International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2015): 6.391

FireMAN: Next Generation Firewall Systems Design using Data Mining

Nareshkumar D. Harale[1], Dr. B. B. Meshram[2]

[1]Department of Computer Science & Engineering, Sant Gadge Baba Amravati University, Amravati, Maharashtra, India
nareshkumar.d.harale[at]gmail.com
[2]Department of Computer Technology, VJTI, Mumbai, Maharashtra, India
Bbmeshram[at]vjti.ac.in

Abstract: ***Network security is main issue of this generation of computing because cyber-attacks are increasing day by day in terms diversity and complexity. Establishing a network is not a big issue for network administrators but protecting the entire network is a big issue. There are various methods and tools are available today for destroying the existing network. In this paper mainly emphasize on the network security also we present some major issues that can affect our network. Also key objective is to build a comprehensive firewall that can secure a computer from outside threat, monitor data that flows into, filter if objectionable and forward if it pertains to the set standards and polices. The user should be able to set various policies for filtering without having to go into the details of its implementation. Also, the user should be having administrative privileges, so as to safeguard the integrity of the system. The defined rules are applied on the packets and its contents flowing into and outside the system In addition to this, the firewall should implement filtering of data at the application level. Also a port scanning utility would be an added advantage in determining the various intrusion attempts into the computing systems. Meanwhile, stress should be given on the ease of use and user friendliness of the system.***

Keywords: Network Attacks, Network Intrusion, Port Scanning, Intrusion Protection, Data Mining, Firewall Systems, Packet Filtering, Data Filtering, Network Security

1. Introduction

Interest and knowledge about computer and network security is growing along with the need for it. This interest is, no doubt, due to the continued expansion of the Internet and the increase in the number of businesses that are migrating their sales and information Channels to the Internet. The growth in the use of networked computers in business, especially for e-mail, has also fueled this interest. Many people are also presented with the post-mortems of security breaches in high-profile companies in the nightly news and are given the impression that some bastion of defense had failed to prevent some intrusion. One result of these influences is that Firewall has become an important part of network security.

Internet Security

Internet security is certainly important issue today. The internet itself an entirely new thing in the world: a market place, a library, even a telephone. It's growth has been astounding. The internet is a large city, not a series of small towns. Anyone can use it and use it nearly anonymously. But the internet is a bad neighborhood. Internet security is not very different from other forms of security. The same concepts used to design castles applied to construction of a web server. The strength of one's computer security defenses has two faces similar to a coin. On one face it should be proportional to the threat from that arena and the other is the cost of security. Computer security is means to achieve information security.

Achieving robust security framework - There is no such thing as absolute security. We should be aware that an attacker doesn't go through security, but around it. We have to keep the level of all our defenses at about the same height and also in layer wise fashion. The security policy should be an integral part of the original design and should be simple. The key to achieve security is through firewalls.

Firewall Systems

The name firewall is derived from the technique used in construction in which a wall is constructed from fire-retardant materials to prevent or slow down the spreading of a fire. A firewall is a hardware or software device which is configured to permit, deny, or proxy data through a computer network which has different levels of trust. Firewalls have the following attribute such as All communications pass through the firewall, A firewall permits, only traffic that is authorized, and A firewall can withstand attacks on itself.

The basic function of a firewall is to screen network communications for the purposes of preventing unauthorized access to or from a computer network. A firewall is a computer or computers that stand between trusted network and untrusted networks. Simply put, a firewall acts as a buffer between a trusted network and untrusted networks.

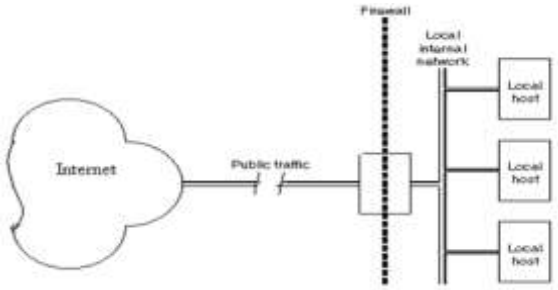


Figure 1: Concept of Firewall Systems

A firewall can be a router, a personal computer, a host or a collection of hosts set up specifically to shield a private network from protocols and services that can be abused

from hosts outside the trusted networks. A firewall system is usually located at a network perimeter such as a site's connection to the internet. However, it can and should be located inside network perimeter to provide additional and more specific protection to a smaller collection of hosts.

Data Mining

Data mining is often described as finding hidden information in a database. The terms knowledge discovery in databases (KDD) and data mining are often used interchangeably. Data mining access of a database differs from a database differs from traditional access in several ways like Query, Data and Output.

Data mining applications - Data mining has a wide variety of applications which include:

- **Finance** : applications include analysis of creditworthiness of clients, segmentation of account receivables
- **Marketing**: applications include analysis of consumer behavior based on buying, determination of market strategies
- **Manufacturing**: Applications involve optimization of resources like machines, manpower and materials
- **Healthcare**: applications include discovering patterns in radiological images.

Data Mining Algorithms - The data mining algorithms can be classified under 3 broad categories – classification, clustering, and association rules. The candidate data mining algorithms under each of these categories which can be applied in firewalls are enumerated here. When compared to the rest of the computing industry, firewall technology is quite an adolescent. The pioneering generation of firewall architectures, called packet filter firewalls, first appeared around 1985 courtesy of the IOS software division of computer networks giant Cisco. Three years later, the first paper on firewall technology came out, authored by Jeff Mogul of the Digital Equipment Corporation (DEC). However, during of all these years, Dave Presetto and Howard Trickey of AT&T Bell Laboratories were developing the second firewall generation - circuit level firewalls. Their work encompassed the decade, starting from 1980 and concluding in 1990.

In 1990 and 1991, Bill Cheswick, Marcus Ranum, and Gene Spafford published papers that described the third generation of firewalls, called application layer firewalls (or proxy-based firewalls). This triumvirate researched and developed the third generation independently of each other, with Ranum receiving the most attention for his work. In 1991, the DEC released the first firewall commercial product, named "SEAL". SEAL was based on the work of Marcus Ranum. The following year, Bob Braden and Annette DeSchlon of the University of Southern California began to develop their own fourth generation packet filter firewall system, called "Visas". The system is the first one to have had a visual integration interface, replete with colors and icons. Visas formed the basis of the commercial product FireWall-1, released in 1994 by the Israeli company Check Point Software.

In 1996, Scott Wiegel of the Global Internet Software Group began work on the fifth generation of firewalls, called the Kernel Proxy architecture. A year later, Cisco released the first commercial product based on the Kernel Proxy technology, the Cisco Centri Firewall. Recently, the industry has seen the convergence of firewall technologies and intrusion-prevention systems. This has led to some people calling the fusion the "Next Generation Firewalls".

1.2 Motivation

The nature of computer crime has changed over the years as the technology has changed and the opportunities for crime have changed. Although thrill-seeking adolescent hackers are still common, the field is increasingly dominated by professionals who steal information for sale and disgruntled employees who damage systems or steal information for revenge or profit. When Willie Sutton was asked why he robbed banks, he replied, "because that's where the money is." People attack computers because that's where the information is, and in our hyper-competitive, hi-tech business and international environment, information increasingly has great value. Some alienated individuals also gain a sense of power, control, and self-importance through successful penetration of computer systems to steal or destroy information or disrupt an organization's activities. A common view of computer security is that the threat comes from a vast group of malicious hackers "out there." The focus of many computer security efforts is on keeping the outsiders out -- through physical and technical measures such as gates, guards, locks, firewalls, passwords, etc. Yet, while the threat from outsiders is indeed as great as generally believed, the malicious insider with approved access to the system is an even greater threat!.

Why use data mining techniques in a firewall?

This is one obvious question that comes to our mind. The task of manually-managing firewall policy rules becomes very difficult and time-consuming. This enormous task addresses the need for the effective management of firewall security with policy management techniques and tools that enable network administrators with ease to optimize and validate firewall rules automatically. Although the deployment of firewall technology is the first important milestone toward securing networks, the effectiveness of firewall security may be limited or compromised by a poor management of firewall policy rules.

1. Problem Definition

The modules of the project consists design and implementation of firewall user interface, Implementation of a framework wherein the user can easily use packet header filtering. (Filtering of packets based on protocols, ip addresses, ports, Ethernet address)

3. Implementation of framework wherein the user can easily does content filtering. (Blocked Website List, Trusted Website list, Web content filtering, User blocking, File download restrictions, Dynamic content filtering), design and Implementation of Port Scanning. The modules are explained below:

1. The user interface being the primary module interacting with the user needs to be complete, easy to and the user should be able to relate it to some real world metaphor. As this product is being targeted for average users, the user interface should be perceived to be efficient and intuitive by the user. It should be easy to learn and it should guide the user along by prompting actions. In other words, the user interface should be responsive to the user needs. The user interface of this firewall provides the following facilities:

- Create a rule, Insert a rule, Delete a rule, Activate data filtering, Port scanning
- Choose different alternatives such as filtering of text, phrase, file extensions, images and videos

2. Packet filtering in short means rejecting data packets from unauthorized hosts and rejecting connection attempts to unauthorized services by extracting the packet headers and deciding their fate based upon a set of user defined rules. The implementation of packet filtering includes use and customization of iptables to control the flow of packets through the netfilter framework available under the LINUX operating system. The iptables utility is a powerful tool for filtering packets at various levels namely –prerouting, postrouting, local in, local out and forward hooks in the netfilter framework.

3. Content filtering is basically the filtering of data at application level. It can be carried out for different applications like HTTP, FTP, SMTP etc. The implementation of data filtering includes the use of the popular web caching proxy named SQUID. The access control list (ACL) maintained by SQUID is used for content filtering, based on Text, Images, Videos, Scripts, URLs, Websites and Regular expressions.

4. Port Scanning has its importance when it comes keep a check on active connections and also on backdoor intrusion which may be align for the system. Port scanning includes scanning and displaying all the open and closed ports on the system.

2. Literature Survey

We have made the extensive literature survey of the following firewall,

2.1 Overview to open source firewall Systems

- SHORELINE, GUARDDOG and JAY"S firewalls are classified as Packet Filtering firewalls.
- PRIVOXY, NET NANNY, CYBER PATROL, GAUNTLET and DANE GUARDIAN firewalls are classified as Data Filtering firewalls.
- Whereas IPCOP and SMOOTHWALL firewalls are both Data Filtering as well as Packet Filtering firewalls.

2.2 Comparative Matrix of different firewall protection systems

Table 1: Comparison Chart of Firewall Systems

Firewall Name Firewall Type	Shoreline	Guarddog	Privoxy	Netnanny	Cyber patrol	Gauntlet	DansGuardian	Jay	Ipcop	Smoothwall
Packet Header Filtering	Y	Y	N	N	N	N	N	Y	Y	Y
Trusted Website List	N	N	Y	Y	Y	Y	Y	N	Y	N
Blocked Website List	N	N	Y	Y	Y	Y	Y	N	Y	N
Port Monitoring	N	N	N	N	Y	N	N	N	Y	Y
User Blocking	N	N	Y	Y	Y	Y	Y	N	Y	Y
Log Rule File	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Proxy Use	N	N	Y	Y	Y	Y	Y	N	Y	Y
Web Content Filtering	N	N	Y	Y	Y	Y	Y	N	N	N
File Download Restrictions	N	N	N	Y	Y	Y	Y	N	N	N
Dynamic Content Filtering	N	N	N	Y	Y	N	Y	N	N	N

2.3 Proposed Features of the Firewall Systems

Packet Header Filtering - This feature is taken from Shoreline, Guarddog, Jay, Ipcop and Smoothwall. Analyzes all incoming and outgoing pieces of data, called packets. It filters packets to control all data's access to your computer. Filtering is done by matching the packet header fields to specified filtering criteria. It lets you block specific data from entering your computer.

Blocked Website List -This feature is taken from Privoxy, NetNanny, Cyber Patrol, Gauntlet, Ipcop and DansGuardian. List of Internet websites the firewall

always blocks access to. These sites may contain data which is not appropriate for the user or in some way or the other it infringes the security laws of the user.

Trusted Website List -This feature is taken from Privoxy, NetNanny, Cyber Patrol, Gauntlet, Ipcop and DansGuardian. List of Internet websites the firewall always allows access to; you create this trusted list over time. These site may contain data which should not be filtered in any way and the user should be able to access the site without any limitations.

Web Content Filtering-This feature is taken from Privoxy, NetNanny, Cyber Patrol, Gauntlet and DansGuardian. Web content filtering (also known as information filtering) is the screening of web pages that is deemed objectionable. It filtering usually works by specifying character strings that, if matched, indicate undesirable content that is to be screened out.

User Blocking -This feature is taken from Privoxy, Netnanny, Cyber Patrol, Gauntlet, Ipcop, Smoothwall and DansGuardian.The firewall can be customized for each individual who uses the computer. For example, you can have the firewall allow some users unrestricted access, but restrict others (unique logins for each person is required).

File Download Restrictions -This feature is taken from Netnanny, DansGuardian, Gauntlet and Cyber Patrol.The firewall can be used to restrict the type of files to be downloaded from the internet. This filtering or restriction is exercised using the extensions of the files. This feature can be very well used to restrict file downloads like images, videos, audios, animations, executables etc.

Log Rule File - This feature is taken from all. This feature is a basic necessity of all firewall. It logs or stores the effects of the rule on the network packets. This is needed for auditing, debugging and fault recovery purposes.

Proxy Use -This feature is taken from Privoxy, Netnanny, Cyber Patrol, Gauntlet, Dansguardian, Ipcop and Smoothwall. The firewall makes use of the proxy server to implement the application layer filtering. Proxy server acts as a middleman between the internet and the host computer, while doing so it filters the data at the application level.

Port Monitoring (Intrusion Detection) -This feature is taken from Cyber Patrol, Ipcop and Smoothwall. Ports are doorways that allow data to travel in and out of your computer. Firewalls prevent port scanning and alert you of port scan attempts. Unmonitored ports are a quiet doorway for intruders to sneak into your computer.

Dynamic Content Filtering -This feature is taken from Cyber Patrol, Netnanny and Dansguardian. It works by looking at all the text that appears on a web page, then giving combinations of words and phrases described in the rules a positive or negative weight value. The values are added together to give the web page a total weight. If the total weight exceeds the Weight Threshold set, then the web page is blocked.

3. FireMAN Protection System Design

The FirMAN systems requirements are analyzed and designed using Object Oriented technques. omponent Diagram helps to model the physical aspect of an Object-Oriented software system. It illustrates the architectures of the software components and the dependencies between them. Those software components including run-time components, executable components also the source code components.

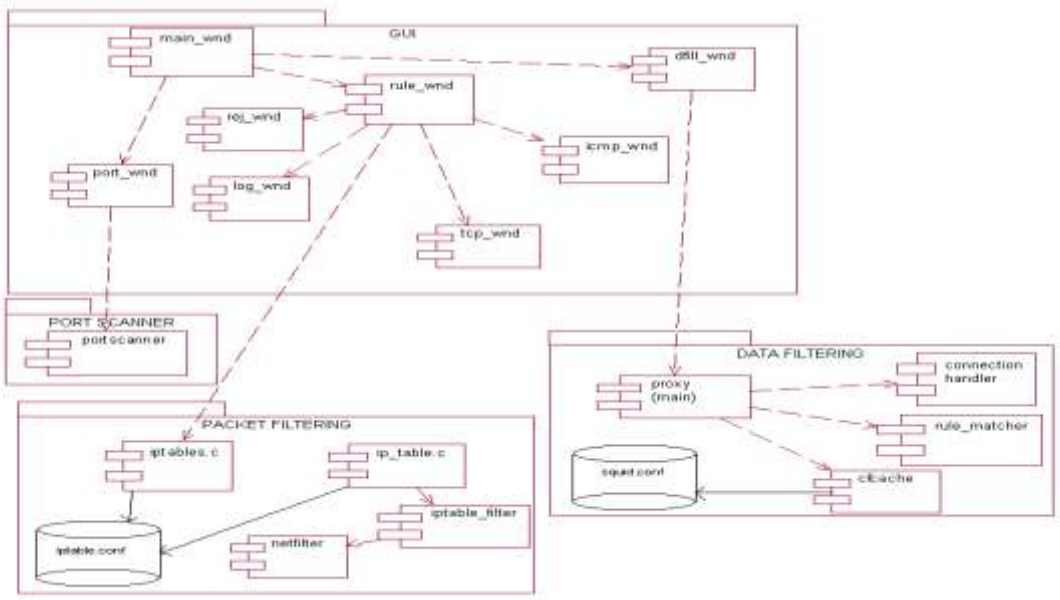


Figure 2: Component diagram for Proposed FireMAN Protection Systems

The component diagram shows that the GUI is split into a number of components. The rule_ wind provides an interface to iptables configuration file and similarly the dfill_wnd provides the interface to the squid configuration file. The netfilter gets the verdict from iptables and accordinlgy it filters the packet whereas the proxy filter works with quid configuration file.

4. Proposed Data structure and algorithms Design

The following algorithms have been used by us for our project. The various modules alongwith their pseudo codes have been given below.

- For Port Scanning
- For Packet header Filtering
- For Ban user
- For Ban Phrase
- For URL Filtering
- For Ban Site, file ext
- For Content Replacement

The algorithm for Packet header filtering is given as below.

INPUT: -User enters the various packet filtering specifications.

ALGORITHM

```
gchar sz_tmp_buffer[64];//used to store values extracted from various fields
gchar command_line[];//stores the command to be executed
GtkWidget *cb_tables;//gives protocol to be matched
GtkWidget *cb_chains;//gives chain in iptables where the rule is to be inserted
GtkWidget *cb_protocol;//gives protocol to be matched
GtkWidget *bt_tcp;//gives other options available with tcp protocol
GtkWidget *bt_icmp;//gives other options available with tcp protocol
GtkWidget *cb_jump;//gives target for the rule on match
GtkWidget *ed_src_address;//gives source address to be matched
GtkWidget *ed_src_port;//gives source port to be matched
GtkWidget *ed_dest_address;//gives destination address to be matched
GtkWidget *ed_dest_port;//gives destination port to be matched
GtkWidget *bt_set_reject;/*gives msg to be given to user when reject is selected as a target*/
GtkWidget *bt_log_rule;//gives log rule facility
GtkWidget *bt_accept;//gives protocol to be matched
GtkWidget *bt_cancel;//gives protocol to be matched
GtkWidget *ed_in_inter;//gives incoming ethernet address to be matched
GtkWidget *ed_out_inter;//gives outgoing ethernet address to be matched
GtkWidget *ed_packages;//gives no. of packets of data that matched the rule
GtkWidget *ed_bytes;//gives no. of bytes of data that matched the rule
GtkWidget *chk_fragment;
GtkWidget *chk_frag_w_options;
```

OUTPUT

```
FILE *p/*contains the rules loaded in iptables*/
create_win();/*This function creates a window to accept the packet filtering parameters from the user(builds the main window for create rule window;
builds the containers required to display the buttons, text fields, labels;
packs the buttons, text fields, labels in the containers;displays the containers used; displays the create rule window)*/

{

/*The statements below are used to build the command for iptables by transfering data from temporary buffer to command_line array*/

strcat(command_line, "iptables -t ");//iptables commands start with this string
strcpy(sz_tmp_buffer, gtk_entry_get_text(main2_wnd.cb_tables));
strcat(command_line, sz_tmp_buffer);

strcat(command_line, " -A ");//used to append to the iptables
strcpy(sz_tmp_buffer, gtk_entry_get_text(main2_wnd.cb_chains));
strcat(command_line, sz_tmp_buffer);

strcat(command_line, " -p ");//used to specify the protocol to be worked upon
strcpy(sz_tmp_buffer, gtk_entry_get_text(main2_wnd.cb_protocol));
strcat(command_line, sz_tmp_buffer);
```

```
strcpy(sz_tmp_buffer, gtk_entry_get_text(main2_wnd.ed_in_inter));
if (strcmp(sz_tmp_buffer, "") != 0)
{
strcat(command_line, " -i ");//used to specify input interface
strcat(command_line, sz_tmp_buffer);
}
strcpy(sz_tmp_buffer, gtk_entry_get_text(main2_wnd.ed_out_inter));
if (strcmp(sz_tmp_buffer, "") != 0)
{
strcat(command_line, " -o ");//used to specify output interface
strcat(command_line, sz_tmp_buffer);
}

if (GTK_TOGGLE_BUTTON(main2_wnd.chk_frag_w_options)->active)
{
 strcat(command_line, "! -f ");//used to specify that one does not have to apply the rules on the individual fragments and it has to be applied only at the beginning

if (GTK_TOGGLE_BUTTON(main2_wnd.chk_fragment)->active)
{
 strcat(command_line, "-f ");//used to specify that one has to apply rule to individual fragments
}

/*it specifies the number of packets to be inspected*/
strcpy(sz_tmp_buffer, gtk_entry_get_text(main2_wnd.ed_packages));
if (strcmp(sz_tmp_buffer, "") == 0)
strcat(command_line, sz_tmp_buffer);

/*it specifies the number of bytes to be inspected*/
strcpy(sz_tmp_buffer, gtk_entry_get_text(main2_wnd.ed_bytes));
if (strcmp(sz_tmp_buffer, "") == 0)
strcat(command_line, sz_tmp_buffer);

 packets accept or reject or drop*/
strcpy(sz_tmp_buffer, gtk_entry_get_text(main2_wnd.cb_jump));
strcat(command_line, sz_tmp_buffer);

system(command_line); /*this is a system command that is called to execute the iptables command which is now in command_line */

ghar *p_command = "iptables -L";/*command to read the iptables rules stored*/
FILE *p;/*file descriptor for file containing rules*/
p=popen(p_command, "r");/*command to execute p_command and read the rules into file*/
}
```

The algorithm for Banned site is given as below.

INPUT

int type;//represents the type of content filtering(bansite, banurl)

OUTPUT

Boolean success;(returns success is the rule was properly added)

ALGORITHM

```
static char label[30] ;
static char filename[50] ;
FILE *f;
gchar *text;
//this text stores the IP address to be blocked given by the user.

if(type == bansite)
//to check whether the type of content filtering is ban site.

 strcpy(label, "BANNED LIST");
 strcpy(filename, "/etc/dansguardian/bannedsitelist");
//here the name of the file to be modified in dansguardian is //copied into the filename.
```

```
text = (gchar *)gtk_entry_get_text(text_field);
//IP address entered by the user is copied into the text

f=fopen(filename, "a+");
//that particular file is opened in append mode to add the banned site.

fprintf(f, "%s", text);
//the ban site is written in the file

ghar *p_command = "/etc/rc.d/init.d/dansguardian -g";
//dansguardian is reconfigured to include the new changes

FILE *p;
/*file descriptor for file containing rules*/

p=popen(p_command, "r");
/*command to execute p_command*/

if(p!=NULL)
return true;
else
return false;
/*if the comm
```

The algorithm for Banned URL is given as below.

```
INPUT
int type;//represents the type of content filtering(bansite, banurl)

OUTPUT
Boolean success;(returns success is the rule was properly added)

ALGORITHM
static char label[30] ;
static char filename[50] ;
FILE *f;
gchar *text;
//this text stores the URL to be blocked given by the user.

if(type == banurl)
//to check whether the type of content filtering is ban site.

 strcpy(label, "BANNED URL");
 strcpy(filename, "/etc/dansguardian/bannedurllist");
//here the name of the file to be modified in dansguardian is //copied into the filename.

text = (gchar *)gtk_entry_get_text(text_field);
//URL entered by the user is copied into the text

f=fopen(filename, "a+");
//that particular file is opened in append mode to add the banned url.

fprintf(f, "%s", text);
//the ban url is written in the file

ghar *p_command = "/etc/rc.d/init.d/dansguardian -g";
//dansguardian is reconfigured to include the new changes

FILE *p;
/*file descriptor for file containing rules*/

p=popen(p_command, "r");
/*command to execute p_command*/

if(p!=NULL)
return true;
else
return false;
//returns false if the rule could not be added
```

The algorithm for Banned phrase is given as below.

INPUT
int type;//represents the type of content filtering(bansite, banurl)

OUTPUT

Boolean success;(returns success is the rule was properly added)

ALGORITHM

```
static char label[30] ;
static char filename[50] ;
FILE *f;
gchar *text;
//this text stores the phrase to be blocked given by the user.

if(type == banphrase)
//to check whether the type of content filtering is ban site.

 strcpy(label, "BANNED PHRASE");
 strcpy(filename, "/etc/dansguardian/bannedphraselist");
//here the name of the file to be modified in dansguardian is //copied into the filename.

text = (gchar *)gtk_entry_get_text(text_field);
//IP address entered by the user is copied into the text

f=fopen(filename, "a+");
//that particular file is opened in append mode to add the banned site.

fprintf(f, "%s", text);
//the ban phrase is written in the file

ghar *p_command = "/etc/rc.d/init.d/dansguardian -g";
//dansguardian is reconfigured to include the new changes

FILE *p;
/*file descriptor for file containing rules*/

p=popen(p_command, "r");
/*command to execute p_command*/

if(p!=NULL)
return true;
else
return false;
```

The algorithm for Banned user is given as below.

INPUT
int type;//represents the type of content filtering(bansite, banuser)

OUTPUT
Boolean success;(returns success is the rule was properly added)

ALGORITHM

```
static char label[30] ;
static char filename[50] ;
FILE *f;
gchar *text;
//this text stores the phrase to be blocked given by the user.

if(type == banuser)
//to check whether the type of content filtering is ban site.

 strcpy(label, "BANNED USER");
 strcpy(filename, "/etc/dansguardian/banneduserlist");
//here the name of the file to be modified in dansguardian is //copied into the filename.
```

```
text = (gchar *)gtk_entry_get_text(text_field);
//name enteres by the system administrator is copied into the text

f=fopen(filename, "a+");
//that particular file is opened in append mode to add the banned site.

fprintf(f, "%s", text);
//the ban user is written in the file

ghar *p_command = "/etc/rc.d/init.d/dansguardian -g";
//dansguardian is reconfigured to include the new changes

FILE *p;
/*file descriptor for file containing rules*/

p=popen(p_command, "r");
/*command to execute p_command*/

if(p!=NULL)
return true;

else
return false;//returns false if the rule was not accepted
```

The algorithm for Banned MIME type is given as below.

```
INPUT
int type;//represents the type of content filtering(banmime, banuser)

OUTPUT
Boolean success;(returns success is the rule was properly added)

ALGORITHM

static char label[30] ;
static char filename[50] ;
FILE *f;
gchar *text;
//this text stores the mime to be blocked given by the user.

if(type == banmime)
//to check whether the type of content filtering is ban site.

 strcpy(label, "BANNED MIME");
 strcpy(filename, "/etc/dansguardian/bannedmimelist");
//here the name of the file to be modified in dansguardian is //copied into the filename.

text = (gchar *)gtk_entry_get_text(text_field);
//mime type entered by the system administrator is copied into the text

f=fopen(filename, "a+");
//that particular file is opened in append mode to add the banned mime

fprintf(f, "%s", text);
//the ban user is written in the file

ghar *p_command = "/etc/rc.d/init.d/dansguardian -g";
//dansguardian is reconfigured to include the new changes

FILE *p;
/*file descriptor for file containing rules*/

p=popen(p_command, "r");
/*command to execute p_command*/

if(p!=NULL)
return true;

else
```

```
return false;//returns false if the rule was not accepted
```

The algorithm for content replacement is given as below.

```
INPUT
int type;//represents the type of content filtering(banmime, banuser)

OUTPUT
Boolean success;(returns success is the rule was properly added)

ALGORITHM
static char label[30] ;
static char filename[50] ;
FILE *f;
gchar *text;
//this text stores the mime to be blocked given by the user

if(type == contentreplace)

 strcpy(label, "CONTENT REPLACEMENT");
strcpy(filename, "/etc/dansguardian/contentregexplist");

text = (gchar *)gtk_entry_get_text(text_field);
//mime type entered by the system administrator is copied into the text

f=fopen(filename, "a+");
//that particular file is opened in append mode to add the banned mime

fprintf(f, "%s", text);
//the replacement is written in the file

ghar *p_command = "/etc/rc.d/init.d/dansguardian -g";
//dansguardian is reconfigured to include the new changes

FILE *p;
/*file descriptor for file containing rules*/

p=popen(p_command, "r");
/*command to execute p_command*/

if(p!=NULL)
return true;

else
return false;//returns false if the rule was not accepted
```

The algorithm for Allowed phrase type is given as below.

```
INPUT
int type;//represents the type of content filtering(banmime, banuser)

OUTPUT
Boolean success;(returns success is the rule was properly added)

ALGORITHM
static char label[30] ;
static char filename[50] ;
FILE *f;
gchar *text;
//this text stores the phrase to be allowed given by the user.

if(type == allowphrase)
//to check whether the type of content filtering is execptionphrase.

 strcpy(label, "ALLOWED PHRASE LIST");
 strcpy(filename, "/etc/dansguardian/exeptionphraselist");
//here the name of the file to be modified in dansguardian is //copied into the filename.

text = (gchar *)gtk_entry_get_text(text_field);
```

```
//exeption phrase entered by the system administrator is copied into the text

f=fopen(filename, "a+");
//that particular file is opened in append mode to add the exeption phrase

fprintf(f, "%s", text);
//the exeption phrase is written in the file

ghar *p_command = "/etc/rc.d/init.d/dansguardian -g";
//dansguardian is reconfigured to include the new changes

FILE *p;
/*file descriptor for file containing rules*/

p=popen(p_command, "r");
/*command to execute p_command*/
if(p!=NULL)
return true;

else
return false;//returns false if the rule was not accepted
```

The algorithm for Allowed site type is given as below.

```
INPUT

int type;//represents the type of content filtering(banmime, allowsite)

OUTPUT

Boolean success;(returns success is the rule was properly added)

ALGORITHM

static char label[30] ;
static char filename[50] ;
FILE *f;
gchar *text;
//this text stores the site to be allowed given by the user.

if(type == allowsite)
//to check whether the type of content filtering is execptionsite

 strcpy(label, "ALLOWED SITE LIST");
 strcpy(filename, "/etc/dansguardian/exeptionsitelist");
//here the name of the file to be modified in dansguardian is //copied into the filename.

text = (gchar *)gtk_entry_get_text(text_field);
//exeption site entered by the system administrator is copied into the text
f=fopen(filename, "a+");
//that particular file is opened in append mode to add the exeption site
 fprintf(f, "%s", text);
//the exeption site is written in the file
ghar *p_command = "/etc/rc.d/init.d/dansguardian -g";
//dansguardian is reconfigured to include the new changes
FILE *p;
/*file descriptor for file containing rules*/
p=popen(p_command, "r");
/*command to execute p_command*/
if(p!=NULL)
return true;
else
return false;//returns false if the rule was not accepted
```

The algorithm for Allowed URL type is given as below.

```
INPUT
int type;//represents the type of content filtering(banmime, allowed url)

OUTPUT
```

```
Boolean success;(returns success is the rule was properly added)

ALGORITHM

static char label[30] ;
static char filename[50] ;
FILE *f;
gchar *text;
//this text stores the URL to be allowed given by the user.

if(type == allowurl)
//to check whether the type of content filtering is execptionurl.

 strcpy(label, "ALLOWED PHRASE LIST");
 strcpy(filename, "/etc/dansguardian/exeptionurllist");
//here the name of the file to be modified in dansguardian is //copied into the filename.

text = (gchar *)gtk_entry_get_text(text_field);
//exeption URL entered by the system administrator is copied into the text

f=fopen(filename, "a+");
//that particular file is opened in append mode to add the exeption phrase

fprintf(f, "%s", text);
//the exeption url is written in the file
ghar *p_command = "/etc/rc.d/init.d/dansguardian -g";
//dansguardian is reconfigured to include the new changes
FILE *p;
/*file descriptor for file containing rules*/
p=popen(p_command, "r");
/*command to execute p_command*/
if(p!=NULL)
return true; else
return false;//returns false if the rule was not accepted
```

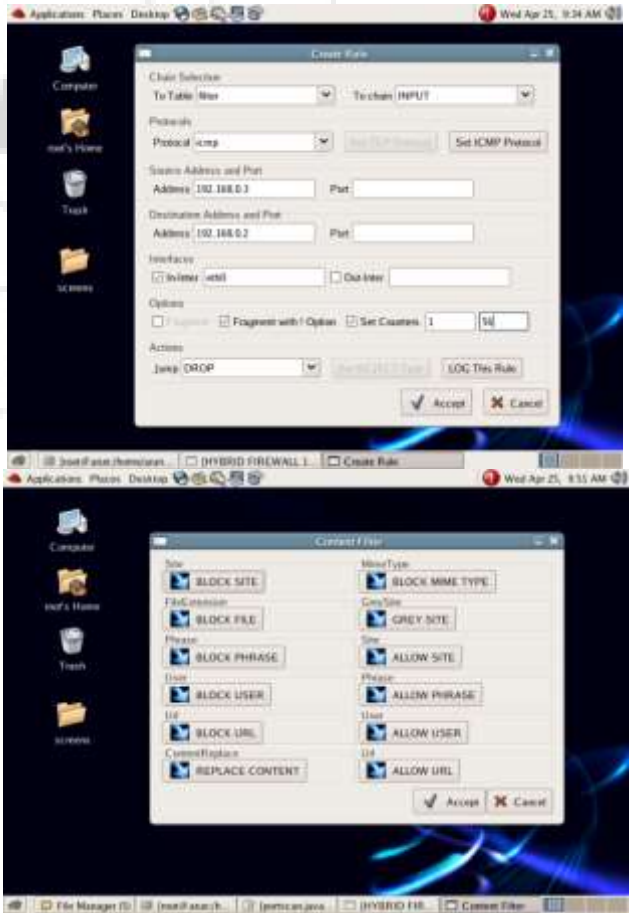
5. FireMAN Systems: Implementation and Results

After the analysis and design, the hybrid firewall is implemented. The implementation of the system is done in the C++ language and on Linux platform. Before the implementation some tools are required for the Hybrid Firewall.

5.1 Input and Output

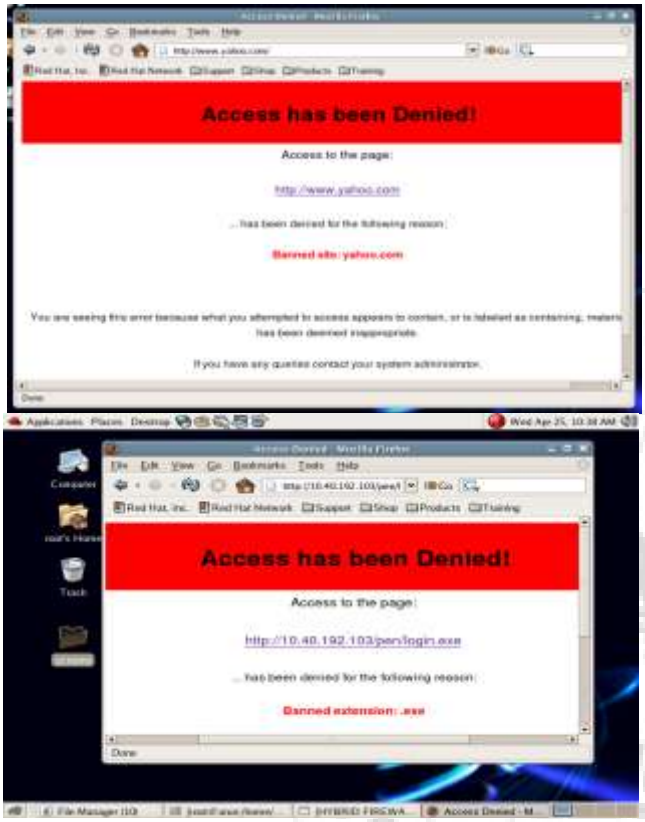
Screenshot 1: Main Window of FireMAN Protection Systems

The screenshot 1 shows the main page of the hybrid firewall which is the default page of this firewall. Here it displays all the rules that are currently active. Along with the rules their details like the source, destination and the target for the rule is also being displayed.



Screenshot 2 and 3: Creating Rule for Firewall and Window for Data Filtering.

The screenshot 2 shows the window for creating a rule for the firewall. This window shows options for chain selection, protocol ICMP. It also shows the source address and port and destination address and port. We can also set the interfaces. Finally it gives the action to be performed for this rule. The screenshot 3 shows the window for selecting a data filtering rule for the firewall. All the individual options are explained in the following screenshots.



Screenshot 4: Output for Banned Site and Extension

The screenshot 9 shows the result of the rules in the firewall which blocks the site "yahoo.com" and displays a message of "Access has been Denied!" in a browser The screenshot 10 shows the window in which a rule for banned extension list is being created. We enter a name of some extensions and then press the ADD button to activate that rule. We can also delete the names of list whenever required.

6. Conclusion and Future Scope

High level programming languages like C++, JAVA, and GTK API, IPTABLES, SQUID were the most important tools required for designing the firewall. The packet headers are extracted, compared and accordingly allowed or denied. The content is checked to see if they match against the policies set for them and accordingly filtered. A firewall system is a necessity without irrespective if you are going to connect your LAN to the Internet or not. Firewalls can examine the traffic coming into and going out of your LAN and make decisions on what kinds of traffic to permit or deny. Techniques such as packet filtering make decisions based on information found in a network packet header, such as the packet's address or the network protocol used. Application proxies act as middle-man and interact with enterprise servers on the Internet so that clients on your internal LAN are never exposed to direct communication and, thus, potential abuse. To decide on what kind of firewall system you will build or buy, you first have to examine your current security policies and evaluate the services users expect to receive after connecting to the Internet. Using your security policy, you can design a firewall strategy, using routers, bastion hosts, and other techniques to secure your network.

This important software is intended for users for whom internet is an easy access to obtain a large amount of information but they are generally unaware of the network attacks. If they are somewhat aware they do not know much about how to prevent it. However in big networks there are those network administrators who know how to prevent it by using commercially available firewalls which are quite expensive and generally out of the reach of general user. Moreover they are designed keeping in mind that the user is an expert in networking, thus laying very less importance on user interface. Our software is designed keeping in mind a novice user. We have laid a great deal of stress on providing a sound user interface so that a general user should learn how to use and control this firewall in simple and easy steps without compromising on functionality and flexibility of the software. The prime motive of our study is to bring together the merits of packet filtering and data filtering under one software called FireMAN -Next Generation Firewall Protection Systems.

Although it is a very comprehensive firewall, it has a wide scope for its development. Various future prospects that we can highlight are as follows:

- Proxy support for other applications using FTP, SMTP, DNS, DHCP, IMAP, POP3, TELNET.
- Providing antivirus support.

References

[1] Brian Komar, Ronald Beekelaar, and Joern Wettern, PhD "Firewall for Dummies" 2nd edition, published by Wiley Publications. www.wiley.com
[2] Robert Zalenski "Firewall Technology", a paper presented in IEEE in February/March 2002 as IEEE potentials.
[3] Rusty Russel "Iptables –HowTo" . This document describes how to filter bad packets for Linux kernel 2.3.15 and beyond. netfilters@lists.samba.org.
[4] Oskar Andreasson "IpTables Tutorial 1.1.9". This tutorial describes the basic working of iptables in detail. blueflux@koffein.net
[5] Terry Ogletree "Practical Firewall", 1st edition June 12, 2000. This book is aimed at beginning to network administrators, it provides information from very basic level.
[6] A manual inbuilt in Linux "Squid HowTo". It informs about how dat can be filtered using the Squid. (www.squid-cache.org)

[7] A manual inbuilt in Linux "Squid Quick Start Guide" which informs how to start using Suid and DansGuardian. (www.squid-cache.org)
[8] David Coulson "Mastering_iptables". This is a tutorial by Professional IPTABLES which shows how 2, 4's new iptables features can be used to keep out crackers. www.linux.format.co.uk
[9] CERT "Attack_trends". In this paper brief overview of recent trends that affect the ability of organizations to use Internet safely (http://www.cert.org/reports/)
[10] Professor Yeali S. Sun from National Taiwan Institute "Firewall-yeah" .This document mainly deals with network security management.
[11] Ian Main and Tony Gale "Gtk tutorial", is a tutorial which briefs about how to start making GUI in GTK.
[12] Philippe Troin "Glade reference manual" .It is the reference manual which deals with Glade features.
[13] http://www.google.co.in used for various purposes like finding various documentations and papers previously presented on Hyrid firewall.
[14] http://www.netfilter.org used while studying packet filtering .
[15] http://www.linux.org is a gamut of knowledge for working on Linus platform.
[16] http://www.kernels.org used for basic information about Linux kernels and their features.
[17] http://www.netnanny.com used while studying Netnanny firewall.
[18] http://www.privoxy.org used while studying Privoxy.
[19] http://firewall-jay.sourceforge.net used while studying Jay's firewall.
[20] http://www.simonzone.com/software/guarddog used while studyin Guarddog firewall.
[21] http://www.shorewall.net/ used while studying Shorewall firewall.
[22] http://www.cyberpatrol.com used while studying Cyberpatrol firewall.
[23] http://www.lib.ru/SECURITY/gauntlet.txt#14 used while studying Gauntlet firewall.
[24] http://dansguardian.org used mainly for acquiring information about data filtering and studying DansGuardian firewall.
[25] http://www.ipcop.org used while studying Ipcop firewall.
[26] http://www.smoothwall.org used while studying Smoothwall firewall