NETWORK SECURITY DEVICE

Field of the Invention

The present invention is directed to a network security device that is connected between a protected computer (the client) and a network and a method for utilizing the network security device. The network security device negotiates a session key with any other protected client. Then, all communications between the two clients are encrypted. The inventive device is self configuring and locks itself to the IP (Internet Protocol) address and MAC address of its client. The client cannot change its IP or MAC address once set. Thus, the inventive network security device does not allow a client to emulate another client by setting a false IP or MAC address.

Background of the Invention

A. Network Architecture

An internet communications network 100 is depicted in FIG. 1 including five transmit or backbone networks A,S,C,D, and E and three stub networks R, Y, and Z. A "backbone" network is an intermediary network which conveys communicated data from one network to another network. A "stub" network is a terminal or endpoint network from which communicated data may only initially originate or ultimately be received. Each network, such as the stub network R, includes one or more interconnected subnetworks I, J, L and M. As used herein, the term "subnetwork" refers to a collection of one or more nodes, e.g., (d), (a)(b,x,y), (q,v)(r,z), (s,u), (e,f,g),(h,i),(j,k,l),(m,n), and (o,p), interconnected by wires and switches for local internodal communication. Each subnetwork may be a local area network or LAN. Each subnetwork has one or more interconnected nodes which may be host computers ("hosts") u,v,w,x,y,z or routers a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s. A host is an endpoint node from which communicated data my initially originate or ultimately be received. A router is a node which serves solely as an intermediary node between two other nodes; the

things, the packetizing of information to be transmitted, the reassembly of received packets into the originally transmitted information, and the scheduling of transmission and reception of packets (see e.g., D. Commer, "Internetworking With TCP/IP", Vol. 1 (1991); D. Commer and D. Stevens, "Internetworking With TCP/IP", Vol. 2 (1991)).

In an exemplary internet protocol call IP, each node of the internet 100 is assigned an internet (IP) address which is unique over the entire internet 100 such as the internet address for the node y shown in FIG. 3B. See, Information Sciences Institute, RFC 791 "Internet Protocol", September, 1981. The IP addresses are assigned in a hierarchical fashion; the internet (IP) address of each node contains an address portion 31 indicating the network of the node, an address portion 32 indicating a particular subnetwork of the node, and a host portion 33 which identifies a particular host or router and discriminates between the individual nodes within a particular subnetwork.

In an internet 100 which uses the IP protocol, the IP addresses of the source and destination nodes are placed in the packet header 42 by the source node. A node which receives a packet can identify the source and destination nodes by examining these addresses.

B. Encryption Techniques

Eavesdropping in a network, such as the network 100 of FIG. 1, can be thwarted through the use of a message encryption technique. A message encryption technique employs an encipherment function which utilizes a number referred to as a session key to encipher data (i.e., message content). Only the pair of hosts in communication with each other have knowledge of the session key, so that only the proper hosts, as paired on a particular conversation, can encrypt and decrypt digital signals. Two examples of encipherment functions are the National Bureau of Standards Data Encryption Standard (DES) (see e.g., National Bureau of Standards, "Data Encryption Standard", FIPS-PUB-45, 1977) and the more recent

Fast Encipherment Algorithm (FEAL)(see e.g., Shimizu and S. Miyaguchi, "FEAL-Fast Data Encipherment Algorithm," Systems and Computers in Japan, Vol. 19, No. 7, 1988 and S. Miyaguchi, "The FEAL Cipher Family", Proceedings of CRYPTO '90, Santa Barbara, Calif., Aug., 1990). Another encipherment function is known as IDEA. One way to use an encipherment function is the electronic codebook technique. In this technique a plain text message m is encrypted to produce the cipher text message c using the encipherment function f by the formula c=f(m,sk) where sk is a session key. The message c can only be decrypted with the knowledge of the session key sk to obtain the plain text message m=f(c,sk).

Session key agreement between two communications hosts may be achieved using public key cryptography. (See e.g., U.S. Patent Nos. 5,222,140, 5,299,263).

Before discussing public key cryptographic techniques, it is useful to provide some background information. Most practical modern cryptography is based on two notorious mathematical problems believed (but not proven) to be hard (i.e., not solvable in polynomial time, on the average). The two problems are known as Factorization and Discrete-Log. The Factorization problem is defined as follows:

Input: N, where N=pq where p and q are large prime numbers

Output: p and/or q.

The Discrete-Log problem is defined as follows:

Input: P,g,y, where $y \equiv g^x \bmod P$, and P is a large prime number

Output: x.

(The Discrete-Log problem can be similarly defined with a composite modulus N=pq).

Based on the Factorization and Discrete-Log problems, some other problems have been defined which correspond to the cracking problems of a cryptographic system.

One system of such a problem which has previously been exploited in cryptography (see, e.g., H.C. Williams, "A Modification of RSA Public-Key Encryption", IEEE Transactions

on Information Theory, Vol. IT-26, No. Nov. 6, 1980) is the Modular Square Root problem, which is defined as follows:

Input: N,y, where $y \equiv x^2 \bmod N$, and N=pg, where p and q are large primes

Output: x.

Calculating square roots is easy if p and q are known but hard if p and q are not known. When N is composed of two primes, there are in general four square roots mod N. As used herein, $z \equiv \sqrt{x} \bmod N$ is defined to mean that x is the smallest integer whereby $z^2 \equiv x \bmod N$.

Another problem is known as the Composite Diffie-Hellman (CDH) problem, which is defined as follows:

Input: N, g, $g^x \bmod N$, $g^y \bmod N$, where N=pq and p and q are large primes.

Output: $g^{xy} \bmod N$.

It has been proven mathematically, that the Modular Square Root and Composite Diffie-Hellman problems are equally difficult to solve as the above-mentioned factorization problem (see, e.g., M.O. Rabin, "Digitalized Signatures and Public Key Functions as Intractable as Factorization", MIT Laboratory for Computer Science, TR 212, Jan. 1979; Z. Shmuely, "Composite Diffie-Hellman Public Key Generating Schemes Are Hard To Break", Computer Science Department of Technion, Israel, TR 356, Feb. 1985; and K.S. McCurley, "A Key Distribution System Equivalent to Factoring:, Journal of Cryptology, Vol. 1, No. 2, 1988, pp. 95-105).

In a typical public-key cryptographic system, each user i has a public key P_i (e.g., a modulus N) and a secret key S_i (e.g., the factors p and q). A message to user i is encrypted using a public operation which makes use of the public key known to everybody (e.g., squaring a number mod N). However, this message is decrypted using a secret operation (e.g., square root mod N) which makes use of the secret key (e.g., the factors p and q)

C. Network Security Devices

At present, the existing network security products are categorized into two classes: (1) firewalls, such as Janus and ANS and (2) software products, such as encrypted mail, secured http, one time password, etc.

The firewall is a dedicated computer, usually running a Unix operating system. It acts as a filter for incoming and outgoing communications. The firewall is placed as a router between the local area network (LAN) and the outside world. The decision whether to pass a packet is made based on the source and/or destination IP address, and the TCP port number. Some firewalls also have the ability to encrypt data, providing that both sides of the communication employ the same brand of firewall. Some firewalls have a personal authentication feature.

Software products are based on the premise that the computer on which they are installed are secured, and protection is only needed outside on the network. Thus, such software products can easily be bypassed by breaking into the computer. A typical scheme is when an intruder implants a "Trojan Horse" on a computer which sends him an unencrypted copy of every transaction. Sometimes, it is even done as a delayed action during the off-hours when the computer is not likely to be supervised.

In addition, there are authentication products designed to maintain the integrity of the computer against intrusion. These products are based on the premise that they are 100% secured. Once the product is compromised, it becomes totally ineffective. Sometimes, careless use by the one user may jeopardize all other users of the product.

Firewalls are more effective in maintaining network security. However they are very expensive. Their price range is between $10,000 and $50,000, plus the price of the hardware. They require a high level of expertise to install and maintain. The most sophisticated and effective firewalls require a specially trained technician or engineer for their maintenance. The special training cost is up to $10,000 per

person, and the salary adds $60,000 to $120,000 or more per annum to the cost.

Firewalls have to be constantly maintained, modified, and monitored in order to yield reasonable security. They only cover the TCP part of the Internet Protocol and not the UDP part. Thus, they do not provide security to NFS (Network File Services) and many client/server applications.

The firewall is a full service computer which can be logged into for maintenance and monitoring. Thus, it can be broken into. Once a firewall is compromised it loses its effectiveness and becomes a liability rather than a security aid. Firewalls only protect the connection between a LAN and a WAN (Wide Area Network). It does not protect against intrusion into a particular host from within the LAN.

In view of the foregoing, it is an object of the present invention to provide a network security device which overcomes the shortcomings of the prior art network security devices.

In particular, it is an object of the present invention to provide a hardware device to provide network security for individual hosts attached to a network.

Summary of the Invention

The network security device of the present invention comprises a first network interface connected to a protected client, a second network interface connected to a portion of a network, and a processing circuit connected to both interfaces. Illustratively, the portion of the network to which the protected client is connected is an Ethernet and the first and second network interfaces are Ethernet interfaces. The processing circuit connected in between the two interfaces may be a conventional CPU such as an Intel 486 DX2-66 or a Pentium. Alternatively, the processing circuit may be implemented as one or more ASICs (Application Specific Integrated Circuits) or a combination of ASICs and a CPU. A communication from the protected client, goes from the client, to the first interface, to the processing circuit, to the second interface and into the network. Similarly, a

communication received from the network, goes from the second interface, to the processing circuit, to the first interface and to the protected client.

Preferably, the network security device is a sealed device and it cannot be logged into. It has the same IP address as the protected client.

A number of important functions are performed by the inventive network security device. The network security device learns the MAC and/or IP address of its client and locks itself to these addresses. To lock, the MAC and/or IP address is stored in a permanent memory of the network security device. A packet arriving from the client will not be passed into the network if the packet has a MAC and/or IP address different from that which is stored in memory. Thus, the protected client is unable to change its MAC and/or IP address. This prevents the protected client from emulating the MAC and/or IP address of another client in the network. When a packet arrives at the first network interface from the client, the processing circuit substitutes a MAC address of the network security device for the MAC address of the protected client. Then the packet is passed to the second interface and into the network. The same translation is performed in reverse for packets arriving at the network security device from the network side.

Packets received from the protected client are encrypted using an encipherment function such as IDEA, FEAL or DES before being transmitted via the network to a destination. Similarly encrypted packets received from a destination are decrypted. Such encryption and decryption requires a common session key to be possessed jointly by the protected client and the destination (the destination being a protected client of another network security device located someplace else in the network).

The common session key is obtained using a public key cryptography technique. Thus, both of the protected clients (hereinafter labeled i and j) have public keys P_i, P_j and secret keys S_i, S_j. The public keys P_i and P_j have a static

part and a dynamic part which is updated periodically. The secret keys S_i, S_j also have a static part and a dynamic part.

To aid in the key exchange, the network security device maintains two databases. A static database contains information about secured hosts or nodes in the network. A secured host or node is a host or node that is protected by a network security device. Each entry in the static database contains information about a particular secured host, i.e., the host IP address, time entered in the database, and the host's permanent public key.

A dynamic data base contains information about secured and unsecured hosts. Each entry in the dynamic database includes a host's IP address, a flag indicating whether or not the host is secured, a flag indicating whether the host is in transition (i.e., in the middle of a key exchange), and a pointer to a common secret session key.

The protocol used by the network security device of host i to agree on a common session key with a network security device of host j is as follows:

Consider a communication from host i to host j. The communication arrives at the network security device of host i from host i. The network security device checks if host j is in the dynamic database. If host j is in the dynamic database, it is determined if the dynamic database has a common session key for communication between host i and host j. If there is such a common session key, the communication from host i is encrypted using the common session key and transmitted to host j. If there is no common session key, then host i sends the dynamic part of its public key P_i to host j and host j replies by sending the dynamic part of its public key P_j to host i. The exchange of dynamic parts of the public keys may be encrypted using the static part of the public keys, which may be obtained from the static databases at host i and host j. The common session key is then calculated according to a Diffie-Hellman technique for example:

Assume that $P_i = g^{si} \bmod N$ where N is either prime or the product of two primes. Assume that $P_j = g^{sj} \bmod N$. After the exchange of public keys, the network security device at host i calculates $\eta = P_j^{si} = g^{sjsi} \bmod N$. Similarly, the network security device at host j calculated $\eta = P_i^{sj} = g^{sisj} \bmod N$. Thus, both hosts i and j are in possession of the same number η. This number may then be used as a common session key or used to derive a common session key.

Note that this assumes that there is an entry for host j in the static database of host i. If there is not, the exchange of dynamic public keys is preceded by an exchange of static public keys and the forming of a database entry for host j in the static database at host i. Moreover, if there is no entry for host j in the dynamic database of host i, such an entry will be generated before the dynamic key exchange.

It should be noted that encryption takes place at the IP level so that TCP and UDP packets are encoded.

In short, the inventive network security device has a number of significant advantages.

Like a firewall, the inventive network security device is a hardware/software combination in a preferred implementation. However, it is a sealed "box" and cannot be logged into. Hence, it cannot be compromised the way a firewall can. It is much cheaper than a firewall. Thus, each node in the LAN can be equipped with it. This way, it provides protection inside the LAN as well as outside. The network security device works directly at the IP level. It therefore, covers all types of IP protocols and requires no special configuration to different network applications. Thus, the inventive network security device is maintenance free.

The inventive network security device senses the IP address of the client host and locks itself to it. It requires no installation. Once locked, it does not allow the client host to change its IP address. The inventive network security device also maintains a permanent database of secured hosts. If a request for key arbitration arrives that conflicts with the database, that host is denied

communication.. The combination of the above two features creates a double authentication of the IP address. The inventive security device creates a physical barrier between the client and the network. Thus, preventing attempts to bypass by direct Ethernet communications.

The inventive security device encrypts all communication to the network, including the exchange of dynamic public keys.

Brief Description of the Drawings

FIG 1 schematically illustrates an internet network.

FIG 2 schematically illustrates the architecture of a host in the network of
FIG 1.

FIGs 3A and 3B illustrate the format of a packet transmitted in the network of FIG 1.

FIG 4 illustrates a network security device for use with a host in the network of FIG 1 in accordance with an embodiment of the present invention.

FIG 5 illustrates an entry in a static database maintained by the network security device of FIG 4.

FIG 6 illustrates an entry in a dynamic database maintained by the network security device of FIG 4.

FIG 7 is a flow chart which illustrates a key exchange algorithm utilized by the network security device of FIG 4.

FIG 8 is a flow chart of an ARP handling algorithm utilized by the network security device of FIG 4.

FIG 9 is a flow chart of an IP packet handling algorithm utilized by the network security device of FIG 4.

Detailed Description of the Invention

FIG 4 schematically illustrates a network security device in accordance with an illustrative embodiment of the invention. The security device 10 comprises a first interface 0 which is connected to the client host 12. Specifically, the interface 0 is connected to a network interface in the client host 12 (e.g., an interface 13 of Fig. 2) via a cable or wire 13. The security device 10 comprises a second interface 1 which is connected to a portion of a network 100.

illustratively, the interface 1 is connected to an Ethernet so that the interfaces 0,1 are Ethernet interfaces such as SMC Elite Ultra Interfaces.

A CPU 14 is connected to the interfaces 0,1. The CPU is for example an Intel 486 DX 62-66. A static memory 16 (e.g. flash EEPROM) is connected to the CPU 14 and a dynamic memory 18 (e.g. RAM) is connected to the CPU 14. An optional encryption module 20 performs encryption and large number arithmetic operations. The encryption unit may be implemented as a programmable logic array. Alternatively, the encryption module may be omitted and its function may be carried out using a software program which is executed by the CPU 14. The interface 0 is put in a promiscuous mode. In this mode, the interface 0 passes all communications from the client host 12 that is sensed on the cable 13 to the CPU 14. The network connection is via the interface 1 which is set to the same IP address as the client 12. The network security device 10 responds to the Address Resolution Protocol by sending its own (rather than the client's) MAC address. This adds a level of security by blocking attempts to bypass the device 10 using the Ethernet protocol.

The CPU 14 maintains two databases. One database is a static database stored in the Flash ROM 16. This database contains permanent information about secured nodes in the network, i.e., the node IP address, time entered into the database, the nodes permanent public key. The structure of an illustrative record for a node in this database is shown in FIG 5.

A second database is a dynamic database. The dynamic database contains information about secured and unsecured nodes, i.e., the node IP address, time last updated, a flag indicating whether the node is secured (e.g.,has its own network security device), a flag indicating whether the node is in transition (i.e., in the middle of a key exchange), a pointer to a common secret key with that node. The structure of an illustrative record for a node in this database is shown in FIG 6. The transition flag has three possible values, 0-

not in transition, 1-pending reply from remote host, and 2-pending computation of common key.

The software executed by the CPU 14 has three components: (1) operating system, (2) networking system, (3) key computation algorithms. The operating system and the networking system are both part of a Unix like kennel. The key computation algorithm reside in memory and are signalled into action by the networking system. The operating system is a lobotomized Linux system with all drives taken out except the RAM, disk and Ethernet interfaces. The networking system is for communication, key exchange, encryption, configuration, etc.

The key exchange algorithm is illustrated in Fig. 7. Consider the case where the host client wants to send a communication to a node in the network whose IP = A. When the communication arrives at the network security device of the host client (step 60 of Fig. 6), the dynamic data base (DDB) is checked to determine if there is an entry for node A in the dynamic data base (step 61). If there is an entry for node A in the dynamic data base, a check is made to see if a common session key for node A and the protected client has expired (step 62).

If the common session key has not expired the packet is encrypted using the session key and an encipherment function such as IDEA (step 63). If the common session key has expired, the dynamic data base entry for the node IP=A is marked in unsecured transition (step 64) which means a key exchange is taking place.

The exchange of the dynamic parts of the public keys of the host client and the node with IP=A proceeds as follows. The host client (i.e., the source) sends its dynamic public key and IP address to the node with IP=A (the destination) (step 65) and waits five seconds for a reply (step 66). The dynamic public key of the host may be encrypted with the static public key of the node with IP=A. The reply is the dynamic public key of the destination node with IP=A. This may be encrypted with the static public key of the host

client. The steps 65 and 66 are repeated three times. If no reply is received (step 68) from the destination, the source network security device marks the transition off (step 67) in the DDB entry for the destination.

If a reply is received the transition flag for the destination in the DDB of the network security device of the host is set to 2 (step 69). Then a common session key for the source and destination is calculated by the network security device of the source (step 70) using for example a Diffie-Hellman technique as described above. The common session key is then entered into the DDB of the source network security device (step 71) and the transition flag for this DDB entry is marked 0 (step 72).

The exchange of dynamic public keys and the calculation of a common session assumes that there is an entry for the destination node with IP=A in the static data base (SDB) of the source network security device and in the dynamic data base of the source network security device. If these entries do not exist, they may be created prior to dynamic public key exchange (steps 65-69).

If there is no DDB entry for node IP=A, (step 61) an entry is created (step 81) and the transition flag for this entry is marked in unsecured transition (step 82). A check is then made to determine if the SDB of the source network security device has an entry for node IP=A (step 83). If there is such an entry, proceed with dynamic key exchange (step 65 etc.). The source network security device transmits the permanent public key of the source to node IP=A (step 84) and waits five seconds for a reply (step 85) . The steps 84 and 85 may be repeated several, e.g., three times.

If a reply is received (step 86), an entry is created in the SDB (step 87). If no reply is received, the transition flag in the entry in the DDB is marked transition off (step 67).

As indicated above in a preferred embodiment, the network security device is a sealed box which cannot be logged into. The network security device 10 senses the IP (and/or MAC)

address of the client host 12 and locks itself to it. Once the network security device is locked to the address, the client is presented by the network security device from changing its IP (and/or MAC) address.

The Address Resolution Protocol (ARP) is the protocol which is used to resolve an IP address into a matching Ethernet machine (MAC) address which is the actual address to which the network interface responds.

The inventive network security device uses ARP (Address Resolution Protocol) to configure itself and hide the client host.

Fig. 8 Shows how the inventive network security device processes an ARP request with IP=B. The request may arrive from the host at interface 0 or form the network at interface 1 (step 100). If the request arrives via interface 0, the network security device determines if it is configured (step 102). If the network security device is not configured, it configures itself (step 103). Configuration involves storing in a permanent memory, the IP address and/or the MAC address of the host. After configuration, the CPU in the network security device replaces the MAC address of the source with the MAC address of interface 1 (step 104) and sends the request to interface 1 (step 105). The request is then transmitted to its destination via the network.

If the host's network security device is already configured (step 102), it is determined if the request is a reply (step 106). This is done by checking the destination MAC field. In a reply this field is not zero. If the request is not a reply, MAC address translation takes place according to step 104 and the request is sent to interface 1 (step 105) for transmission into the network. If the request is a reply, it is determined if the source IP address in the request matches the IP address now permanently stored in memory (step 107). If there is a match, the request is dropped (step 108) since the network security device has already answered the ARP. If there is not a match, the system is shut down (step 109).

It the request arrives via the network at interface 1, the request is passed to interface 0 (step 111). If the request is a reply (step 112), pass the request to the interface 0 (step 113). If the request is not a reply, the request is answered using the MAC address of interface 1 (step 114).

Fig. 9 illustrates a packet handling algorithm utilized by the inventive network security device. Illustratively the packet arrives with the source address IP=C (step 200). The packet may arrive from the host at interface 0 from the network at interface 1.

First consider the case where the packet arrives from the host at interface 0. If the packet carries an ICMP (Internet Control Message Protocol) or IGMP (Internet Gateway Message Protocol) identification (step 201), the packet is passed to the interface 1 without encryption. However, the source MAC address in the packet is translated to the MAC address of interface 1 (step 202). ICMP and IGMP Packets are not addressed to a destination host. Rather these packets are utilized by intermediate entities in the network, e.g., routers, for various functions.

If the destination to which the packet is addressed is insecure, the packet is dropped (step 203, 204). The device may be in a secured/unsecured mode (special order). In such case the packet will be sent unchanged.

Next, it is determined if the packet contains a part of a message that has been fragmented (step 205). If the packet contains a fragment, the fragments are collected (step 206) and the message is encrypted (step 207). The encryption takes place using the common session key and an encipherment function. If the encrypted message is too long for the particular LAN (step 208), it is fragmented (step 209). An encrypted packet is then transmitted to interface 1 for transmission into the network (step 210).

An encrypted packet carries a signature in the protocol IS part of the IP header. This indicates that the packet is encrypted. The IP address of a packet is not encrypted,

otherwise the packet could not be routed through the network.

The case where the packet arrives via the network at interface 1 is now considered. If the packet is an ICMP or IGMP packet (step 220) no decryption is necessary and the packet is sent to interface 0 (step 221). If the packet is a key exchange packet (step 222) the packet is processed according to the key exchange protocol (step 223). If the packet is not encrypted (step 224) the packet is dropped (step 225). The device may be in a secured/unsecured mode (special order). In such cse the packet will be sent to the client unchanged. If the packet is encrypted but the network security device does not have the key (step 226), the key exchange protocol is carried out (step 227) and the packet is dropped (step 228). If the key is available in the dynamic data base of the network security device, the packet is decyrpted (step 229) and sent to interface 0 (step 230).

For packets received from the network the MAC address of the network security device is translated into the MAC address of the client. For packets received from the protected client, the MAC address of the client is translated into the MAC address of the network security device.

In short, a unique network security device has been disclosed. Finally, the above described embodiments of the invention are intended to be illustrative only. Numerous alternative embodiments may be devised by those skilled in the art without departing from the scope of the following claims.

CLAIMS

1. In combination,

a node in a network, and

a security device connected between said node and a network,

said security device preventing said node from changing its IP address so that said node cannot emulate another node in said network.

2. The combination of claim 1 wherein said security device stores said IP address in a permanent memory and blocks packets received from said node with an IP address different from said stored IP address from entering said network.

3. The combination of claim 1 wherein said security device comprises a first network interface connected to said node, a second network interface connected to said network, and a processing circuit connected to said first and second interfaces for encrypting user data but not IP addresses.

4. A method for preventing a node in a network from emulating another node in said network comprising the steps of:

(1) in a memory in a network security device connected between said node and network, permanently storing a characteristic address of said node, and

(2) using said network security device to block packets received at said network security device from said node with a source address difference

* C L M S *

from the permanently stored address from entering said network.

5. The method of claim 4 wherein said characteristic address is an IP address or a MAC address.