An Efficient Approach to Secure VPN based on Firewall using IPSec & IPtables

Mr. J. Mark Jain ME, MISTE ^{*1},N. Ayyamari Devi $^{\#2}$, K. Jancy Rani$^{\#3}$, A.Meenakshi $^{\#4}$, S.Balaselvam$^{\#5}$

^{*1} CSE Dept.,Dr,Sivanthi Aditanar College of Engineering
Tiruchendur,TamilNadu
$^{\#}$ Dr,Sivanthi Aditanar College of Engineering,
Tiruchendur,TamilNadu

Abstract— **Virtual Private Network is used to establish discrete network connection within a network. The traditional VPN has many limitations regarding the security issues. To overcome this, the secure VPN methodology is proposed which provide confidentiality, sender authentication and message integrity etc. In this proposed mechanism, the VPN network is established using SSL, SSH and IPSEC Tunneling protocols to provide efficient data transmission. SSL is designed to make use of TCP to provide reliable end to end secure service. The SSH will authenticate the remote login and allow it if necessary. IPSEC Tunneling protocol is used for transporting encrypted data over a public network. An efficient firewall is designed to allow or deny applications and network users and thus protect the VPN network from unauthorized users. This firewall protection is based on IPTABLES which is coded by us and managed through LINUX kernel. This design of secure VPN is more comfortable for transmitting the data in more reliable and secure way by using the above said protocols and protects data from external threats.**

Keywords— Ipsec,Iptables,ssh,ssl,firewall,vpn

I. Introduction

A virtual private network VPN[1] is a private network running on public network. It is the extension of a private network that incorporates links over public networks. It allows you to communicate among computers across a public network in a way that uses a private connection. Enterprises or organizations are adopting VPNs to ensure Internet security. Information transmitted on a VPN is either encrypted or plain text. VPNs offer various advantages such as reduced cost and scalability.

Virtual private networks can be a cost effective and secure way for different corporations to provide users access to the corporate network and for remote networks to communicate with each other across the Internet[3]. VPN connections are more cost-effective than dedicated private lines; usually a VPN involves two parts: the protected or "inside" network, which provides physical and administrative security to protect the transmission; and a less trustworthy, "outside" network or segment (usually through the Internet).

Virtual private networking has become more of a necessity than a luxury for business users who need a way to access files on an office network when they're on the road, working from home or otherwise physically separated from the network[2]. If the remote computer has Internet connectivity and the office network has a permanent connection to the Internet ,the most cost effective way for remote users to connect is by tunneling through the public network. VPN technologies use tunneling protocols to create the connection and encryption protocols to provide the "private" part, allowing you to securely access a VPN server [10]on the company network.

There are four popular VPN protocols in use, and each has advantages and disadvantages.

PPTP is an extension of the Internet standard Point-to-Point protocol (PPP), the link layer protocol used to transmit IP packets over serial links. PPTP uses the same types of authentication as PPP (PAP, SPAP, CHAP, MS-CHAP, EAP)[10]. PPTP establishes the tunnel but does not provide encryption. It is used in conjunction with the Microsoft Point-to-Point Encryption (MPPE) protocol to create a secure VPN. PPTP has relatively low overhead, making it faster than some other VPN methods.

L2TP requires the use of digital certificates. User authentication can be performed via the same PPP authentication mechanisms as PPTP, but L2TP[6] also provides computer authentication. This adds an extra level of security.

L2TP has several advantages over PPTP[10]. PPTP gives you data confidentiality, but L2TP goes further and also provides data integrity, authentication of origin, and replay protection. On the other hand, the overhead involved in providing this extra security can result in slightly slower performance than PPTP.

SSH[5]uses public-key cryptography to authenticate the remote computer and allow it to authenticate the user. Anyone can produce a matching pair of different keys (public and private). The public key is placed on all computers that must allow access to the owner of the matching private key. While authentication is based on the private key, the key itself is never transferred through the network during authentication. SSH only verifies if the same person offering the public key also owns the matching private key. Hence, in all versions of SSH it is important to verify unknown public keys. Accepting an attacker's public key without validation will authorize an unauthorized attacker as a valid user. SSH also supports password-based authentication that is encrypted by automatically generated keys.

II. Related Work

The Concept behind "An Efficient Approach to Secure VPN based on Firewall using IPSec & Iptables" is, VPN is build between selected systems by using security mechanisms such as SSH and IPSEC. After the implementation of the IPsec VPN all the data transfer happenings inbetween the systems will be sent through this IPsec Tunnel[1]. Then, while transmitting data through the FTP or HTTP protocols there is little differences performed in the Encryption and Authentication method.A second level of security such as

Firewall is build between the gateway within VPN's.Then any type of data sent or received within VPN and the incoming and the outgoing packets from other systems except VPN systems are also enter by means of Firewall only.

Most systems currently are open to attempts by outside users to gain unauthorized access by setting up an illegal connection, by pretending to be a valid user. To implement a firewall ,we provides a series of rules to govern what kind of access you want to allow on your system. Encryption protects transmissions originating from authorized remote users, and authentication verifies that a user requesting access has the right to do so.

Security parameter index(SPI)			
Sequence number			
Initialisation vector(IV)			
Data			
Data	**Padding**	**Padding length**	**Next header**
Hash message authentication code			

Here we use Packet filtering Firewall[4]. This can be implemented by using Netfilter. Netfilter is one of the Linux Kernel module. The Allowing and Denying of the packets can be decided only by the Netfilter module. Netfilter module which includes the packet filtering and NAT tasks[4]. This can be implemented by using the IPTABLES. Here we can update the user defined rules into the iptables. Packet filtering method is the process of deciding whether a packet received should be passed on into the network. Packet filtering mechanism checks the source and the destination addresses of the packet and accepts the packet if its allowed.

An additional task performed by firewall is NAT(Network address translation).It redirects packets to appropriate destination. It performs tasks such as redirecting packets to certain hosts, forwarding packets to other networks, and changing the host source of packets to implement IP Masquerading[9]. Rules are combined into different chains. The kernel uses chains to manage packets it sends and receives. A chain is simply a checklist of rules. A target be another chain of rules, even a chain of user-defined rules. A packet could be passed through several chains before finally reaching a target. Then, all the administration within VPN can be periodically monitor by various command and control like rlogin, rcp, scp, rsync, rsh & ssh[8].

III. System Development

The proposed system has three modules namely Ipsec,Firewall, Monitoring & Control.

A. Ipsec Module

Internet Protocol Security (IPsec) is a protocol suite for securing Internet Protocol(IP) communications by authenticating and encrypting each IP packet of a communication session and also provide secure exchange of encryption keys.This is principally designed to assist in the implementation of VPNs between hosts or networks. IPSec can itself be used as a tunneling protocol, and is in fact considered by many to be the "standard" VPN solution, especially for gateway-to-gateway[2].

The set of services offered by IPSEC which includes are:

Access control, Connectionless integrity, Protection against replays, Data origin Authentication, Confidentiality(Encryption), Limited flow traffic confidentiality[6].

IPsec uses the following mechanism to perform various functions:

Authentication Header(AH)provide connectionless integrity and data origin authentication for IP datagrams and provides protection against replay attacks[9].

Encapsulating Security Payloads(ESP) provide confidentiality data origin, authentication ,connectionless integrity, an anti-replay service (a form of partial sequence integrity), and limited traffic flow confidentiality.

Security Association

A security association is simply the bundle of algorithms and parameters used to encrypt and authenticate a particular flow in one direction. Security associations are established using Internet Security Association and key Management Protocol (ISAKMP)[6]. ISAKMP is implemented by manual configuration with pre-shared secrets, Internet key exchange(IKE).

IPsec can be implemented in a host-to-host transport mode, as well as in a network tunnel mode.

Transport mode

In transport mode, only the payload of the IP packet is usually encrypted or authenticated.

Tunnel mode

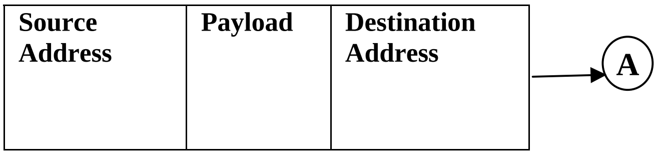
In tunnel mode, the entire IP packet is encrypted and/or authenticated. It is then encapsulated into a new IP packet with a new IP header. Tunnel mode is used to create VPN for network-to-network communications and host-to-host communications[7].

Next header	**Payload length**	**Reserved**
Security Parameter Index(SPI)		
Sequence number		
Hash message authentication code		

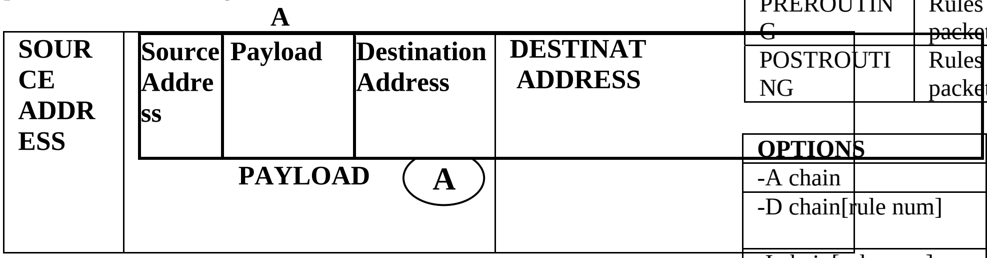
Module Explanation

In this module, the IPSEC can be configured to one system to another system using a host-to-host connection by using the Pre-shared key of key_value55. This type of connection uses the network to which each host is connected to create a secure tunnel between each host. A host-to-host IPsec connection is an encrypted connection between two systems, both running IPsec with the same authentication key. With the IPsec connection active, any network traffic and data transmissions between the two hosts is always encrypted[9].

Normal Data packet:



During the packet transmissions between Ipsec tunnel, the packets will be changed as follows:



In the receiver side the additional source address and destination address will be discarded and the original data packet will be taken into account .

B. *Firewall Module*

Firewall is a means of protecting a local system or network from network based threats. An efficient firewall is designed to allow or deny the network users and thus protect the VPN network from unauthorized users[4]. Firewall protection is based on IPTABLES which is coded and managed through LINUX kernel. The firewall allow or deny packet transmissions through it based on the configuration. Four techniques are used. They are the service control, direction control, user control, and behaviour control. The Service control determines the types of internet services. Then the Direct control determines the direction in which particular service request may be initiated. The User control, it can be performed controls access to a service according to which user is attempting to access it.

The list of targets in IPTABLES, are the Accept, Reject, Drop etc. The Accept will allow packet to pass through the firewall. The Drop will deny access by the packet. The Reject will deny access and notify the sender. The list of Netfilters built in chains, are the Input, Output, Forward, Prerouting, Postrouting etc. The Input sets rules for incoming packets. Output sets rules for outgoing packets. Forward sets rules for forwarded packets[4].

The Prerouting sets the rules for redirecting or modifying incoming packets. Postrouting sets rules for redirecting or modifying outgoing packets, NAT table only. One of the important features built on top of the Netfilter framework is connection tracking. Connection tracking allows the kernel to keep track of all logical network connections or sessions and iptables can use this information to act as a stateful firewall. The kernel uses chains to manage packets it sends and receives out. A chain is simply a checklist of rules. A target be another chain of rules, even a chain of user-defined rules. A packet could be passed through several chains before finally reaching a target.

TARGET	FUNCTIONS
ACCEPT	Allow packet to pass through.
DROP	Deny access by the packets.
REJECT	Deny access and notify the sender.

CHAIN	DESCRIPTION
INPUT	Rules for incoming packets.
OUTPUT	Rules for outgoing packets.
FORWARD	Rules for forward packets.
PREROUTING	Rules for redirecting incoming packets.
POSTROUTING	Rules for redirecting outgoing packets.

OPTIONS	FUNCTIONS
-A chain	Appends a rule to a chain.
-D chain[rule num]	Deletes matching from a chain.
-I chain[rule num]	Insert in chain as rule num(default 1=first).
-R chain[rule num]	Replaces rule rulenum(1=first)in chain.
-L[chain]	List rules in chains or all chains.
-F[chain]	Flushes all rules in chain or all chains.
-R[chain]	Replaces a rule; rule are numbered from 1.
-E[chain]	Renames a chain.
-P[chain] targets	Changes policy on chain to targets.

-s -->source address attached to packet.
-d -->destination address attached to packet.
-j -->target of the rule.
-i -->input device in Input, Forward chains.
-o-->output device in Output, Forward chains.

OPTIONS	FUNCTIONS
-p [!] protocol	Specifies a protocol TCP, UDP, ICMP.
-s [!] address [1mask][!] [port:[port]]	Specifies source address to match with port.
--sport [!] [port:[port]]	Specifies source port.
--dport [!] [port:[port]]	Specifies destination port.
--icmp-type [!] typename	Specifies ICMP type.
-j target[port]	Specifies the target for a rule.
-n	Specifies numeric output of address and ports
-m	Specifies a module to use, such as state.
--state	Specifies options for the state module

C. *Monitoring & Control Module*

The monitoring and controlling operations are used to improve the system performance. Controlling operations are more important. We can control other systems via remote access using RLOGIN, RCP, RSYNC, RSH & SSH[8].

1) ***RLOGIN*** The RLOGIN means remote login. Here we can access to any account in the network without the use of password. It can be used instead of telnet. For RLOGIN we need the ipaddress of the system and username of the system.

For e.g., rlogin client1.org (or) rlogin –l tom client1.org (or) rlogin client1.org –l tom

2) ***RCP*** RCP means remote copy. The RCP copies files or directories from one machine to another without using http, ftp, and telnet.. For RCP we need the ipaddress and username of the system. For instance, I can copy a file named sample.dat from the remote machine client1.org to my local machine client2.org. For e.g., rcp client1.org :sample.dat It also work the other way. Remote copy to client1.org: rcp sample.dat tom@client1.org:sample.dat

3) ***RSH*** Remote shell is the most powerful remote command, as it can be integrated into a pipe. The RSH generate data streams through several mechines accessing local disk and tape drivers For e.g., rsh client1.org 'ls-l' you should get a listing from your home directory on client1.org

4) ***SCP*** The SCP means secure copy. It is a better option than RCP because it uses encryption and authentication just like SSH program. Has to provide login and password of the target computer.

The linux SCP syntax to send file or directory to a remote computer,

SCP –r [/path/filename] [login name @ipaddress]:.

The linux SCP syntax to retrieve file or directory from a remote computers

SCP –r [login name @ ipaddress]:[/path/filename].

5) ***RSYNC*** RSYNC stands for remote sync. It is used to perform the backup operation in LINUX. RSYNC utility is used to synchronize the file. The RSYNC is established by using the syntax,

Rsync -zvr /home/source username/filename dest.username @dest.ip :/home/dest.ipaddress

In the syntax, the –z used to enable compression, -v used for verbose and –r indicates recursive.

6) ***SHH*** SSH is a very powerful remote command because it gives direct access to the command prompt.We can perform multitude of operations thereby.We can even shutdown one system via another system. SSH root @destination ipaddress

We need to enter the password for authentication and then the syntax for shutdown are follows

Shutdown –r 1 shut -r will force a restart

Shutdown –h

Shutdown –c -c “comment” will force a comment to appear Reboot

7) ***Monitoring*** In this monitoring section, the server system will monitor the entire administration. And also it supervise the entire client systems that are configured into the VPN network[7]. Within VPN network, the authentication, data transmission, connection to other networks and all other security issues are periodically watching by this module.

- **Ps** command will list every process associated with the specific user. This is helpful if you run into problems and need to kill a particular process that is stuck in memory.
  For eg, Ps –A | more Lists every process running on the server one screen at a time.
- **Nslookup** checks the domain name and IP information of a server. It is also a very useful security where you can lookup DNS information regarding a particular host IP
- **Netstat** simply summarises the network connections and status of sockets.
  For eg, Netstat –r This option gives the network routing addresses.
- **Traceroute** will traces the existing network routing for a remote or local server. This is a very powerful network that basically gives the exact route between your machine and a server.
  For eg, Traceroute hostname
- **Whois** allows you to check the internic database for proper host names.
  For eg, Whois –f 10.1.1.1 Replace the 10.1.1.1 with a specific server IP
- **Host** is a simple utility for performing DNS lookups.
- **Socklist** reports all existing sockets in the system.
- **Ifconfig** allows you to check and configure your server’s network cards, assigning IP, DNS and gateway addresses.
  For eg, Ifconfig eth0 10.1.1.1 (Replace to 10.1.1.1 with an actual IP address)
- **Ping** sends test packets to a specific server and checks if there is a response.
  For eg, Ping 10.1.1.1 (Replace 10.1.1.1 with the specific IP address of the server)

IV EXPERIMENTAL SETUP

A. *Ipsec Tunneling*

We need to configure a file named ifcfg-ipsec55 in system A and that file should be saved in etc/sysconfig/network-scripts. A unique name has to be given to identify the particular user connection . For System A, the IP address of System B – *130.0.1.8* need to be provided. For System B, the IP address of System A *130.0.1.9.*

For Authenticating both systems ,create another file named keys-ipsec55 which is having the content of Pre-shared key and save it in etc/sysconfig/network-scripts. The contents of this file should be identical on both systems A and B, and only the root user should be able to read or write this file[10].

Ifcfg-ipsec55:

```
DST=X.X.X.XTYPE=IPSEC
ONBOOT=no
IKE_METHOD=PSK
```

Keys-ipsec55:

```
IKE_PSK=Key_Value55
```

Files named as psk.txt, x.x.x.x.conf and racoon.conf will be automatically generated and located in etc/raccoon/. The /etc/racoon/racoon.conf files should be identical on all IPsec nodes *except* for the include"/etc/racoon/*X.X.X.X*.conf" statement. This statement is generated when the IPsec tunnel is activated. For System A, the *X.X.X.X* in the include statement is System B's IP address. The opposite is true of System B.

The following shows a typical x.x.x.x.conf file when the IPsec connection is activated.

remote ***X.X.X.X***{ exchange_mode aggressive, main;
 my_identifier address; proposal
 { encryption_algorithm des;
 hash_algorithm sha1;
 authentication_method pre_shared_key; dh_group 2
 ; }}

- *remote X.X.X.X:*Specifies that the file apply only to the remote node - *X.X.X.X* IP address.
- *exchange_mode aggressive:*Lowers the connection overhead
- *my_identifier address:* Specifies the identification method to use.
- *encryption_algorithm 3des:*Specifies the encryption cipher used.Default is 3DES.
- *hash_algorithm sha1;*Specifies the hash algorithm used. By default, SHA-1
- *authen__method pre_shared_key:*Specifies the authentication used : pre-shared keys.
- *dh_group 2:*Specifies the Diffie-Hellman group number for dynamically-generated session keys.

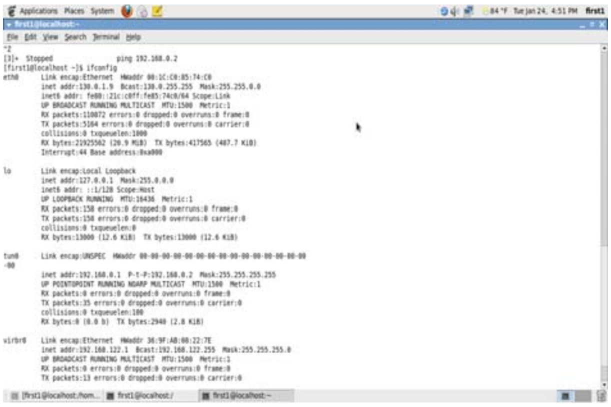
The following shows a typical racoon.conf file when the IPsec connection is activated it will be generated automatically.

racoon.conf

# Racoon IKE daemon configuration file.
path include "/etc/racoon";
path pre_shared_key"/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";
sainfo anonymous
{ pfs_group 2;lifetime time 1 hour ;
 encryption_algorithm 3des,blowfish 448, rijndael ;
 authentication_algorithm mac_sha1, hmac_md5 ;
 compression_algorithm deflate ;}
include"/etc/racoon/X.X.X.X.conf";

- *sainfo anonymous:*SA can anonymously initialize with peer that credentials match.
- *pfs_group 2:*Defines the Diffie-Hellman key exchange protocol.
- *lifetime time 1 hour:*Specifies the lifetime of an SA by default, a one hour lifetime.
- *encryption_algorithm 3des, blowfish 448, rijndael:*3DES, 448-bit Blowfish,Rijndael .
- *authentication_algorithm hmac_sha1, hmac_md5:*sha1 and md5 for auth_codes.
- *compression_algorithm deflate:*Defines Deflate compression for faster transmission

To start the connection, use the following command on each System **# /sbin/ifup ipsec55.**To test the IPsec connection, run the tcpdump utility to view the network packets being transferred between the hosts and verify that they are encrypted via IPsec[1]. Tcpdump will prints out the header of the packets on a network interface that match the boolean expression .**# tcpdump -n -i eth0 host ipsec55.** Packet transfer can be monitored during HTTP access and FTP file transfer, through the tunnel in an encrypted manner. This can be acknowledged by seeing the AH and ESP parameter, which proves IPSEC tunnel mechanism.



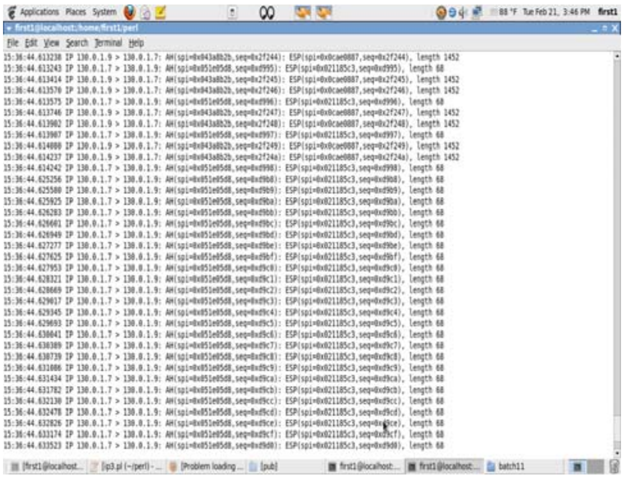
Applications Places System
84 °F Tue Jan 24, 4:51 PM first1
first1@localhost:~
File Edit View Search Terminal Help
[first1@localhost:/hom... first1@localhost:/ first1@localhost:~

```
^Z
[3]+  Stopped                 ping 192.168.0.2
[first1@localhost ~]$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1C:C0:85:74:C0
          inet addr:130.0.1.9  Bcast:130.0.255.255  Mask:255.255.0.0
          inet6 addr: fe80::21c:c0ff:fe85:74c0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:110072 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5164 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:21925502 (20.9 MiB)  TX bytes:417565 (407.7 KiB)
          Interrupt:44 Base address:0xa000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:158 errors:0 dropped:0 overruns:0 frame:0
          TX packets:158 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:13000 (12.6 KiB)  TX bytes:13000 (12.6 KiB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
-00
          inet addr:192.168.0.1  P-t-P:192.168.0.2  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:35 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b)  TX bytes:2940 (2.8 KiB)

virbr0    Link encap:Ethernet  HWaddr 36:9F:AB:00:22:7E
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
```

Fig-4.1(a): Tunnel Established on both ends

B. *Firewall Configuration*

# Firewall configuration
*nat
:PREROUTING ACCEPT [0:0];:OUTPUT ACCEPT [0:0];:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -o eth0 -j MASQUERADE
-A POSTROUTING -o tun+ -j MASQUERADE

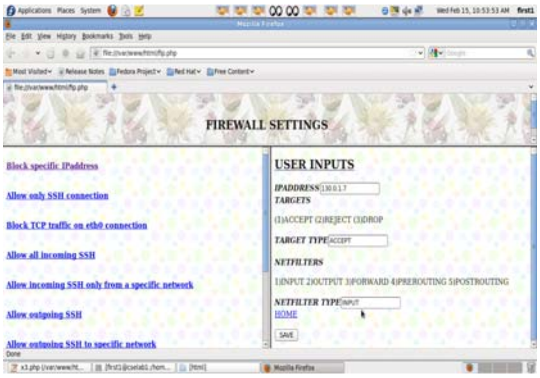


Applications Places System
88 °F Tue Feb 21, 3:46 PM first1
first1@localhost:/home/first1/perl
File Edit View Search Terminal Help
[first1@localhost... ip3.pl (~/perl) - ... [Problem loading ... [pub] first1@localhost:... first1@localhost:... batch11

```
15:36:44.613238 IP 130.0.1.9 > 130.0.1.7: AH(spi=0x043a8b2b,seq=0x2f244): ESP(spi=0x0cae0887,seq=0x2f244), length 1452
15:36:44.613243 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd995): ESP(spi=0x021185c3,seq=0xd995), length 68
15:36:44.613414 IP 130.0.1.9 > 130.0.1.7: AH(spi=0x043a8b2b,seq=0x2f245): ESP(spi=0x0cae0887,seq=0x2f245), length 1452
15:36:44.613570 IP 130.0.1.9 > 130.0.1.7: AH(spi=0x043a8b2b,seq=0x2f246): ESP(spi=0x0cae0887,seq=0x2f246), length 1452
15:36:44.613575 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd996): ESP(spi=0x021185c3,seq=0xd996), length 68
15:36:44.613746 IP 130.0.1.9 > 130.0.1.7: AH(spi=0x043a8b2b,seq=0x2f247): ESP(spi=0x0cae0887,seq=0x2f247), length 1452
15:36:44.613902 IP 130.0.1.9 > 130.0.1.7: AH(spi=0x043a8b2b,seq=0x2f248): ESP(spi=0x0cae0887,seq=0x2f248), length 1452
15:36:44.613907 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd997): ESP(spi=0x021185c3,seq=0xd997), length 68
15:36:44.614080 IP 130.0.1.9 > 130.0.1.7: AH(spi=0x043a8b2b,seq=0x2f249): ESP(spi=0x0cae0887,seq=0x2f249), length 1452
15:36:44.614237 IP 130.0.1.9 > 130.0.1.7: AH(spi=0x043a8b2b,seq=0x2f24a): ESP(spi=0x0cae0887,seq=0x2f24a), length 1452
15:36:44.614242 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd998): ESP(spi=0x021185c3,seq=0xd998), length 68
15:36:44.625256 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9b8): ESP(spi=0x021185c3,seq=0xd9b8), length 68
15:36:44.625580 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9b9): ESP(spi=0x021185c3,seq=0xd9b9), length 68
15:36:44.625925 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9ba): ESP(spi=0x021185c3,seq=0xd9ba), length 68
15:36:44.626283 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9bb): ESP(spi=0x021185c3,seq=0xd9bb), length 68
15:36:44.626601 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9bc): ESP(spi=0x021185c3,seq=0xd9bc), length 68
15:36:44.626949 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9bd): ESP(spi=0x021185c3,seq=0xd9bd), length 68
15:36:44.627277 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9be): ESP(spi=0x021185c3,seq=0xd9be), length 68
15:36:44.627625 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9bf): ESP(spi=0x021185c3,seq=0xd9bf), length 68
15:36:44.627953 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9c0): ESP(spi=0x021185c3,seq=0xd9c0), length 68
15:36:44.628321 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9c1): ESP(spi=0x021185c3,seq=0xd9c1), length 68
15:36:44.628669 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9c2): ESP(spi=0x021185c3,seq=0xd9c2), length 68
15:36:44.629017 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9c3): ESP(spi=0x021185c3,seq=0xd9c3), length 68
15:36:44.629345 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9c4): ESP(spi=0x021185c3,seq=0xd9c4), length 68
15:36:44.629693 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9c5): ESP(spi=0x021185c3,seq=0xd9c5), length 68
15:36:44.630041 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9c6): ESP(spi=0x021185c3,seq=0xd9c6), length 68
15:36:44.630389 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9c7): ESP(spi=0x021185c3,seq=0xd9c7), length 68
15:36:44.630739 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9c8): ESP(spi=0x021185c3,seq=0xd9c8), length 68
15:36:44.631086 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9c9): ESP(spi=0x021185c3,seq=0xd9c9), length 68
15:36:44.631434 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9ca): ESP(spi=0x021185c3,seq=0xd9ca), length 68
15:36:44.631782 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9cb): ESP(spi=0x021185c3,seq=0xd9cb), length 68
15:36:44.632130 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9cc): ESP(spi=0x021185c3,seq=0xd9cc), length 68
15:36:44.632478 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9cd): ESP(spi=0x021185c3,seq=0xd9cd), length 68
15:36:44.632826 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9ce): ESP(spi=0x021185c3,seq=0xd9ce), length 68
15:36:44.633174 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9cf): ESP(spi=0x021185c3,seq=0xd9cf), length 68
15:36:44.633523 IP 130.0.1.7 > 130.0.1.9: AH(spi=0x051e05d8,seq=0xd9d0): ESP(spi=0x021185c3,seq=0xd9d0), length 68
```

Fig-4.1(b) : Data transmission through Ipsec tunnel

-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i eth0 -j ACCEPT
-A INPUT -i ppp+ -j ACCEPT
-A INPUT -i tun+ -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 21 -j ACCEPT
-A INPUT -p ah -j ACCEPT
-A INPUT -p esp -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 1194 -j ACCEPT
-A FORWARD -p icmp -j ACCEPT
-A FORWARD -i eth0 -j ACCEPT
-A FORWARD -i tun+ -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited;
COMMIT



File Edit View History Bookmarks Tools Help
file:///var/www/html/fp.php
Most Visited
Release Notes
Fedora Project
Red Hat
Free Content

FIREWALL SETTINGS

Block specific IPaddress

Allow only SSH connection

Block TCP traffic on eth0 connection

Allow all incoming SSH

Allow incoming SSH only from a specific network

Allow outgoing SSH

Allow outgoing SSH to specific network

USER INPUTS

IPADDRESS 130.0.1.7

TARGETS

(1)ACCEPT (2)REJECT (3)DROP

TARGET TYPE ACCEPT

NETFILTERS

1)INPUT 2)OUTPUT 3)FORWARD 4)PREROUTING 5)POSTROUTING

NETFILTER TYPE INPUT

HOME

SAVE

Fig -4.b: Firewall using Iptables

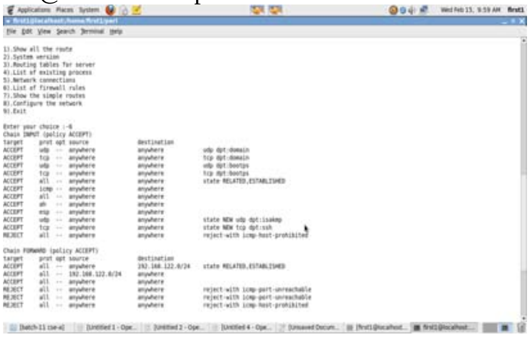
C. *Monitoring and Controlling*

1) Monitoring

Ps –A | more ; Netstat –r ;
Traceroute hostname; Whois –f 10.1.1.1 ;
Ifconfig eth0 10.1.1.1 ;
Ping 10.1.1.1

2) Controlling

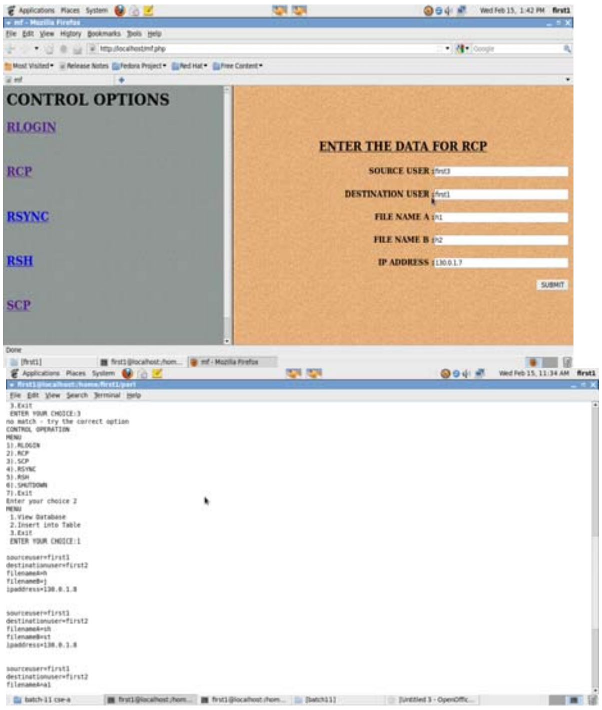
rlogin client1.org ; rcp client1.org :sample.dat
rsh client1.org 'ls-l'
SCP –r [/path/filename] [login name @ipaddress]:.
SSH root @destination ipaddress



Applications Places System
first1@localhost:/home/first1/perl
File Edit View Search Terminal Help

```
1).Show all the route
2).System version
3).Routing tables for server
4).List of existing process
5).Network connections
6).List of firewall rules
7).Show the simple routes
8).Configure the network
9).Exit

Enter your choice :-6
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     udp  --  anywhere             anywhere            udp dpt:domain
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:domain
ACCEPT     udp  --  anywhere             anywhere            udp dpt:bootps
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:bootps
ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     ah   --  anywhere             anywhere
ACCEPT     esp  --  anywhere             anywhere
ACCEPT     udp  --  anywhere             anywhere            state NEW udp dpt:isakmp
ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             192.168.122.0/24    state RELATED,ESTABLISHED
ACCEPT     all  --  192.168.122.0/24     anywhere
ACCEPT     all  --  anywhere             anywhere
REJECT     all  --  anywhere             anywhere            reject-with icmp-port-unreachable
REJECT     all  --  anywhere             anywhere            reject-with icmp-port-unreachable
REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited
```

Fig -4.c(i): VPN Monitoring Application



Applications Places System
mf - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/mf.php

CONTROL OPTIONS

RLOGIN

RCP

RSYNC

RSH

SCP

ENTER THE DATA FOR RCP

SOURCE USER : first3

DESTINATION USER : first1

FILE NAME A : h1

FILE NAME B : h2

IP ADDRESS : 130.0.1.7

SUBMIT

Done
Applications Places System
first1@localhost:/home/first1/perl
File Edit View Search Terminal Help

```
 3.Exit
 ENTER YOUR CHOICE:3
no match - try the correct option
CONTROL OPERATION
MENU
1).RLOGIN
2).RCP
3).SCP
4).RSYNC
5).RSH
6).SHUTDOWN
7).Exit
Enter your choice 2
MENU
 1.View Database
 2.Insert into Table
 3.Exit
 ENTER YOUR CHOICE:1

sourceuser=first1
destinationuser=first2
filenameA=h
filenameB=j
ipaddress=130.0.1.8


sourceuser=first1
destinationuser=first2
filenameA=ih
filenameB=st
ipaddress=130.0.1.8


sourceuser=first1
destinationuser=first2
filenameA=a1
```

Fig -4.c(ii): VPN Controlling Application

V. SYSTEM ANALYSIS

A. *Security Aspect*

Consider, Vpn 1 - Normal VPN . Vpn 2 - Secure VPN using Ipsec & Iptables

After capturing and analysing the data collected through Ipsec tunnel(VPN-2), we compared it with ordinary VPN data transmission(VPN-1) and plotted the graph. This graph describes the differences, the x-axis indicates the packet sent rate with respect to time and the y-axis which will indicates the performance of the systems. The major question arise in normal VPN is whether the data is secure or not. Because the normal VPN has no encryption and authentication. Due to the absence of authentication and encryption the unauthorized users can easily steal or hack the data in the systems during transmissions[9].

The two security drawbacks in normal VPN will be overcome by the secure VPN using IPSEC & IPTABLES. Here for the encryption and authentication mechanism we uses the MD5,SHA-1 and AES algorithms[5]. So the data packets transferred between the secure VPN systems will be initially authenticate and encrypted then only the packets must be sent. By using the tcpdump the data packets transferring between the secure VPN systems are confirmed to be very much safe and secure.

B. *Data Integrity*

In normal VPN, we does not check the data integrity. So, while transferring the data packets using normal VPN, a time delay will be happened and sometimes the data packets will be dropped or modified intentionally[6].But in secure VPN, there is no means the data can be modified, because of the encryption and authentication algorithms being used by Ipsec tunnel along with Iptable rules. This graph describes the differences, the x-axis indicates the packet sent rate with respect to time and the y-axis which will indicates the performance of the systems.

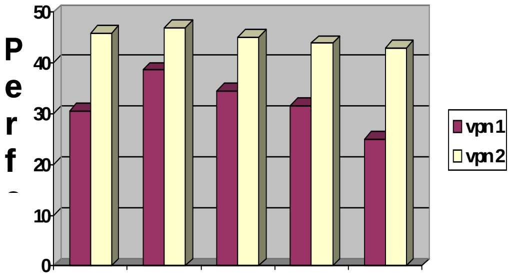


Fig -5.a Performance Analysis

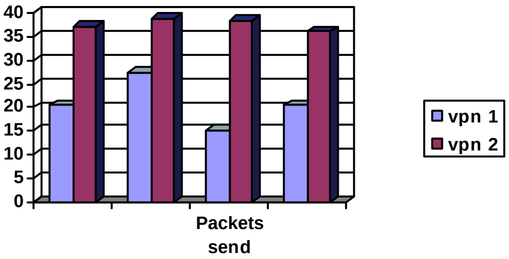


Fig -5.b Data Integrity Analysis

VI. Conclusion

By using the security mechanisms such as SSL, SSH, IPSEC Tunnelling protocol, the unauthorized users cannot misuse the VPN since we are denying the access to our secure VPN. Also, an efficient firewall which provides overall security to the entire VPN network based on the set of rules described using Iptables. Only the authenticated users can access our VPN network for transferring encrypted information and resource sharing through secure transmission channel.

Acknowledgment

We thank our management for providing the infrastructure and resources to fulfill this work that we undergo in the area of Network Security.

References

[1]Byeong Ho kong and Maricel O.Balitanas "Vulnerabilities Of VPN using IPSEC and Defensive Measures" International Journal of Advanced Science and Technology, July 2009.

[2] Suresh Limkar and Dhiren Patel "Geographically Secured SSL-VPN Using GPS" International Journal of Computer Applications, 2010.

[3] H. Bourdoucen, A. Al Naamany and A. Al Kalbani "Impact of Implementing VPN to Secure Wireless LAN" International Journal of Computer and Information Engineering , 2009.

[4]Bhisham Sharma,Karan Bajaj," Packet Filtering using IP Tables in Linux" International Journal of Computer Science Issues, July 2011.

[5] S.R.M.Krishna,Paradeep singh jamwal,K.padma priya,P.Hema Vishnu, "Enhancing the Communication Channel Through Secure Shell And Irrational DES"International Journal on Computer Science and Engineering (IJCSE), Mar 2011.

[6]Steven Noel, Sushil Jajodia, Lingyu Wang, Anoop singhal,"Measuring Security Risk of Networks Using Attack Graphs" International Journal of Next-Generation Computing, July 2010.

[7] Jingli Zhou, Hongtao Xia, Xiaofeng Wang, and Jifeng Yu, "A New VPN Solution Based on Asymmetrical SSL Tunnels", Proceedings of the Japan-China Joint Workshop on Frontier of Computer Science and Technology (FCST'06), Nov.2006.

[8] Sea Shuan Luo, "A Comparative Study about Linux Laboratory Environment", International Journal of Cyber Society and Education, Dec 2009.

[9] Rajeswari Goudar,Pournima more, "Multilayer Security Mechanism in Computer Networks", International journal of scientific and research publications, Jan 2012.

[10] T.MohanRaj, S.Shahul Hammed, A.Amala Deepan, " Enhancing Security Measures by Tunnelling Protocol in Distributed Grid Network ", International Journal of Computer Application, January 2012 .

Author's Bibliography:

A.Meenakshi, S.Balaselvam, K.Jancy Rani, N.Ayyamari Devi are B.E CSE final year students of Dr.Sivanthi Aditanar College of Engineering,Tiruchendur. Their area of interest is **"Network Security".** They have published two papers in national conferences in the area of Network Security

Mr. J. Mark Jain has completed his B.E degree from Dr.Sivanthi Aditanar College of Engineering,in 2002.He has completed his M.E degree in 2009 from the same college. His research interests include Network Security & Network Forensics. He is a life member of ISTE. He is currently doing R & D funded project. He has published many National and International conferences and journals.