International Journal of Combined Research & Development (IJCRD)
eISSN:2321-225X;pISSN:2321-2241 Volume: 5; Issue: 3; March -2016

VULNERABILITY STUDY OF REMOTE LOGIN PROTOCOLS: TELNET & SSH AND A PROPOSED METHOD FOR SECURE COMMUNICATION USING TELNET OVER IPSEC IN WINDOWS PLATFORM

Dr.J Sebastian Nixon[1], Dr. A Francis Saviour Devaraj[2]
[1& 2] Professor, Department of CS & IT,
Wolaita Sodo University, Federal Democratic Republic of Ethiopia.

Abstract. Protocols are a set of rules designed to communicate the devices on the network. Different kinds of protocols are used for different purpose. Some protocols have native security some may not. Protocols like HTTP & Telnet are vulnerable to man-in-the-middle attack because they exchange data in a non encrypted form and also they do not perform mutual authentication on both ends of the communication. A subset of those insecure protocols has secure counterparts that we can use to replace them like HTTPS instead of HTTP and SSH instead of Telnet but some of them do not have a secure alternative. In this paper, we created & configured a Network Topology, implemented Telnet & SSH separately and done the vulnerability studies using GNS3 with Wire Shark & Oracle Virtual Box. ***Then we proposed instead of replacing Telnet, how to use Telnet in a secure manner by implementing Telnet over IPsec [IP Security] in Windows Platform. Finally we have proposed a method to configure Telnet over IPsec on a Windows Server***.

Keywords: Telnet, SSH, IPsec, MMC.

1. Introduction

In the early 1960's, researchers independently published papers describing the idea of building computer networks [1]. At the same time, the telecommunication and computer industries became interested in computer networks. In parallel, the ISO with support from the governments worked on developing an open Suite of networking protocols. In the end, TCP/IP became the de facto standard. Among the different reference models, the **O**pen **S**ystems **I**nterconnection (OSI) model [2] became familiar. The word protocol is derived from the Greek word "***protocollon***" which means a leaf of paper glued to manuscript volume. In computing, protocol means a set of rules, a communication language or set of standards between two or more computing devices. Protocols exist at several levels o the OSI (**O**pen **S**ystem **I**nterconnectivity) layers model. The TCP/IP protocols suite consists of transmission control protocol, internet protocol, file transfer protocol, dynamic host configuration protocol, Border gateway protocol and a number of other protocols.

1.1 Properties of the protocol

Different protocols perform different functions so it is difficult to generalize the properties of the protocol. The following are some basic properties of most of the protocols.

- Detection of the physical [wired or wireless connection]
- Handshaking
- How to format a message
- How to send and receive a message
- Negotiation of the various connections
- Correction of the corrupted or improperly formatted messages.
- Termination of the session.

The widespread use of the communication protocols is a prerequisite to the internet. The term TCP/IP refers to the internet protocols suite and a pair of the TCP and IP protocols are the most important internet communication protocols. Most protocols in communication are layered together where the various tasks are divided and performed.

Protocol stacks refer to the combination of the different protocols. [3], defines the requirements for Internet hosts, mentions four different layers. Starting from the top, these are:

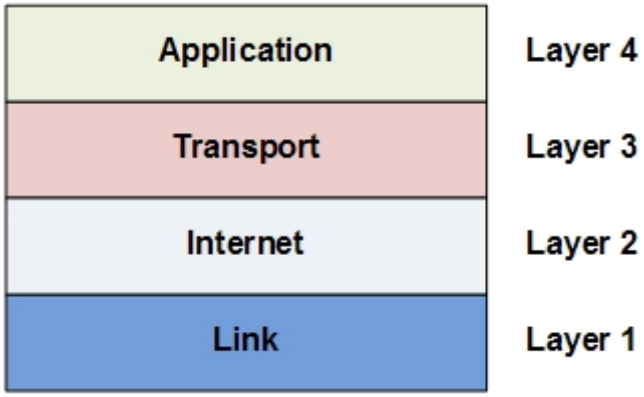

Figure 1: TCP/IP layer

• an Application layer which is equivalent to application layer, presentation layer and session layer of OSI model.

• a Transport layer which is equivalent to transport layer of OSI model

• an Internet layer which is equivalent to the network layer of OSI model.

• a Link layer which combines the functionalities of the physical and data link layers of OSI model.

1.2 Open Systems Interconnection Reference Model [OSI]

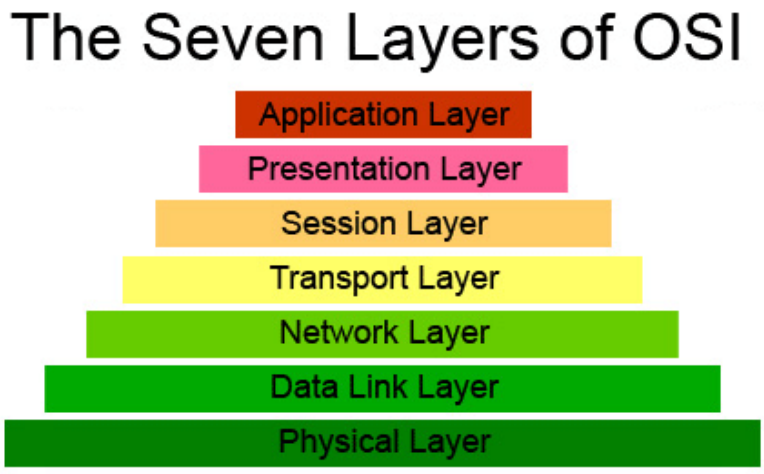

Figure 2: The OSI model

The OSI reference model is the conceptual model that is used to represent the protocol stacks. The OSI reference model defined in [4] is divided in seven layers.
Layer 1 - **Physical:** Defines mechanical and electrical interfaces and the transmission medium.
Layer 2 - **Data Link:** Defines methods for ensuring data integrity such as error correction.

Layer 3 - **Network:** Defines how packets of data are routed from source to destination.
Layer 4 - **Transport:** Defines the organization of data passing to and from the lower layers.
Layer 5 - **Session:** Defines the procedure for different communications equipment to establish dialogues.
Layer 6 - **Presentation:** Defines the syntax and semantics of transmitted information.
Layer 7 - **Application:** Defines procedures for file transfers, access methods and management of messages.

1.3 How Protocols Work?

The protocol steps must be carried out in a consistent order that is the same on every computer in the network. In the sending computer, these steps must be executed from the top down. In the receiving computer, these steps must be carried out from the bottom up.

The Sending Computer

Protocols at the sending computer:

1. Break the data into smaller sections, called **packets** that the protocol can handle.
2. Add addressing information to the packets so that the destination computer on the network can determine that the data belongs to it.
3. Prepare the data for transmission through the NIC (**N**etwork **I**nterface **C**ard) and out onto the network cable.

The Receiving Computer

Protocols at the receiving computer carry out the same series of steps in reverse order.

1. Take the data packets from the cable.
2. Bring the data packets into the computer through the NIC.
3. Strip the data packets of all the transmitting information that was added by the sending computer.
4. Copy the data from the packets to a buffer for reassembly.
5. Pass the reassembled data to the application in a usable form.

1.4 How Network Protocols Are Implemented

Modern operating systems like Microsoft Windows contain built-in services or daemons that implement support for some network protocols. Applications like Web browsers contain software libraries that support the high level protocols necessary for that application to function. For some lower level TCP/IP and routing protocols, support is implemented directly in hardware (silicon chipsets) for improved performance.

A group of network protocols that work together at higher and lower levels are often called a ***protocol family.***

2. Types of Protocols & their uses in TCP/IP

Table 1: list of layers, protocols and their uses

SNO	LAYER	PROTOCOL	PROTOCOLS USAGE
1	Link layer	Address Resolution Protocol (ARP), Neighbor Discovery Protocol (NDP)	Frame physical network functions like modulation, line coding and bit synchronization, frame synchronization and error detection, LLC and MAC sub layer functions.
2	Internet layer	IP, ICMP and IGMP	traffic routing, traffic control, fragmentation and logical addressing.
3	Transport Layer	Transport Control Protocol (TCP) and User Datagram Protocol (UDP).	message segmentation, acknowledgement, traffic control, session multiplexing, error detection and correction (resends) and message

			reordering.
4	Application Layer	NetBIOS, MIME, TLS, SSL, FTP, DNS, HTTP, SMTP	session establishment, maintenance and termination, character code translations, data conversion, compression and encryption, remote access & network management.

3. IPSec

Internet Protocol Security (IPsec) is used for securing **I**nternet **P**rotocol (**IP**) communications by *authenticating* and *encrypting* each IP packet of a communication session. IPsec also includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session. Outgoing packets are encapsulated, encrypted and authenticated just before being sent to the network and incoming packets are verified, decrypted and decapsulated immediately upon receipt [20]. IPsec implementations employs simple filter-based languages that specify routing rules for handling packets that match bit patterns in packet headers based on such parameters as incoming and outgoing addresses , ports, services, packet options, etc.[21].

3.1 Use of IPSec

- Used to fulfill security requirements or simply enhance the security of applications.
- It allows us to add IP restrictions and TCP/UDP level encryption to applications which may not otherwise support it.

Kerberos is used in initial authentication [22]. Kerberos is widely used in Microsoft Windows networking and other applications. Authentication proofs for each stage of Kerberos rely on the secrecy guarantees of keys set up in earlier stages, while the secrecy proofs similarly rely on previously proved authentication guarantees an alternation first pointed out in [23]. The Kerberos protocol involves four roles—***the Client, the Kerberos Authentication Server (KAS), the Ticket Granting Server (TGS) and the Application Server***.

4. Creation & Configuration of Network Topology

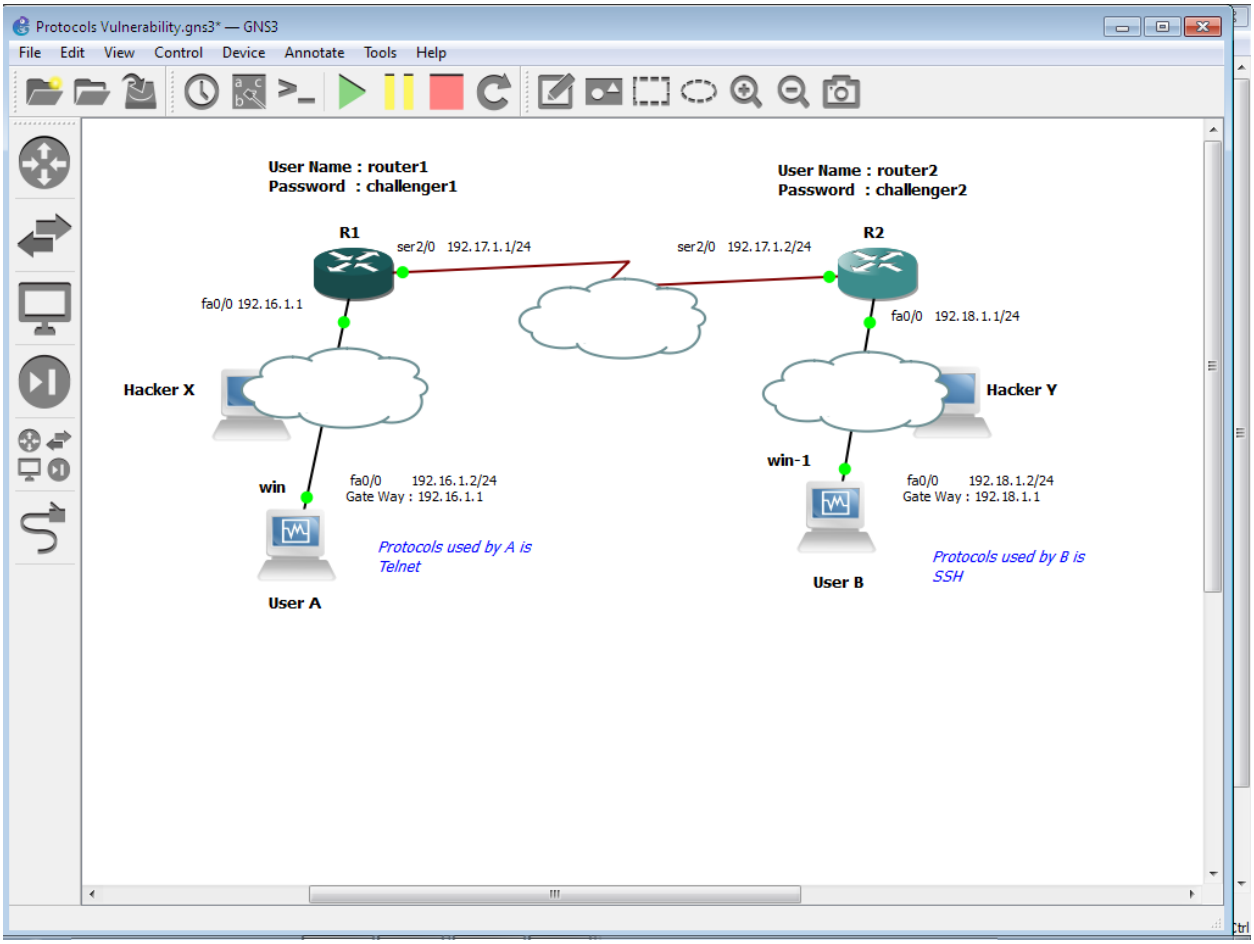

Figure 3: Network Topology

5. SSH – [uses TCP port 22]

Secure Shell is a protocol that provides authentication, encryption and data integrity to secure network communications. SSH is a client/server application that provides secure remote login [19]. The SSH protocol is officially specified in a set of five RFCs, namely RFC 4250-4254. Implementations of Secure Shell offer the following capabilities:

- A secure command-shell
- Secure file transfer
- Remote access to a variety of TCP/IP applications via a secure tunnel.

Secure Shell offers a good solution for the problem of securing data sent over a public network. In applications like telnet, a malicious user can eavesdrop on the network and to collect all communicated information [14,15]. To curb eavesdroppers, security researchers designed the Secure Shell (SSH), which offers an encrypted channel between the two hosts and strong authentication of both the remote host and the user [16 , 17, 18].

In the following section, we demonstrate how remote communication is secure using SSH, in the network topology [refer Figure 3].

Scenario 1: The hacker is running snipping tool [wireshark] to analyze the network traffic.

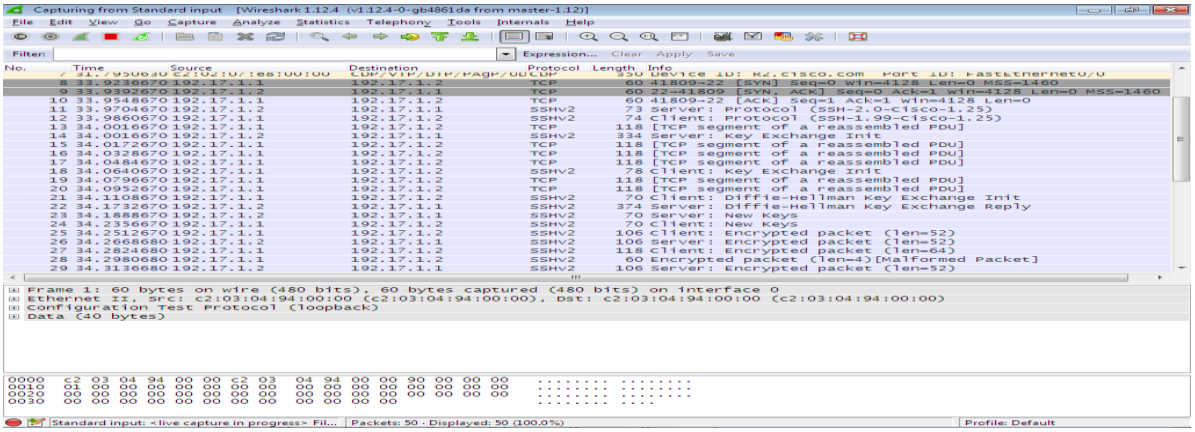

Figure 4: Hacker Y is running snipping tool

Scenario 2: User B Configuring SSH parameters in PuTTY on his/her computer to access Router [R2].

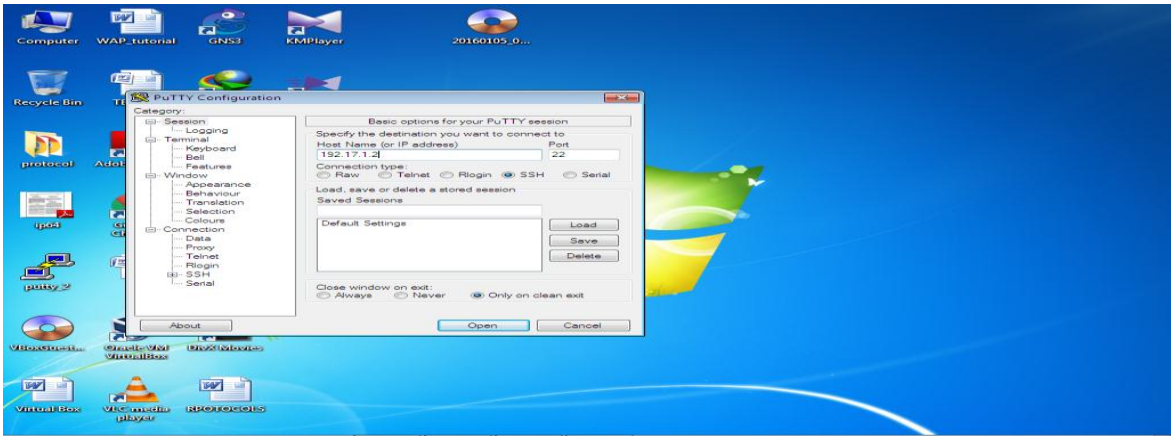

Figure 5: SSH configuration

Scenario 3: User B successfully logged into Router [R2] using **SSH**.

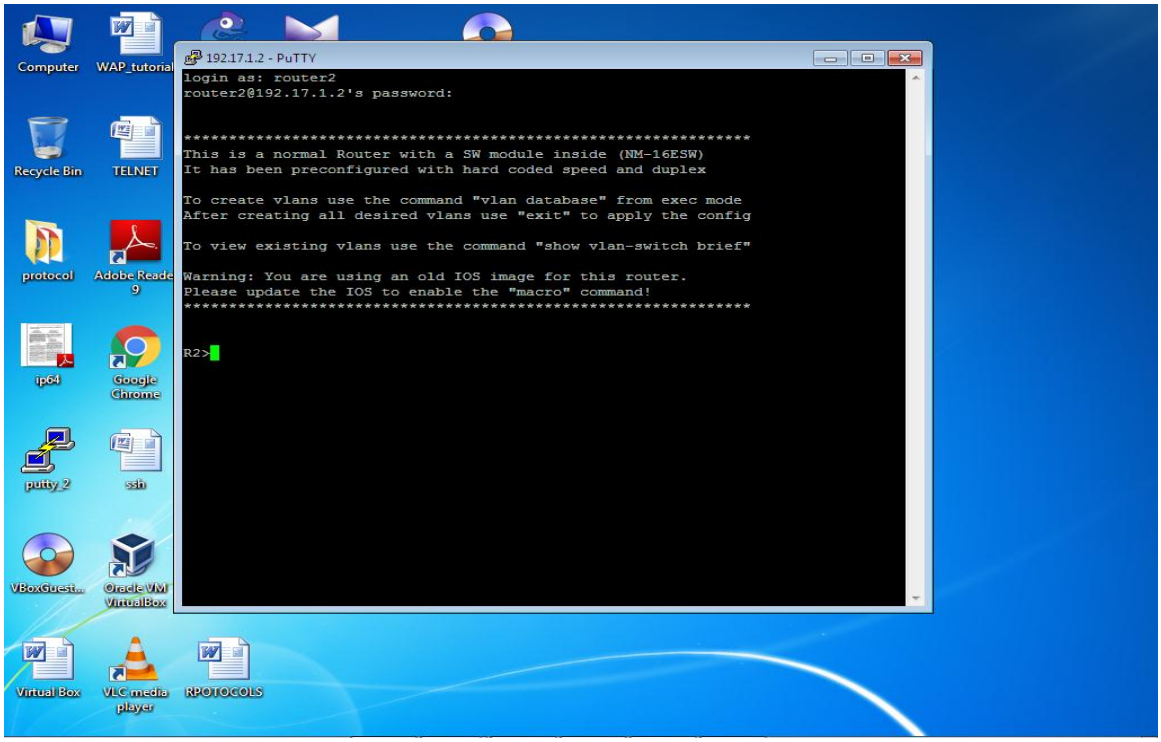

Figure 6: PuTTY screen shot that displays successful login using SSH.

Scenario 3: the hacker is filtering the SSH packets only

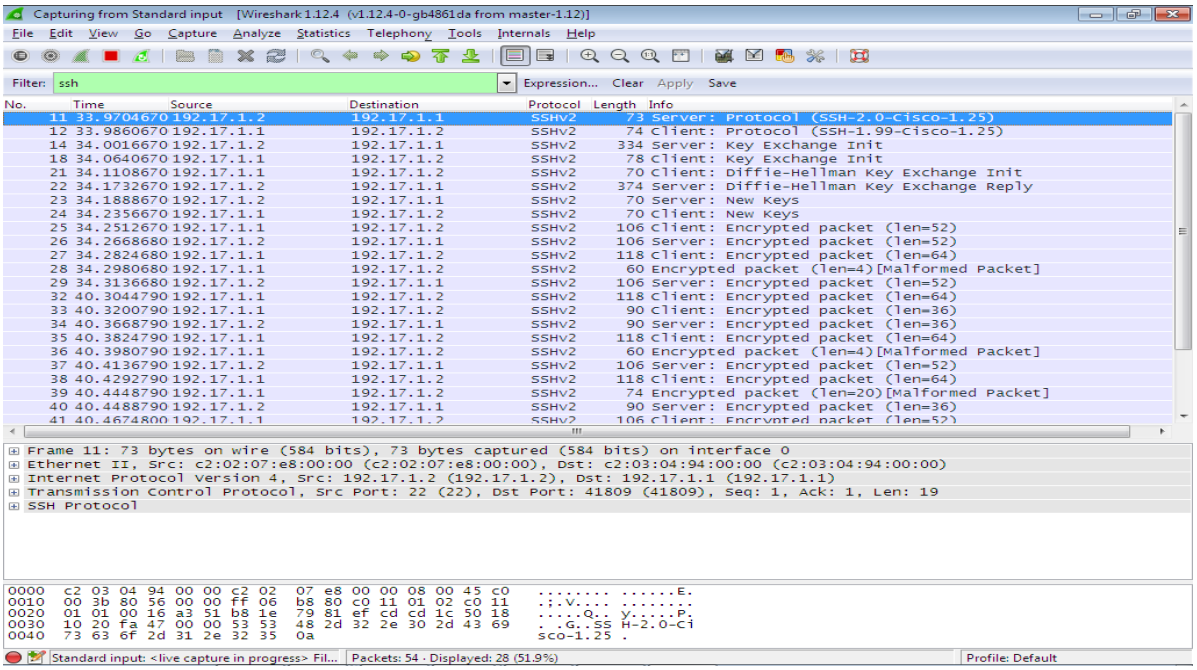

Figure 7: Hacker filtering SSH packets only

Scenario 3 (contd..) Hacker selecting TCP stream Option to open the packets & to view the SSH contents

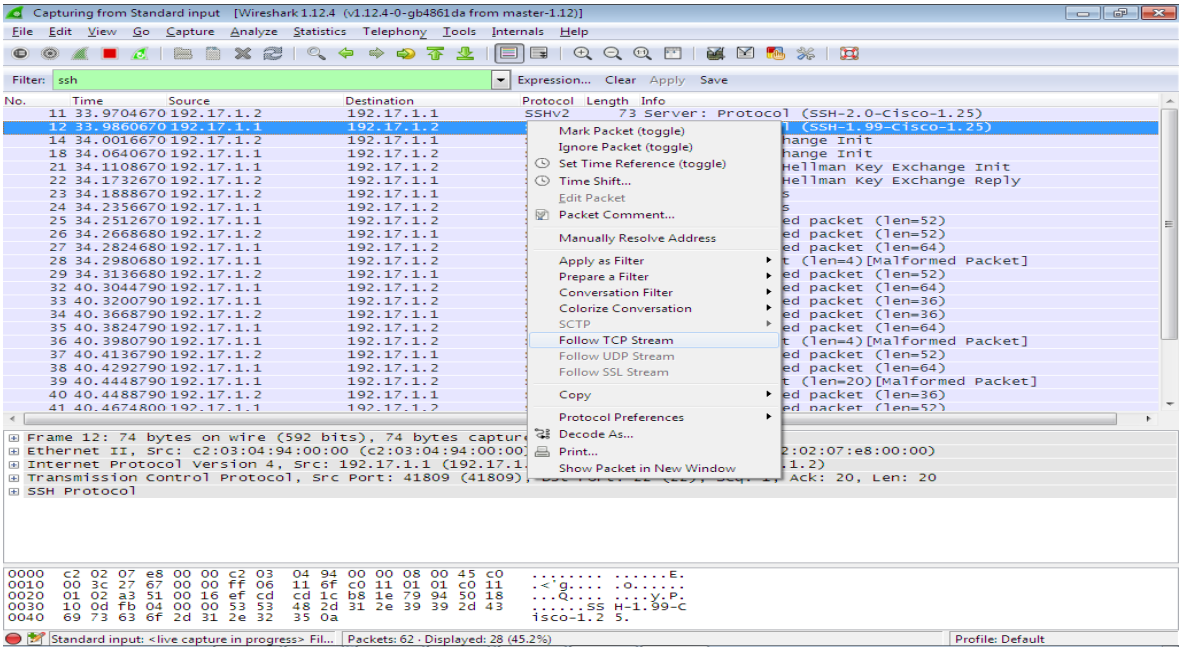

Figure 8: selecting TCP stream to open the packets

Scenario 4: SSH packets are encrypted, hence the hacker could not view the content.

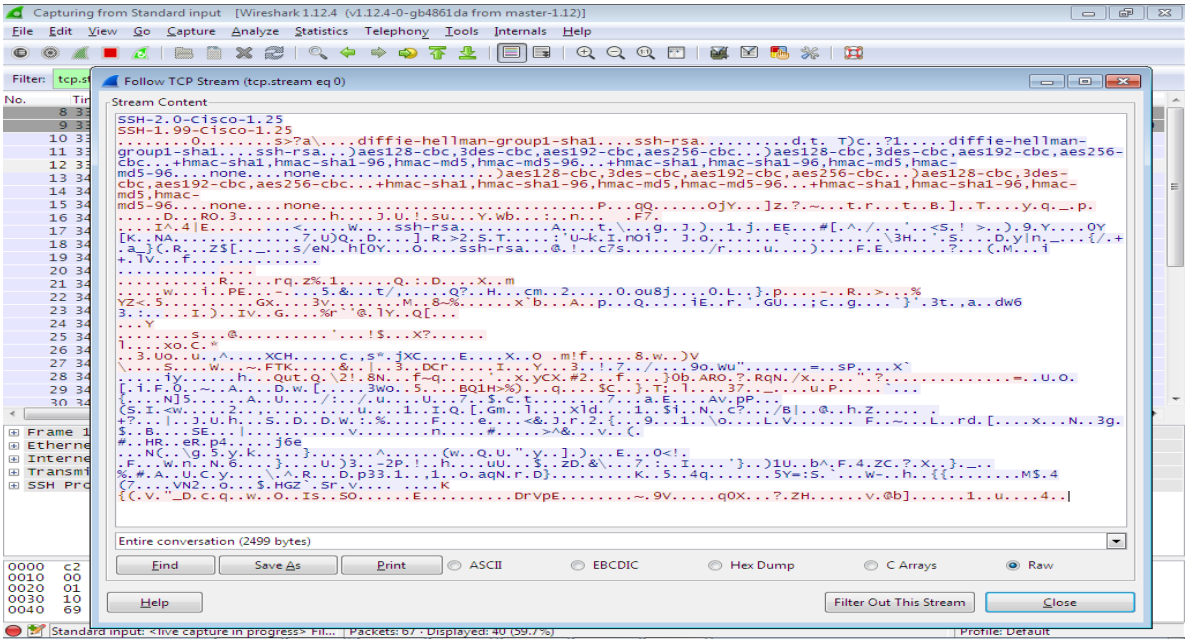

Figure 9: SSH encrypted content

From the scenario 4, we can conclude that Remote login using SSH is secure.

6. Telnet – [uses TCP port 23]

The Telnet protocol enables TCP/IP connections to a host. Telnet is an abbreviation for Terminal Network. Telnet allows a user at one site to establish a TCP connection to a login server at another site and then passes the keystrokes from one device to the other. Telnet can accept either an IP address or a host name as the remote device address. Telnet listens at TCP well known port 23. Telnet is terminal emulator software and is used to gain access to a command-line interface on a remote machine. The Telnet protocol is defined in RFC 854. Telnet client applications are available for all operating systems. Telnet means to establish a connection with the Telnet protocol, either with command line client or with a programmatic interface [9].

6.1 Telnet Security

When you telnet from one machine to another, the information is sent across the network. It is possible that other machines on the network can watch the traffic as it is exchanged between two machines.

If encryption is not negotiated in the telnet protocol, the traffic is not encrypted and it can be quite easy for other machines on the network to eavesdrop on the communication and record such things as passwords and other sensitive data. The person doing the eavesdropping might later use this information to

- impersonate you,
- destroy data on your machine(s),
- plant backdoors on your machine to allow future access,
- start running programs on your machine to promiscously listen on the network for more passwords and generally just play havoc.

This eavesdropping is not a theoretical attack. It is something that has happened multiple times in any facility. If you send your password across the network without encrypting, your password will be compromised. If you routinely send your password across the network without encryption, your machines could already be compromised.

6.2 Possible Issues when using Telnet for Remote Access

Using GNS3 with Wire Shark & Oracle Virtual Box we can explain the possible attacks as follows:

Scenario: 1 The Hacker X is Running packet snipping tool [wireshark] to Analyze & capture the packets on the network.

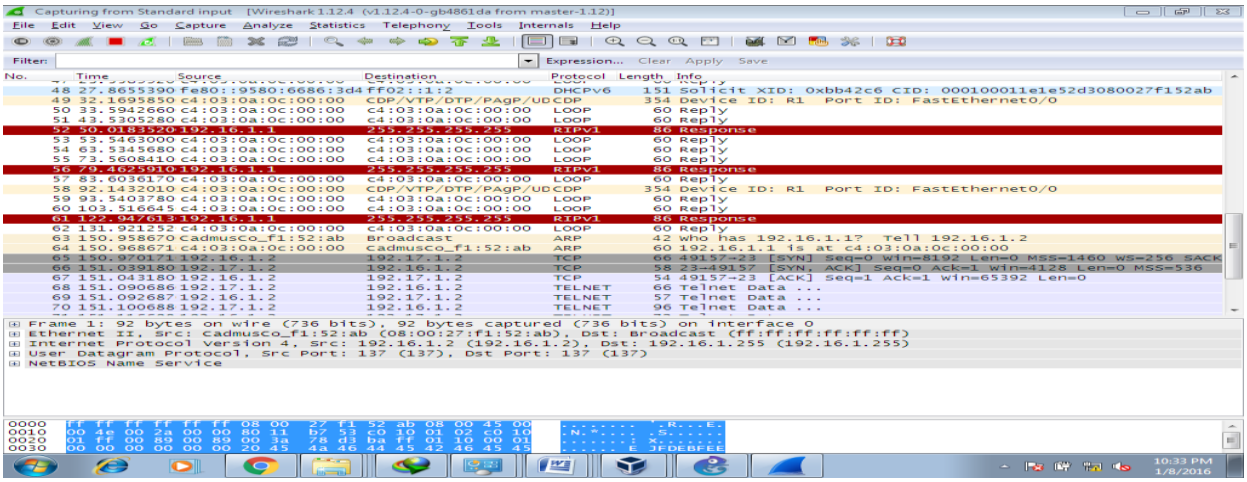

Figure 10: screenshot of packet captured by hacker X.

Scenario: 2 The **User A** on PC - **WIN** Accessing the Router **R2** using **TELNET** by giving the **router R2's password [challenger2].**

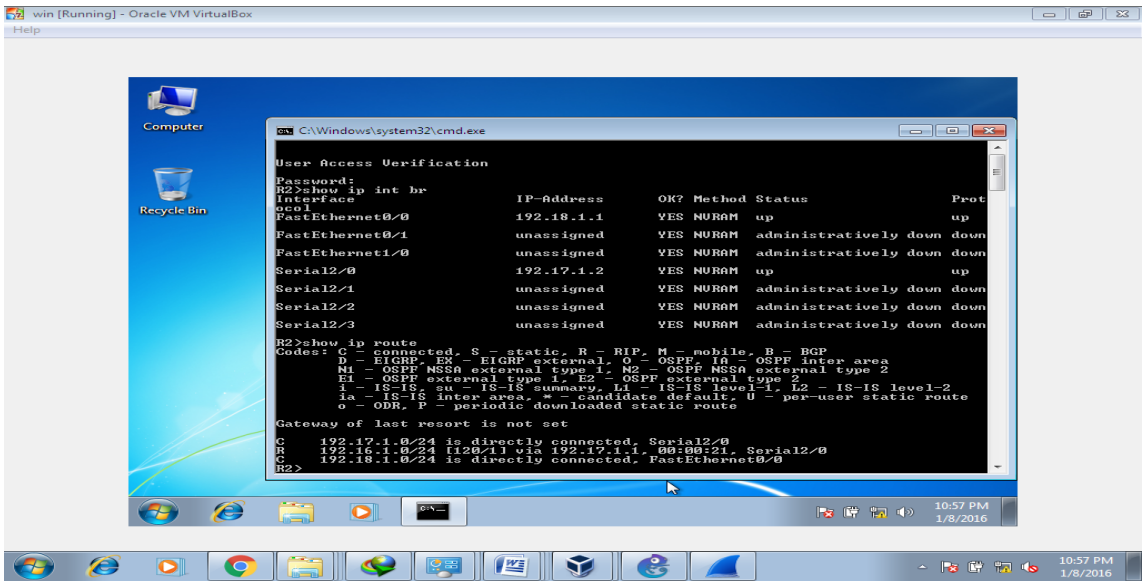

Figure 11: Console of Router R2

Scenario : 3 – The Hacker X is Filtering ***Telnet Packets only***

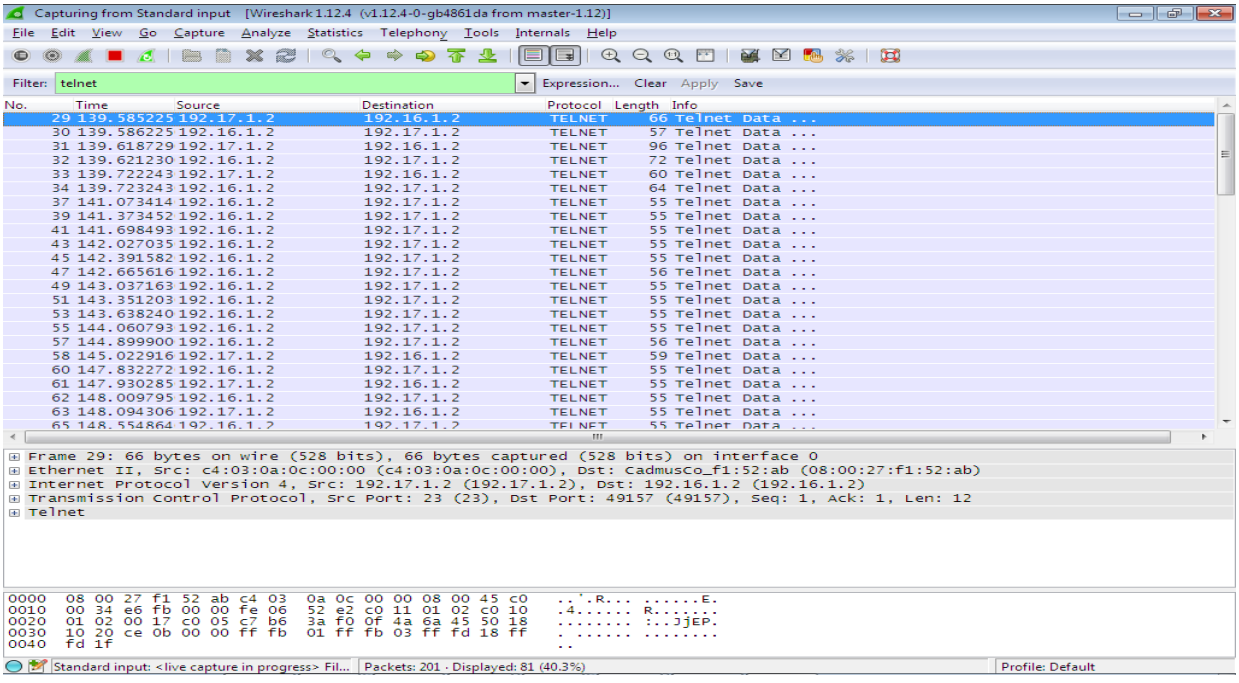

Figure 12: screenshot of Telnet packets

Scenario 3 (contd..) By Right Click on the Telnet Packet & Selecting Follow **TCP Stream,** the hacker X can view the Router R2's Password & other details as shown in Figure13 & Figure 14

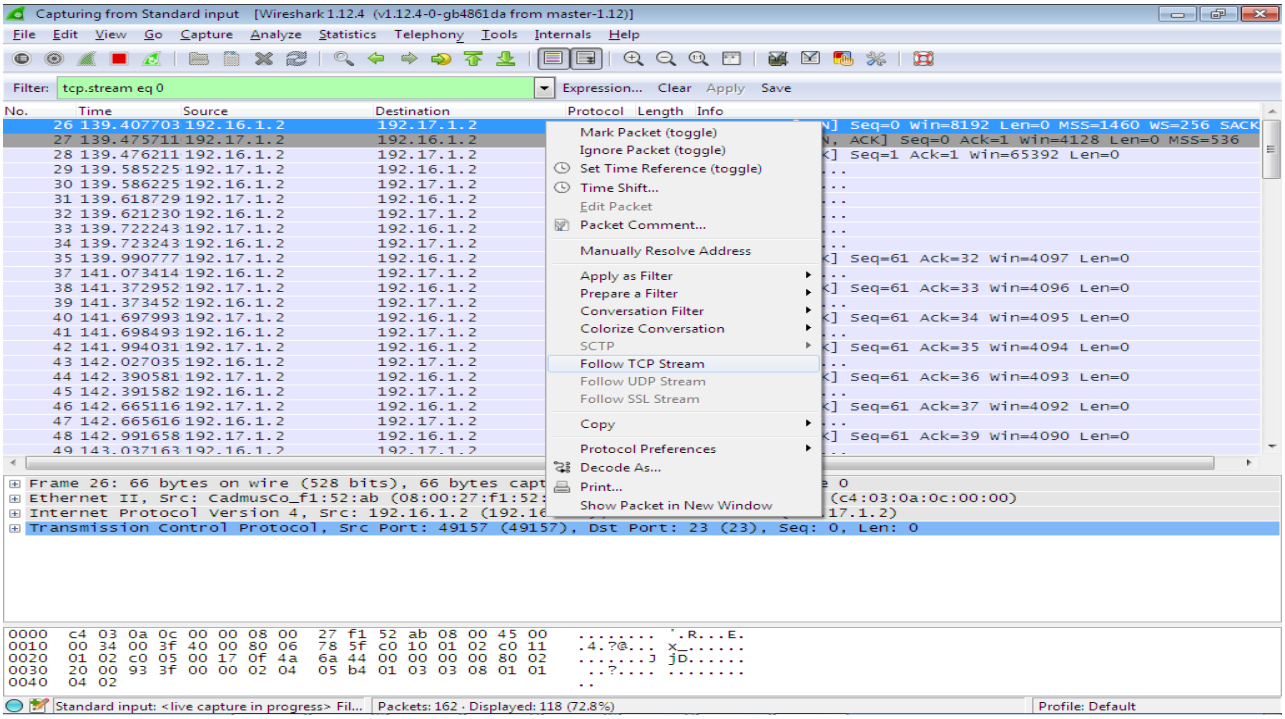

Figure 13 : screen shot to choose TCP stream

Scenario 3 (contd..)

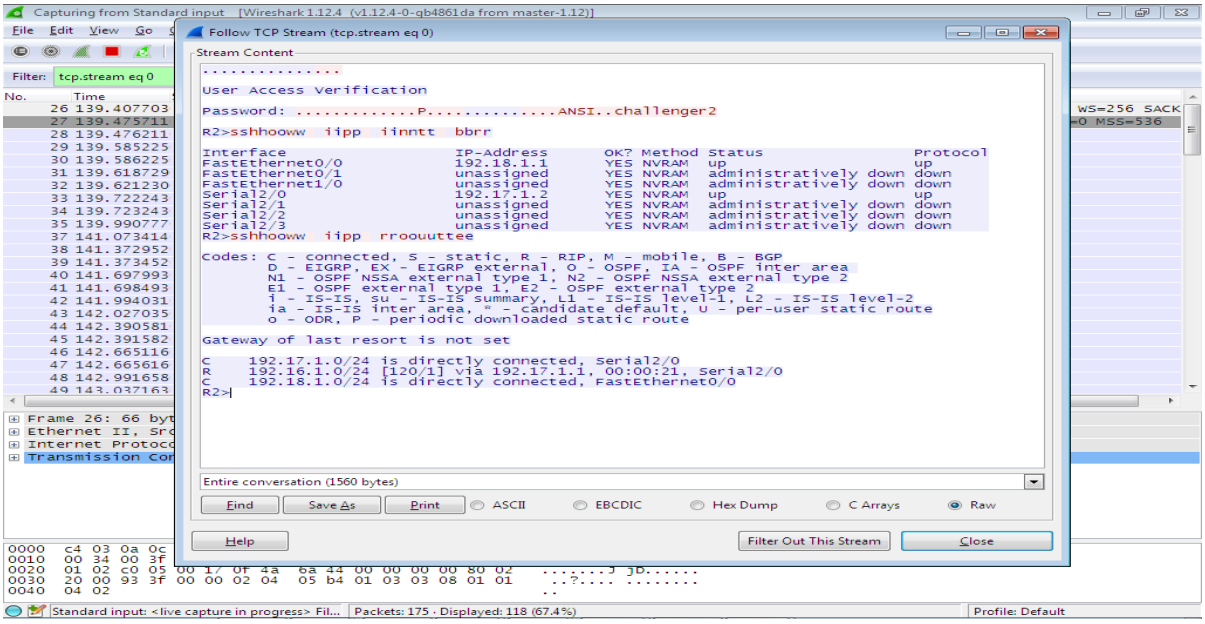

Figure 14 screenshot of TCP stream revealing the Telnet password.

The Router R2's Password is: ***challenger2***, The Command Executed on R2 was: ***show ip int br & show ip route***. So, from the above scenario we can conclude that ***communicaton using TELNET is insecure.***

That's why, most of the people are using SSH instead of Telnet. At the same time not all the SSH versions are free of cost. Henc, we proposed a secure method "Telnet over IPSec" for remote login in windows.

7. Proposed Method - Telnet Over IPsec

We can make the Telnet communication in a secure manner by using IPSec. In Windows 2000 & later OS include a troubleshooting tool called **M**icrosoft **M**anagement **C**onsole [**MMC**] IP Security Monitor snap-in to troubleshoot failed IPsec connections.

It contains Filters, Policies and Security Associations for both Main Mode IPSec and Quick Mode IPsec. IPsec used in conjunction with Telnet dramatically improves Telnet Security. IPSec [5], [6] provides a method to protect IP datagrams. For both IPv4 and IPv6 it offers the choice of two protocols:

- **E**ncapsulating **S**ecurity **P**ayload [ESP] [7]
- **A**uthentication **H**eader [AH] [8].

The Internet Protocol Security architecture [IPSec] [10] has been proposed by IETF to provide

- Integrity
- Confidentiality
- Authentication of data communications over IP networks.

The IPSec policy consists of lists of rules that designate the traffic to be protected, the type of protection, such as authentication or confidentiality and the required protection parameters, such as the encryption algorithm [12]. Packets are sequentially matched against the rules until one (single-trigger) or more (multiple-trigger) matching rules are found [11, 13].So, ***it is possible to encrypt the Telnet traffic using IPsec to perform remote login in a secure manner***.

In this proposed method, we used Kerberos, the Windows IPsec default authentication method. When we used this method, the client and server must reside in the same / trusted domains. The IPsec also supports **C**ertificates **A**uthentication method [CA**]**. The protection offered by IPSec to certain traffic is based on requirements defined by security policy rules defined and maintained by the system administrator [11, 6].

To encrypt Telnet traffic over IPsec, We have to create the following objects.

- IPsec Policy
- IPsec Security Rule
- IP Filter List
- Filter Action

Procedure

- Start the Telnet service on the windows server machine then perform the following steps to create the IPsec Policy to Encrypt Telnet traffic using windows server MMC:

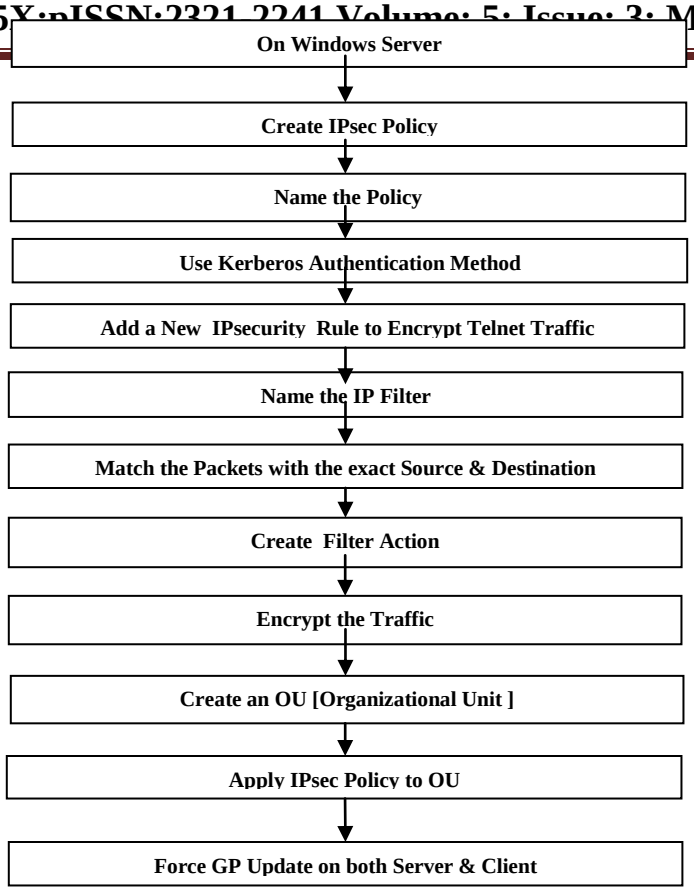

Figure 15: Flow chart outlining about the configuration of Telnet over IPSec

Conclusion:

In this paper, we have discussed and proved that SSH is secure than Telnet using GNS3 with wireshark and Oracle virtual box tools by capturing and opening the packets. Also we have proposed to use Telnet securely by using IPsec. In future we would implement and show the configuration of Telnet over IPSec and study about the performance of the network.

References

[1] Licklider, J., Memorandum For Members and Affiliates of the Intergalactic Computer Network,
[2] Zimmermann, H., OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection, IEEE Transactions on Communications, vol. 28, no. 4, April 1980, pp. 425 - 432.
[3] Braden, R., Requirements for Internet Hosts - Communication Layers, RFC 1122
[4] ITU-T, recommendation X.200, Open Systems Interconnection - Model and Notation
[5] S. Kent and R. Atkinson, "Security architecture for the internet protocol," RFC 2401 (Proposed Standard), Internet Engineering Task Force, Nov. 1998, updated by RFC 3168.
[6] N. Doraswamy and D. Harkins, IPSec: the new security standard for the Internet, intranets, and virtual private networks, 1st ed. Prentice Hall, 1993
[7] S. Kent and R. Atkinson, "IP encapsulating security payload," RFC 2406 (Proposed Standard), Internet Engineering Task Force, Nov. 1998.
[8] "IP authentication header," RFC 2402 (Proposed Standard), Internet Engineering Task Force, Nov. 1998.
[9] N. G. Duffield, W. A. Massey, and W. Whitt. "A nonstationary offered-load model for packet networks. Telecommunication Systems", 16(3-4):271–296, 2001.
[10] S. Kent and R. Atkinson. Security architecture for the internet protocol. IETF RFC-2401, November 1998.

[11] N. Doraswamy and D. Harkins. IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks. Prentice Hall PTR, second edition, March 2003.
[12] K. Jason, L. Rafalow, and E. Vyncke. IPsec configuration policy information model. IETF RFC-3585, August 2003.
[13] S. Kent and R. Atkinson. Security architecture for the internet protocol. IETF RFC-2401, November 1998.
[14] William R. Cheswick and Steven M. Bellovin. Firewalls and Internet Security – Repelling the Wily Hacker. Professional Computing Series. Addison-Wesley, 1994. ISBN 0-201-63357-4
[15] Simson Garfinkel and Gene Spafford. Practical UNIX & Internet Security. O'Reilly & Associates, 1996
[16] ¨ Tatu Ylonen. ¨ SSH – Secure Login Connections over the Internet. In Sixth USENIX Security Symposium, San Jose, California, July 1996.
[17] IETF Secure Shell Working Group (SECSH). http://www.ietf.org/html. charters/secsh-charter.html, 2001.
[18] T. Ylonen, ¨ T. Kivinen, M. Saarinen, T. Rinne, and S. Lehtinen. SSH protocol architecture. Internet Draft, Internet Engineering Task Force, May 2000.
[19] Tatu Yl¨onen. SSH – Secure Login Connections over the Internet. In Proceedings of the 6th USENIX Security Symposium, pages 37–42, July 1996.
[20] J. Ioannidis and M. Blaze. The Architecture and Implementation of Network-Layer Security Under Unix. In Fourth Usenix Security Symposium Proceedings. USENIX, October 1993.
[21] S. McCanne and V. Jacobson. A BSD Packet Filter: A New Architecture for User-level Packet Capture. In Proceedings of USENIX Winter Technical Conference, pages 259–269, San Diego, California, Jan. 1993.
[22] L. Zhu and B. Tung. Public Key Cryptography for Initial Authentication in Kerberos (PKINIT). RFC 4556 (Proposed Standard), June 2006.
[23] I. Cervesato, C. Meadows, and D. Pavlovic. An encapsulated authentication logic for reasoning about key distribution protocols. In CSFW, pages 48–61, 2005

AUTHORS PROFILE

Dr. Sebastian Nixon .J. is a Professor , department of CS & IT, CNCS, Wolaita Sodo University, Federal Democratic Republic of Ethiopia. He is a Microsoft Certified Sysem Engineer **(MCSE)**. He is specialized in Network Security. He has around 19 + Years of teaching experience. He is a life member in **CSI, ISOC & CRSI**.

Dr. A Francis Saviour Devaraj is a Professor , department of CS & IT, CNCS, Wolaita Sodo University, Federal Democratic Republic of Ethiopia. He is a Cisco Certified Network Associate **(CCNA)**. He is specialized in Information Security. He has around 15 plus years of teaching experience. He is a life member in **CSI, ISOC, CRSI & ISTE**.