

# User Control on Diskless Client Server Platform

B. S. Sonawane<sup>1</sup>, R. R. Deshmukh<sup>2</sup>, S. D. Waghmare<sup>3</sup>, Pushpendra Chavan<sup>4</sup>

<sup>1</sup>Research Fellow, Dept. of CSIT, Dr. B. A. M. University, Aurangabad

<sup>2</sup>Professor, Dept. of CSIT, Dr. B. A. M. University, Aurangabad

<sup>3</sup>Research Fellow, Dept. of CSIT, Dr. B. A. M. University, Aurangabad

<sup>4</sup>Principal, Tech Support Engineer, Red Hat India Pvt. Ltd, Pune,

**Abstract-** This paper addresses the approach of controlling user access on the Linux systems used in Disk Less Client server for various purposes. User control is very much necessary in today's world as one shouldn't be given full access to underlying system in case its important to protect it from malicious activities and threats. Controlling user what to execute and what not to execute is called as user control. There are various methods to achieve it, one of them is very popular i.e. sudo access which is used to allow user run the superuser binaries which is completely opposite to the motive of this paper, instead of allowing user to execute superuser binaries, this paper addresses controlling users to execute the normal user binaries themselves and allowing access to only certain binaries.

The purpose here is to protect the underlying system and reduce the monitoring efforts on the system which only allows the certain tasks per user on a disk less client server which are booted of the network on disk less systems into Memory.

**General Terms-**The Disk Less Client is a technology used for ease of administration. The scope of this paper will be only for Software oriented Virtual Disk Less clients.

**Keyword-**Disk less client server, minimal server, automation, user control, chmod, linux administration, DHCP, PXE, ftpboot , users and files attributes, jailed users.

## I. INTRODUCTION

Diskless Client Server Technology is used to reduce the cost of maintenance of the servers and centralized resource management where all users will be local and centralized. This allows users to login from network by loading the operating system into memory of the hardware systems in network without any underlying disks into picture. These loaded Operating system instances into memory will allow users to login on their console and after landing up into their home directories, users are allowed to execute certain binaries, say in 100s in number. Controlling these binary accesses is much necessary because an user for task A isn't supposed to run/execute/participate in task B and if no access control available, the user will perform the supposed tasks he/she has been asked to perform. This is called as User Control on Diskless Client. This paper explain the control management of user which is in the network of computer systems. The control can be provided through remote places from the server side scripting. The linux platform is the simplified way to control the user through client server architecture.

### 1.1 Terminologies:

User Control Mechanism :

In Linux , there are many ways users can be controlled to access the system, by default a normal system user can't access the superuser binaries, and other user's home directories. All rest normal binaries present in /bin and /usr/bin are accessible by the user. The binary called as /usr/bin/sudo is used to get the superuser privileges to execute superuser binaries. The scope of this paper isn't for sudo but for exactly opposite, i.e. controlling users even to execute certain binaries from /bin and /usr/bin directories as well as not be able to access any other given location in the system except for the home directories assigned to them.

Shell : Shell is an integral part of a Linux System which allows user to execute the commands and then passes the signals to kernel in order to create system calls and then accepts output from kernel and pass it on to the user who executed the command. The shells are of multiple types and can then be assigned to users by default by setting as a default shell of login in the system in /etc/passwd file which is a local user database of a Linux system. Mostly used shells are /bin/bash , /bin/sh, /bin/ksh and /bin/tcsh shells. There are many more, but /bin/bash is the one mostly gets used.

Restricted Shell : The restricted shell is a shell which doesn't allow the change directory operation for the user it's assigned too at all. This is exactly same shell like bash but doesn't actually allow user to execute cd command.

Environment : Environment in Linux system is the set of variables and there values. This environment can be fetched by executing the env command and can shoot out all the details about the most important variable values like PATH, LD\_LIBRARY\_PATH, LANG, MAIL, USER and SHELL etc.

Links : Links are simply the shortcuts, and can be of two types, soft link and hard link. The Soft links can exist between the two filesystems whereas hard links cannot.

File Attributes and File permissions : File attributes are special attributes like immutable, append only, user ownership. These are special permissions assigned to the files/directories which control actions on top of them. For example, the immutable attribute won't let user/root to modify/delete/move the file/directory having immutable attribute. File permissions are mainly read, write and execute. The permissions can be set to file/directory owner, the group and to others covering everybody else except the assigned owner and assigned group

## II. LITERATURE REVIEW

The hypervisor is a program of virtualization which handles the protection among virtual machines; hence applications can be easily migrated on different virtual machines without isolation of host integrity. Virtual machine architecture having six major types i.e. Full Virtualization, Hardware assisted virtualization, Para Virtualization, Operating System Level Virtualization, Application Level Virtualization and Network virtualization, in this research paper we are using the concept of Network virtualization to implement the diskless client and zero hardware environment for computer laboratories[1]. Distributed resource manager concept helps us to improve the scalability of operating systems which will be accessed by the client during run time environment. The Operating system will be residing on the Host OS on Remote normal/RAID storage as per the requirements. This will be shared using NFS protocol and will make sure to be accessible outside the HOST Operating System [2]. The Network used here will be of 10GBPS at least, because the OS will be transferred over the network itself. This will be a local area network which will take care of the total transfer of the data over Ethernet. The Diskless Clients will boot from network and will request for the DHCP IP address by sending an ARP (Address Resolution Protocol) packets broadcast over the network. This DHCP server will provide the IP address using DORA process and along with the DHCP IP address, the next-server IP address will also be shared here which will be the IP of the same HOST OS which shares the Guest OS image using NFS [3] in this article operating system having dynamic IP address so that user not having rights to change the IP, after restarting the operating system IP can be changed every time.

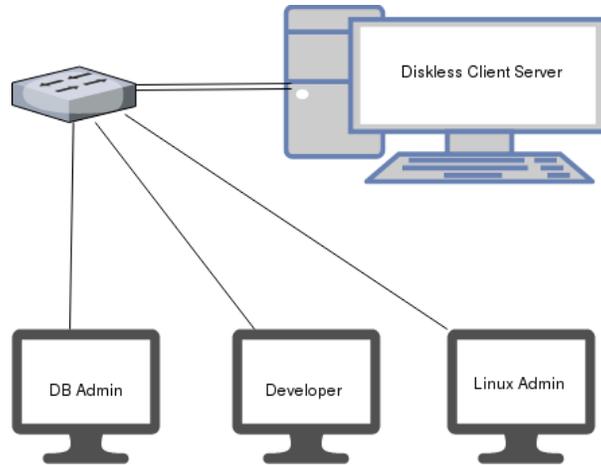
The TFTP Server will host the PXE kernel as well as the Guest OS kernel along with the network initrd.img files in the menu listing. The PXE kernel will be transferred via tftp server and that kernel image will read out the menu file thus moving to the NFS shared image of the Guest OS. The NFS protocol will be used to share the Guest Image of the OS for Disk Less Clients. Once the menu listing is loaded, the system will transfer the NFS hosted Guest OS image over network. The Client machines are the machines which just have Central Processing Unit, Monitor, Keyboard, Mouse but not the HDD. The system will boot using network, so network boot enabled NICs are required too. In order to keep the Host OS secure and robust, the Proxy and firewall is required. The firewall will have ports open for DHCP, NFS and TFTP services [3].

Server virtualization brings revolutionary changes to the data centers and it will provide the high speed connectivity during accession of operating systems from remote locations, currently Amazon provides full operating systems as pay as per use concept but not affordable to the small scale and other offices. Integrating servers by virtualization can significantly reduce the space and power consumption, enhances the IT services, simplify the hardware management so that virtualization term is very much popular in the business market.[4]. Cloud computing, and particularly the use of public clouds, brings advantages on the technical, The previous paper shows the way to provide the control of client which is in the network and manage the resources of the system from server end. My system having control calling provision over the network and put the call through RPC and complete the call lock mechanism which help to provide the control to the specified system.

Environmental and business sides, allowing multiple under-utilized systems to be consolidated within fewer physical servers hosting them. A cloud provider can manage physical resources in a very efficient way by scaling on the several hundreds and thousands of customers [1, 4].

### 2.1 Implementation

A diskless Client Server will contain the Base OS to host the actual chrooted Linux OS in some directory shared using Network File System protocol, and inside that directory, the Diskless Client Users database file /etc/passwd will be present. All the operations below has to be performed inside the chrooted OS. A restricted Shell has to be created in order to disallow user to change directories and only selected binaries will be given access to the users based on their tasks. For example, the DB admin will only get sql, tar and zip, unzip permissions, whereas a developer will get only vi and gcc permissions.



In Order to have a working prototype of a Disk Less Client in-house, following requirements needs to be full filled. A CentOS6 as a base machine is required to be used as a HOST OS which will serve as a Parent holding almost all the services which will be required by the Disk Less Client as shown in above Diagram. The same server will host the Local Repository of the Packages which will be deployed during GUEST image creation. The Services like DHCP (Dynamic Host Control Protocol), PXE (Pre-boot), TFTP and NFS will be running on the same HOST OS Machine too.

## 2.2 The Proof of Concept can be implemented as follows.

1. Install CentOS 6 machine with minimum 20GB free space and make sure it has a static IP address 192.168.0.254 (or whatever), we took 192.168.0.254/24 as the IP address for this Server
2. After installation, attach the CentOS 6 DVD to the machine, and mount it on the serv Diskless Client and RAID 1:

Diskless client server's TFTPBOOT directory which hosts the chrooted Operating system image for booting purpose of the diskless clients, can be created as a RAID 1 - Mirroring Device which will give maximum fault tolerance over the site in case of any disk goes faulty. No downtime required for any operations here for recovery. The architecture will look as follows.

Post implementation the diagram will look as above.

Implementation code.

```
[root@usercontrol ~]# cp /bin/bash /bin/restricted-bash
[root@usercontrol ~]# useradd db-admin -s /bin/restricted-bash
[root@usercontrol ~]# useradd developer -s /bin/restricted-bash
[root@usercontrol ~]# echo testpass | passwd --stdin db-admin
Changing password for user db-admin.
passwd: all authentication tokens updated successfully.
[root@usercontrol ~]# echo testpass | passwd --stdin developer
Changing password for user developer.
passwd: all authentication tokens updated successfully.
[root@usercontrol ~]# mkdir /home/db-admin/DB-BINS
[root@usercontrol ~]# mkdir /home/developer/DEV-BINS
[root@usercontrol ~]# vi /home/db-admin/.bash_profile
[root@usercontrol ~]# vi /home/developer/.bash_profile
[root@usercontrol ~]# egrep PATH /home/db-admin/.bash_profile /home/developer/.bash_profile
/home/db-admin/.bash_profile:PATH=$HOME/DB-BINS
/home/db-admin/.bash_profile:export PATH
/home/developer/.bash_profile:PATH=$HOME/DEV-BINS
/home/developer/.bash_profile:export PATH
[root@usercontrol ~]# su - db-admin
[db-admin@usercontrol ~]$ ls
-restricted-bash: ls: command not found
[db-admin@usercontrol ~]$ tar
```

```
-restricted-bash: tar: command not found
[db-admin@usercontrol ~]$ exit
logout
[root@usercontrol ~]# su - developer
[developer@usercontrol ~]$ ls
-restricted-bash: ls: command not found
[developer@usercontrol ~]$ date
-restricted-bash: date: command not found
[developer@usercontrol ~]$ exit
logout
[root@usercontrol ~]# ln -s /bin/tar /home/db-admin/DB-BINS/tar
[root@usercontrol ~]# ln -s /usr/bin/gcc /home/developer/DEV-BINS/gcc
[root@usercontrol ~]# ls -l /home/db-admin/DB-BINS/tar /home/developer/DEV-BINS/gcc
lrwxrwxrwx. 1 root root 8 Apr  5 19:57 /home/db-admin/DB-BINS/tar -> /bin/tar
lrwxrwxrwx. 1 root root 12 Apr  5 19:58 /home/developer/DEV-BINS/gcc -> /usr/bin/gcc
[root@usercontrol ~]# chattr +i /home/db-admin/.bash_profile /home/developer/.bash_profile
[root@usercontrol ~]# lsattr /home/db-admin/.bash_profile /home/developer/.bash_profile
----i-----e- /home/db-admin/.bash_profile
----i-----e- /home/developer/.bash_profile
[root@usercontrol ~]# su - db-admin
[db-admin@usercontrol ~]$ tar --version
tar (GNU tar) 1.23
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Written by John Gilmore and Jay Fenlason.
[db-admin@usercontrol ~]$ clear
-restricted-bash: clear: command not found
[db-admin@usercontrol ~]$ exit
[root@usercontrol ~]# su - developer
[developer@usercontrol ~]$ gcc --version
gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-18)
Copyright (C) 2010 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
[developer@usercontrol ~]$ clear
-restricted-bash: clear: command not found
[developer@usercontrol ~]$ tar
-restricted-bash: tar: command not found
[developer@usercontrol ~]$ exit
logout
```

### III. COMPARATIVE ANALYSIS

Table: 1 Comparative Analysis

Characteristics	Thin Client	Thick Client	Diskless Client using RAID
Low Watt Power Supply	Good	Poor	Good
Energy Efficiency	Good	Poor	Good
CPU Capacity	Poor	Good	Good
Cost	Medium	Poor	Excellent
Availability	Medium	Good	Good
Support	Good	Good	Good

We have checked the performance of diskless client using open source architecture along with existing technologies available in the market, we can connect more than 1000

Many of the aspect related with the speed, Availability, Graphics, Energy our solution is good. The most important part to understand in the POC is the dracut network image. If the root partition is on a network drive, one has to have the network dracut modules installed to create a network aware init image. This `initramfs.img` gets downloaded using `tftp` protocol and then creates an environment in Physical Memory in order to create a feasible environment for mounting the further file systems.

Why we are promoting open source because we all know paid operating systems cost is not affordable to any section like school, colleges, universities, offices, industries etc. and this cost is increasing day by day, we need have a stable operating system and stable hardware cost also, but the scenario is different the cost of hardware is also not affordable, if we installed this solution to above said area it will save cost energy and affordable to the mass education and every on can dream for free operating systems and every one can happy to learn the computer system without any trouble.

#### IV. CONCLUSION

The design of Diskless client with the RAID 1 Mirroring will be highly recommended to use for its 100% fault tolerance and faster read operations. This will enable the users to get access to their data with highest speed and system will make sure that even if one disk gets corrupted, the data is still there for further operations.

#### V. REFERENCES

- [1] Lan Wu and Wei Zhang, 2013. A Case study of hybrid WCET analysis for Xen Virtualization 978-1-4799-0053-4 @2014 IEEE.
- [2] Jibin Wang, Bohan Wang, 2016, A hybrid Main Memory applied in Virtualization Environments. 978-1-4673-8515-2 @2016 IEEE.
- [3] Shingo Takada, Akira Sato, Yasushi Shinjo, Hisashi Nakai, Akiyoshi Sugiki and kozo Itano," @2013 IEEE. A P2P Approach to Scalable Network-Booting, Third International Conference on Networking and Computing.
- [4] Noki Tanida, Kei Hiraki, Mary Inaba, "Efficient disk-to- disk copy through long-distance high-speed networks with background traffic", Fusion Engineering and Design, [www.elsevier.com/locate/fusengdes](http://www.elsevier.com/locate/fusengdes).
- [5] Ebrahim Ghazisaeedi, Ning Wang, Rahim Tafazolli, Link Sleeping Optimization for Green Virtual Network. 978-14673-4941-3 @ 2012 IEEE
- [6] G. Clarco, M. Casoni, 2012. "On the Effectiveness of Linux Containers for network virtualization. Simulation modeling practice and theory. [www.elsevier.com/locate/simpat](http://www.elsevier.com/locate/simpat).
- [7] Jinqian Liang, Xiaohong Guan, 2006 "A Virtual Disk Environment for providing file system recover" Science Direct.[www.elsevier.com/locate/cose](http://www.elsevier.com/locate/cose).