procedure to Create Encrypted File System

Loading of Modules

The Figure 4.13 shows that three modules are used to encrypt the file system that is, loop, cryptoloop, aes. These modules can be loaded to kernel by using modprobe. Modprobe is a program that is used to append a loadable Kernel module (LKM) into Linux kernel.

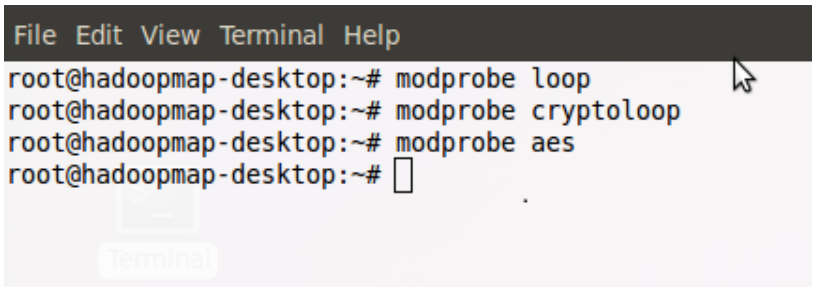


Figure 4.13: Loading Modules

Creation of Loopback Device

The Figure 4.14 shows that an empty image file is created called as newfs image where the input file is any random file and output file is new_fs having size of 20 MB . dd if=/dev/urandom is used to create a device file with random bits, “of ”parameter defines output file with maximum size of (20*1024*1024=20971520/ 512 =40960) 20

MB.

Figure 4.14: Creation of Loopback Device

Binding Image To Loopback Device

The Figure 4.15 shows that the device file can be attached to loopback device. Losetup is used to connect loopback device with regular file. It will attach the newfs_image to /dev/loop0 device and uses aes encryption to encrypt /dev/loop0 device and will ask for the password that can be used in future to mount and unmount the partition.

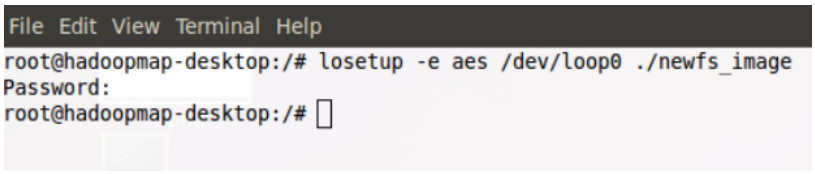


Figure 4.15: Binding Device file to Loopback Device

File System Creation

The Figure 4.16 shows that file system can be created by making use of /dev/loop0 device. Reiserfs file system is used to format file system.

Mounting Loopback device

The Figure 4.17 shows that the file system is mounted permanently to /file system with encryption and acl features. It will ask for the password to encrypt the image.

Encrypted File System

The Figure 4.18 shows once encrypted file system is mounted permanently on a partition, user can switch to encrypted file system which will shows lost+found directory. It is helpful in recovering the files from system due to improper shutdown of system or power failure. This file is created for each partition by the Operating System dur-

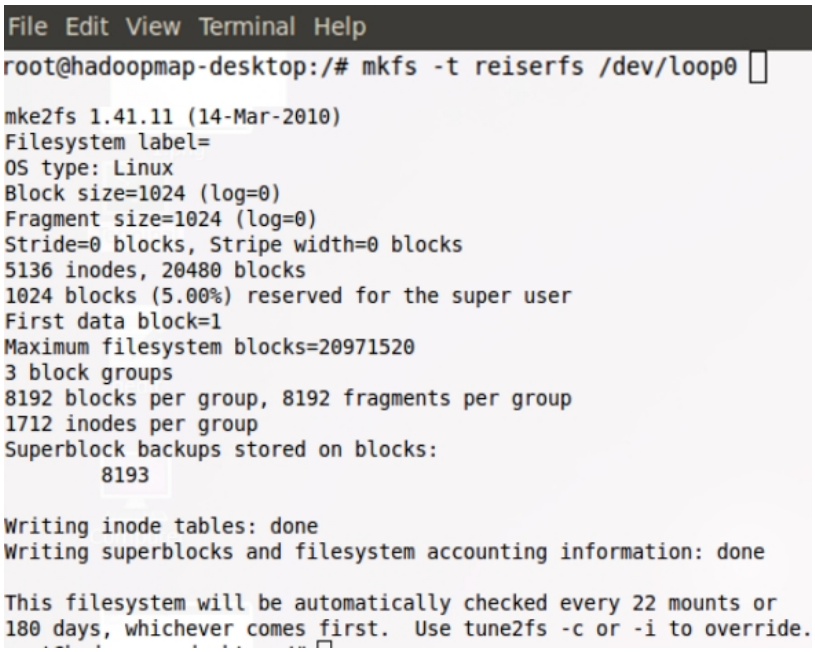


Figure 4.16: File System Creation

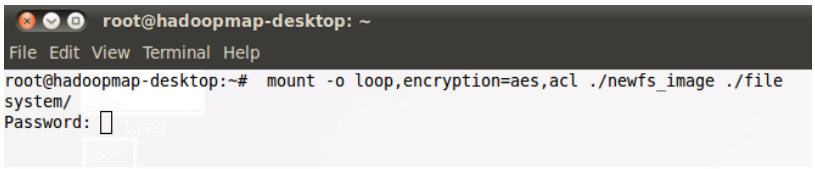


Figure 4.17: Mounting Loopback Device Permanently

ing its installation. Mounted Partition consists of lost+found folder. Fsck keeps this folder to recover the file.

4.4 SSH Key Login Pair

In traditional user authentication system, user proves its authentication by providing username and password to a server. Then server will validate user-authentication by matching its credentials. If server gets hacked then all the credentials will be stolen by hacker that can be used by him to gain access of a system. So, this problem can be overcome by using ssh login key pair.

In this technique we have used SSH network protocol. Secure Socket Shell is a

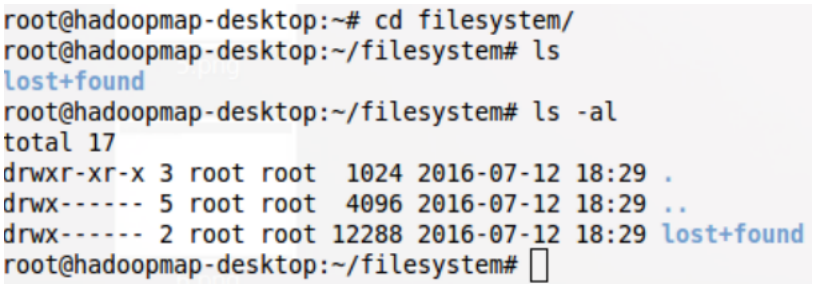


Figure 4.18: Switching to Encrypted File System

protocol that provides security services to insecure network and secure remote login. When a user sends data over network data will be encrypted automatically by SSH and when the data will reach to destination data will be decrypted automatically by SSH. SSH provides transparent encryption as user is unaware of how data is encrypted and communicated in a secure way. SSH comprises of three components:

- **Transport layer protocol**

  This protocol provides authentication, integrity and confidentiality of server. Compression may also be provided by SSH. Perhaps it also provides compression optionally. The transport layer runs over a TCP/IP connection distinctively

- **User Authentication Protocol**

  The User Authentication Protocol [SSH-USERAUTH] authenticates the user of client-side to the server. It runs over the transport layer protocol.

- **Connection Protocol**

  The Connection Protocol [SSH-CONNECT] complexes the encrypted tunnel into few logical channels. It runs over the user authentication protocol.

A service request sends by the client once a secure connection has been established by the transport layer. A second service request is sent after the completion of user authentication. This gives an authorization to the new protocols to be defined and coexist with the above listed protocols.

On the proposed framework only SSH based password-free login procedure is established. In public key authentication, a key pair is generated which consists of

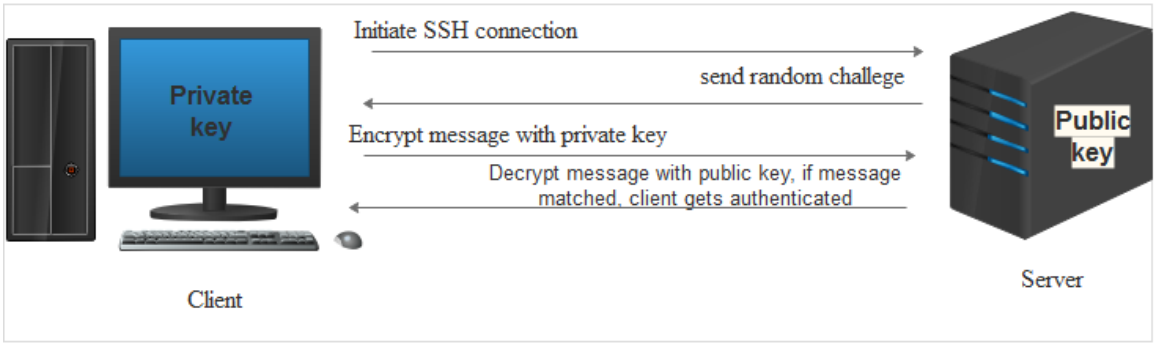


Figure 4.19: SSH Authentication

public key and a private key. The signature can be created by using private key which cannot be forged by anyone. The signature can be verified by only those who have public key. Therefore, key pair is generated on the local host and the public key can be copied onto server. Any compromise to the private key would permit an attacker to gain access to server because server has been configured by using associated public key. By using server's public key, message can be encrypted and it can be decrypted only by using client's private key. This explains the authentication procedure by using key pair.

Procedure to Generate SSH key pair

- **SSH Key Pair Generation**

  Figure 4.20 shows SSH key pair is generated on client machine by using ssh-gen utility which is present in OpenSSH suite of tools. This Figure depicts that first login is made into client machine 192.168.10.156 which will generate key pair. SSH-gen utility will ask for the location to save the keys and keys will get stored in /.ssh directory that is inside user's home directory. The public key is id_dsa.pub and the private key is id_dsa. After saving the file, prompt will ask to enter the pass-phrase for the key. To encipher the private key file on the disk, an optional passphrase can be used.

- **Public key contents**

  The Figure 4.21 show contents of the public key is obtained which will be copied onto server. Thus allowing user to log into the server from client machine using

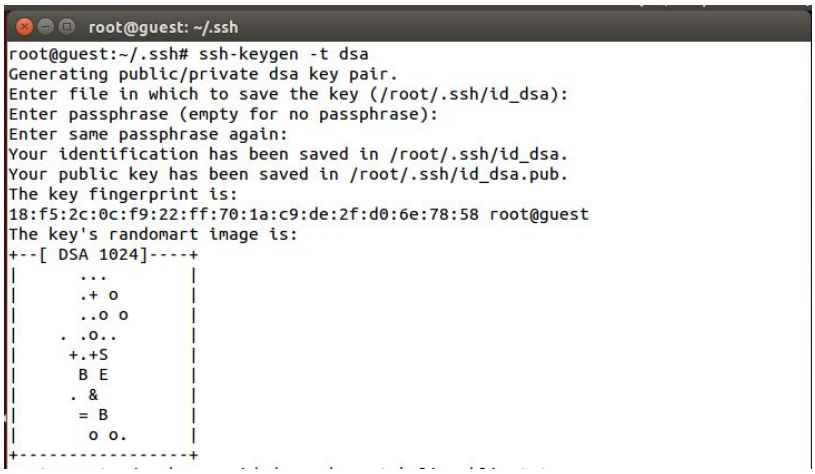


Figure 4.20: SSH Key Pair Generation

private key.

Figure 4.21: Public key Contents

- **Copying Public key to SSH Server**

  The Figure 4.22 Once the public key has been generated it can be copied onto server by using secure copy (scp) command. By using scp command files can be copied to or from different hosts. It use SSH to transfer data securely and needs password for authentication.

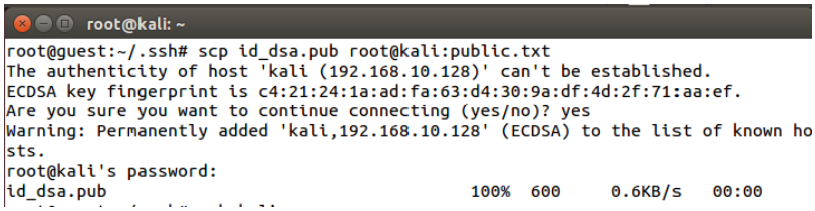


Figure 4.22: Copying Public Key to Server

- **Authorized key File appended**

  This figure 4.23 shows that the authorized key file appended with public key of hosts.

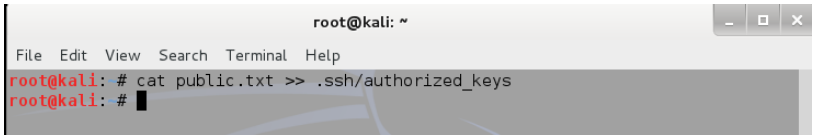


Figure 4.23: Authrozied keys Appended

- **Successful Login to machine**

  This Figure 4.24 depicts successful login of the machine without password/ pass-phrase from root@guest machine into root@kali machine.

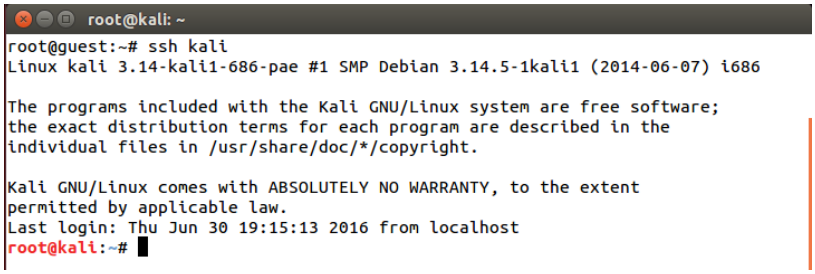


Figure 4.24: Successful Login Without Password

Chapter 5

Results and Discussion

System's Security is one of the major concerns to protect the data from intruder. The flaws in system's security invite cyber-criminals to breach the security of a system. Compromised security can lead to system's degradation. File system is also one of the targets for an attacker to compromise the security. There can be different reasons to attack on a file system like attacker may want to gain access of system to execute its own code, to steal the credentials or important information of user or to defame the company.

Therefore, the security of file system has been enhanced by providing security at core level by restricting physical access, which involves creation of restrictive shell, authentication of user through SSH Key login pair and encrypting file system. Shell has been restricted by using two methods, by manual setup and by shell script. This results in offering restrictive shell to user in which only particular specified commands can be executed. Thus, providing security to shell so that an unwanted user cannot execute malicious code to breach the security of system.

The Figure 5.1 represents manual setup to create restrictive shell. User is given a jailed environment , where only the specified commands can be executed. In this we have executed chroot command with chroot jail director which results into creation of new restrictive shell that can be assigned to a user account to have limited access

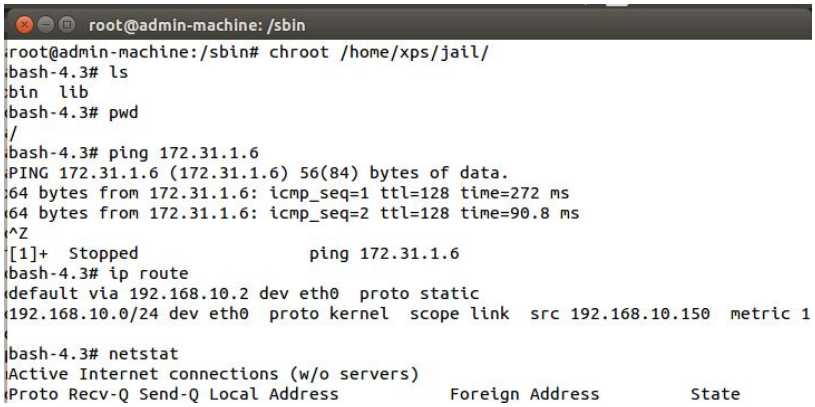


Figure 5.1: Restrictive Shell Through Manual setup

of shell. This inhibits an unauthorized user's from executing malicious code or its own set of commands thus, enhancing the security of system.

The Figure 5.2 represents creation of restrictive shell through shell script. This shell is being offered automatically, when user will login into account and can execute only small set of commands defined by admin. The admin console screen is shown as:

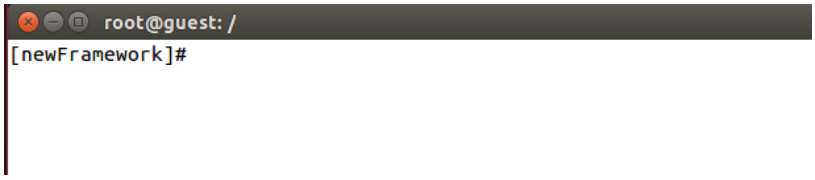


Figure 5.2: New Restrictive Shell Through Shell Script

The Figure 5.3 represents when a user want to execute a command which is not defined by admin in shell script then this message appears, which shows that only these commands are available for execution.

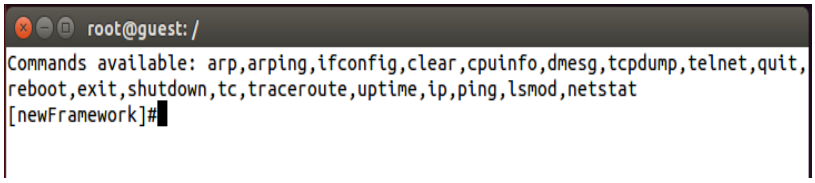


Figure 5.3: Admin Console Screen

The Figure 5.4 shows successful login of root@guest into root@kali machine successfully without entering password. This prevents the attacker from stealing the keys

from server as the private keys are not stored on server and are kept by client only.

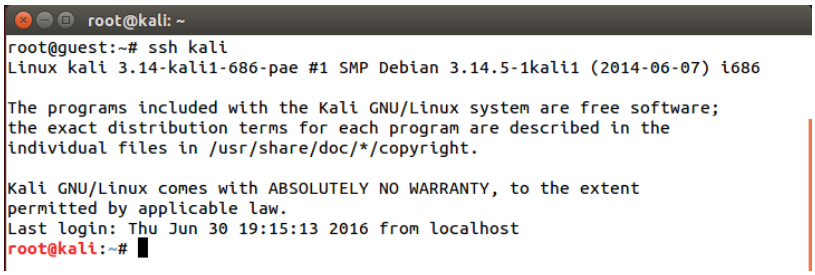


Figure 5.4: Login to machine without password

This Figure 5.5 shows that file system is encrypted with acl features to provide extra layer of security to file system that will inhibits the unauthorized from reading the private information or important data in case of data theft.

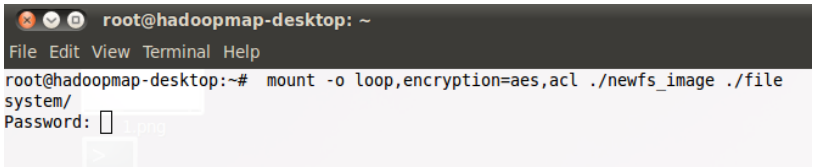


Figure 5.5: Encrypted File System

.

Chapter 6

Conclusions and Future Work

6.1 Conclusion

In this work, a modular framework has been proposed that aims at enhancing security of a file system. For the Framework, console minimal access is provided to Operating System environment by creating jail environment through manual setup and by establishing a startup script for the user. This allows only certain command privilege avoiding direct interaction with other binary programs on the framework.This inhibits the intruder from executing malicious shell script or code thus enhancing security of system.

Other module of framework involves successful authentication of user by login into server without entering password . For successful authentication of user, only user-name is provided, as in traditional system user proves its authentication by validating its credentials stored on the server(both user-name and password). Another module results in encrypted file system thus avoiding the unauthorized user from accessing data in several cases like , if laptop or hard-drive stolen by unwanted user, he may try to read the confidential data by mounting the hard-drive on some other system. But unauthorized user may not be able to read the data as the encrypted file system has different mount point thus this inhibits mounting of file system on another system.This results in enhancing security of File system from the attacker.

6.2 Future Work

In cloud computing, as data remain on open cloud anyone can access data without any restriction due to weak security policies. In future, we can plan to apply this framework in the area of cloud computing in which each user will have secure and restrictive shell to execute its set of commands, restrictive file system access and user authentication without entering password.

References

[1] Jacobs, Stuart. Engineering information security: the application of systems engineering concepts to achieve information assurance. Vol. 14. John Wiley & Sons, 2011

[2] "Operating System Security", tutorial point, [Online]. Available: http://www.tutorialspoint.com/operating_system/os_security.html. Accessed on [5 02 2016]

[3] File Allocation Table, Available at https://technet.microsoft.com/en-us/library/cc938438.aspx., Accessed on[10 2 2016]

[4] File Allocation Table, Available at https://en.wikipedia.org/wiki/File_Allocation-_Table, Accessed on [10 2 2016]

[5] Encrypting File system, Available at https://en.wikipedia.org/wiki/Encrypting_File-_System, Accessed on [12 02 2016]

[6] How permission works, NTFS permissions Available at, https://technet.microsoft.com/en-us/library/cc783530 (v=ws.10).aspx, Accessed on [15 02 2016]

[7] Access token, Technet Available at https://technet.microsoft.com/en-us/magazine/2005.11.howitworksntfs.aspx, Accessed on 18 6 2016, Accessed on [16 02 2016]

[8] Windows access tokens, bsodtutorial, [Online] Available at https://bsodtutorials.wordpress.com/2014/08/09/windows-access-tokens-token-and-_token/, [Accessed on 16 02 2016]

[9] "Primary Access token", Available at https://msdn.microsoft.com/en-us/library/windows/desktop/aa374909(v=vs.85).aspx, [Accessed on 16 02 2016]

[10] "Access control entries", Available at https://msdn.microsoft.com/enus/library/--windows/desktop/aa374868 (v=vs.85).aspx. , [Accessed on 16 2016] .

[11] "EXT2 file system", Availabe at, https://en.wikipedia.org/wiki/Ext2, Accessed on [19 02 2016]

[12] Ext4 disk layout,Available at, https://ext4.wiki.kernel.org/index.php/Ext4_Disk_Layout, Accessed on [20 02 2016]

[13] Anatomy of ext4, Availabe at ., http://www.ibm.com/developerworks/library/l-anatomy-ext4/

[14] Roberts, Thomas J. "Efficient method for encrypting superblocks of data." U.S. Patent No. 5,008,935. 16 Apr. 1991.

[15] Anderson, Ross, Roger Needham, and Adi Shamir. "The steganographic file system." International Workshop on Information Hiding. Springer Berlin Heidelberg, 1998.

[16] McDonald, Andrew D., and Markus G. Kuhn. "StegFS: A steganographic file system for Linux." Information Hiding. Springer Berlin Heidelberg, 1999.

[17] Self-certifyingFilesystem, https://www.usenix.org/legacy/event/usenix01/-freenix01/full_papers/cattaneo/cattaneo_html/node6.html

[18] D. Mazires, M. Kaminsky, M. F. Kaashoek, E. Witchel "Separating key management from file system security", Proceedings of 17th ACM Symposium on Operating System Principles (SOSP '99), Kiawah Island, South Carolina, December 1999.

[19] Blaze, Matt. "A cryptographic file system for UNIX." Proceedings of the 1st ACM conference on Computer and communications security. ACM, 1993

[20] "dm-crypt": a device-mapper crypto target for Linux. [Online]. Available at http://www.saout.de/misc/dm-crypt/.

[21] Sharma, Satyam, Rajat Moona, and Dheeraj Sanghi. "Transcrypt: A secure and transparent encrypting file system for enterprises." 8th International Symposium on System and Information Security. 2006

[22] Raghavan, Arun. "File System Independent Metadata Organization for TransCrypt." Diss. Master's thesis, Indian Institute of Technology Kanpur, India (2008).

[23] Woodhouse, David. "JFFS: The journaling flash file system." Ottawa linux symposium. Vol. 2001. 2001

[24] CramFS document. [Online]. Available: http://lxr.linux.no/source/fs/cramfs- -/README .

[25] 24. Hyun, Seunghwan, Hyokyung Bahn, and Kern Koh. "LeCramFS: an efficient compressed file system for flash-based portable consumer devices."Consumer Electronics, IEEE Transactions on 53.2 (2007): 481-488.

[26] Miller, Ethan, et al. "Strong security for distributed file systems."Perforance, Computing, and Communications, 2001. IEEE International Conference on.. IEEE, 2001.

[27] Ranish Partition Manager, http://web.mit.edu/netbsd/i386/utils/partition- -_manager/PART.HTM

[28] "TrueCrypt", [Online] Available at ,https://en.wikipedia.org/wiki/TrueCrypt. Accessed on [28 03 2016]

[29] Zhou, Li. Information Diffusion on Twitter. Diss. University of Dayton, 2015.

[30] Disk duplicator, Example to backup data using Linux commands.

http://www.thegeekstuff.com/2010/10/dd-command-examples/?utm_source=feedburner&utm_medium=email&utm_campaign=Feed.

[31] Hard disk duplicator,[Online], Avaibale at Guide, http://www.aleratec.com /hard-disk-duplicator-buyers-guide.html, Accessed on [20 03 2016]

[32] "fdisk", [Online,] formatting of hard-disk, Available at, https://wiki–archlinux.org/index.php/Fdisk.

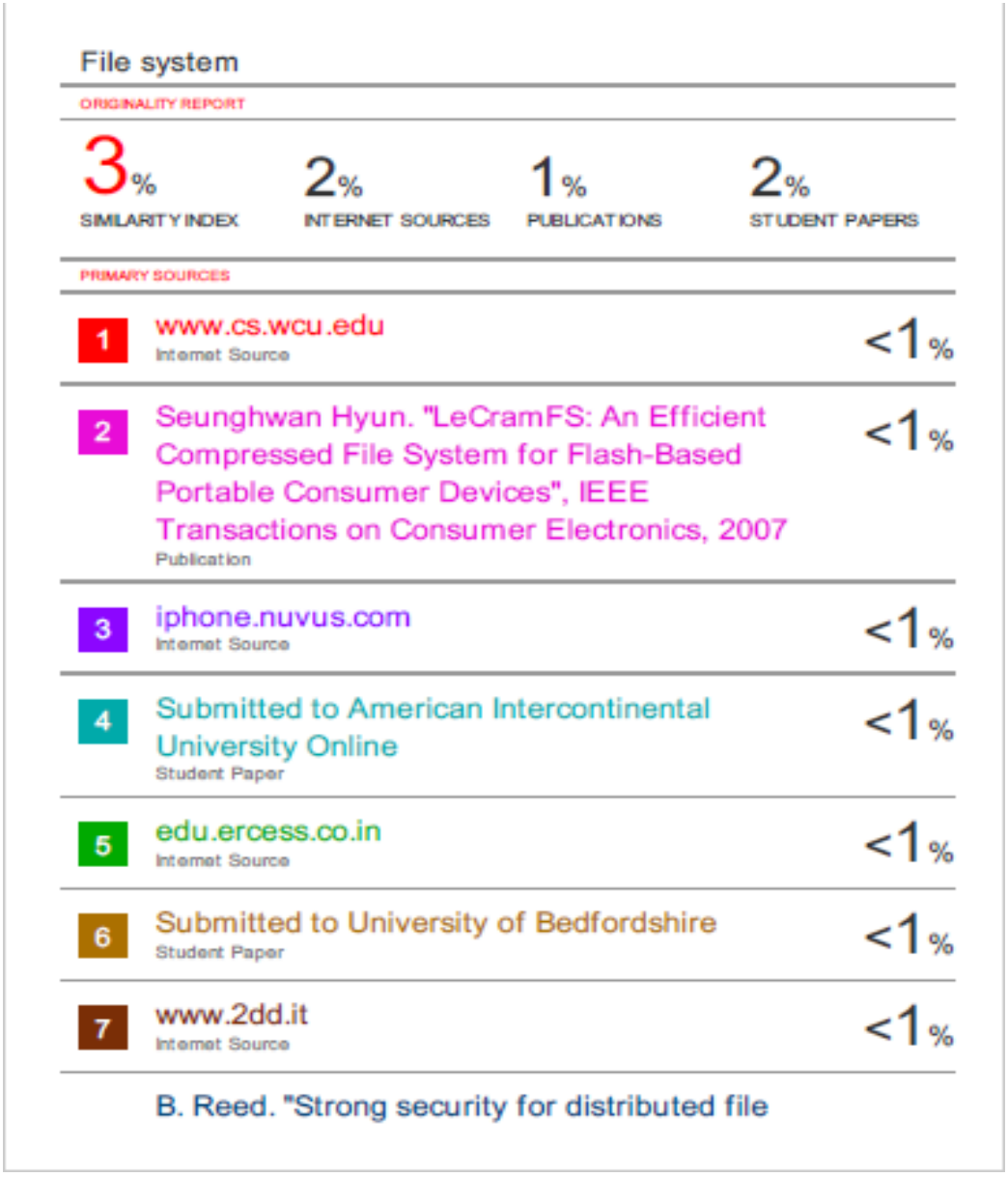
Publication

Navneet Kaur, Maninder Singh, "Improved File System Security Through restrictive Access", *International Conference on Inventive Computation Technologies (ICICT 2016) to be held from August 26-27, 2016 Tamil Nadu, India*[Accepted],ICICT-PaperID: E-321.

Video Link

url: “https://www.youtube.com/channel/UC49ChGux9u4d8NXNaOU_ZZA”

Plagiarism Report



File system

ORIGINALITY REPORT

3%	2%	1%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.cs.wcu.edu Internet Source	<1%
2	Seunghwan Hyun. "LeCramFS: An Efficient Compressed File System for Flash-Based Portable Consumer Devices", IEEE Transactions on Consumer Electronics, 2007 Publication	<1%
3	iphone.nuvus.com Internet Source	<1%
4	Submitted to American Intercontinental University Online Student Paper	<1%
5	edu.ercess.co.in Internet Source	<1%
6	Submitted to University of Bedfordshire Student Paper	<1%
7	www.2dd.it Internet Source	<1%

B. Reed. "Strong security for distributed file