REMOTE BASED LINUX ADMINISTRATION[1]

***Gaurav S. Jadhav, **Vishal B. Tambe, ***Nikita Deshmukh**

**Student, Vellore Institute of Technology, Vellore*

***Student, Indian Institute of Technology, Guwahati*

****Student, Maharashtra Academy of Engineering*

ABSTRACT

The research work presented here includes a web-based interface for configuring all the internals the Linux operating system. It also includes work that has been done in the field of configuring Linux operating system from web. Here the proposed method is to configure Linux operating system from web.

WebLix is a system configuration[1] tool for Unix-like systems. It has a web-based interface for configuring all the internals the operating system. WebLix is basically use for Remote Administration of Servers. Remote Administration refers to any method of controlling a computer from a remote location. It consist of a simple web server, and a number of CGI programs which directly update system files. All the modules and web server is coded in PERL 5. Why there is a need of remote administration? Companies like ISPs and Telecoms need this facility to deal with their servers in case of faults. Fault can be defined as "An incorrect step, process, or data definition in a computer program" as we all know that software can possess more faults then hardware because they are much vulnerable to user abuse. WebLix allows moderately experienced users to manage their UNIX system through a web browser interface, instead of editing configuration files directly. The most recent version supports Apache, Squid and BIND, Samba and many other servers and services. It supports multiple operating systems and distributions, different languages, multiple users each with different levels of access, and SSL encryption.

Keywords— **Samba, SSL encryption, Remote Administration, WebLix.**

INTRODUCTION

For the inexperienced user, UNIX system administration can be daunting. Almost all services have configuration files that must be edited manually and often have complex formats. While these files are usually well documented in main pages, it is often unclear exactly how different sections and directives fit together. Furthermore, a single mis-spell or missing punctuation character can ruin an entire configuration file.

OBJECTIVES

Administrating a Linux network is such a tedious task now a day for Administrator in terms of location, complexity, flexibility. The main objective of this work is to simplify the process of managing a Linux or Unix system. Normally you need to manually edit configuration files and run commands to create accounts, set up a web server. The purpose is to make such a system that will

[1] How to cite the article:
Jadhav S.G., Tambe B.V., Deshmukh N., Remote Based Linux Administration, *International Journal of Advances in Engineering Research*, April 2013, Vol 3, Issue 4, 59-64

allow you to perform these tasks remotely through an easy to use web interface, and automatically updates all of the required configuration files for you. By using any modern web browser, you can setup user accounts, Cron Jobs, SAMBA configuration[2], file sharing and much more. This will make the job of administering your system much easier.

This research work is not just to make system administration easy, but also makes the problems more visible and the solutions more consistent.

PROPOSED WORK

These techniques prescribed in previous work are used to automatically generate a large number of images for a specified object class. The simple architecture for the proposed system is:

WebBrowser: WebLix is also accessible, in the sense that it can be used successfully from nearly any browser. Text mode browsers, small screen displays, and nearly anything else can be accommodated through the appropriate use of themes and numerous configurable display parameters. For each user new session id will be created and maintain till the user logged in.

Configuration Type: WebLix allows you to configure most of the important aspects of WebLix itself, as well as install new modules, upgrade existing modules, and upgrade WebLix itself. It also provides a means to configure modules such as User Management, Disk and Network File system, File manager, Process management and configure servers.

WebLix Server: The WebLix Server category is for administration of the server applications that run on a system to provide some service to clients on the network. One example is the Apache web server daemon[4]. The WebLix server will exchange the data from Linux administrator who work from remote place and the Linux.

Run CGI Scripts: CGI[3], or Common Gateway Interface, provides a means for a website to have dynamic, program generated, content on a web page. CGI programs can interact with the user through the use of input fields and can provide different data based on the information returned. CGI programs can be written in nearly any language, though it is most common for them to be written in Perl, PHP, Python, Java, C, and bash or other shell scripting languages [8]. The CGI Programs module provides an interface to the global CGI options of Apache.

Configure Files: WebLix allows finer grained control over many modules, and this functionality is becoming more flexible with every release. For example, a user with permission to configure the Cron Job Module can be denied the ability to edit some specific aspects the configuration. WebLix knows how to interact with the system based on configuration files for each module, that are selected based on the operating system configured here.

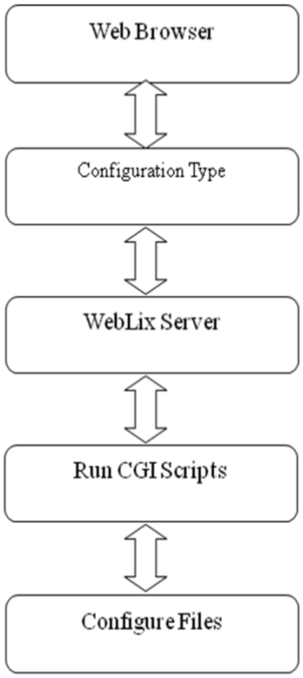

Fig 1: Simple Architecture For Proposed System

SYSTEM IMPLEMENTATION

WebLix is implemented as a large number of Perl CGI programs, arranged into modules. Each module handles the configuration of some Unix service, such as System User Management,, Process management, File manager, Network Management. Each module has one or more functionality to configure Linux system. Typically, these functions deal with the actual configuration files.

Web based interface for system service administration for Linux : It consists of a simple web server, and a number of CGI programs which directly update system files. The web server and all CGI programs are written in Perl, and use no nonstandard Perl modules. By using any browser that supports tables and forms, you can setup user accounts, Apache, DNS, file sharing etc.

Role Based Implementation :-System administrator assigns various privileges to different users that logging to system, WebLix will consider these privileges while logging into system and also you can manage these privileges by using WebLix.

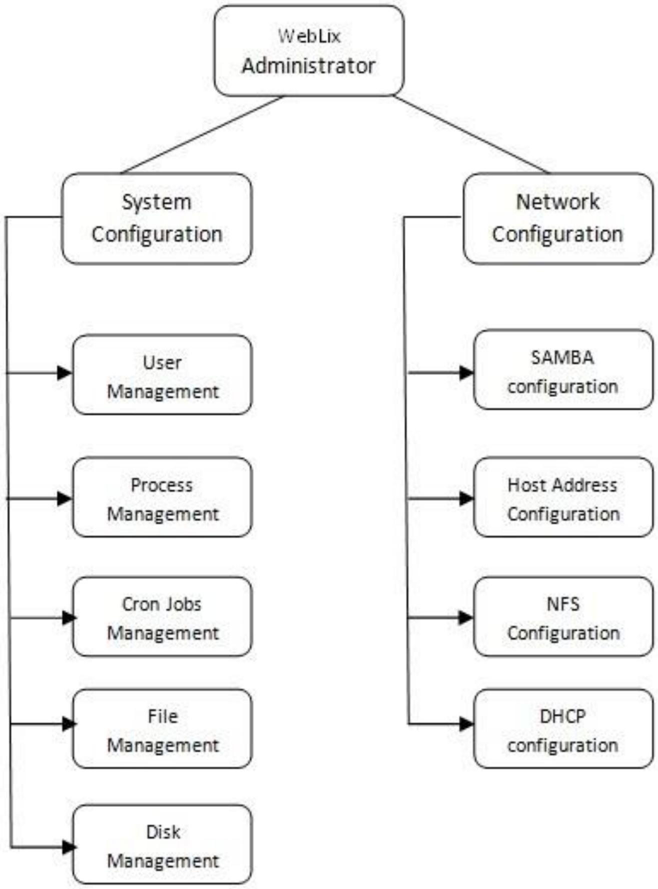

Fig 2: Hierarchy of proposed system

A. System Configuration[6]:-

User management:-

In WebLix, you can use the Users and Groups module (covered on Users And Groups) to edit all details of a user, including his password. However, if you just need to change passwords on a regular basis, or want to give less-trusted admin permissions to only change passwords, the Users and Groups module is un-necessarily complex.

Process Management:-This module displays all available information about the process, including its full command line, parent command and any sub-processes. You can just to the information page for the parent by clicking on its command, or to the page on any of the sub-processes by clicking on its processID. A list of files that the process has open & network connections that it is currently using can be viewed by clicking the Files and Connections button. The process can be stopping using a TERM signal by clicking the Terminate Process button. Because this can be ignored by some commands, the Kill Process button can be used to send a KILL signal if the termination fails. Unless the process is hung inside a kernel system call, killing it is guaranteed to succeed.

Scheduled cron jobs:-Cron jobs are very useful for performing regular system tasks, such as cleaning up log files, synchronizing the system type, backing up files and so on. Most Linux distributions will have several Cron jobs that were setup by default as part of the operating system install process for doing things like removing unneeded kernel modules, updating the database used by the locate command and rotating log files.

File Manager:-In WebLix, you can use File Manager module to View, edit and change permissions on files and directories on your system with a Windows-like file manager.

Disk and Network file system:-UNIX systems support many different kinds of file system, some for files stored on local hard disks and some for files on networked file servers. On Linux, the file systems on your hard disks will probably be in ext2 or ext3 format. Many other local file system types exist, such as iso-9660 for CD-ROMs, vfat for Windows partitions, and xfs and reiserfs for high performance file access. Every local file system type uses a different format for storing data on disk, so if a partition has been formatted as a file system of a particular type, then it must be mounted as that type. In WebLix, you can use the Disk and Network file system module to mount and eject the file system.

B. Network Configuration

Samba windows file sharing[2]:-This WebLix module allows you to specify directories and printers to be shared to Windows clients using the SMB protocol. It can be found in the Servers category, and when its link is clicked the main page as shown in the screenshot below will be displayed. All existing shares are listed, along with their paths and the users that they are available to. Below them are icons for setting various global options that apply to all shares, links for managing Samba users and a button for starting or re-starting the server processes.

Host address configuration[6]:- Every Unix system has a hostname, which appears in the login prompt, system logs, outgoing email and on every WebLix page. Normally the hostname is the same as or part of the DNS name for the system's IP address, but this does not have to be the case, especially if the system is not connected to a network or only connects occasionally via dialup. However, for permanently connected systems the hostname should be the hosts fully qualified DNS name (like server1.foo.com), or just the first part (like server1). Anything else is likely to cause confusion and possibly network problems.

NFS Configuration[9]: The Network File System is certainly one of the most widely used network services. Network file system (NFS) is based on the Remote procedure call which allows the client to automatically mount remote file systems and therefore transparently provide an access to it as if the file system is local.

DHCP server management [7]:-DHCP is a protocol that allows hosts to request and be assigned an IP address on a local area network. It is used to simplify the process of IP address assignment, as a single server can manage the addresses of multiple clients. It is also useful for systems such as laptops that are moved between multiple networks, as they do not need to be re-configured for each LAN that they connect to.

CONCLUSION

WebLix can address more quickly the administration of a Linux system. The interface allows the different modules offered to assist the administrator to go further in the configuration of individual services or system. WebLix makes it possible to start and grow more easily in this area. However it is important to do as often as possible parallel with the console command line to properly identify actions at a lead in a module configuration. The WebLix tool has single-handedly brought Linux administration to a state of simplicity. Although there are numerous Linux administration tools, none of them come close to the ease of use and flexibility of WebLix. WebLix is not only server-only administration tool. WebLix is also very useful for remotely working with your desktop machine.

ACKNOWLEDGMENT

Success is never achieved single handedly so our duty to acknowledge all those who have provided timely guidance and helping hand in making our project a success.
We would like to thank our guide Prof. R.C.Sanap and head of the department for the encouragement and support that they have extended. I would also like to thank the anonymous reviewers who provided helpful feedback on my manuscript.

REFERENCES

[1] Jack Tackett, Jr. Steve Burnett, (1999), *Special Edition Using Linux*, pp. 14-26,1999.
[2] Robert Eckstein, David Collier-Brown, Peter Kelly, (November, 1999), *Using Samba*; 1st Edition; pp.46-73
[3] Kirrily Robert, Paul Fenwick, Jacinta Richardson (2001-2004), *CGI Programming with Perl*
[4] Ken Coar & Rich Bowen, (2008), *Apache Cookbook: Solutions and Examples for Apache Administrators*.
[5] Craig Hunt; *Linux DNS Server Administration*; Craig Hunt Linux Library
[6] Wale Soyinka, (September 2008), *Linux Administration: A Beginner's Guide*, Fifth Edition.
[7] Ralph Droms, Ted Lemon, (October 2002), *DHCP Handbook*; The Adobe Reader.
[8] Steve Parker; (August 2011), *Shell Scripting: Expert Recipes for Linux, Bash and more.*
[9] Erez Zadok, *Linux NFS and Automounter Administration.*