Secure Communication Process between Two Versions of Internet Protocols

Sandeep Shiravale
Department of Information Technology
MIT Academy of Engineering
Alandi (D),Pune 411052

Sumedha Sirsikar
Department of Information Technology
Maharashtra Institute of Technology Pune 411038,
India

ABSTRACT

The Network performance optimization, configuration depends on proper design of complete network. The network layer is concerned with getting packets from source all the way to destination. The packets may cross many networks to reach the destination. To achieve its goals, the network layer must know about the topology, communication subnets and the version of the IP addresses used. Considering the present scenario, we are using IP v4 which needs to communicate with IP v6.

The present networks are using only IP version 4 it is necessary to design and deploy certain methods which will help in communicating with the future generation of IP addresses. The present applications are still using IP version 4 to communicate and hence it is necessary to securely migrate them with the latest IP version. We present a secure tunneling mechanism that has some advantages over the other models. The model discussed here can be used to establish hybrid communications between the two different versions of Internet protocol in both ways.

General Terms

IPv4, IPv6, Intrusion Detection System, KDD dataset, SVM Classifier.

1. INTRODUCTION

The present networks are using IP version 4 to communicate with the other systems across the globe. The last batch of IP version 4 was released in February 2011 by IANA and therefore it is necessary to migrate to the new version of IP address. Further, it is also necessary that both the versions of IP addresses must co-exist. The major difference between the two addresses is their packet header. There are different incompatibilities as follows:

1) Network host to router incompatibility.

2) Network router to another network router incompatibility.

3) Network to network incompatibility.

The design process must consider the mentioned incompatibilities. There is common transition mechanisms exist in the market. 1) Dual Stack: The network nodes are stacked to both the versions of protocols. The drawback of this method is all the network components must maintain dual stack which makes them less efficient. 2) Protocol Translation: This mechanism eliminates the problems caused due to dual stack mechanism. A complete IP v6 network can communicate with IP v4 nodes with the concept of translation. This concept is divided again further. It involves again dual stack and Network Address Translator which increases the complexity and we need to analyze different networks before implementing this concept. 3) Tunneling: IP v6 traffic is carried further using the current IP v4 infrastructure. Though this technology provides accurate results, it is somehow manual process and may not provide necessary security during transition phase.

The future networks should communicate using an IPv4 infrastructure, an IPv6 infrastructure, or an infrastructure that is a combination of IPv4 and IPv6. Here we discuss a mechanism which is different compared with the existing technologies. We design a secure tunneling mechanism that can transfer IP v4 or IP v6 packets to any network using the existing infrastructure.

2. SCOPE OF WORK

IPv6 over IPv4 tunneling is the encapsulation of IPv6 packets with an IPv4 header so that IPv6 packets can be sent over an IPv4 infrastructure.

Following are the phases in which we can implement the required solution.

1) Test bed development: To develop test bed for creating a tunnel which will have 2 IP v6 networks connected with IP v4 network via 6 to 4 routers.
2) Set up communication: To send packets from an IP v6 network to another IP v6 network through IP v4 to IP v4 network with the help of tunnel.
3) Intrusion Detection System: The system will filter out unwanted traffic thus providing proper security. [1] [7]

The various phases are as follows:

2. Developing a test bed: The tunnel has been created in between router1 and client1. Further it is necessary to create it between router1 to router2.

3. Packet Capture: The traffic flowing in between router1 and router 2 is tunneled encapsulated traffic, which is necessary to diagnose.

4. De capsulation: After capturing the packets we have to de capsulate the IP v6 packets which are encapsulated in IPv4 packet.

5. Diagnosis for maliciousness: (IDS)In this phase we will extract the features of de capsulated packet and we will decide whether the packets are suspicious or normal. Log of the captured and diagnosed packets should be created. [2]

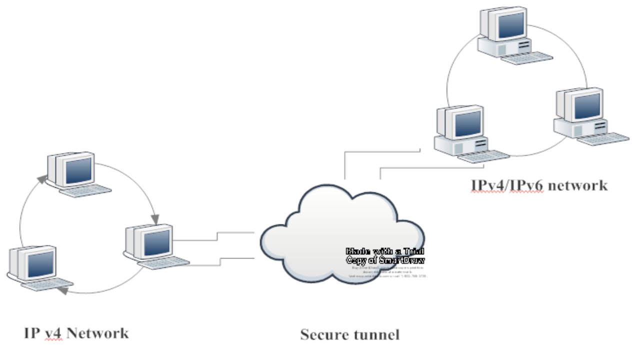


Figure 1: Test network

3. DESIGN PROCESS

The sample system architecture is shown in the diagram as below. The two systems which are equipped with Windows 2003 server operating System will perform various functions. The Active Directory is implemented on both the servers so that we can enable “Routing and Remote Access” functionality. Further, we can integrate Domain Name Server and Dynamic Host Configuration Protocol as well.

The Intrusion Detection System is implemented on both the servers. We have used Java (JDK) and Net beans 7.1 to configure IDS. The two servers will function as Router Server, IDS Server and so on. The packets are routed through the router server and the IDS will check if there are any intrusions. The design of IDS employs swing class with which various modules have been integrated. The clients are running windows XP operating system on which we have installed Jfree chat application. The clients can communicate with the chat server and route the packets to the other destination systems with the help of the stated application.

Java is a programming language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible.

Net Beans refers to both a platform framework for Java desktop applications, and an integrated development environment (IDE) for developing with Java, JavaScript, PHP and others. The Net Beans Platform is a reusable framework for simplifying the development of Java Swing desktop applications. The Net Beans IDE bundle for Java contains what is needed to start developing Net Beans plug-in and Net Beans Platform based applications. [2]-[4]

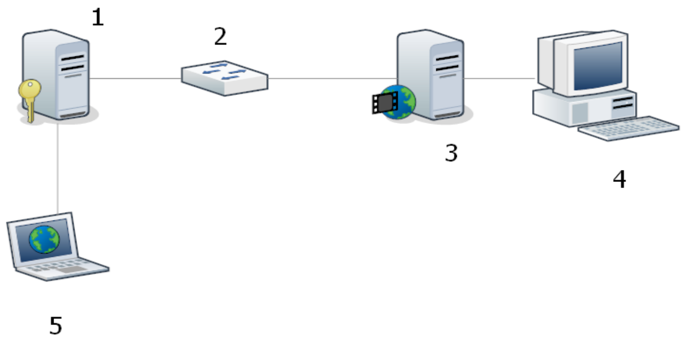


Figure 2: System Architecture

The components of the architecture are described as below:

1: Secure communication Server

2: Switch

3: Chat Server

4: Sender-Client

5: Receiver-Client

4. HIGH LEVEL DESIGNS

4.1. Intrusion Detection System

Intrusion detection is a classification task that attempts to discern if a given request for network service is an intrusion attempt or a safe request to the server. Algorithms for Intrusion Detection have been classically categorized into two main techniques: Anomaly Detection and Misuse Detection. A useful combination of these techniques uses three-tier architecture for their intrusion detection system. They are follows.

1) Anomaly Detection mechanism,

2) Misuse Detection mechanism,

3) Multi-class SVM that classifies the detected intrusion into one of the known attack types. [5] [8]

Intrusion detection systems currently in use typically require human input to create attack signatures or to determine effective models for normal behavior. Support for learning algorithms provides a potential alternative to expensive human input. The main task of such a learning algorithm is to discover appropriate models from the training data for characterizing normal and attack behavior. The ensuing model is then used to make predictions regarding unseen data.

4.2. KDD 99 Intrusion Detection data set

Knowledge Discovery Dataset (KDD) 99 intrusion detection datasets, which are based on DARPA 98 dataset, provides labeled data for researchers working in the field of intrusion detection and is the only labeled dataset publicly available. The KDD 99 intrusion detection benchmark consists of three components, which are listed in Table 5.1. In the International Knowledge Discovery and Data Mining Tools Competition, only “10% KDD” dataset is employed for the purpose of training. This dataset contains 22 attack types and is a more concise version of the “Whole KDD” dataset. It contains more examples of attacks than normal connections and the attack types are not represented equally.

Table 1: BASIC CHARACTERISTIC OF THE KDD 99 INTRUSION DETECTION DATASET IN TERMS OF NUMBER OF SAMPLES

Dataset	DoS	Probe	U2r	R2l	Normal
“10% KDD”	391458	4107	52	1126	97277
“Corrected KDD”	229853	4166	70	16347	60593
“Whole KDD”	3883370	41102	52	1126	972780

4.3. Support Vector Machines Classifier

Support Vector Machines (SVM) was developed to solve classification problem but they have been extended to the other domains as well. A support vector machine (SVM) is a concept used for classification and regression analysis. SVM classifies with support vector methods and support vector regression is used to describe regression.

SVM selects certain parameters from the selected dataset and provides us with the necessary predictions. The NCC KDD dataset used in the project comprises of 41 features. We have selected certain features like duration, protocol type, service, flag, source bytes, destination bytes, land, wrong fragment, type of attack and so on. [6] [9] [11]

The dataset with the above mentioned features is trained. The accuracy of the classification should be high nearing 90 percent. The said accurate dataset is further utilized for the communication between the clients using both the versions of internet protocol.

5. RESULTS OF THE DESIGNED SYSTEM

The system is implemented using a secure tunnel for communication. The design is divided into three main categories, project, files and services. The Intrusion Detection System is configured on the server and it will help in encapsulating IPversion6 with IPversion4.

The Intrusion Detection System comprises of different modules like IDS, packet, KDD protocol. It is configured using Net beans IDE 7.0 and it works well on most windows environment. The IDS once started on the server will provide Graphical Interface.

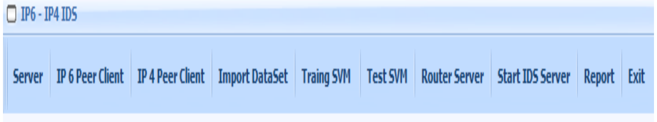


Figure 3: GUI of Intrusion Detection System

Figure 3 drawn above informs about the GUI (Graphical User Interface). Server is the chat server that is generally implemented on the router server. The clients can run on IP version 4 or version 6. IDS server is configured on the server end and it will detect any kind of attacks or threats. The various options available in IDS are as follows.

Server: This is TCP chat server used for communicating with the clients. The said server is started on one of the server operating system.

IP v6 peer clients: The clients running Internet Protocol version 6.

IP v4 peer clients: The clients running Internet Protocol version 4.

Import Dataset: KDD 99 intrusion detection datasets used for detecting and preventing any kind of intrusions.

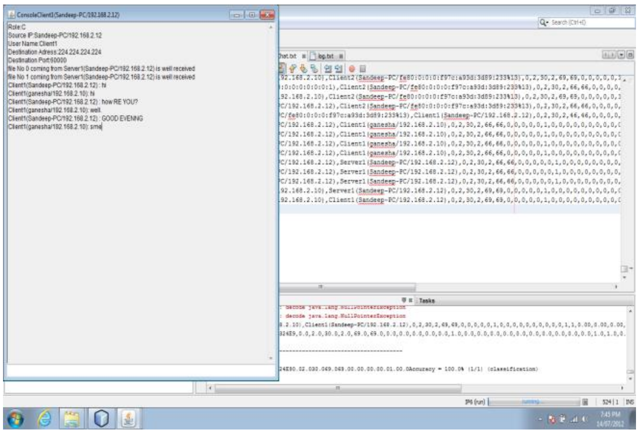
Training SVM: A support vector machine (SVM) is a concept in statistics and computer science for a set of related supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis.

Test SVM: Used to test and analyze the dataset.

Router Server: It is used to route the packets between end networks.

IDS Server: Intrusion Detection System. It is used to detect any kind of intrusion.

Fig 5 indicates the secure communication process between two clients and figure 6 informs about the features present in KDD dataset.



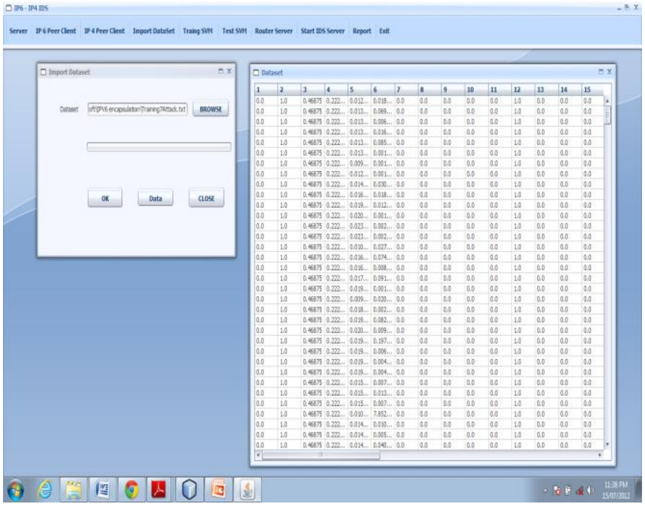
ConsoleClient1(Sandeep-PC/192.168.2.12)

```
Role:C
Source IP:Sandeep-PC/192.168.2.12
User Name:Client1
Destination Adress:224.224.224.224
Destination Port:60000
file No 0 coming from Server1(Sandeep-PC/192.168.2.12) is well received
file No 1 coming from Server1(Sandeep-PC/192.168.2.12) is well received
Client1(Sandeep-PC/192.168.2.12) : hi
Client1(ganesha/192.168.2.10): hi
Client1(Sandeep-PC/192.168.2.12) : how RE YOU?
Client1(ganesha/192.168.2.10): well
Client1(Sandeep-PC/192.168.2.12) : GOOD EVENING
Client1(ganesha/192.168.2.10): sme
```

```
92.168.2.10),Client2(Sandeep-PC/fe80:0:0:0:f97c:a93d:3d89:233%13),0,2,30,2,69,69,0,0,0,0,0,1
):0:0:0:0:0:0:1),Client2(Sandeep-PC/fe80:0:0:0:f97c:a93d:3d89:233%13),0,2,30,2,66,66,0,0,0,0,
92.168.2.10),Client2(Sandeep-PC/fe80:0:0:0:f97c:a93d:3d89:233%13),0,2,30,2,69,69,0,0,0,0,0,1
PC/192.168.2.12),Client2(Sandeep-PC/fe80:0:0:0:f97c:a93d:3d89:233%13),0,2,30,2,66,66,0,0,0,0,
PC/fe80:0:0:0:f97c:a93d:3d89:233%13),Client1(Sandeep-PC/192.168.2.12),0,2,30,2,46,46,0,0,0,0,
PC/192.168.2.12),Client1(ganesha/192.168.2.10),0,2,30,2,66,66,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0
PC/192.168.2.12),Client1(ganesha/192.168.2.10),0,2,30,2,66,66,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0
PC/192.168.2.12),Client1(ganesha/192.168.2.10),0,2,30,2,66,66,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0
PC/192.168.2.12),Client1(ganesha/192.168.2.10),0,2,30,2,66,66,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0
PC/192.168.2.12),Server1(Sandeep-PC/192.168.2.12),0,2,30,2,66,66,0,0,0,0,0,1,0,0,0,0,0,0,0,0,
PC/192.168.2.12),Server1(Sandeep-PC/192.168.2.12),0,2,30,2,66,66,0,0,0,0,0,1,0,0,0,0,0,0,0,0,
PC/192.168.2.12),Server1(Sandeep-PC/192.168.2.12),0,2,30,2,66,66,0,0,0,0,0,1,0,0,0,0,0,0,0,0,
92.168.2.10),Server1(Sandeep-PC/192.168.2.12),0,2,30,2,69,69,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0
92.168.2.10),Client1(Sandeep-PC/192.168.2.12),0,2,30,2,69,69,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0
```

```
: decode java.lang.NullPointerException
: decode java.lang.NullPointerException
8.2.10),Client1(Sandeep-PC/192.168.2.12),0,2,30,2,69,69,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,
324E9,0.0,2.0,30.0,2.0,69.0,69.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,1.0,0.
-----------------------------------
24E90.02.030.069.069.00.00.00.00.01.00.0Accuracy = 100.0% (1/1) (classification)
```

Figure 4: Secure communications using a tunnel



Dataset: off\IPV6 encapsulation\Training7Attack.txt

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.0	1.0	0.46875	0.222...	0.012...	0.018...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.013...	0.069...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.013...	0.006...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.013...	0.016...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.013...	0.085...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.013...	0.001...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.009...	0.001...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.012...	0.001...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.014...	0.030...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.016...	0.018...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.019...	0.012...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.020...	0.001...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.023...	0.002...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.023...	0.002...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.010...	0.027...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.016...	0.074...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.016...	0.008...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.017...	0.091...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.019...	0.001...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.009...	0.020...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.018...	0.002...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.019...	0.082...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.020...	0.009...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.019...	0.197...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.019...	0.006...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.019...	0.004...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.019...	0.004...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.015...	0.007...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.015...	0.013...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.015...	0.007...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.010...	7.852...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.014...	0.010...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.014...	0.005...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	1.0	0.46875	0.222...	0.014...	0.040...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0

Figure 5: Forty one features of the dataset

The table below describes different test cases for IDS server. The malicious packets are always blocked whereas normal packets are allowed.

Table 2: TEST CASE TABLE FOR IDS SERVER

Sr.No.	**Test steps**	**Test data**	**Expected result**	**Actual result**	**Pass/Fail**
1	IDS Server	Malicious packet	Block the packet	Block the packet	Pass
2	IDS Server	Normal packet	Allow the packet	Allow the packet	Pass

Table 3 describes various kinds of attacks. The attacks are chosen from KDD dataset. Most attacks are blocked thus the design of the system is verified.

Table 3: TEST CASE FOR BLOCKING THE ATTACK

Sr. No.	**Test steps**	**Test data**	**Expected result**	**Actual result**	**Pass/Fail**
1	Normal Attack	KDD dataset	Block normal attack	Block normal attack	Pass
2	Smurf Attack	KDD dataset	Block smurf attack	Block smurf attack	Pass
3	Multihop Attack	KDD dataset	Block Multihop attack	Block Multihop attack	Pass
4	Neptune Attack	KDD dataset	Block Neptune attack	Block Neptune attack	Pass
5	Portsweep Attack	KDD dataset	Block Portsweep attack	Block Portsweep attack	Pass

6. CONSTRAINTS

Transitioning technologies can poses security threats. The security related products (firewall/IDS/IPS) needs to be configured to inspect IPv6 packets in depth thus cannot allow malicious packets to pass through.

Throughput values for various operating systems with the two transition mechanisms must be comparable. Delay in transmission must be as small as possible. Ideally it should be negligible.

The designed system works well on windows platform. This system is not tested on other platforms and the results are thus limited to a specific operating system.

The hardware and software components must be compatible with IP v6. Old hardware and operating systems does not support IP v6 and hence they are eliminated. The complete network architecture must have latest configuration to support this tunneling mechanism. [14]

7. CONCLUSION AND FUTURE SCOPE.

The proposed system discusses two different tunnels with the help of which security level can be increased.

The design involves KDD 99 or DARPA data set used to design an Intrusion Detection System. Further, SVM Classifier helps in classifying training data set.

Certain parameters were considered with which data can travel securely within a tunnel. We select the features that like, duration, protocol type, service, flag, source bytes, destination bytes, land, wrong fragment and so on. The features are utilized for a secure tunnel creation.

There are concerns related to data security. The issues are as follows: Dual-stack related issues, Header manipulation issues, IPSEC Relies on PKI , Not yet fully Standardized, No many firewalls in market with V6 capable, Flooding issues, Mobility, Packet Spoofing, ARP and DHCP attacks, Viruses, Worms and automated attack tools and so on. It is necessary to design solution to address all the above issues.

IP v6 protocol is not yet compatible with various operating system and active network components. The future scope involves such compatibility.

8. REFERENCES

[1] J020 B.D. Cabrerat B. Ravichandran and Raman K. Mehra, “ Statistical Traffic Modelling for Intrusion Detection” 0-7695-0728-WOO$-10.00 0 2000 IEEE 466-473

[2] Wanming Luo, Baoping Yan , Xiaodong Li, Wei Mao "Network-Processor-Based IPv4/IPv6 Translator: Implementation and Fault Tolerance" Feb. 17-20, 2008 ICACT 2008 488

[3] Mohammadreza Ektefa "Intrusion Detection using Data Mining techniques" 978-1-4244-5651-2/10/$26.00 ©2010 IEEE 200-203

[4] Shaneel Narayan(Member IEEE), Shailendra S. Sodhi, Paula R. Lutui, Kaushik J. Vijaykumar "Network Performance Evaluation of Routers in IPv4/IPv6 Environment A testbed analysis of software routers" 978-1-4244-5849-3/10/$26.00 ©2010 IEEE

[5] Lee Ling , Kasmiran Jumari, Mahamod ismail and Khairil Anuar, Joong-Hee Leet, Jong-Hyouk Leet "Effective Value of Decision Tree with KDD99 Intrusion Detection dataset for Intrusion Detection System" Feb. 17-20, 2008 ICACT 2008 1170-1175

[6] Yuhai Liu l, Hongbo Liu2 "The internet Traffic classification an online SVM approach"

[7] Chunling Wei "Research on Campus Network IPV6 Transition Technology". 978-0-7695-4480-9/11 $26.00 © 2011 IEEE DOI 10.1109/ISIE.2011.138

[8] Jiang Xie and"Case Study of Mobility Support for IPv4/IPv6 Transition Mechanisms over IPv6 Backbone networks"

[9] Thanh-Nghi Do " A Novel Speed-up SVM Algorithm for massive classification tasks" 978-1-4244-3279-8/08/$25.00 © 2008 IEEE 215-220

[10] Debajyoti Mukhopadhayay, Byung-Jun-Oh, Sang-Heon Shim, Young-Chon Kim, "A Study on recent Approaches in Handling DDoS Attacks" Cornell University, The computing Research Repository 1012.2979, Dec 2010

[11] Sheng Liu, Na Jiang "SVM Parameters optimization Algorithm and its Application" 978-1-4244-2632-4/08/$25.00 IEEE 509-513

[12] Mahbod Tavallaee, Ebrahim Bagheri, wei Lu, and Ali A. Ghorbani "A detailed Analysis of the KDD CUP 99 Data Set" 978-1-4244-3764-1/09/$25.00