New Jersey Institute of Technology
Digital Commons @ NJIT

Theses

Theses and Dissertations

Fall 2002

Firewall interface for java FTP SMTP and HTTP servers

Ratan Kumar Erukulla
New Jersey Institute of Technology

Follow this and additional works at: https://digitalcommons.njit.edu/theses

Part of the Electrical and Electronics Commons

Recommended Citation
Erukulla, Ratan Kumar, "Firewall interface for java FTP SMTP and HTTP servers" (2002). *Theses*. 594.
https://digitalcommons.njit.edu/theses/594

This Thesis is brought to you for free and open access by the Theses and Dissertations at Digital Commons @ NJIT. It has been accepted for inclusion in Theses by an authorized administrator of Digital Commons @ NJIT. For more information, please contact digitalcommons@njit.edu.

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

NJIT
New Jersey's Science & Technology University

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

FIREWALL INTERFACE FOR
JAVA FTP SMTP AND HTTP SERVERS

by

Ratan Kumar Erukulla

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Department of Electrical and Computer Engineering

January 2003

ABSTRACT

FIREWALL INTERFACE FOR JAVA FTP, SMTP AND HTTP SERVERS

by
Ratan Kumar Erukulla

The objective of my thesis is to develop a firewall interface for FTP, HTTP and SMTP servers. A firewall interface is a graphical user interface (GUI) to configure the properties of the server. Using this interface, we can spy a user, block and unblock the IP address of a client, display statistics of user and server. We can track all the user actions while he is connected to the server. These actions are logged and based on the log data we can deny access to users performing illegal operations. All the servers and interfaces are written in Java. Java was chosen because it is platform independent.

Blank Page

APPROVAL PAGE

FIREWALL INTERFACE FOR JAVA FTP SMTP AND HTTP SERVERS

Ratan Kumar Erukulla

Dr. Constantine Manikopoĺous, Thesis Advisor　　Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Sotirios Ziavras, Committee Member　　Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Bin He, Committee Member　　Date
Systems Analyst, XPRT Solutions Inc.,

BIOGRAPHICAL SKETCH

Author: Ratan Kumar Erukulla

Degree: Master of Science

Date: January 2003

Undergraduate and Graduate Education

- Master of Science in Electrical Engineering
  New Jersey Institute of Technology, Newark, NJ, 2003

- Bachelor of Engineering in Electronics and Communications Engineering
  Jawaharlal Nehru Technological University, Hyderabad, India, 2001

Major: Electrical Engineering

This thesis is dedicated to my

beloved parents and other family members.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Dr. Constantine Manikopolous, who not only helped me as my thesis advisor, but also for his invaluable guidance, support and encouragement throughout the research.

Special thanks to Dr. Sotirios Ziavras and Dr. Bin He for actively participating as members of the committee.

I am grateful for the help from Mr. Ravi Kiran Yalamarthy.

I would like to thank Dr. Ronald Kane and Ms. Clarisa Gonzalez-Lenahan for their time and the help they rendered in the documentation of this thesis.

TABLE OF CONTENTS

Chapter Page

1 INTRODUCTION 1

1.1 Need for Firewalls 1

1.2 FTP, SMTP & HTTP Servers 2

1.2.1 FTP 2

1.2.2 SMTP 2

1.2.3 HTTP 2

1.2 Firewall Interface 3

2 INSTALLING JAVA 4

2.1 Getting Java 4

2.2 Setting PATH and CLASSPATH 4

3 FIREWALLS 9

3.1 What is a Firewall? 9

3.2 Why do I Want a Firewall? 9

3.3 What Can a Firewall Protect Against? 10

3.4 What Can't a Firewall Protect Against? 11

3.5 Types of Firewalls 11

3.5.1 Network Layer Firewalls 12

3.5.2 Application Layer Firewalls 12

4 FTP SERVER 13

4.1 What is a FTP Server? 13

4.2 FTP Fundamentals 14

TABLE OF CONTENTS
(Continued)

Chapter **Page**

4.2.1 FTP Fundamentals 15

4.2.2 Anonymous and Non-Anonymous FTP 15

5 FTP FIREWALL INTERFACE 16

5.1 Installing the Server 16

5.2 Snapshots and Description 17

5.2.1 User Screen 21

5.2.3 IP Screen 24

5.2.3 Connection Screen 26

5.2.4 Statistics 29

5.2.5 Upload Screen 31

5.2.6 Download Screen 34

5.2.7 Delete Screen 35

6 SMTP SERVER 37

6.1 What is SMTP? 37

6.2 The SMTP Model 38

6.3 Implemented Features 39

6.4 Sizes and Limitations 40

7 SMTP SERVER INTERFACE 41

7.1 Installing the Server 41

7.2 Snapshots and Description 41

TABLE OF CONTENTS
(Continued)

Chapter **Page**

7.2.1 General Settings 42

7.2.2 User Settings 44

7.2.3 Log Settings 46

7.2.4 Remote POP Accounts 47

7.2.5 JDBC Settings Panel 49

7.2.6 Statistics 51

7.2.7 Saving the Changes 52

7.3 Using Auto Reply Feature 53

8 HTTP SERVER 55

8.1 Types of HTTP Requests 55

8.2 HTTP Request Methods 55

8.2.1 GET Method 56

8.2.2 HEAD Method 56

8.2.3 PUT Method 56

8.2.4 DELETE Method 57

8.2.5 LINK Method 57

8.2.6 UNLINK Method 57

8.3 Starting the HTTP Server 57

8.4 Implemented Features 58

8.5 Content Types Supported 58

8.6 Directory Browsing and Index Pages 60

TABLE OF CONTENTS
(Continued)

Chapter **Page**

8.7 Logging of Requests 61

8.8 To Block IP Address 63

8.9 Miscellaneous Features 65

9 CONCLUSIONS 67

REFERENCES 68

LIST OF TABLES

Chapter **Page**

8.1 HTTP Requests 55

8.2 Content Types Supported 58

8.3 Status Request Codes 62

LIST OF FIGURES

Figure **Page**

2.1 Getting System Properties 5

2.2 System Properties 6

2.3 Environment Variables Screen 7

2.4 Edit System Variables Screen 8

5.1 Command Prompt at FTP Server 16

5.2 FTP Start Screen 17

5.3 Getting FTP Configuration File 18

5.4 FTP Configuration Display 19

5.5 FTP User Screen 21

5.6 Anonymous User Screen 23

5.7 FTP Ban IP Screen 24

5.8 FTP List of Banned IP 25

5.9 FTP Connection Screen 26

5.10 FTP Connected Users 27

5.11 FTP User Screen 28

5.12 FTP Spy User Screen 29

5.13 FTP Statistics Screen 30

5.14 FTP Upload Access Denied Screen 31

5.15 FTP Upload Screen 32

5.16 FTP Uploaded Files Screen 33

5.17 FTP Download Screen 34

LIST OF FIGURES
(Continued)

Figure **Page**

5.18 FTP Delete Screen 35

5.19 FTP Deleted Files Screen 36

6.1 The SMTP Model 39

7.1 SMTP General Settings 42

7.2 SMTP User Settings 44

7.3 SMTP Log Settings 46

7.4 SMTP Remote Settings 47

7.5 SMTP JDBC Settings Panel 49

7.6 SMTP Statistics 51

8.1 HTTP Directory Browsing 61

8.2 HTTP Command Prompt 62

8.3 HTTP Starting 63

8.4 HTTP Firewall Screen 63

8.5 HTTP Firewall Screen with Banned IP's 64

8.6 HTTP Logging Screen 65

CHAPTER 1

INTRODUCTION

1.1 Need for Firewalls

Firewalls try to protect computer users from security breaches. Any computer connected to internet is vulnerable to hackers. Firewalls try to remedy the risk of hackers through software or hardware.

A firewall is a collection of components placed at the edges of a network. All traffic going into or out of the network must pass through the firewall. Firewalls can be software based or hardware based. Hardware based firewalls are much faster but more expensive. The firewalls can do a variety of things such as virus protection, filtering packets, and blocking hackers from open ports.

Firewalls cannot be used with modems so in the past home users had no use for firewalls, but now that DSL and cable modems are becoming prominent there is a need for firewall use in the home too. These connections are open at all times and are more susceptible to hackers because the hackers have unlimited time to hack at their computers.

Besides providing protection, firewalls can capture and log a lot of data about the traffic that is coming to and from your computer. Based on this log data we can determine the level of threat that we have been exposed to. Depending on that we can take necessary action to strengthen our security and report the intruders IP address to the proper authority.

1.2 FTP, SMTP&HTTP Servers

1.2.1 FTP

FTP stands for the acronym File Transfer Protocol.

A FTP server is a way off sharing files with other people on the Internet. You setup a FTP server and other people can download files you have on your hard disk. This provides you with an easy way off sharing files. A FTP Server allows Uploading which means a user is sending files to your FTP Server and Downloading which means a user is transferring files from your FTP Server.

1.2.2 SMTP

SMTP stands for the acronym Simple Mail Transfer Protocol.

SMTP, a process can transfer mail to another process on the same network or to some other network via a relay or gateway process accessible to both networks. Within the Internet, email is delivered by having the source machine establish a TCP connection to port 25 of the destination machine. Listening to this port is an email daemon that "speaks" SMTP. This daemon accepts incoming connections and copies messages from them into the appropriate mailboxes. If a message cannot be delivered, an error report containing the first part of the undeliverable message is returned to the sender.

1.2.3 HTTP

HTTP is an acronym for HyperText Transfer Protocol

The HTTP protocol consists of two fairly distinct items: the set of requests from browsers to servers and the set of responses going back from the server.

1.3 Firewall Interface

My thesis is about developing a user interface for configuring FTP, HTTP & SMTP servers. All the servers are written purely in Java. Configuring the server involves the following:

- Setting the administrator for the server.
- Blocking and unblocking IP address of a client. This is useful if we want to deny access to a client.
- Displaying the statistics of a user. Statistics include login time, number of uploads & downloads (for FTP).
- Displaying the current clients connected to the server.
- Spying a user. This is used to track user actions while connected to the server.
- Displaying the statistics of the server. Statistics include when the server was started, number of users currently logged in etc.

CHAPTER 2

INSTALLING JAVA

2.1 Getting Java

All the servers are written in Java. Due to my zeal towards Java I used java for developing software for server with a neat user interface. I choose Java because it is the fastest growing technology and I think we can do wonders with Java.

The basic requirement to install the servers is little knowledge of networking and Java programming. To run the servers our system must have JDK1.3 (Java Development Kit) installed on it. We can download the latest JDK (Java Development Kit) kit at http://java.sun.com/downloads. Right now the latest Java Development Kit is JDK1.4. It is available as executable file of around 30MB. Just download the .exe file and run the file and follow the on screen instructions. The JDK will come with JRE (Java Run Time Environment) which is useful when browsing internet. Also it is always better to download the documentation for Java Programming. The documentation consists of Java API's using which we can write Java Programs according to our needs. There are some thousands of classes and interfaces in the API which can be used according to the program requirements.

2.2 Setting PATH and CLASSPATH

After downloading the JDK (Java Development Kit) kit and the project files, before running the Server. bat, we need to do one more thing. We need to set the

PATH and CLASSPATH for our JDK so that it knows where it has to look if a user types a particular command.

To set the PATH and CLASSPATH in Windows NT, Windows 2000 and Windows XP –

- Right click on "My Computer" and click "Properties"

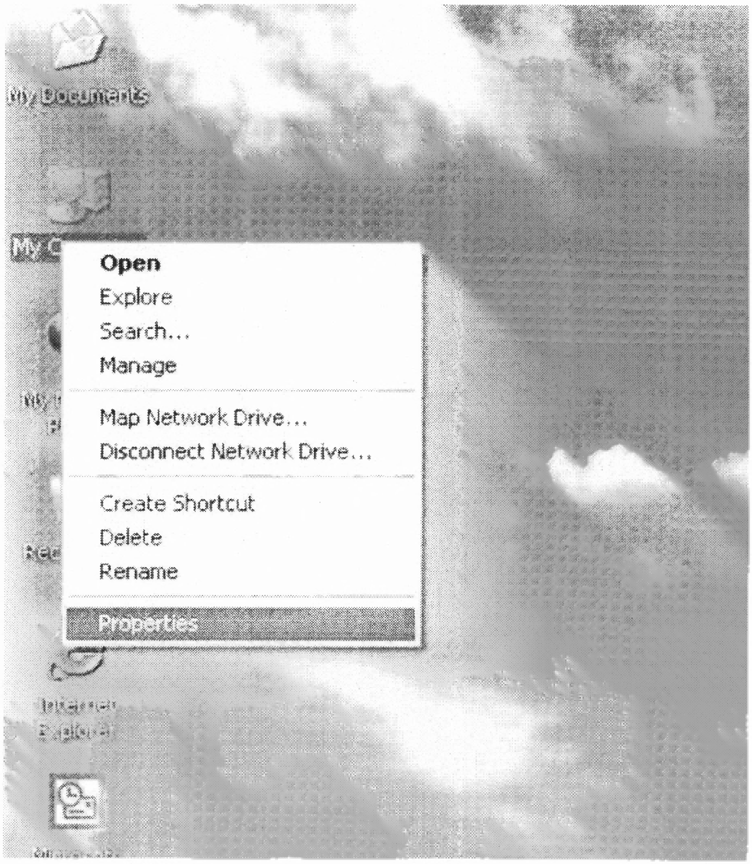
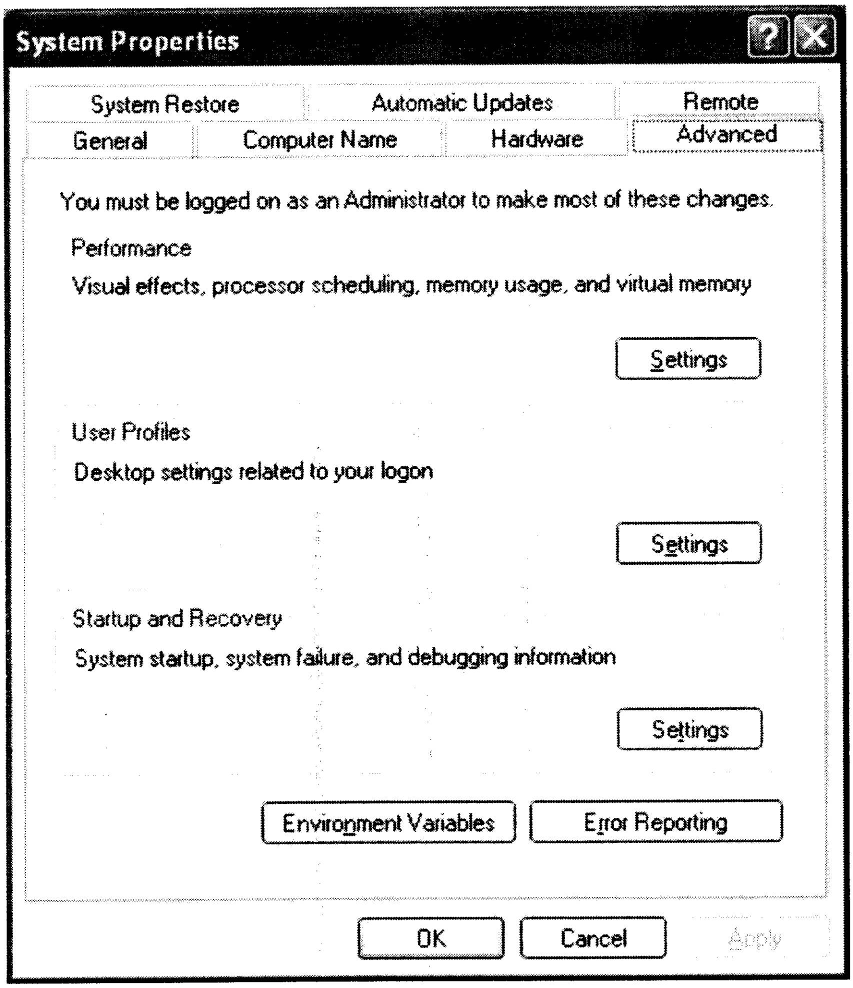


Figure 2.1 Getting System Properties

- In the dialog box select "Advanced" tab as shown below



You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles

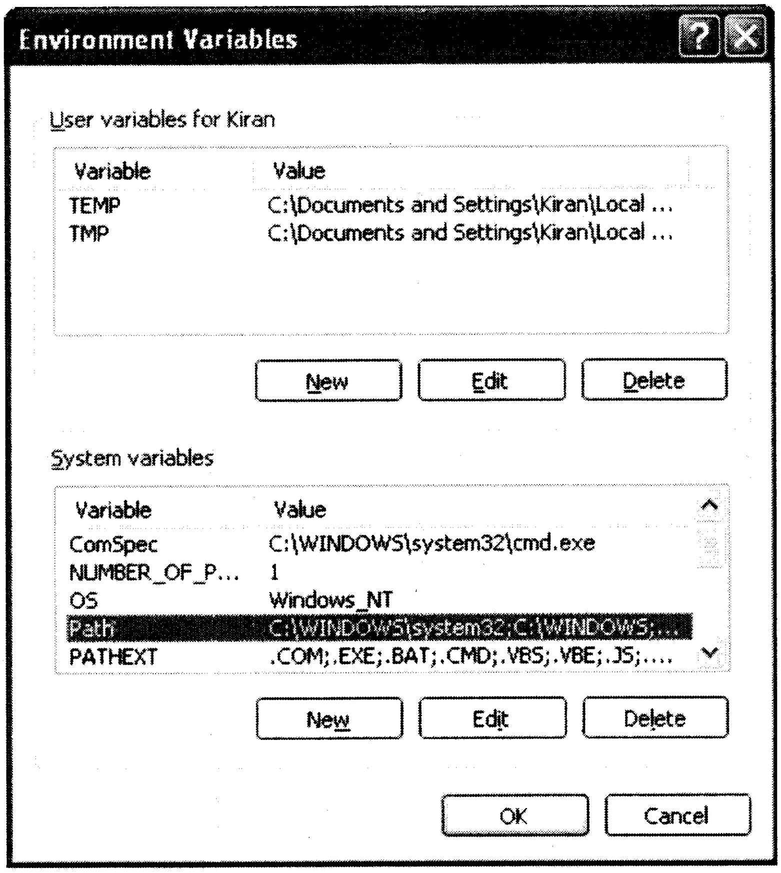
Desktop settings related to your logon

Startup and Recovery

System startup, system failure, and debugging information

Figure 2.2 System Properties

- Click on “Environment Variables” button
- Select the “PATH” variable as shown



Environment Variables

User variables for Kiran

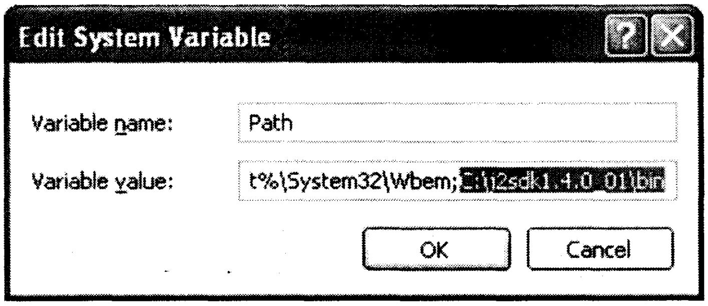
Variable	Value
TEMP	C:\Documents and Settings\Kiran\Local ...
TMP	C:\Documents and Settings\Kiran\Local ...

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
NUMBER_OF_P...	1
OS	Windows_NT
Path	C:\WINDOWS\system32;C:\WINDOWS;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

Figure 2.3 Environment Variables Screen

- Double Click on “PATH” and in a dialog box will come. Here at the end add you java installation directory, for me something like this, C:\j2sdk1.4.0_01\bin and press OK.



Variable name: Path

Variable value: t%\System32\Wbem;C:\j2sdk1.4.0_01\bin

Figure 2.4 Edit System Variable Screen

- Under “User Variables” if “CLASSPATH” is present double click on that and add the path of your java class files. If you can’t see variable named “CLASSPATH” then click on “New” and enter “Variable Name” as “CLASSPATH” and “Variable Value” as “your path to java class file for e.g. C:\Documents Settings\Administrator\Desktop\mycode\
- Press ok. This will most probably will solve the “NoClassFoundError” errors when we compile the Java programs.

Once we setup Java, we are ready to install our servers. I will explain how to set up each server in next chapters.

CHAPTER 3

FIREWALLS

3.1 What is a Firewall?

A firewall is a system or group of systems that enforces an access control policy between two networks. The actual means by which this is accomplished varies widely, but in principle, the firewall can be thought of as a pair of mechanisms: one, which exists to block traffic, and the other, which exists to permit traffic. Some firewalls place a greater emphasis on blocking traffic, while others emphasize permitting traffic. Probably the most important thing to recognize about a firewall is that it implements an access control policy. If you don't have a good idea of what kind of access you want to allow or to deny, a firewall really won't help you. It's also important to recognize that the firewall's configuration, because it is a mechanism for enforcing policy, imposes its policy on everything behind it. Administrators for firewalls managing the connectivity for a large number of hosts therefore have a heavy responsibility.

3.2 Why do I Want a Firewall?

The Internet, like any other society, is plagued with the kind of jerks who enjoy the electronic equivalent of writing on other people's walls with spray paint, tearing their mailboxes off, or just sitting in the street blowing their car horns. Some people try to get real work done over the Internet, and others have sensitive or proprietary data they must protect. Usually, a firewall's purpose is to keep the jerks out of your network while still letting you get your job done.

Many traditional-style corporations and data centers have computing security policies and practices that must be adhered to. In a case where a company's policies dictate how data must be protected, a firewall is very important, since it is the embodiment of the corporate policy. Frequently, the hardest part of hooking to the Internet, if you're a large company, is not justifying the expense or effort, but convincing management that it's safe to do so. A firewall provides not only real security--it often plays an important role as a security blanket for management.

Lastly, a firewall can act as your corporate ``ambassador" to the Internet. Many corporations use their firewall systems as a place to store public information about corporate products and services, files to download, bug-fixes, and so forth. Several of these systems have become important parts of the Internet service structure (e.g.: UUnet.uu.net, whitehouse.gov, gatekeeper.dec.com) and have reflected well on their organizational sponsors.

3.3 What Can a Firewall Protect Against?

Generally, firewalls are configured to protect against unauthenticated interactive logins from the ``outside" world. This, more than anything, helps prevent vandals from logging into machines on your network. More elaborate firewalls block traffic from the outside to the inside, but permit users on the inside to communicate freely with the outside. The firewall can protect you against any type of network-borne attack if you unplug it.

Firewalls are also important since they can provide a single ``choke point" where security and audit can be imposed. Unlike in a situation where a computer system is being attacked by someone dialing in with a modem, the firewall can act as an effective ``phone tap" and tracing tool. Firewalls provide an important logging and auditing function; often they provide summaries to the administrator about what kinds and amount of traffic passed through it, how many attempts there were to break into it, etc.

3.4 What Can't a Firewall Protect Against?

- Firewalls can't protect against attacks that don't go through the firewall. *For a firewall to work, it must be a part of a consistent overall organizational security architecture.* Firewall policies must be realistic and reflect the level of security in the entire network
- Another thing a firewall can't really protect you against is a traitor inside your network
- Lastly, firewalls can't protect against tunneling over most application protocols to poorly written clients. Tunneling ``bad" things over HTTP, SMTP, and other protocols is quite simple and trivially demonstrated. Security isn't ``fire and forget".

3.5 Types of Firewalls

Basically firewalls operate on two layers

1. Network layer
2. Application layer

3.5.1 Network Layer Firewalls

They generally make their decisions based on the source, destination addresses and ports in individual IP packets. A simple router is the "traditional" network layer firewall, since it is not able to make particularly sophisticated decisions about what a packet is actually talking to or where it actually came from. Network firewalls are typically used when speed is essential. Since packets are not passed to the application layer and the contents of the packet are not being analyzed, packets can be processed quicker. This can be advantageous for firewalls that scan for connections to web and email servers, especially ones that have high amounts of traffic.

3.5.2 Application Layer firewalls

They generally are hosts running proxy servers, which permit no traffic directly between networks, and which perform elaborate logging and auditing of traffic passing through them. Since the proxy applications are software components running on the firewall, it is a good place to do lots of logging and access control. Application layer firewalls can be used as network address translators, since traffic goes in one ``side" and out the other, after having passed through an application that effectively masks the origin of the initiating connection. Major benefit of application firewalls is that they typically support the ability to report to intrusion detection software.

CHAPTER 4

FTP SERVER

FTP is an acronym for File Transfer Protocol.

The objectives of FTP are

1) To promote sharing of files (computer programs and/or data).

2) To encourage indirect or implicit (via programs) use of remote computers.

3) To shield a user from variations in file storage systems among hosts.

4) To transfer data reliably and efficiently. FTP, though usable directly by a user at a terminal, is designed mainly for use by programs.

4.1 What is a FTP Server?

A FTP server is a way off sharing files with other people on the Internet. You setup a FTP server and other people can download files you have on your hard disk. This provides you with an easy way off sharing files. A FTP Server allows Uploading which means a user is sending files to your FTP Server and Downloading which means a user is transferring files from your FTP Server.

FTP stands for "file transfer protocol" and it is a method of transferring files between two computers. FTP allows you to get access to files stored in the disk directory of a remote computer that is connected to the Internet. Sometimes we forget that the Internet, when you strip away all the hype, is really still just a large network of computers. One of the primary purposes of a network is to allow different computers on the network to share resources, including files.

At least some form of FTP is available to anyone who has access to the Internet. FTP is an information "service" available to any Internet user, just like the web and e-mail. FTP, like many other services, is independent of the software you use to access it. For example you can use a browser or other programs, like the Windows 95 FTP client, WS_FTP LE, or Fetch for the Mac to access Internet files using FTP.

4.2 FTP Fundamentals

4.2.1 Uploading and Downloading

Simply put, FTP allows you to enter a directory on a computer connected to the Internet and transfers a file to or from that directory to a directory on another computer. Normally, you will be transferring the file to or from a large, multi-user computer and your own computer. Files can be transferred in either direction. "Downloading," refers to a transfer of a file from a remote computer to your computer. "Uploading" refers to a transfer of a file from your computer to a remote computer.

In order to upload or download a file by FTP, you need to do four things:

- Login into a remote computer that has been configured as an FTP server.
- Submit a username and a password to gain access to the remote system.
- Change to the particular directory on the remote system which contains the file you wish to download or upload.
- Transfer the file to or from the system in question.

4.2.2 Anonymous and Non-Anonymous FTP

There are two types of FTP connections available on the Internet: "anonymous" and "non-anonymous." The most widely used type is anonymous FTP. In fact, you may have used it without even knowing it. Many web pages contain links to files that you can download. Often these links point to a file in an anonymous FTP directory. If a file is stored in an anonymous FTP directory virtually anyone with Internet access and an FTP program of some sort, even a web browser, can download the file. Uploading, on the other hand, is not usually possible with anonymous FTP. Anonymous FTP, therefore, is used primarily to give the Internet public download access to a particular directory of files. Anyone can download files from the directory, but only the "owner" of directory can upload to the directory.

When you connect to any FTP directory, the host system asks for your username and password before allowing you access to the directory (this process is done behind the scenes when you use a web browser to access an FTP directory). With an anonymous FTP directory, any user can gain access because the username is always "anonymous" and the password is always the user's e-mail address.

Non-anonymous FTP, on the other hand, requires a unique username and password for the FTP directory in question. Normally, you will use non-anonymous FTP to get access to directories that YOU OWN on the web server.

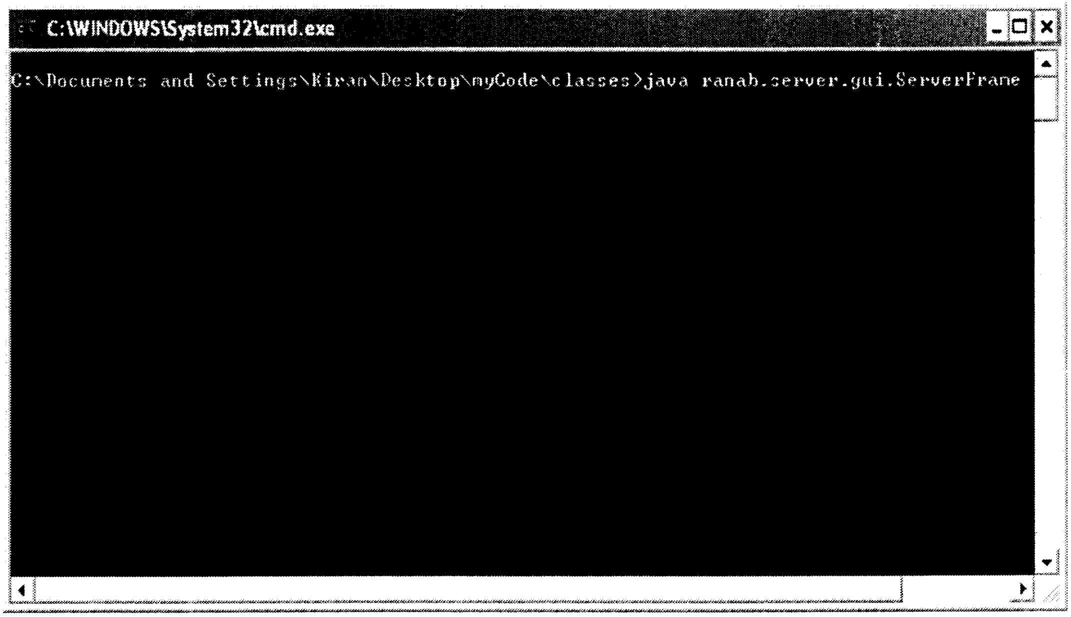
CHAPTER 5

FTP FIREWALL INTERFACE

5.1 Installing the server

Copy all the Project files included in the Floppy disk or Zip disk. You will need just the Class files as I already compiled all the java files. Also in the "bin" directory I wrote a batch file, which will run the corresponding Java program, which will start the FTP Server and displays User Interface to perform various actions.You can run the program from the command prompt also. You have to got the directory of classes and there type

```
java ranab.server.gui.ServerFrame
```



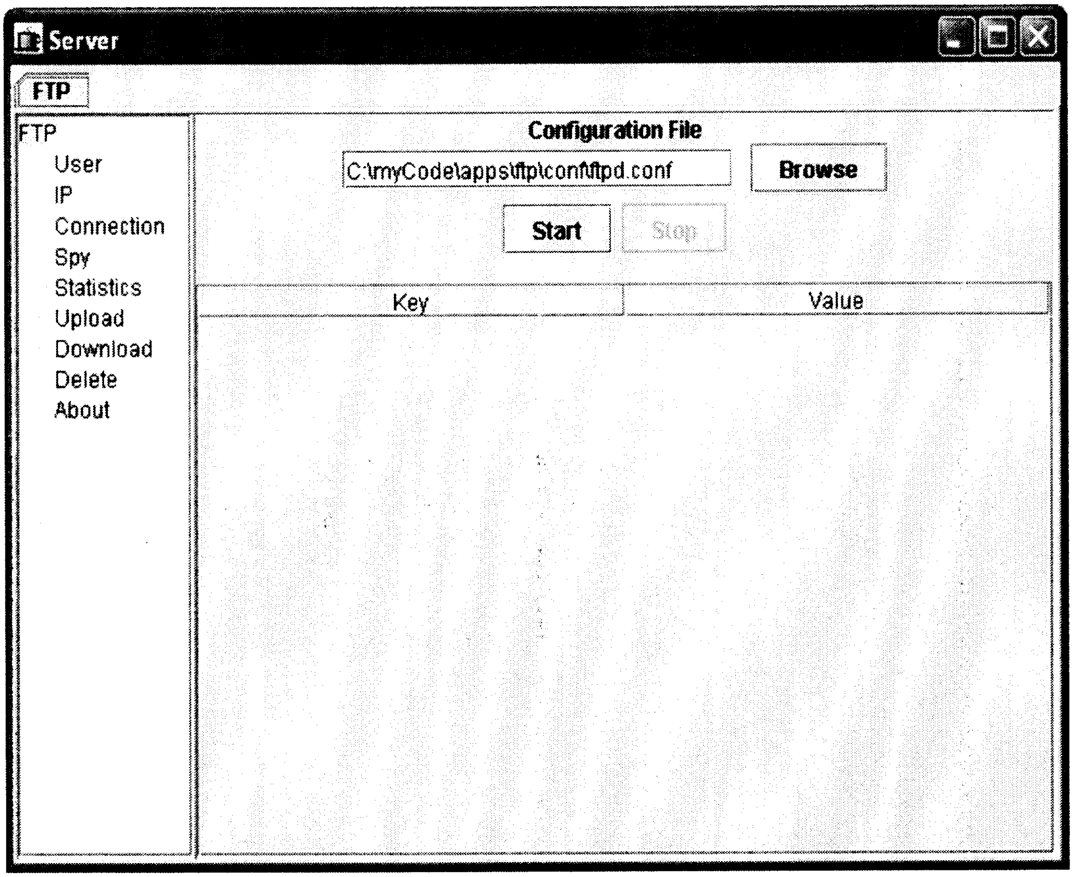
```
C:\Documents and Settings\Kiran\Desktop\myCode\classes>java ranab.server.gui.ServerFrame
```

Figure 5.1 Command Prompt at Ftp Server

This command will start the User Interface for the FTP Server.

5.2 Snapshots and Description

We can start the running the program by opening the file "Server. bat" included in \bin directory. As soon as we open "Server. bat" file a "Command Prompt" is opened automatically and User Interface for the Java FTP Server will appear.

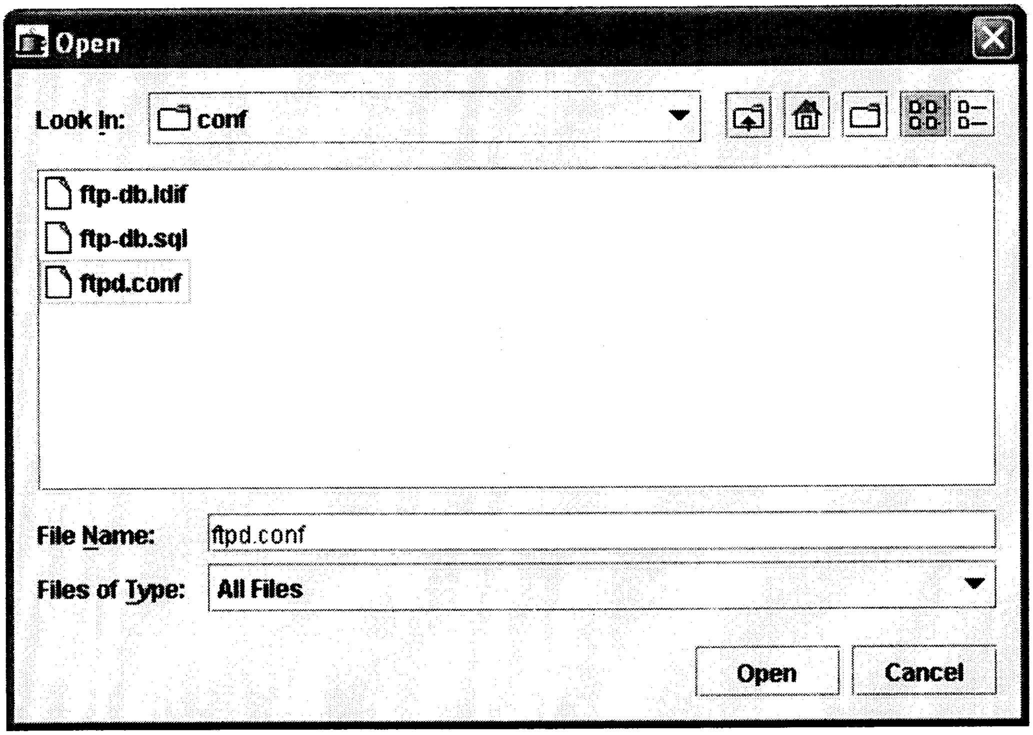


C:\myCode\apps\ftp\conf\ftpd.conf

Figure 5.2 FTP Start Screen

I wrote a "Configuration File" in which I set some default preferences for the FTP Server. To start the Java FTP Server we need to give the path of the configuration file "ftpd.conf".

Click on “Browse” button and locate the configuration file and press “OPEN”



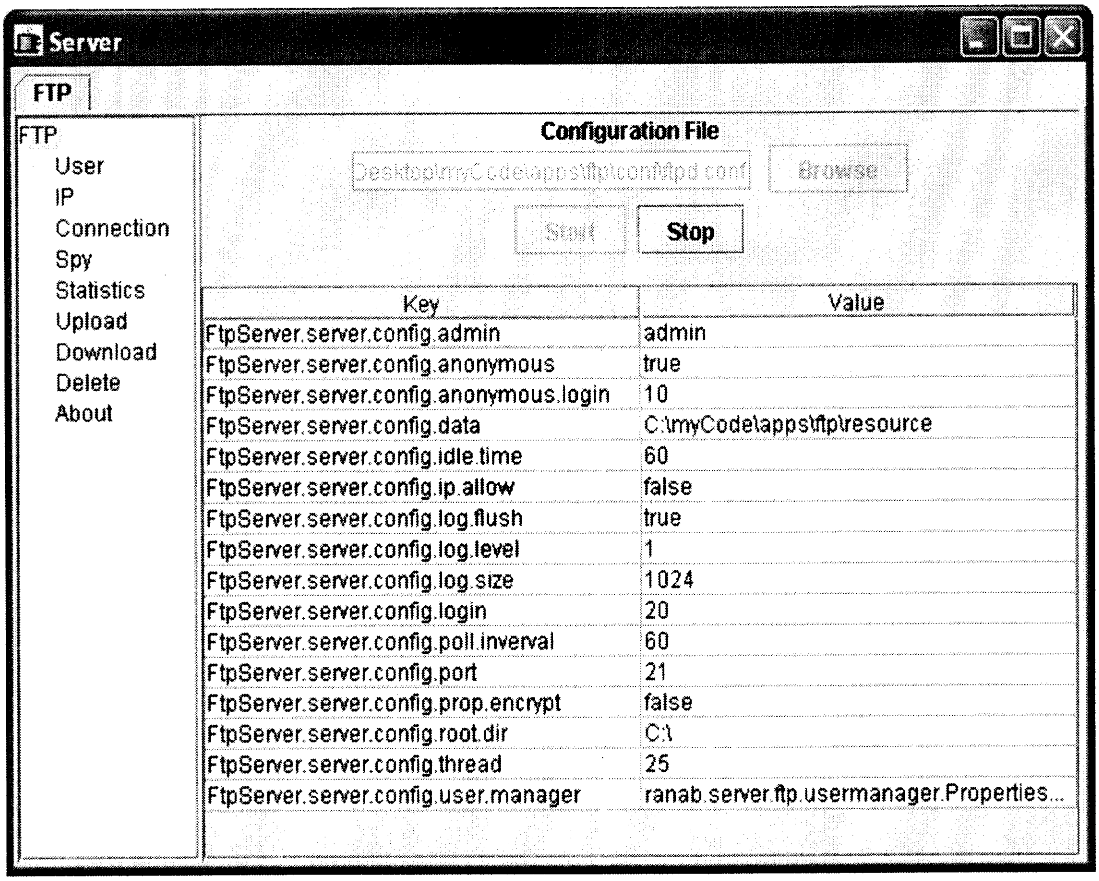
Look In: conf

File Name: ftpd.conf

Files of Type: All Files

Figure 5.3 Getting FTP Configuration File

After locating the file click the “Start” button in the main Java FTP Server Window, which will start our Java FTP Server for other users. At this point any one who knows our IP address can access our FTP Server anywhere in the world by just typing IP address for e.g. ftp://196.168.1.100.



Configuration File

Desktop\myCode\apps\ftp\conf\ftpd.conf

Key	Value
FtpServer.server.config.admin	admin
FtpServer.server.config.anonymous	true
FtpServer.server.config.anonymous.login	10
FtpServer.server.config.data	C:\myCode\apps\ftp\resource
FtpServer.server.config.idle.time	60
FtpServer.server.config.ip.allow	false
FtpServer.server.config.log.flush	true
FtpServer.server.config.log.level	1
FtpServer.server.config.log.size	1024
FtpServer.server.config.login	20
FtpServer.server.config.poll.inverval	60
FtpServer.server.config.port	21
FtpServer.server.config.prop.encrypt	false
FtpServer.server.config.root.dir	C:\
FtpServer.server.config.thread	25
FtpServer.server.config.user.manager	ranab.server.ftp.usermanager.Properties...

Figure 5.4 FTP Configuration Display

Important points to be noted regarding configuration file are as follows:

- The default FTP Server host is the "localhost"
- The default FTP Server port number is "21"

Now let us see in sequence the variables I used in the Configuration file to start the Java FTP Server.

- FTP Server administrator name is admin
- By default the server allows "anonymous" login to the Java FTP Server. We can change the value by just changing the "true" value to "false" and

start the FTP Server. This time our FTP Server will not allow "anonymous" logins.

- Maximum anonymous simultaneous logins. Should be less than or equal to the number of server connections. It will not be used if the sever does not allow anonymous login. The default value is 10.
- FTP resource directory. IP restrictor file, log files etc. will be stored in this directory or subdirectories.

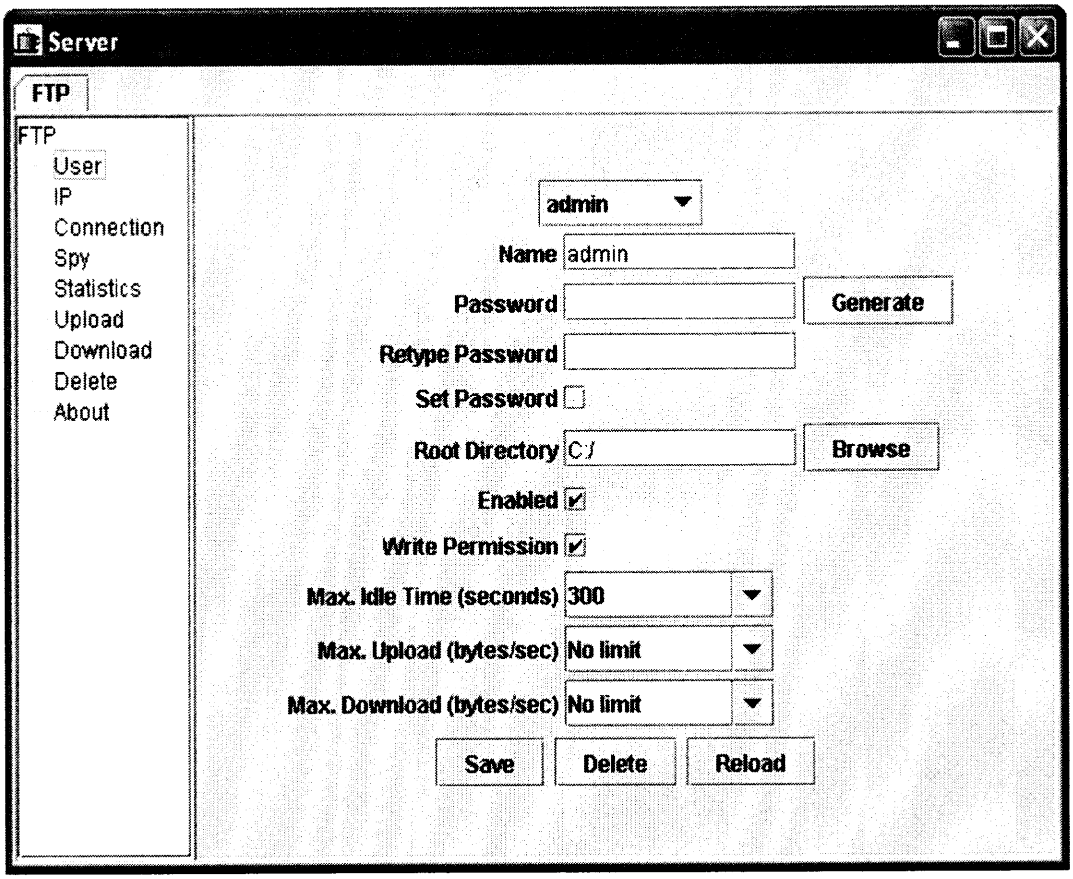
  FtpServer.server.config.data = C:\myCode\apps\ftp\resource

  We can change the directory to desired location. Just edit the configuration file and Restart the FTP Server. The changes will be effective.
- Pooling interval in seconds to disconnect idle users. The default value is 60 seconds.
- Allow/deny IPs. If it is true only allow listed IPs. Else ban the listed IPs.
- Flag to indicate to flush log every time after writing. The default value is false.
- Log file maximum size (0 means no limit) in Kbytes. The default maximum log file size is 1024KB.
- Maximum simultaneous logins. Should be less than or equal to the thread pool count. The default value is 20.
- Pooling interval in seconds to disconnect idle users. The default value is 60 seconds.

- Default root directory. When you start the ftp server for the first time, two users (admin, anonymous) will be created automatically. This directory will be their root directory.
- Threadpool count. This is the maximum number of simultaneous connections. The default value is 25.

5.2.1 User Screen

On the left hand side when we navigate to "User" the following screen will appear.



admin

Name admin

Password

Retype Password

Set Password ☐

Root Directory C:/

Enabled ☑

Write Permission ☑

Max. Idle Time (seconds) 300

Max. Upload (bytes/sec) No limit

Max. Download (bytes/sec) No limit

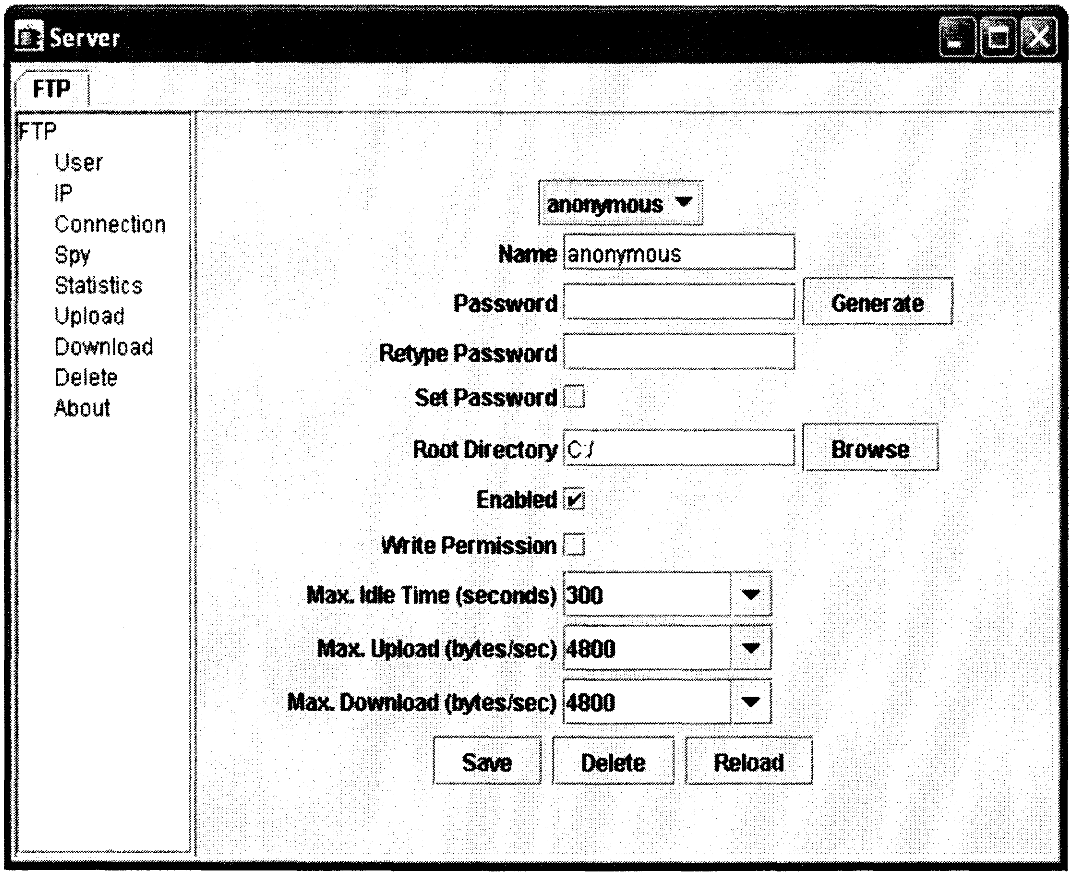
Figure 5.5 FTP User Screen

As said earlier two users "admin" and "anonymous" are created by default. We can see the properties for "admin" login as follows. As seen above we didn't assign any password. If we need any password, just type the password twice and check the box "Set Password" and press "Save". This will create user, "admin" with password.

- Root Directory is the directory, which can be viewed by the user. Here the user "admin" can access anything in the directory C:\ and anything above the directory C:\. It is better to store all the important files below the root directory so that they are not accessible to the user. We can change the "Root Directory" by just pressing the "Browse" button and select the directory and just click ok.
- Enabled box suggests whether the particular user is enabled or not. If "Enabled" is not checked the user with that login cannot access the FTP Server.
- Write Permission is very important when considering security. If the user has Write Permission he can write anything on to the local hard drives. So be sure when using this option.
- Max. Idle Time (seconds) specifies, how much time a user can stay idle and can maintain connection with the server. This is expressed in seconds.
- Max Upload (bytes/sec) specifies the transfer rate of the particular user. Pull down menu specifies the selections we can choose.
- Max Download (bytes/sec) is the speed at which a user can download from our FTP server.

- The "Save", "Delete", and "Reload" buttons saves the modified information, deletes the corresponding user and Reloads the user profile respectively.

A similar profile for an "anonymous" user can be shown as below with some modifications.



anonymous ▾

Name anonymous

Password

Retype Password

Set Password □

Root Directory C:/

Enabled ☑

Write Permission □

Max. Idle Time (seconds) 300 ▾

Max. Upload (bytes/sec) 4800 ▾

Max. Download (bytes/sec) 4800 ▾

Figure 5.6 Anonymous User Screen

5.2.2 IP Screen (Firewall)

When we navigate down the tree left side of the window we can see “IP”. When you click on “IP” you will see a screen similar to one shown below

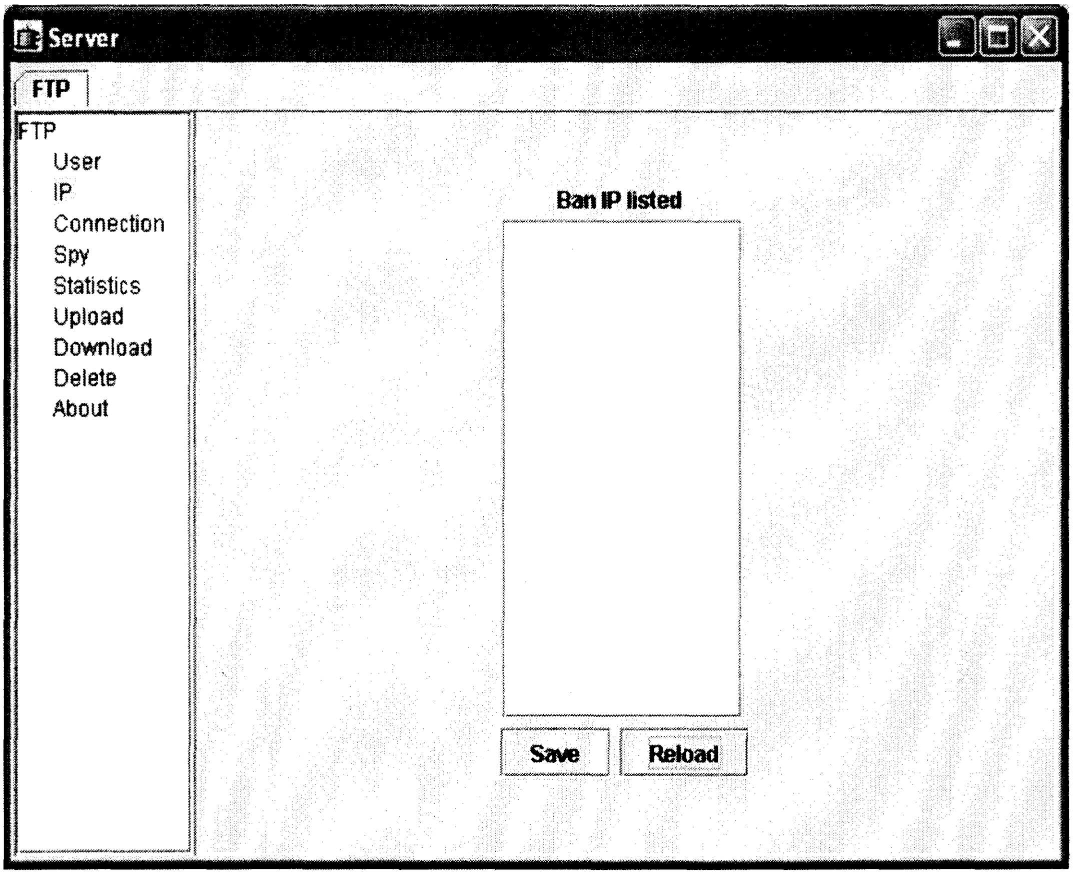
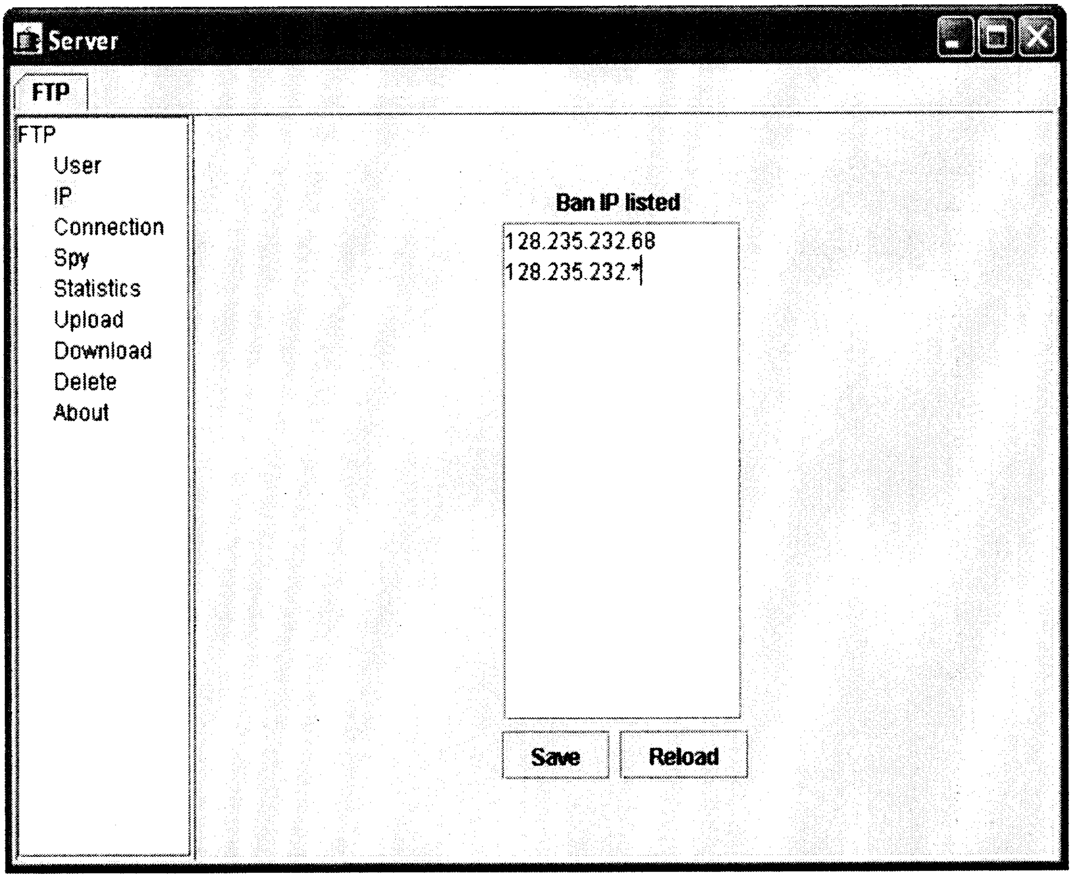


Figure 5.7 FTP Ban IP Screen

This is very important when we consider our FTP Server. Initially you will see, under “Ban IP listed” you will not see any IP addresses. Suppose if we want to restrict some users from accessing our FTP Server, we should have some means to stop users accessing the FTP Server without stopping the entire server. From this screen you can specify IP addresses, so that the respective users cannot access the FTP Server.

We can specify two types of notation here to restrict the users from accessing our FTP Server. It can be shown as



128.235.232.68
128.235.232.*

Figure 5.8 FTP List of Banned IP

As shown above the IP address 128.235.232.68 cannot access the FTP Server. The second notation 128.235.232.* specifies that any IP address with prefix 128.235.232. cannot access our FTP Server and making our FTP Server secure from destructive users.

Just type the IP address in the text area and press "Save" button. As soon as you press "Save" the IP address is blocked and cannot access our FTP Server. The "Reload" button is used to reload the FTP Server after altering the IP list. Only if we press, "Reload" the changes will be effective.

5.2.3 Connection Screen

The next option we can see is the "Connection". The "Connection" screen shows the present connections to our FTP Server. It is show below

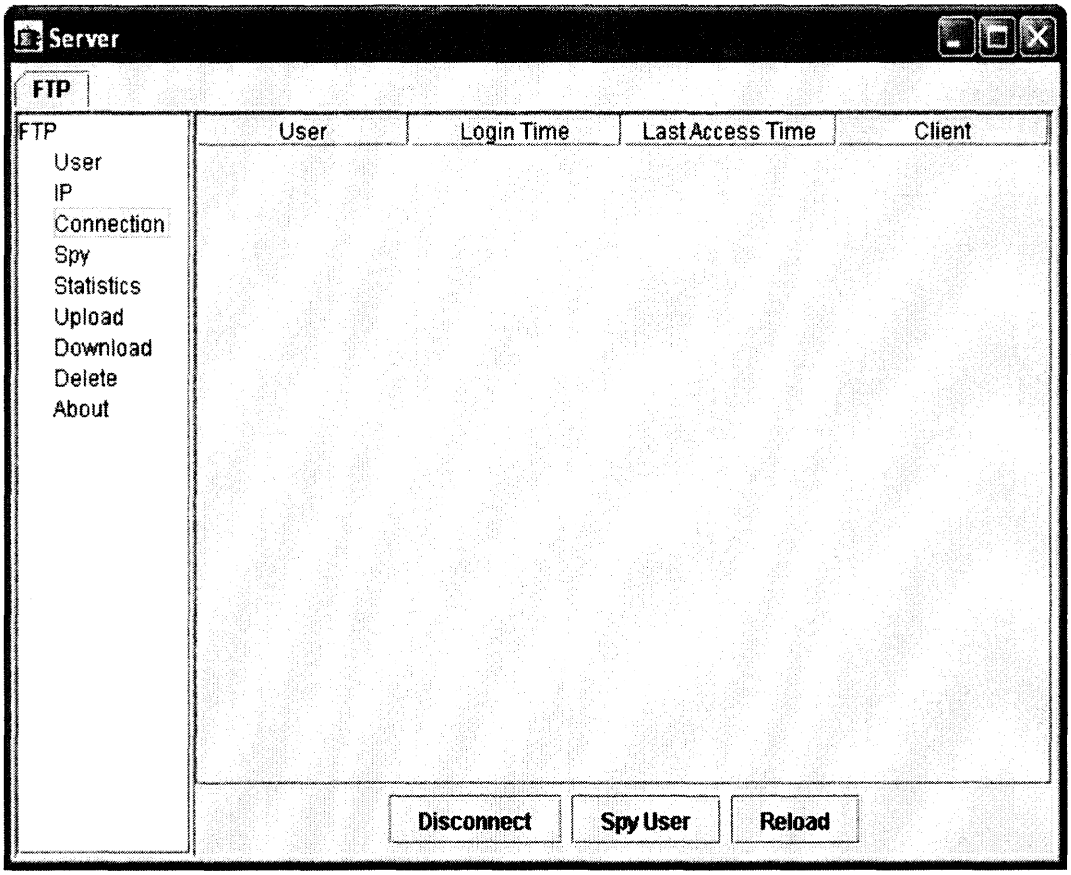
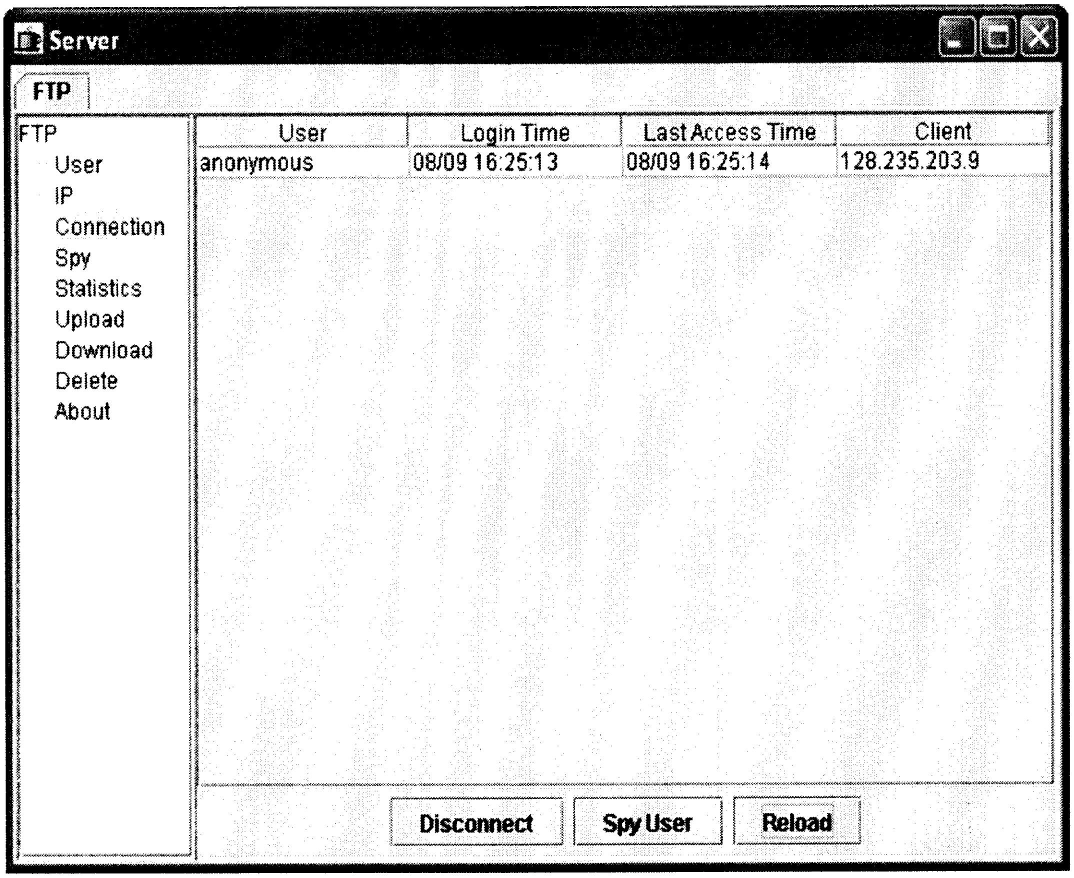


Figure 5.9 FTP Connection Screen

As shown above, right now no user is connected to our FTP Server.

Let us see the screenshot of users connected to our server.



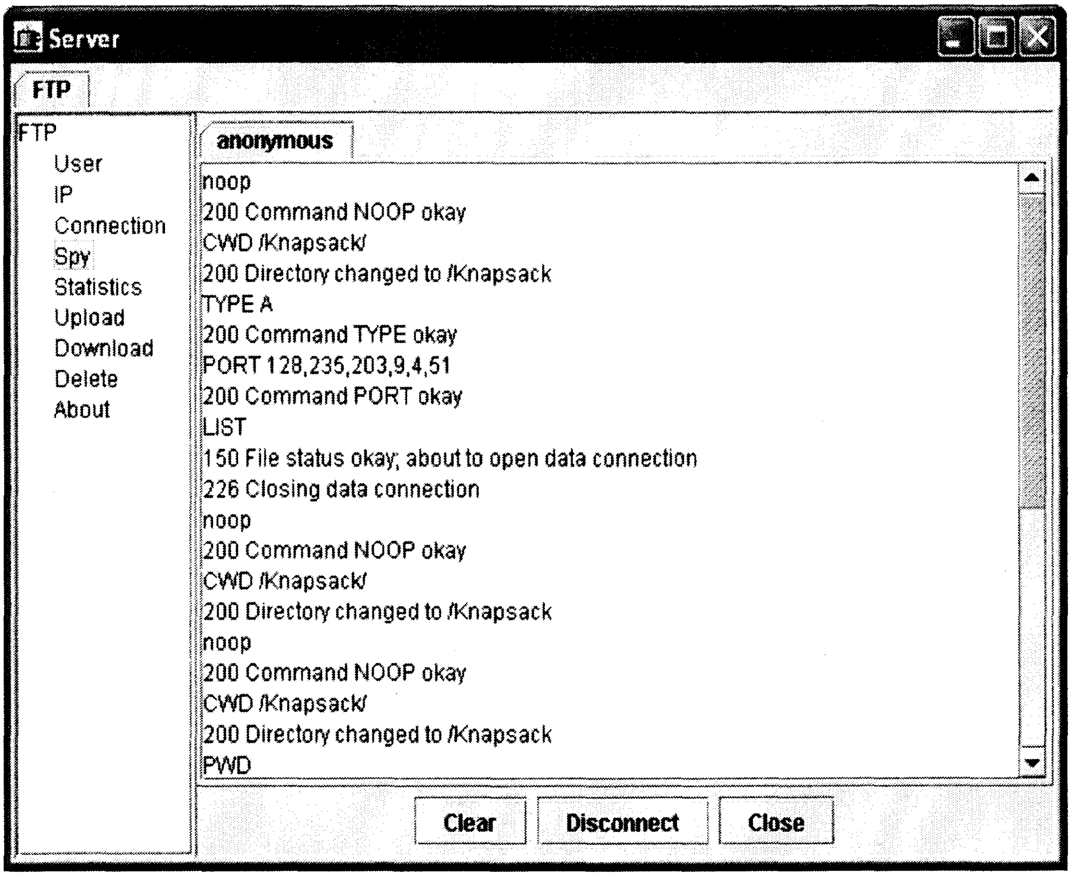
User	Login Time	Last Access Time	Client
anonymous	08/09 16:25:13	08/09 16:25:14	128.235.203.9

Figure 5.10 FTP Connected Users

Now you can see that there is one user connected to our FTP Server. And the User is logged in anonymously and his Login Time is 08/09 16:25:13 and Last Access Time is 08/09 16:25:14 and IP Address of the User is 128.235.203.9.

Here if we can select any user listed and if we press "Disconnect" the user will be disconnected from the FTP Server. He will be no longer can access the FTP Server. After we press "Disconnect" just press "Reload" button to make changes effective.

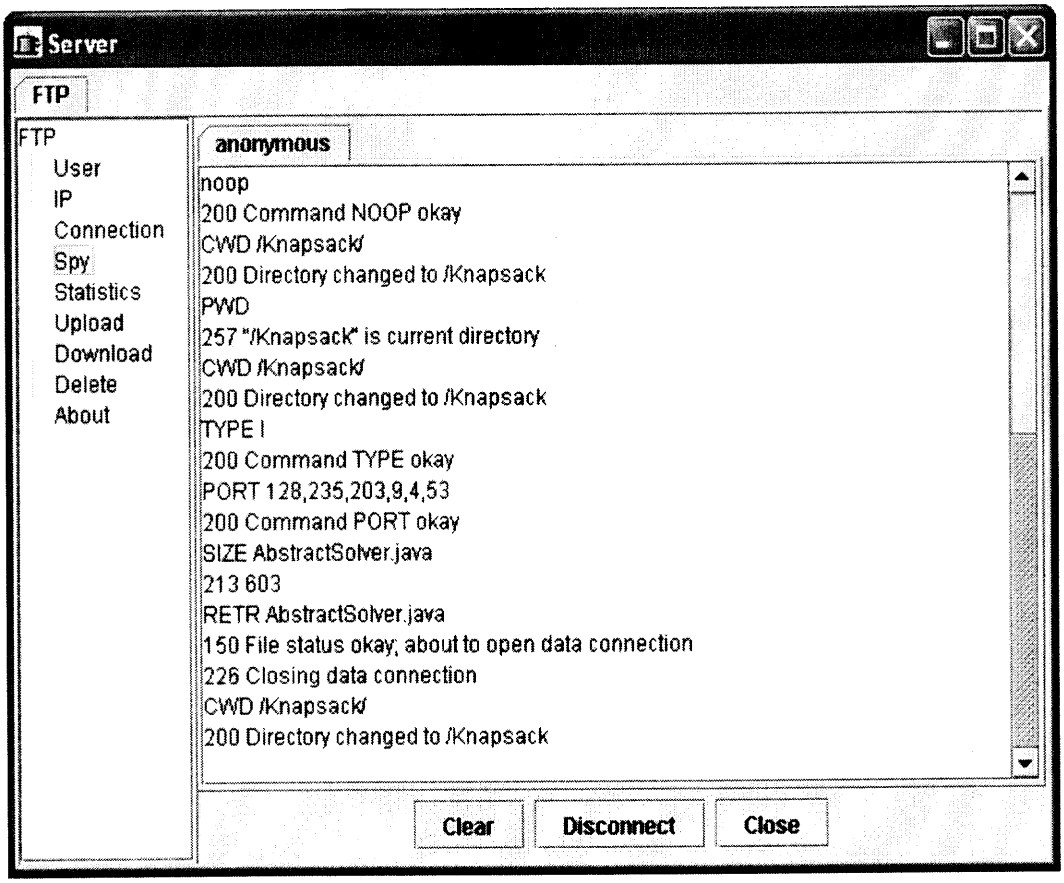
"Spy User" is used to track the user actions while he is connected to our FTP Server. Select a connection and then press "Spy User" button. The output can be viewed by pressing "Spy" on the left hand side of the window.



```
noop
200 Command NOOP okay
CWD /Knapsack/
200 Directory changed to /Knapsack
TYPE A
200 Command TYPE okay
PORT 128,235,203,9,4,51
200 Command PORT okay
LIST
150 File status okay; about to open data connection
226 Closing data connection
noop
200 Command NOOP okay
CWD /Knapsack/
200 Directory changed to /Knapsack
noop
200 Command NOOP okay
CWD /Knapsack/
200 Directory changed to /Knapsack
PWD
```

Figure 5.11 FTP Spy User Screen

Here it shows that the user navigated through various folders and transferred a file and it was successful. "NOOP" means no operation.



```
noop
200 Command NOOP okay
CWD /Knapsack/
200 Directory changed to /Knapsack
PWD
257 "/Knapsack" is current directory
CWD /Knapsack/
200 Directory changed to /Knapsack
TYPE I
200 Command TYPE okay
PORT 128,235,203,9,4,53
200 Command PORT okay
SIZE AbstractSolver.java
213 603
RETR AbstractSolver.java
150 File status okay; about to open data connection
226 Closing data connection
CWD /Knapsack/
200 Directory changed to /Knapsack
```

Figure 5.12 FTP Spy User Screen

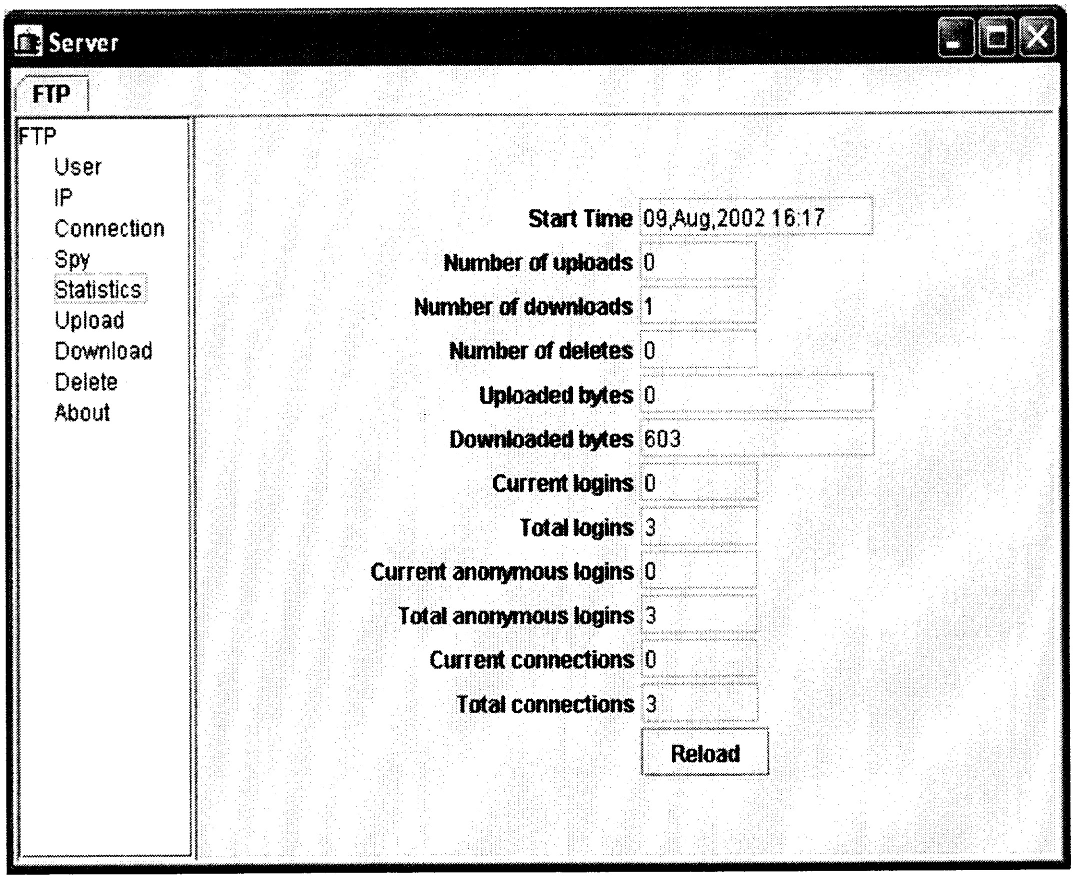
5.2.4 Statistics

“Statistics” screen shows the complete details of the server information.

It shows

- When the server was started
- Number of Uploads took place
- Number of Downloads from the server
- Number of deletes done on the server
- Total amount of uploaded bytes
- Total amount of downloaded bytes
- Current number of Logins

- Total number of logins up to the point
- Current anonymous user logins at this point
- Total anonymous logins that took place in this session
- Current connections
- Total connections which include anonymous and admin logins



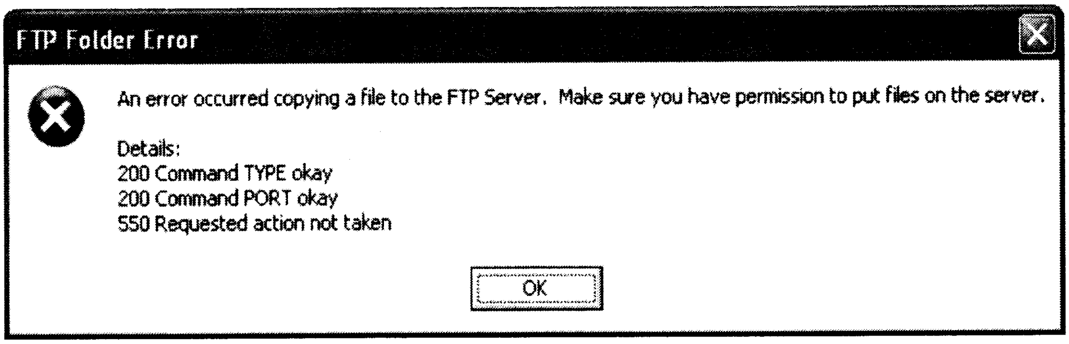
Start Time 09,Aug,2002 16:17
Number of uploads 0
Number of downloads 1
Number of deletes 0
Uploaded bytes 0
Downloaded bytes 603
Current logins 0
Total logins 3
Current anonymous logins 0
Total anonymous logins 3
Current connections 0
Total connections 3

Figure 5.13 FTP Statistics Screen

By pressing “Reload” button we can update the data in the boxes.

5.2.5 Upload Screen

The upload screen illustrates how many uploads are done to the FTP Server. We have to make sure that user has the access to write to the server. Otherwise error message like this will pop up saying user don't have rights to do the operation.

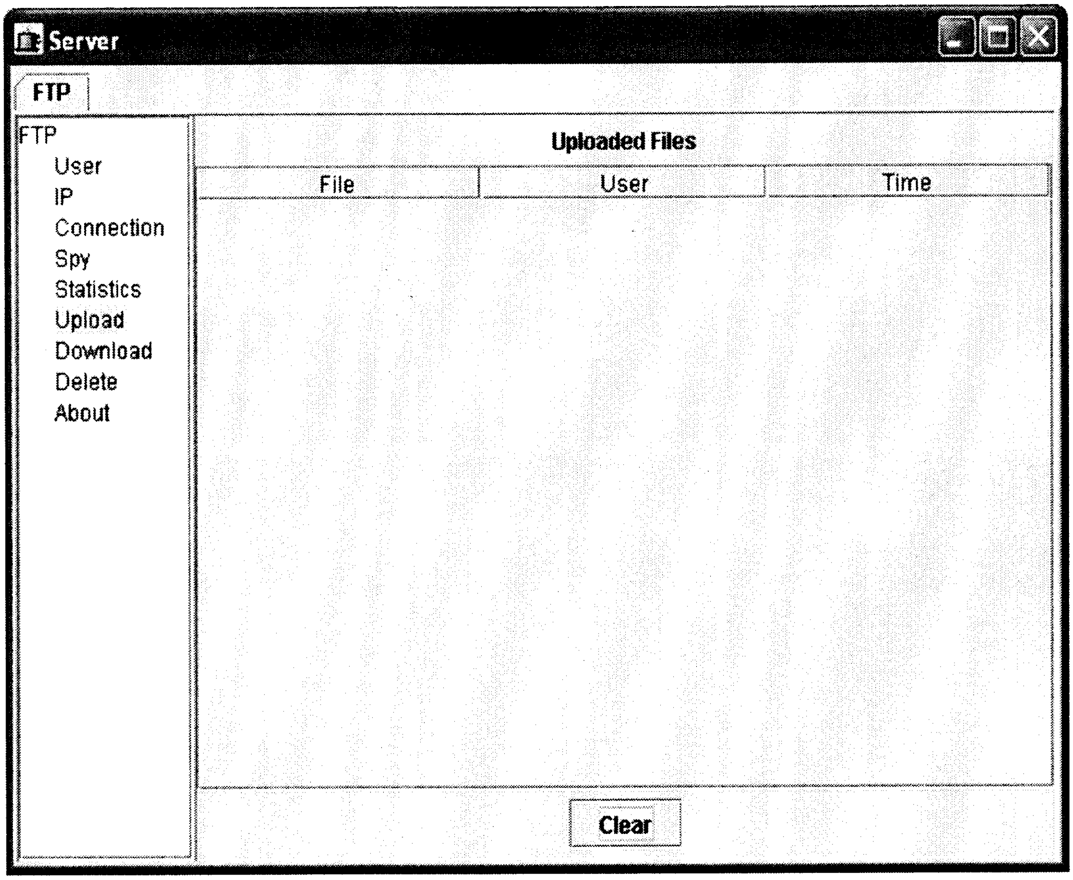


An error occurred copying a file to the FTP Server. Make sure you have permission to put files on the server.

Details:
200 Command TYPE okay
200 Command PORT okay
550 Requested action not taken

Figure 5.14 FTP Upload Access Denied Screen

In the above case the user was anonymous and we didn't assign him the write permission. So he can't copy any data or files on to the server. He can only download from the server.

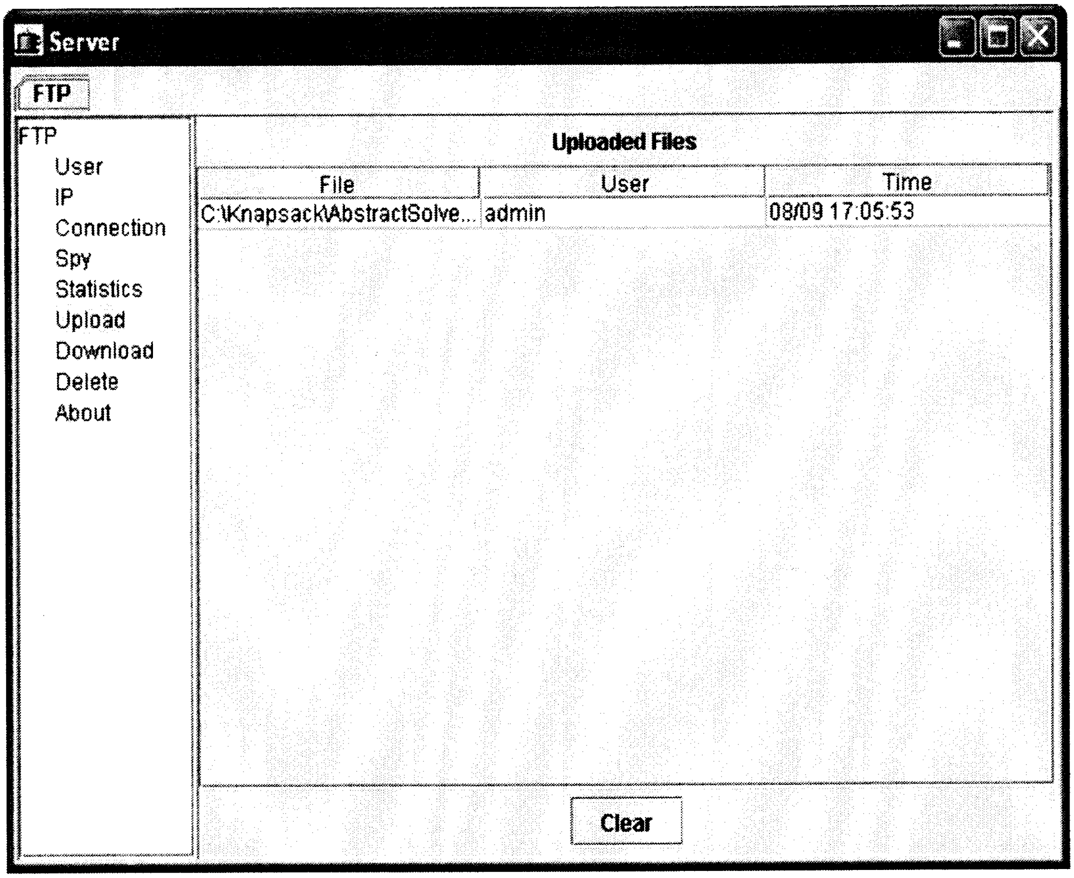


File	User	Time

Figure 5.15 FTP Upload Screen

The above screen shows that there is no file uploads done to the server.

Here we can see that there was one upload to our FTP Server from the user "admin" and the "time" of upload was 08/09 17:05:53.



Uploaded Files

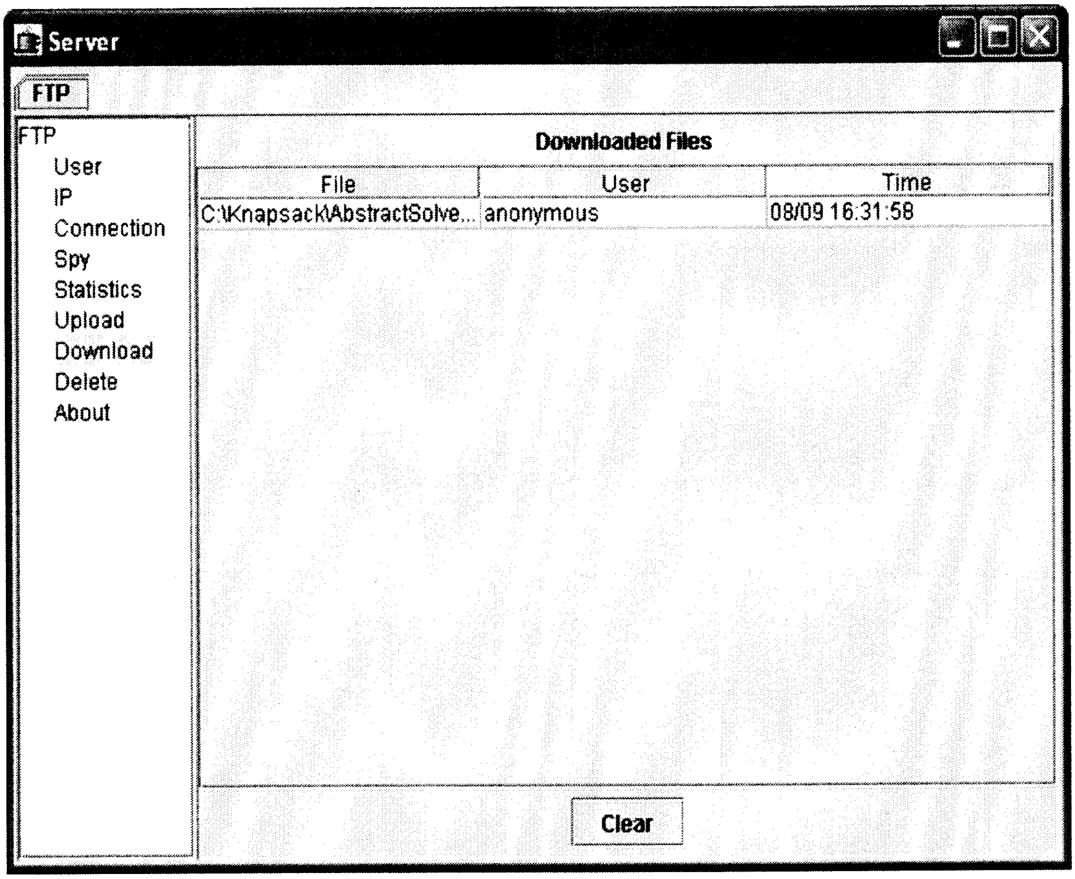
File	User	Time
C:\Knapsack\AbstractSolve...	admin	08/09 17:05:53

Figure 5.16 FTP Uploaded Files Screen

If we press "Clear" the display table will be cleared.

5.2.6 Download Screen

Similar to the Upload screen the download screen shows how many files have been downloaded from the FTP Server and who are the users.



Downloaded Files

File	User	Time
C:\Knapsack\AbstractSolve...	anonymous	08/09 16:31:58

Figure 5.17 Download Screen

Here we can see that there was one download from the FTP Server by the "admin" user at 08/09 16:31:58

5.2.7 Delete Screen

Similar to the Upload and Download screens we can see how many files have been deleted from our FTP Server and who are the users. It keeps track of every action user performs as long as he is connected to the FTP Server.

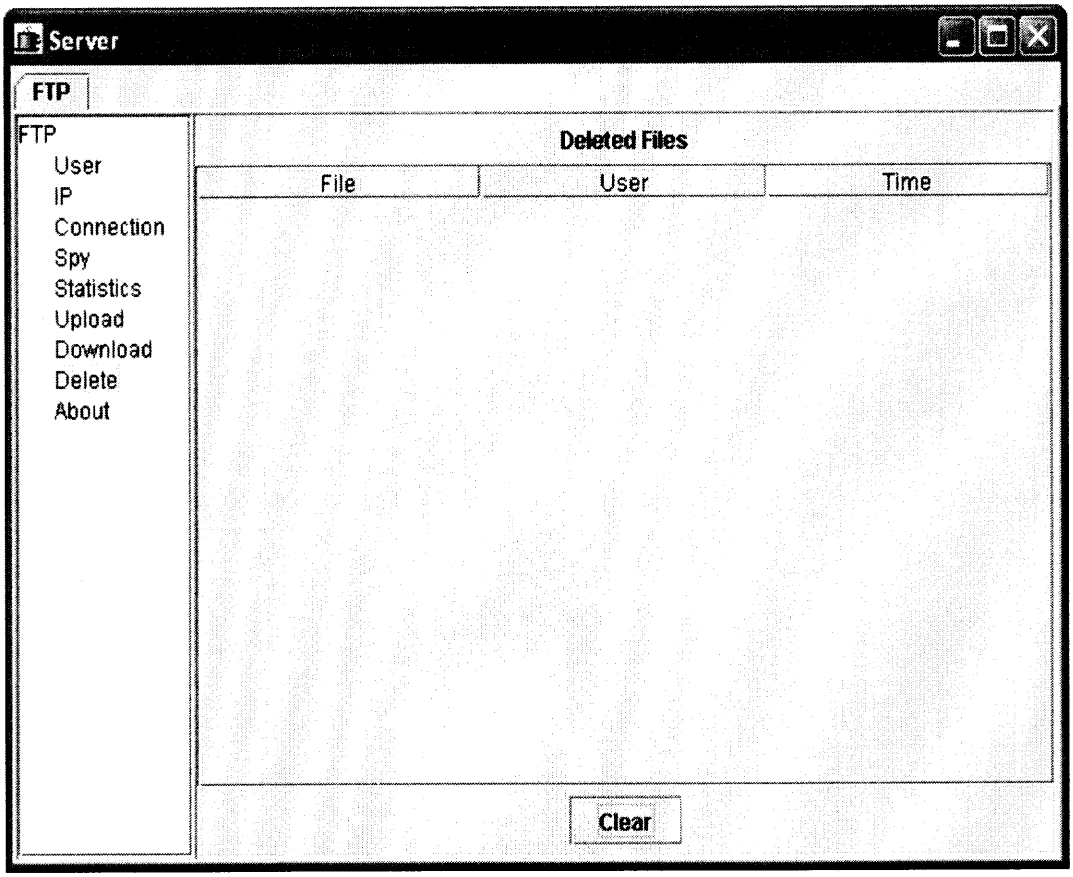
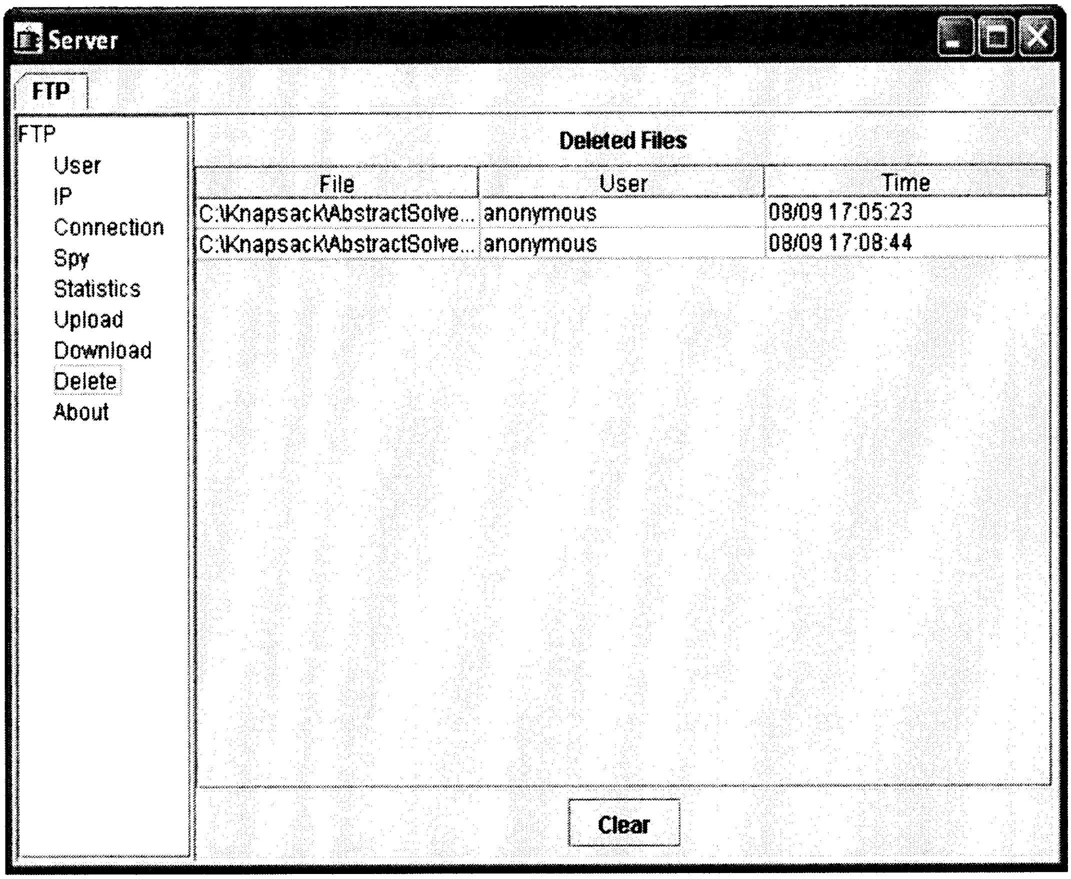


Figure 5.18 FTP Delete Screen

As of now there are no Deletes made by any of the Users on the FTP Server.

The below screenshot shows that there are 2 deletes from the FTP Server.



Deleted Files

File	User	Time
C:\Knapsack\AbstractSolve...	anonymous	08/09 17:05:23
C:\Knapsack\AbstractSolve...	anonymous	08/09 17:08:44

Figure 5.19 FTP Deleted Files Screen

This is how we configure the FTP server using the interface developed.

CHAPTER 6

SMTP SERVER

SMTP is an acronym for Simple Mail Transfer Protocol.

6.1 What is SMTP?

SMTP, a process can transfer mail to another process on the same network or to some other network via a relay or gateway process accessible to both networks. Within the Internet, email is delivered by having the source machine establish a TCP connection to port 25 of the destination machine. Listening to this port is an email daemon that "speaks" SMTP. This daemon accepts incoming connections and copies messages from them into the appropriate mailboxes. If a message cannot be delivered, an error report containing the first part of the undeliverable message is returned to the sender.

SMTP is a simple ASCII protocol. After establishing the TCP connection to port 25, the sending machine, operating as the client, waits for the receiving machine, operating as the server, to talk first. The server starts by sending a line of text giving its identity and telling whether or not it is prepared to receive mail. If it is not, the client releases the connection and tries again later.

If the server is willing to accept email, the client announces whom the email is coming from and whom it is going to. If such a recipient exists at the destination, the server gives the client the go-ahead message. Then the client sends the message and the server acknowledges it. No checksums are generally needed because TCP provides a reliable byte stream. If there is more email, that is now sent. When all the email has been exchanged in both directions, the connection is released.

6.2 The SMTP Model

The exchange of mail using TCP/IP is performed by a *message transfer agent* (MTA). Users normally don't deal with the MTA. The system administrator is responsible to set up the local MTA. The SMTP protocol describes how two MTAs communicate with each other using a single TCP connection.

SMTP uses the concept of spooling. The idea of spooling is to allow mail to be sent from a local application to the SMTP application, which stores the mail in some device or memory. Once the mail has arrived at the spool, it has been queued. A server checks to see if any messages are available and then attempts to deliver them. If the user is not available for delivery, the server may try later. Eventually, if the mail cannot be delivered, it will be discarded or perhaps returned to the sender. This is known as an *end-to-end delivery system*, because the server is attempting to contact the destination to deliver, and it will keep the mail in the spool for a period of time until it has been delivered.

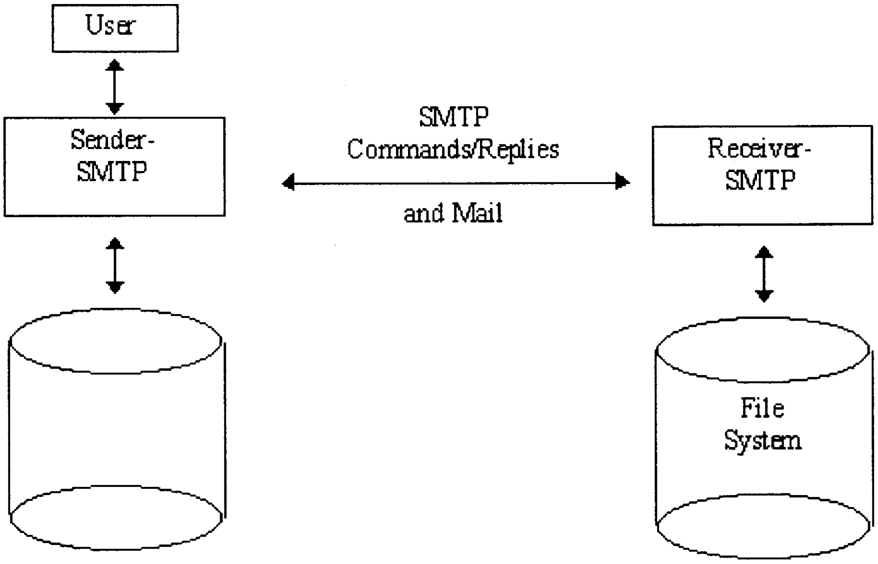


Figure 6.1 the SMTP Model

6.3 Implemented Features

Features implemented in our mail server are

- SMTP Server that collects Email from Internet and Intranet.
- POP3 that can be used by Intranet Users to retrieve their Email.
- Sender Utility that forwards collected Email to your ISPs Mail MTA, if it is addressed to non-local users. (Can run as background daemon).
- Fetcher Utility that retrieves Email from different POP3 Servers in the Internet. (Can run as background daemon).
- Alias List configurable through configuration file.
- Auto reply feature.

6.4 Sizes and Limitations

There are several objects that have required minimum maximum sizes. That is, every implementation must be able to receive objects of at least these sizes, but must not send objects larger than these sizes.

User

The maximum total length of a user name is 64 characters.

Domain

The maximum total length of a domain name or number is 64 characters.

Path

The maximum total length of a reverse-path or forward-path is 256 characters (including punctuation and element separators).

Command Line

The maximum total length of a command line including the command word

Reply Line

The maximum total length of a reply line including the reply code

Text line

The maximum total length of a text line is 1000 characters.

Recipients Buffer

The maximum total number of recipients that must be buffered is 100 recipients.

CHAPTER 7

SMTP SERVER INTERFACE

7.1 Installing the Server

Copy the project files from floppy or zip disk. You will just need the class files as I have already compiled the java files. Also I have written necessary batch files, which will run the corresponding java files. The following are the batch files

- Jmailserver.bat The main Pop3 and SMTP Server
- Sender. bat Sends the queue directory to the Internet
- Fetcher. bat Fetch E-Mail accounts from the Internet
- AddUser.bat Command line tool to add users accounts.
- Config.bat GUI based tool to configure the server.

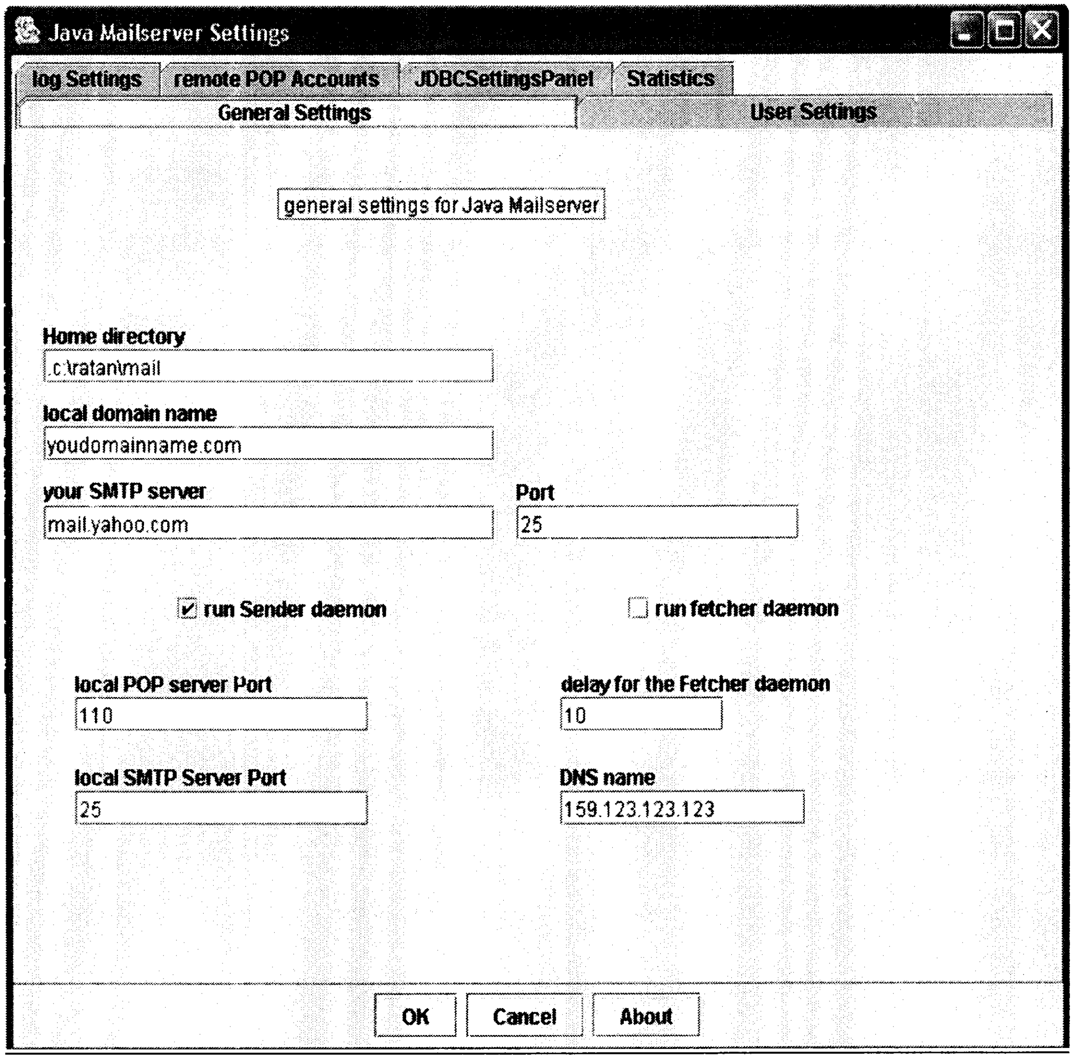
I wrote a configuration file (Jmailsrv.cfg) in which I have default preferences for the SMTP server.

7.2 Snapshots and Description

We start running the program by opening the "config.bat". As soon as we open the "config.bat" automatically a command prompt will open and the User interface of the Java MailServer will start. Every time I make any changes to the mail server the configuration file "Jmailsrv.cfg" will be updated.

7.2.1 General Settings

The general settings window is as shown in the following figure



Home directory
c:\ratan\mail

local domain name
youdomainname.com

your SMTP server
mail.yahoo.com

Port
25

☑ run Sender daemon

□ run fetcher daemon

local POP server Port
110

delay for the Fetcher daemon
10

local SMTP Server Port
25

DNS name
159.123.123.123

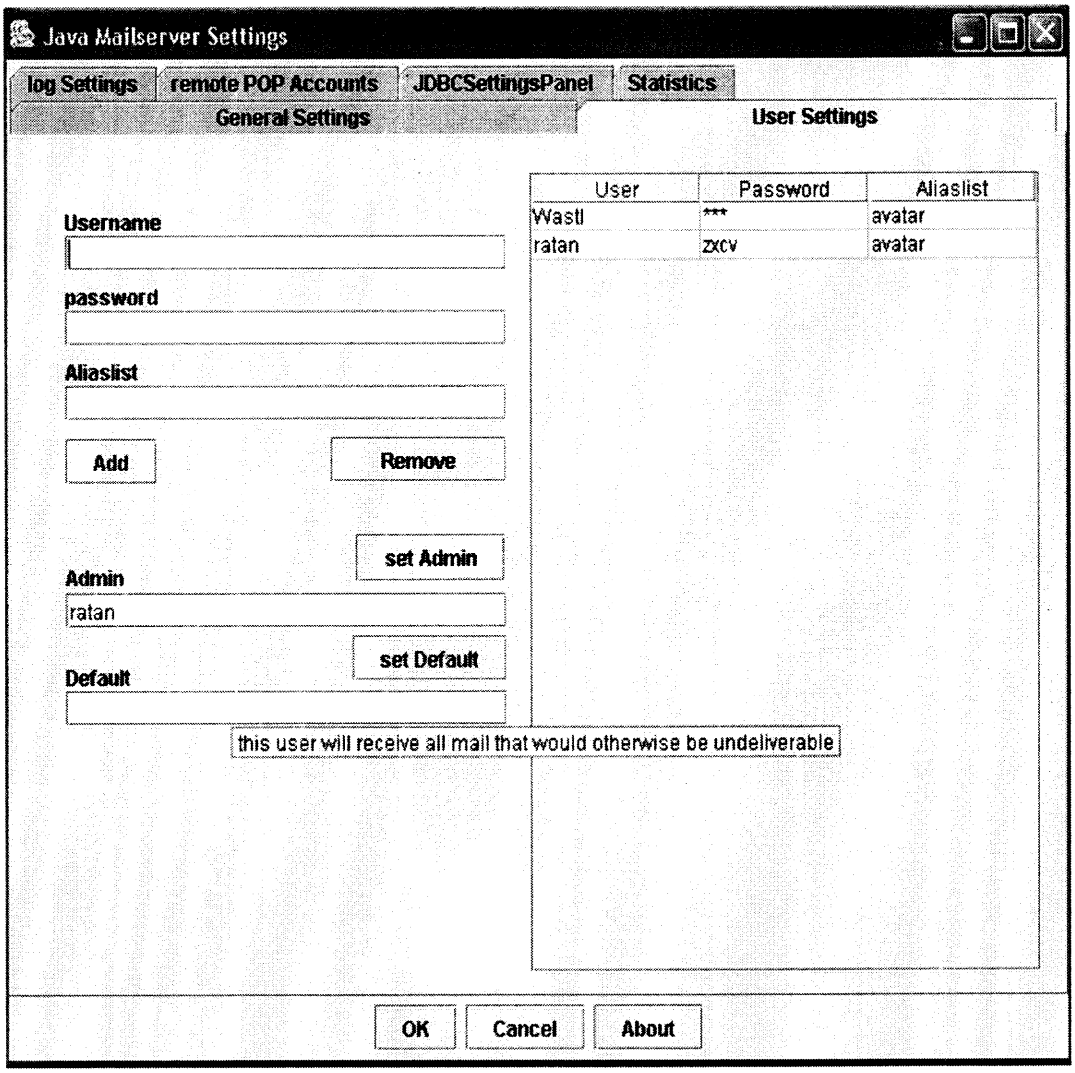
Figure 7.1 SMTP General Settings Screen

- Home Directory –Here we can give path to the directory where the user mail directories and the status files should be created.
- Local Domain Name- It is the domain name of the local domain. Local domain will be used to differentiate between local and remote mails.

- Your SMTP Server- It is your ISP's SMTP server where all outgoing mails will be forwarded to, if no DNS Server is specified.
- Local SMTP Server Port -This is the port for the local SMTP server, for incoming requests. The port number for SMTP server is 25.
- Local POP Server Port – This is the port for the local POP server, for retrieving mails from other server. The default port number is 110.
- Fetcher Daemon- It is used to fetch mail from remote POP accounts automatically. The usernames used for this accounts must be valid mail server users.
- Run Fetcher Daemon- It is used to enable/disable fetcher daemon, which is to collect remote mail locally. We should use this setting only if the mail server has permanent access to the Internet.
- Run Sender Daemon- It is used to enable/disable sender daemon, which is to send outgoing mail immediately. We should use this setting only if the mail server has permanent access to the Internet.
- Delay for the Fetcher Daemon- This is the delay between the accesses to the remote mail box by the fetcher daemon. Default value of delay is 10 minutes. (Here time is measured in minutes).
- DNS Name- It is the IP address of the DNS Server. It should be used for name resolution. The default value of DNS name is 159.123.123.123

7.2.2 User Settings

The user settings window is shown in the following figure



User	Password	Aliaslist
Wastl	***	avatar
ratan	zxcv	avatar

Admin: ratan

Default

this user will receive all mail that would otherwise be undeliverable

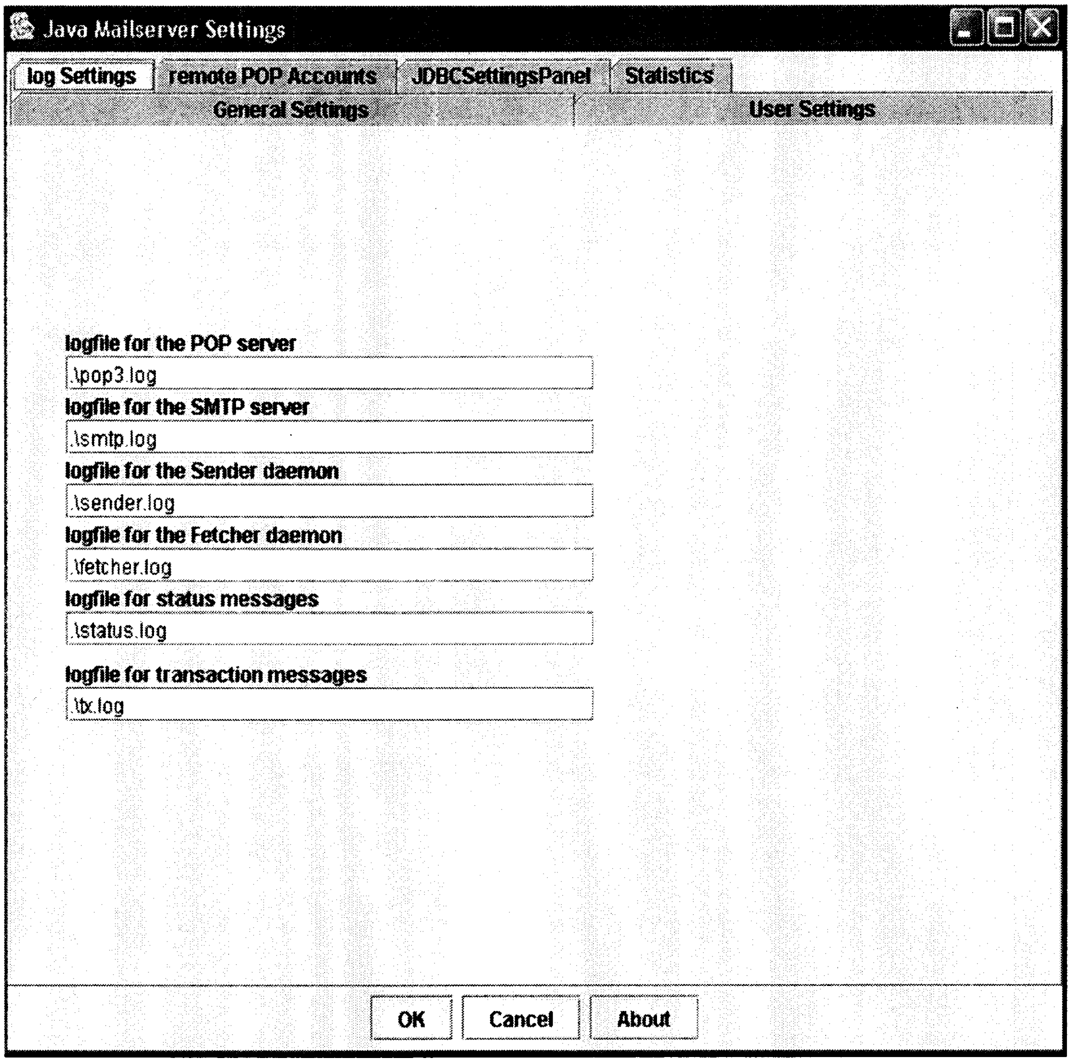
Figure 7.2 SMTP User Settings

Here we can add users to the mail servers. I have added two users namely “wastl” and “ratan”. These two users are added to the same alias list named “avatar”. User “ratan” is set as “Administrator”. Administrator is like a post mater to whom all alerts and bug reports are sent.

- Once when we enter the username, password and Aliaslist when Add button is clicked the user will be added to the list. Usernames are stored in lower case letters so it should be specified in lower case. We can specify several aliases for an existing user account.
- To remove a user, select the user from the list and click Remove button. Only administrator can remove a user.
- Set Admin button is used to set administrator for the server. Select a user from the list and click on the “set Admin” button. Here user “ratan” is set as administrator.
- Admin text box shows the current administrator. To change the administrator, delete the text in Admin text box and set a new administrator.
- Set Default button is used to set a default user to whom all undeliverable local mails will be sent to.

7.2.3 Log Settings

The log settings window is shown in the following figure



logfile for the POP server
.\pop3.log
logfile for the SMTP server
.\smtp.log
logfile for the Sender daemon
.\sender.log
logfile for the Fetcher daemon
.\fetcher.log
logfile for status messages
.\status.log
logfile for transaction messages
.\tx.log

Figure 7.3 SMTP Log Settings

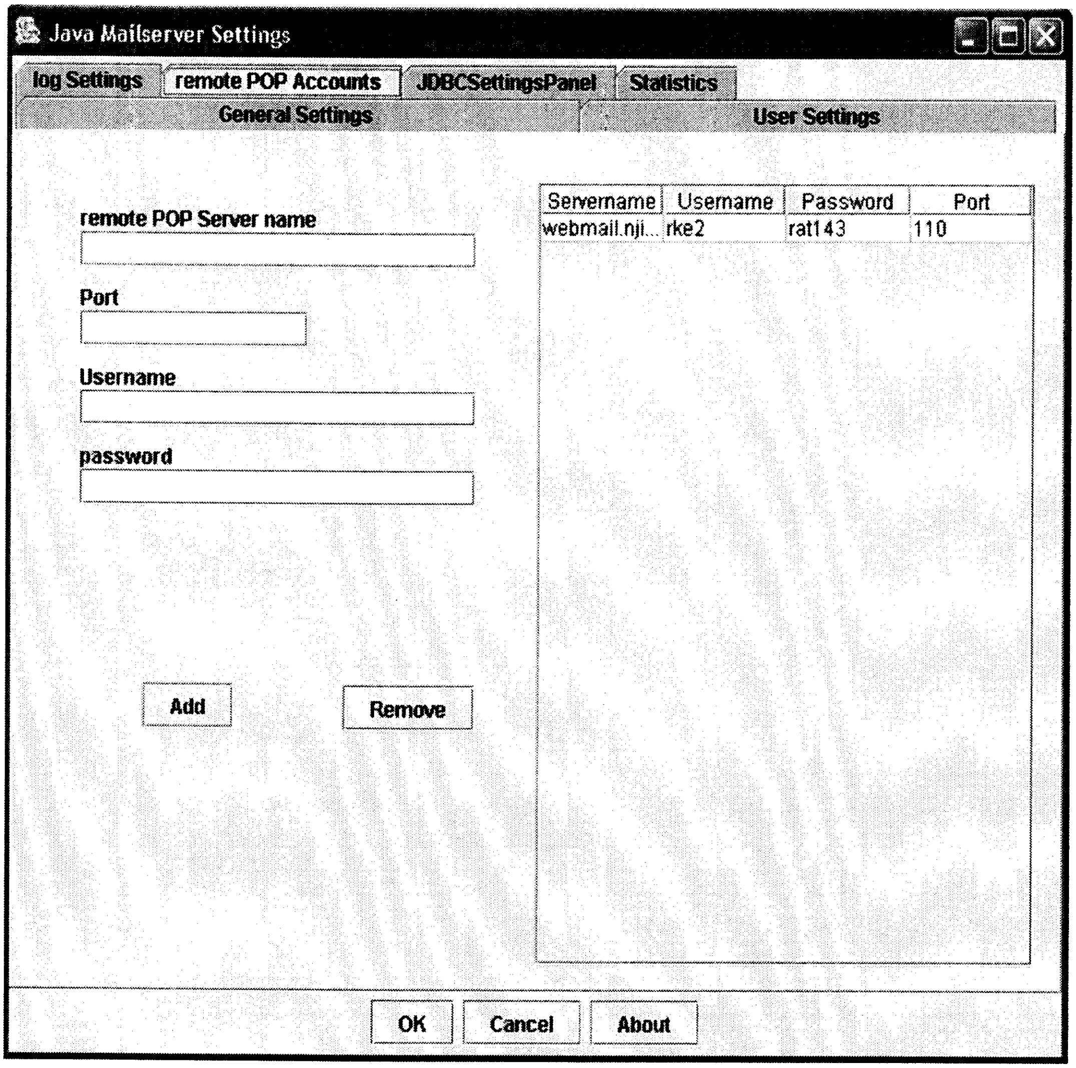
Here we can set the path to log files, which stores the logs of different scenarios.

- Pop3.log is the log file for POP3 traffic.
- Smtp.log is the log file for SMTP traffic

- Sender.log is the log file for outgoing mail traffic.
- Fetcher.log is the log file for mail fetched from remote mail accounts.
- Status.log is the log file for status reports
- Transaction.log is the log file for transaction logs.

7.2.4 Remote POP Accounts

The user interface for remote POP accounts is shown below



Servername	Username	Password	Port
webmail.nji...	rke2	rat143	110

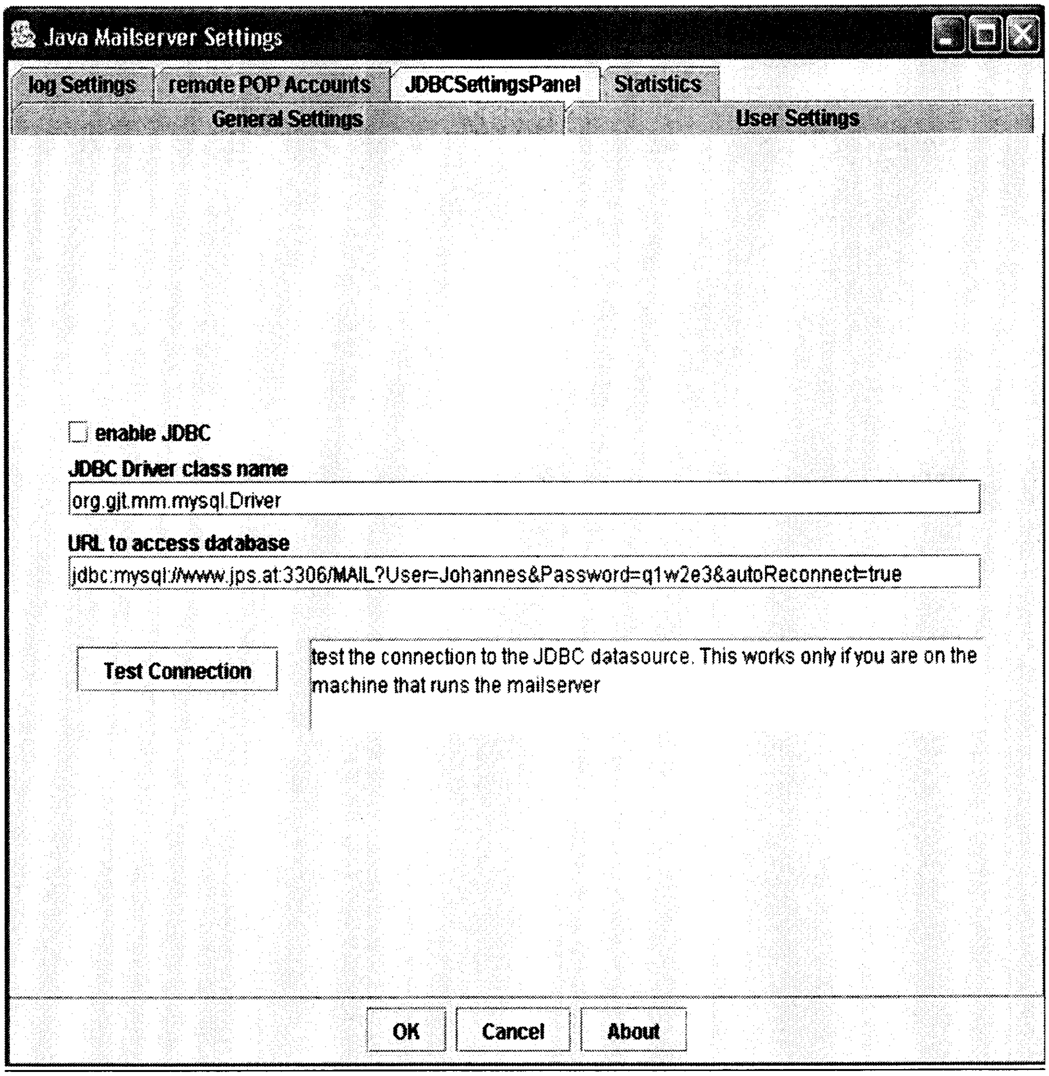
Figure 7.4 SMTP Remote Settings Screen

POP3 is an acronym for Post Office Protocol. It is used for fetching email from a remote mailbox. Using this user can log in, log out, fetch messages and delete

messages. The point of POP3 is to fetch email from remote mailbox and store it on the user's local machine to be read later.

- In the remote POP server name text box we must enter domain name of a valid mail server. For example webmail.njit.edu
- To connect to the remote mail server we must enter a existing user id and password.
- The default port to connect to remote mail server is 110.
- To add a POP account, we must enter all the above details and click on "add" button.
- To remove a POP account, select the account in the list and click on "remove" button.

7.2.5 JDBC Settings Panel



□ enable JDBC

JDBC Driver class name

org.gjt.mm.mysql.Driver

URL to access database

jdbc:mysql://www.jps.at:3306/MAIL?User=Johannes&Password=q1w2e3&autoReconnect=true

test the connection to the JDBC datasource. This works only if you are on the machine that runs the mailserver

Figure 7.5 SMTP JDBC Settings Panel

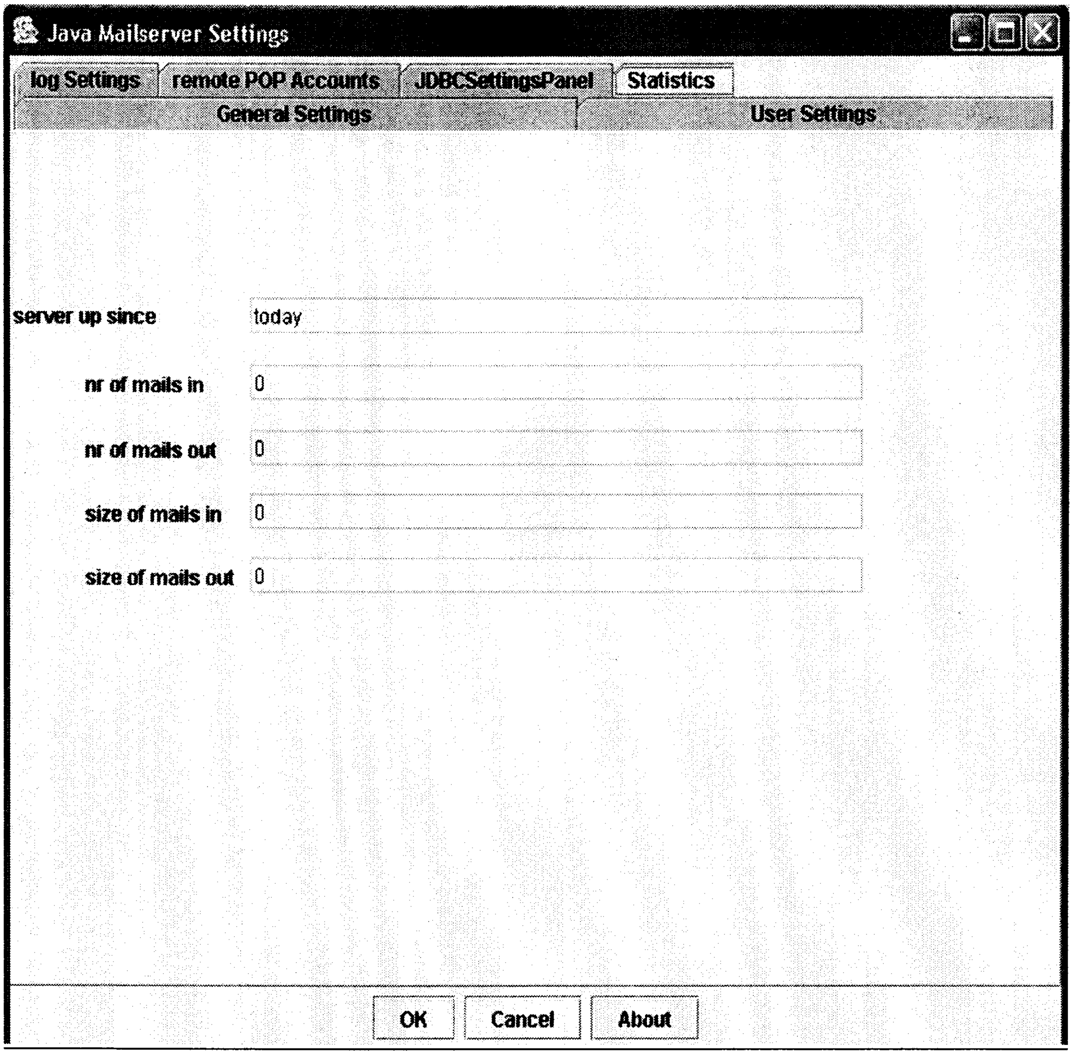
If we want to use a database as mail store instead of our local file system, we need to connect to a database.

Using JDBC (Java Database Connectivity) we can connect to a database like SQL server or Microsoft Access.

- Enable JDBC - It is used to enable/disable JDBC connection. If JDBC connection is enabled, mails will be stored in the database, which is specified in the URL to access the database.
- JDBC Driver Class Name- Here we should enter the full-qualified class name of the JDBC driver. The driver class has to be specified in the class path of the system's environmental variables.
- URL to access Database- Specify the URL of the database where the mails are to be stored
- Test Connection- After specifying the URL to access database and the appropriate driver to be used, we can test the database connection by clicking on "Test Connection" button. This works only if you are on the machine that runs the mail server.

7.2.6 Statistics

Statistics window is as shown below.



server up since: today

nr of mails in: 0

nr of mails out: 0

size of mails in: 0

size of mails out: 0

Figure 7.6 SMTP Statistics Screen

"Statistics" screen displays the complete details about the following

- When the server was started.
- Number of incoming mails.
- Number of outgoing mails.
- Size of incoming mails.
- Size of outgoing mails.

7.2.7 Saving the Changes

To save changes click on "OK" button. This will make changes to the "Jmailsrv.cfg" file. Content of "Jmailsrv.cfg" is as follows

```
User.admin=ratan

Fetcher.runDaemon=false

Home=.c\:\\ratan\\mail

SMTP.server=mail.yahoo.com

LocalDomain=youdomainname.com

Sender.wait_timeout=20

Fetch1=pop.ISP.com;username1;password1;110

JDBC.URL=jdbc\:mysql\://www.jps.at\:3306/MAIL?User\=Johannes&Password\
=q1w2e3&autoReconnect\=true

Fetch0=webmail.njit.edu; rke2; rat143; 110

Logfile.SMTP=.\\smtp.log

Logfile.Status=.\\status.log

RegisterRMIInterface=false

Fetcher.delay_time=10

User.unknown=ratan

SMTP.localPort=25

User.user=alias; alias2; alias3

SMTP.remotePort=25

SMTP.timeout=30000
```

```
Logfile.POP3=.\\pop3.log

User.user1=aliasforuser2; aliasforuser21; aliasforuser22

ShowStatusWindow=false

POP3.timeout=30000

Logfile.Fetcher=.\\fetcher.log

SMTP.filterClass=at.jps.mailserver.ProhibitRelayFilter

JDBC.enabled=false

JDBC.driver=org.gjt.mm.mysql.Driver

Sender.runDaemon=true

POP3.localPort=110

DNServer=159.123.123.123
```

From the configuration file it is evident that

1. SMTP server socket timeout is 3000 milliseconds.
2. POP3 server port timeout is 3000 milliseconds.

7.3 USING AUTO REPLY FEATURE

Java Mailserver can be setup to send a response to an incoming mail automatically. For suppose if you are going out for vacation, we can send a vacation response to all the incoming mails.

- To turn on this feature simply send an email with the contents you want the reply message to contain in the following format

  TO:autoresponder@yourdomain.com

  Subject: username ;password ;on

You will receive a confirmation email. As long as this feature is turned on every incoming email will be answered using the provided message.

- To turn off this feature simply send a message in the following format

  TO:autoresponder@yourdomain.com

  Subject: username ;password ;off

  You will receive a confirmation email and the stored auto reply message will be deleted.

CHAPTER 8

HTTP SERVER

HTTP is an acronym for HyperText Transfer Protocol

The HTTP protocol consists of two fairly distinct items: the set of requests from browsers to servers and the set of responses going back from the server.

8.1 Types of HTTP Requests

HTTP supports two kinds of requests:

1. Simple Request
2. Full Request.

Simple Request: A simple request is a single line request naming the page desired without the protocol version. The response is just the raw page with no headers and no encoding. This mechanism is needed for backward compatibility. Its use will decline as browsers and servers based on full requests become standard.

Full Request: Full Requests are indicated by the presence of the protocol version on the request line. Requests may consist of multiple lines. The first line of a full request contains the command, the page desired and protocol. Subsequent lines contain headers.

8.2 HTTP Request Methods

The following are the built-in HTTP request methods.

Table 8.1 HTTP Request Methods

Method	**Description**
GET	Request to read a Web page

HEAD	Request to read a Web page's header
PUT	Request to store a Web page
POST	Append to a named resource (e.g., a Web page)
DELETE	Remove the Web page
LINK	Connects two existing resources
UNLINK	Breaks an existing connection between two resources

8.2.1 GET Method

The GET method requests the server to send the requested page. However, if the GET request is followed by an *If-Modified-Since* header, the server only sends the data if it has been modified since the date supplied. Using this mechanism, a browser that is asked to display a cached page can conditionally ask for it from the server, giving the modification time associated with the page. If the cache page is still valid, the server just send back a status line announcing that fact, thus eliminating the overhead of transferring the page again.

8.2.2 HEAD Method

The HEAD method just asks for the message header, without the actual page. This method can be used to get a page's time of last modification, to collect information for indexing purposes. Conditional HEAD requests do not exist.

8.2.3 PUT Method

The PUT method writes the page. This method makes it possible to build a collection of web pages on a remote server. The body of the request contains the page. The lines following the PUT might include *content type* and authentication headers, to prove that the caller indeed has permission to perform the requested operation.

8.2.4 DELETE Method

DELETE removes the web page. The authentication and permission play a major role here. There is no guarantee that DELETE succeeds, since even if the remote HTTP server is willing to delete the page, the underlying file may have a mode that forbids the HTTP server from modifying or removing it.

8.2.5 LINK Method

The LINK method connects the existing resources.

8.2.6 UNLINK Method

The UNLINK method breaks an existing connection between two resources.

8.3 Running the HTTP Server

Copy all the project files from floppy disk or Zip disk. The HTTP server is packaged in a single small jar file for ease of use.

To execute the HTTP server, issue the following command at the command prompt: -

```
"java -jar WebServerLite.jar"
```

This will start the HTTP server on the default port (80) and your web directory will be the current directory.

To change the port number and location of the web root directory, you can execute the web server with optional parameters: -

```
"java –jar WebServerLite.jar C:\inetpub\wwwroot 8080"
```

8.4 Implemented Features

Features implemented in our HTTP server are

- It can deal with multiple requests at the same time
- Support for a variety of content-type (images, videos, HTML etc)
- Directory Browsing Features
- Index page retrieval without specifying full path
- Logging of the requests.

8.5 Content Types Supported

The HTTP server is aware of the following content types.

Table 8.2 Content Types Supported

Content type	**Recognized filename extensions**
application/postscript	ai ps eps
application/rtf	rtf
audio/basic	au snd
application/octet-stream	bin dms lha lzh exe class
application/msword	doc
application/pdf	pdf
application/powerpoint	ppt
application/smil	smi smil sml

application/x-javascript	js
application/zip	zip
audio/midi	midi kar
audio/mpeg	mga mp2 mp3
audio/x-wav	wav
image/gif	gif
image/ief	ief
image/jpeg	jpeg jpg jpe
image/png	png
image/tiff	tiff, tif
model/vrml	wrl, vrml
text/css	css
text/html	html htm shtml shtm stm sht
text/plain	txt inf nfo
text/xml	xml dtd
video/mpeg	Mpeg mpg mpe
video/x-msvideo	avi

Any unrecognized file types are delivered with the application/octet-stream content type.

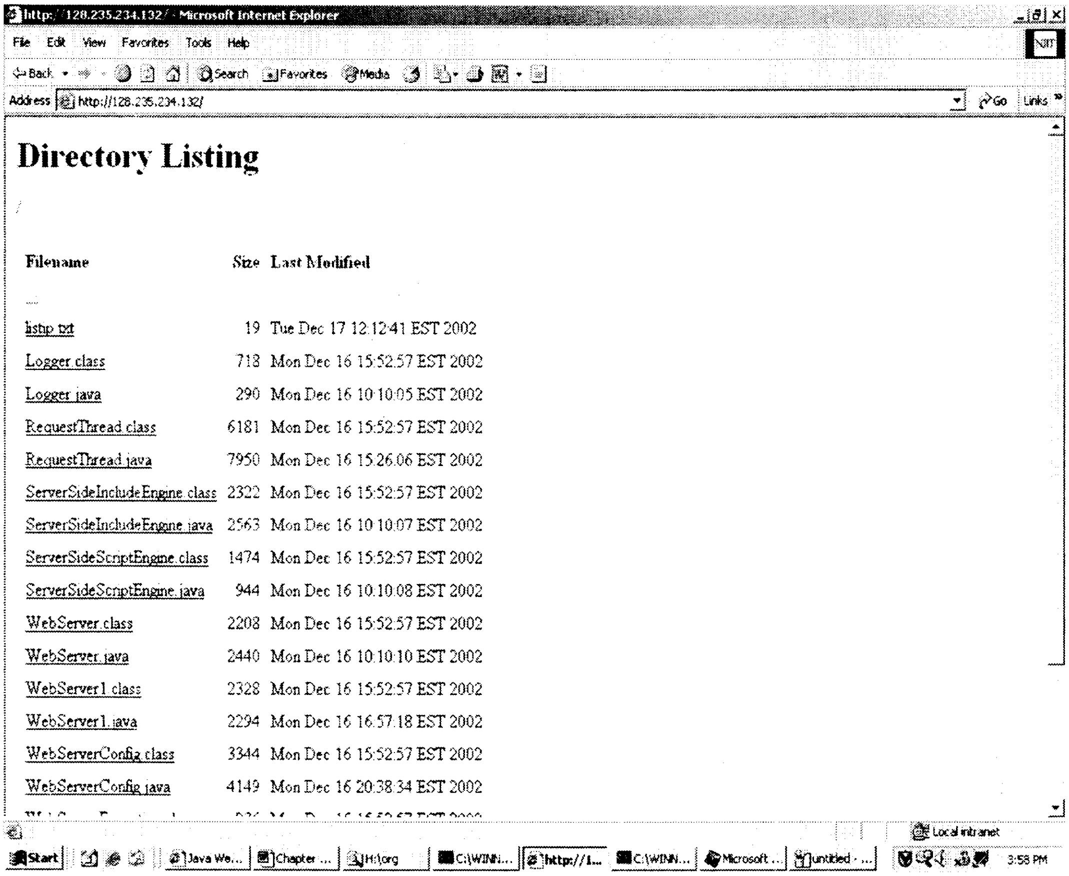
8.6 Directory Browsing and Index Pages

Accessing a directory via the HTTP server will present you with a browsable directory listing of all files in the logical web directory. Users will be prevented from obtaining any directory listings or viewing any files that are not reachable by descending from the web root directory.

The exception to this is if the directory contains a file recognized to be an index page, in which case the contents of that file will be delivered to the client. The known index page names are (in order): -

1. Index.html
2. Index.htm
3. Index.shtml
4. Index.shtm
5. Index.stm
6. Index.sht

The following snapshot shows the browsable directory listing of the files in the logical web directory. It displays all the files with size and last modified date and time.


http://128.235.234.132/ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://128.235.234.132/
Go Links
Local intranet
Start Java We... Chapter ... H:\org C:\WIN... http://1... C:\WIN... Microsoft ... untitled - ...
3:58 PM

Directory Listing

/

Filename	Size	Last Modified
..		
listip.txt	19	Tue Dec 17 12:12:41 EST 2002
Logger.class	718	Mon Dec 16 15:52:57 EST 2002
Logger.java	290	Mon Dec 16 10:10:05 EST 2002
RequestThread.class	6181	Mon Dec 16 15:52:57 EST 2002
RequestThread.java	7950	Mon Dec 16 15:26:06 EST 2002
ServerSideIncludeEngine.class	2322	Mon Dec 16 15:52:57 EST 2002
ServerSideIncludeEngine.java	2563	Mon Dec 16 10:10:07 EST 2002
ServerSideScriptEngine.class	1474	Mon Dec 16 15:52:57 EST 2002
ServerSideScriptEngine.java	944	Mon Dec 16 10:10:08 EST 2002
WebServer.class	2208	Mon Dec 16 15:52:57 EST 2002
WebServer.java	2440	Mon Dec 16 10:10:10 EST 2002
WebServer1.class	2328	Mon Dec 16 15:52:57 EST 2002
WebServer1.java	2294	Mon Dec 16 16:57:18 EST 2002
WebServerConfig.class	3344	Mon Dec 16 15:52:57 EST 2002
WebServerConfig.java	4149	Mon Dec 16 20:38:34 EST 2002

Figure 8.1 HTTP Directory Browsing

8.7 Logging of Requests

All HTTP Requests whether successful or not will be logged to the standard output and to a text file name "log.txt".

Each log entry is time stamped and is in the following format: -

```
[Fri Jan 10 15:53:26 EST 2003] 128.235.234.132 "GET / HTTP/1.1" 200
```

Each line in the log consists in order of

- Time stamp
- IP address of the remote host
- The raw request

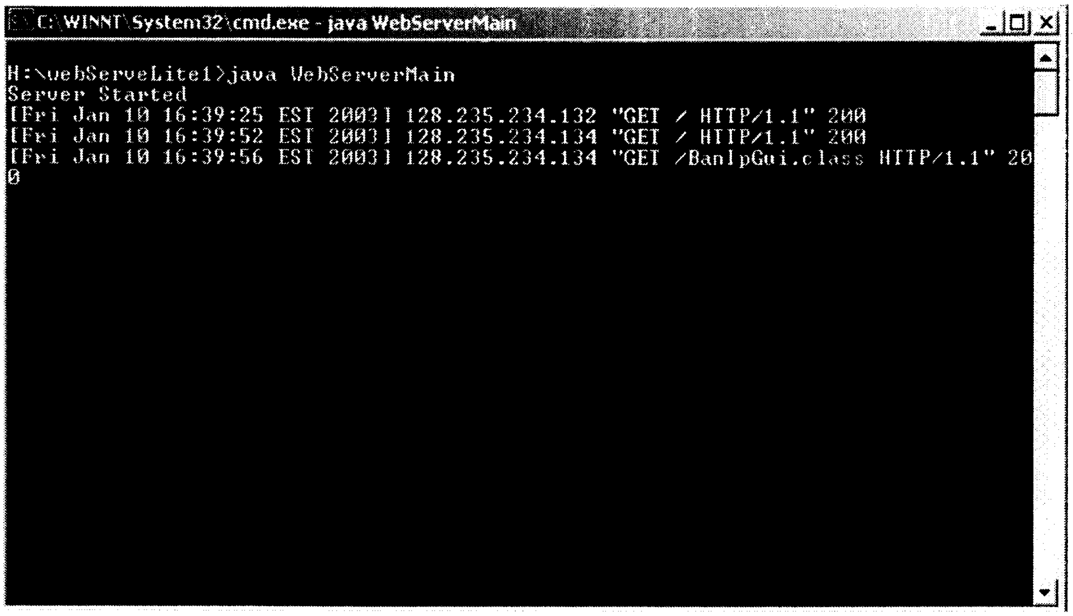
- A number to represent the status of the request.

Typical request status values are:

Table 8.3 Status Request Codes

Code	Meaning
200	OK
403	Forbidden
404	File Not Found
405	Method Not Allowed
000	Reserved for error messages from the web server

The following snapshot shows the logging at the standard output

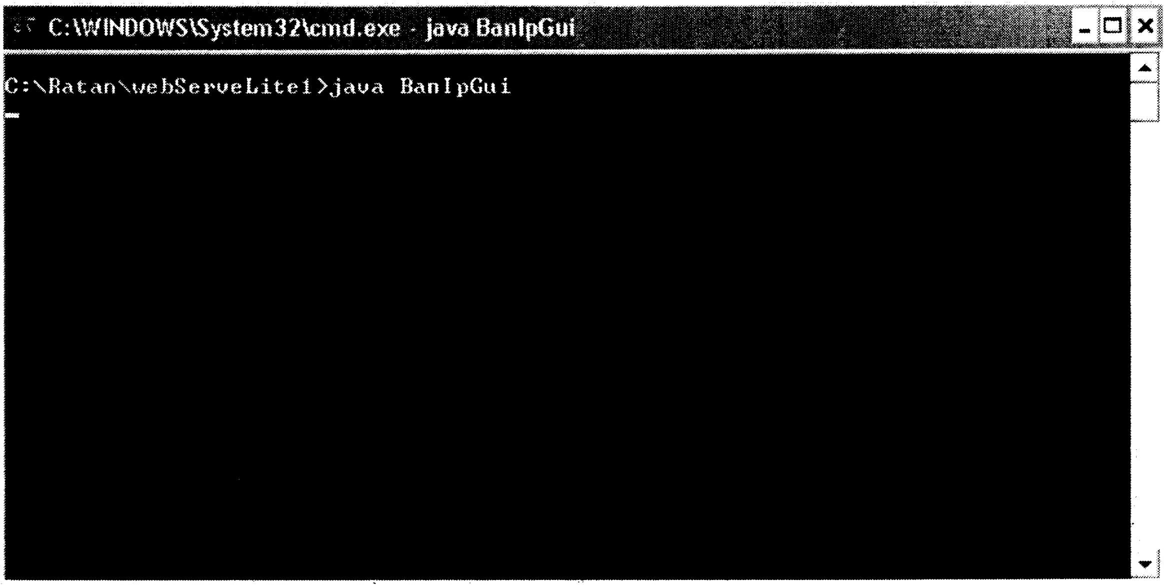


```
H:\webServeLite1>java WebServerMain
Server Started
[Fri Jan 10 16:39:25 EST 2003] 128.235.234.132 "GET / HTTP/1.1" 200
[Fri Jan 10 16:39:52 EST 2003] 128.235.234.134 "GET / HTTP/1.1" 200
[Fri Jan 10 16:39:56 EST 2003] 128.235.234.134 "GET /BanIpGui.class HTTP/1.1" 20
0
```

Figure 8.2 HTTP Command Prompt

8.8 To Block IP Address

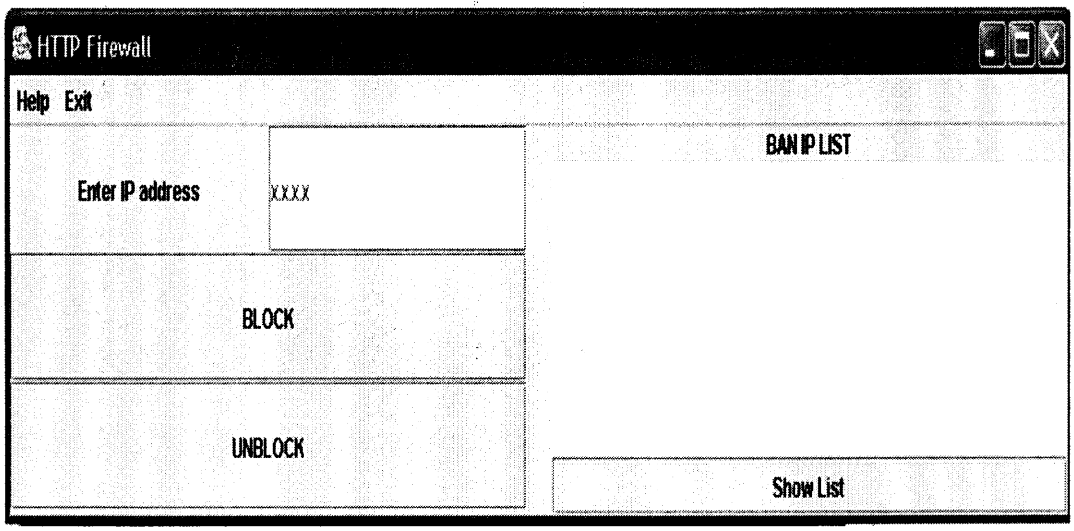
To block IP address, open the command prompt and go to the directory in which the HTTP server is installed. At the command prompt type "java BanIpGui" as shown below



```
C:\Ratan\webServeLite1>java BanIpGui
```

Figure 8.3 HTTP Starting

The command will start the user interface to Block IP Address which is as shown

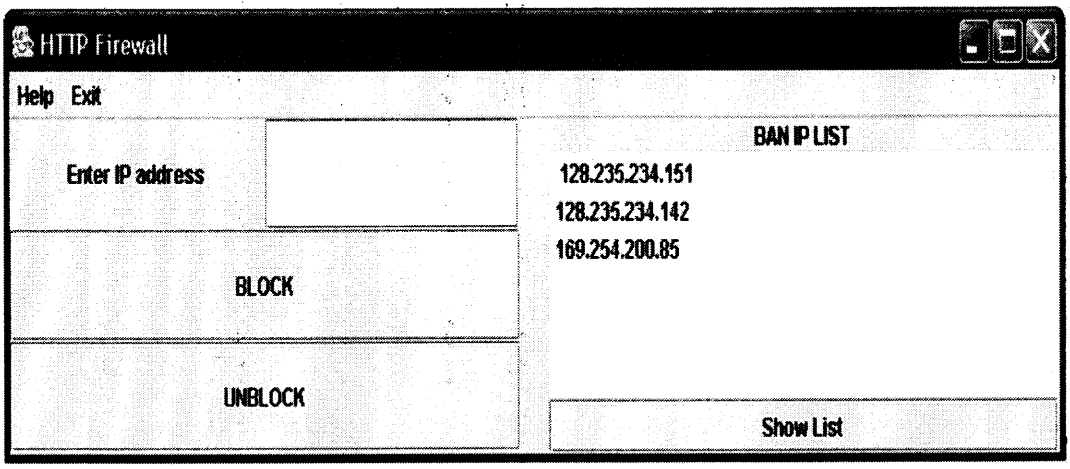


Enter IP address: XXXX

Figure 8.4 HTTP Firewall Screen

- In the text box adjacent to "Enter IP address" we must enter the IP address that should be blocked.
- Once we enter the IP address to be blocked, click on "Block" button to block the IP address. The blocked IP address cannot access our HTTP server.
- To see the list of banned IP's we need to click on "Show List" button.
- To remove an IP from banned list, we need to select the IP from the list and then click on "Unblock" button.

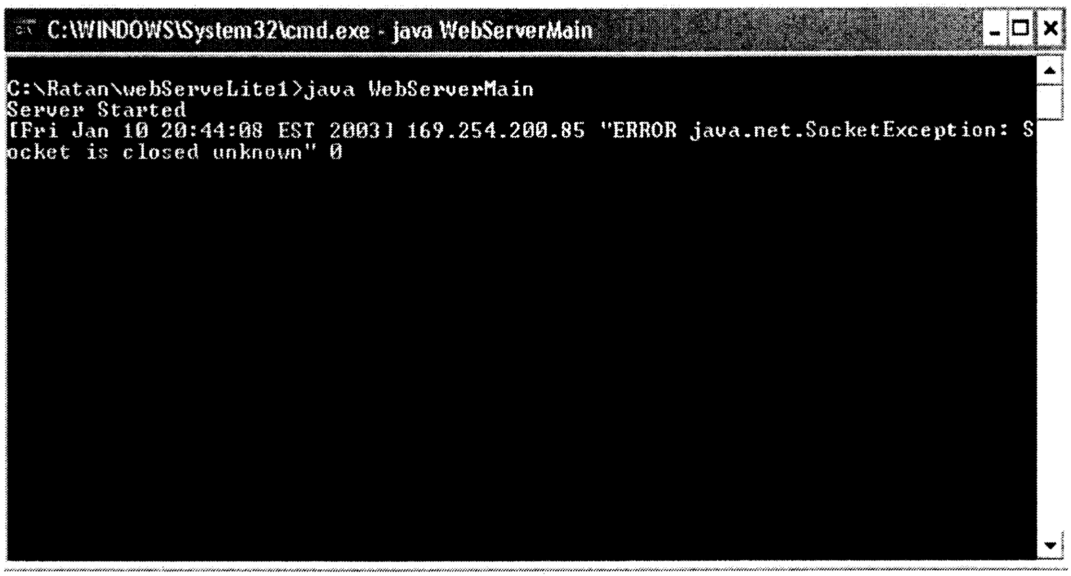
The following snapshot shows that the IP's "128.235.234.151", "128.235.234.142" and "169.254.200.85" are blocked.



128.235.234.151
128.235.234.142
169.254.200.85

Figure 8.5 HTTP Firewall Screen with Banned IP's

Suppose if a client from IP address "169.254.200.85" tries to access the server then we will see the following at the command prompt



```
C:\Ratan\webServeLite1>java WebServerMain
Server Started
[Fri Jan 10 20:44:08 EST 2003] 169.254.200.85 "ERROR java.net.SocketException: S
ocket is closed unknown" 0
```

Figure 8.6 HTTP Logging Screen

We will get the following error message

"ERROR java.net.SocketException: Socket is closed unknown".

This means that the server will ignore socket connection to the banned client.

This message will be logged to "log.txt" file which can be used for later analysis.

8.9 Miscellaneous Features

- The HTTP server responds to HTTP/1.0, HTTP/1.1 and GET requests with HTTP/1.0. Connection to the client is terminated if an unrecognized request is received.
- An HTTP 200 (OK) response is issued in response to the client's request if the file or directory was found and readable. If the directory does not contain an index page, the server then proceeds to deliver an HTML-formatted page describing the files in the directory. Otherwise, the appropriate file is delivered to the client with a suitable content type.

- Attempts to trick the server into requesting a file or directory that is not reachable by descending the web root directory are dealt with by issuing the client with an HTTP 403 (Forbidden) response.
- If a file or directory does not exist, the server issues the client with an HTTP 404 (File Not Found) response.
- There is currently no limit to the number of simultaneous connections that the HTTP server can accept.

CHAPTER 9

CONCLUSIONS

- Using the firewall interface we can deny access to destructive users.
- If we wish to maintain database to verify “username” an “password”, the design will be a 2-tire architecture which will be more useful.
- As all the applications are completely written in Java, they can be run on any platform.
- If we have a domain name, we can build a website using our HTTP server and we can give mail access to our users with our SMTP server and also the users can share their files using our FTP Server.

REFERENCES

1. Andrew S. Tanenbaum, Computer Networks, Prentice Hall, Englewood Cliffs NJ, Third Edition, 1997.

2. http://www.interhack.net/pubs/fwfaq/ on January 16, 2003.

3. http://java.sun.com

4. http://raddist.rad.com/networks/1998/smtp/smtp.htm on January 16, 2003.

5. http://www.intranetjournal.com/ on January 16, 2003.

6. http://www.ietf.org/rfc/rfc959.txt on January 16, 2003.

7. http://www.ietf.org/rfc/rfc0821.txt on January 16, 2003.
8. http://jetty.mortbay.org/ on January 16, 2003.

9. http://www.w3c.org/ on January 16, 2003.