MultiCraft

International Journal of Engineering, Science and Technology
Vol. 4, No. 1, 2012, pp. 125-128

INTERNATIONAL JOURNAL OF ENGINEERING, SCIENCE AND TECHNOLOGY

www.ijest-ng.com
www.ajol.info/index.php/ijest

Public Key Infrastructure (PKI) enhanced file transfer over secure sockets in Linux environment

Gyanendra Kumar Pal[1]*, A.K Malviya[1], Lokender Tiwari[2], Prashant Yadav[3]

[1]*Department of Computer Science & Engineering, Kamla Nehru Institute of Technology, Sultanpur, U.P, INDIA*
[2] *Department of Computer Science & Engineering, Veer Bahadur Singh Purvanchal University, Jaunpur, U.P, INDIA*
[3] *Department of Computer Science & Engineering, United College of Engineering & Research, Allahabad, U.P, INDIA*
[*]*Corresponding Authors: e-mail: loke1602@gmail.com, gyanpal@gmail.com .*

Abstract

Public Key Infrastructure (PKI) provides an intensive security mechanism for securing data communication over network. Generally transferring a file over a network is not secure if the network is wireless network or it consists of hubs as a networking device. Because then packets are broadcasts to every other computers over the network. A hub does not remember what all devices are attached to it. It just sends the packets to all its ports. Same in case of wireless networks the data packets are broadcasted. In general scenario the data packets are received by only those clients which are supposed to receive it, but it may be happen that a third party too, called "Sniffers" capture or "sniffed" the data packets during file transaction even if they are not supposed to accept it. In this work we try to enhance the security of file transfer by merging file transfer over secure socket along with Public Key Infrastructure (PKI). If we implement file transfer along with asymmetric key cryptography then there is another problem arises, known as man-in-middle attack. This attack exposes the problem of key validity, in which the attacker intercept the first message and sends its own public key to each. By doing this, attacker pretends to be the other person and hence can read the stream of decrypted traffic and can modify it. To provide privacy and security to file transfer we use Secure Socket Layer (SSL) a communication layer protocol.

Keywords: PKI, Secure Sockets, Tunneling, Man-in-Middle Attack, Socket Programming, Packet Sniffer.

DOI: http://dx.doi.org/10.4314/ijest.v4i1.14S

1. Introduction

This paper is designed with the aim of defining enhanced architecture of file transfer over tunneled connection in linux environment using Secure Socket Layer (SSL) with enhanced security and prevents the man-in-middle attack using Public Key Infrastructure (PKI) and certificates. The intended audiences are all people who rapidly transfer confidential data over Internet through TCP/IP layer. Whole architecture is implemented in Linux environment using C-language socket programming as it provides basic framework and header files. Playfair cipher is used as encryption/decryption algorithm. First, we develop a file transfer program using socket programming in Linux platform then, for encryption and decryption we use above algorithm and enhanced its authentication mechanism by implementing it along PKI architecture with the assurance that if supposed data packets can be "sniffed" by third party then it cannot be interpreted or decode into original data and prevent man-in-middle attack by using certificates (Davis *et al*, Apress, 2004) (Singh *et al*, *INDIACom-2010)*

2. Client- Server Model

Client-Server interaction is the basis of computer communication. The fundamental motivation for the client-server architecture arises from the problem of rendezvous, and solved by asserting that in any pair of communicating applications, one side must start execution and wait (indefinitely) for the other side to connect. Applications that initiate communication is called *clients*, and a *server* is one that wait for incoming communication request from clients (Comer *et al*, Prentice Hall of India, 2001). In next section we describe overview of sockets.

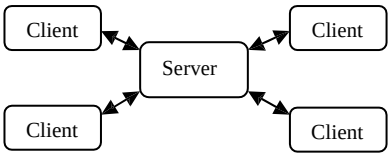

Figure 1. Client-Server 2 Model

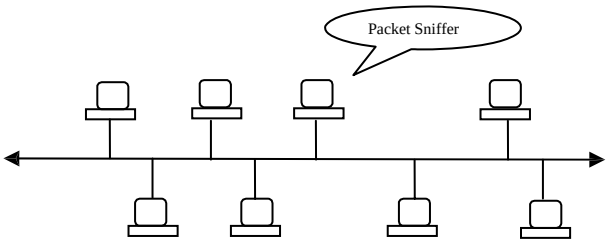

Figure 2. Packet Sniffing

3. Sockets- The Socket Interface

A socket is an abstraction for network communication, just as a file is an abstraction for file system communication. The interface that resulted from the task that is assigned by Advanced Research Group Projects Agency (ARPA) to a group at the University of California, Berkeley , to port TCP/IP software to the UNIX operating system (Stevens, 2005) become known as the *socket interface* sometimes *Berkeley socket interface.* The Berkeley socket interface extends the concept of a file descriptor to that of *socket descriptor*. A socket structure contains information such as socket type, port being used by socket, local address, and the remote address and port that will receive communication from the socket. Sockets can be used in two different ways, once we create a socket it can wait for the connection, or it can initiate a connection to another remote host or its own local host. A socket used by client program to initiate a connection to the server called *active socket.* While the socket that acts as a server and wait for the connection is known as *passive client* (Davis *et al,* Apress, 2004)((Comer *et al,* Prentice Hall of India, 2001)

4. Packet Sniffing- An Overview

Packet sniffing is done by software's called "sniffers". These are the programs that have the ability to intercept the traffic that passes over a network. Every computer that communicate over network have at least one network interface card (NIC) and each card is uniquely recognized by their MAC address and IP address. Data packets that are to be transmitted over network contains communication overhead information such as source and destination address and port numbers to route the packets to correct computer. In usual manner data packet is only received by that computer which is supposed to be and all other reject it. But a computer connected to a network as shown in figure 2, that is "sniffer" can see all the data traffic over the wire, this is done by sniffer by putting their own NIC in promiscuous mode and can capture data packets and reconstruct into original data packet (Singh *et al, INDIACom-2010).*

5. Tunneled Connection- Tunneling

Tunneling provide secure connection mechanism between client and server. It can be used to secure data packets from sniffers. First we create a secure connection after then other network communication forwarded or "tunneled" over that secure connection.

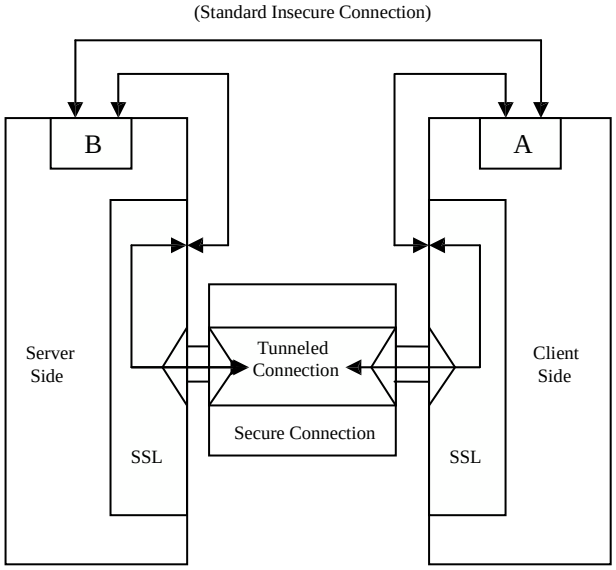

Figure 3. Tunneling Over Secure Connection

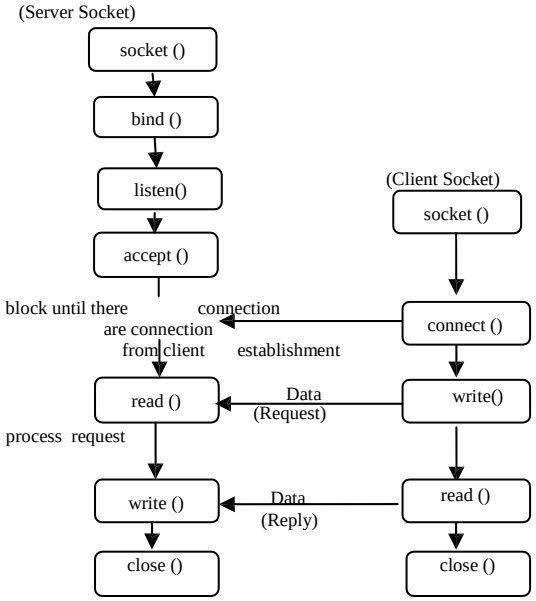

Figure 4.TCP/IP Client-Server Socket Programming Overview

In this work we use OpenSSL libraries to implement a Secure Sockets Layer/ Transport Layer Security (SSL/TLS) encrypted channel. We take an insecure ftp application, and tunnel it over a Secure Socket Layer (SSL) connection. It required a signed SSL certificate, which is generated with the OpenSSL toolkit, or it may be purchased from some certificate issuing companies. An overview of how tunneling works shown in figure 3. (Davis *et al*, Apress, 2004)

6. TCP/IP Client- Server Socket Interface

In this work to implement file transfer both client and server begin with a call to socket, bind it to a port and returning a socket descriptor after that server wait for incoming client connections. Clients then call connect, while server call the accept. After creating an active connection both client and server exchange data securely. All this socket programming is done over tunneled connection using Secure Socket Layer (SSL). A functional Client-Server socket interface along with socket function calls is shown in figure 4. (Davis *et al,* Apress, 2004)

7. Public Key Infrastructure (PKI)

PKI provides a hierarchical framework for managing the digital security attributes of entities that will engage in secured communication. Each PKI participants holds a digital certificate that has been issued by a Certificate Authority (CA). There are two types of key cryptography used for encryption: Symmetric Key cryptography and Asymmetric Key cryptography. Symmetric Key Cryptography employs a mathematical function that uses a single unique key for both encryption and decryption.

While in asymmetric key cryptography we use concept of two keys, one for encryption purpose called public key other one is private key used for decryption of the encrypted data. Asymmetric Key Cryptography makes it possible to share a public key without compromising secure connection. While, an attack known as man-in-middle attack exposes another problem of key-validity (Davis *et al*, Apress, 2004) (Stallings, *4th ed*, Prentice Hall, 2006).

8. Playfair Cipher

Playfair cipher is a stream cipher technique use 5x5 matrix and a keyword. The 5x5 matrix contains all 26 English alphabets with i and j in same box. Playfair cipher encrypts two letters at a time and is much harder to break then a mono alphabetic cipher (Stallings, *4th ed*, Prentice Hall, 2006).

9. Man-In-Middle Attack

In this attack, the attacker intercepts the first message and sends its own public key to each. By doing this each client thinks that the attacker is the other person. Due to this an attacker can read, modify, and use the stream of decrypted traffic (Davis *et al*, Apress, 2004).

10. Certificates

To prevent man-in-middle attack we use certificates which are included in PKI standards. A certificate is used to confirm that certain data has been sent by the organization or person that claims to have sent it. A certificate contains a digital signature that is applied and verified with symmetric cryptography by a trusted third party. When a session initiated client and server exchange public keys and certificates and this can be verified by third party. In this work we generate self certificates for testing purpose (Davis *et al*, Apress, 2004).

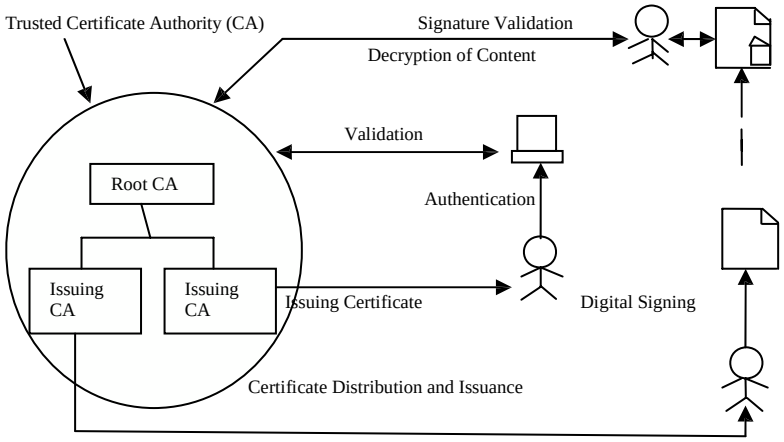

Figure 5. Public Key Infrastructure (PKI)

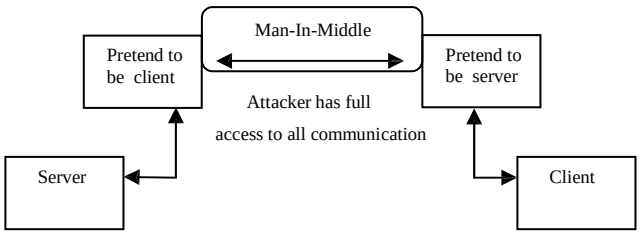

Figure 6. Man-In Middle Attack

11. Conclusion

Data packets are broadcasted over the network when network is wireless network or contains hubs as a networking devices, file transfer over a network can be seen by everyone who is connected to network or can be sniffed by sniffers as they start accepting data packets even though they are not intended to accept these data packets. Therefore we encrypt the data packets before transmitting it over the network, after that if sniffer capture these data packets then it is only the garbage meaningless data or it cannot be decode into original data. In this work we use the playfair cipher to encrypt the data before transmission. To provide more secure data exchange we implement Public Key Infrastructure (PKI), which enhanced the security mechanism, and to prevent man-in-middle attack we use the certificates along with symmetric key cryptography which sort the problem of key validity. In future we can enhance the security mechanism by changing the encryption method or algorithm such as RSA, Elliptic Curve Encryption etc to provide more security to communicating data.

References

Stevens Richard. W, Fenner Bill and Rudoff M. Andrew, 2005 . *UNIX Network Programming-The Sockets Networking API*, Vol. 1 3rd ed : Pearson Education.

Comer E. Douglas and Stevens L. David, 2001. *Internetworking with TCP/IP- Client-Server Programming and Applications (BSD Socket Version)*", Vol. 3, 8th ed , Prentice Hall of India.

Davis Keir, Turner W. John and Yocom Nathan,2004 .The *Definitive Guide to Linux Network Programming*, Apress.

Singh Abhipal, Sethi Singh Gurneet, Oberoi Kaur Kavleen and Kaur Jasleen,2010.File Transfer Using Secure Sockets in Linux Environment, *Proc. of the 4th National Conference: INDIACom-2010.*

Stallings William, 2006.*Cryptography and Network Security*, 4th ed, Prentice Hall.

http://www.tcpdump.org/#documentation

http://www.openssl.org

Biographical notes

Gyanendra Kumar Pal was born at Jaunpur ,(U.P),in India. He received the B.Tech degree in Information Technology in 2007 from, U.N.S Institute of Engineering & Technology, Veer Bahadur Singh Purvanchal University , Jaunpur, (U.P) INDIA .He is currently pursuing M.Tech in Computer Science and Engineering from Kamala Nehru institute of Technology Sultanpur, U.P. His research interests are Software Engineering, Cryptography &Network Security, Distributed Computing.

Dr. Anil Kumar Malviya is an Associate Professor in the Computer Science & Engg. Department at Kamla Nehru Institute of Technology, (KNIT), Sultanpur. He received his B.Sc. & M.Sc. both in Computer Science from Banaras Hindu University, Varanasi respectively in 1991 and 1993 and Ph.D. degree in Computer Science from Dr. B.R. Ambedkar University; Agra in 2006.He is Life Member of CSI, India. He has published about 30 papers in International/National Journals, conferences and seminars. His research interests are Data mining, Software Engineering, Cryptography & Network Security.

Lokender Tiwari was born at New Delhi in India. He is a Research Scholar at Department of Computer Science & Engineering, U.N.S Institute of Engineering & Technology, Veer Bahadur Singh Purvanchal University , Jaunpur, (U.P) INDIA . His current area of research includes Socket Programming, Cryptography & Network Security, Digital Forensics, Client-Server Computing, Computer Network Routing Technologies and Distributed Computing.

Prashant Yadav was born at Jaunpur, (U.P.), in India. He received the B.Tech. degree in Information Technology from , U.N.S Institute of Engineering & Technology, Veer Bahadur Singh Purvanchal University , Jaunpur, (U.P) INDIA . He is currently pursuing M.Tech in Computer Science and Engineering from United College of Engineering & Research, Allahabad, (U.P) India. His area of research includes Cryptography, Operating System, Automata Theory.

Received January 2012
Accepted February 2012
Final acceptance in revised form March 2012