Security vulnerabilities in the Internet and possible solutions

1. Introduction

The foundation of today's Internet is the TCP/IP protocol suite. Since the time when these specifications were finished in early 1980's, the Internet has grown from a network connecting a small community of researchers to its present state - a huge global network connecting people of all types. The TCP and IP protocols were designed assuming the users trusted each other and lacks many features desirable in an insecure network. The widespread availability and use of Internet has exposed many of the security flaws in these basic building blocks.

The following section provides examples of attacks carried on TCP/IP protocol suite. Section 3 identifies requirements for a secure network. Section 4 and 5 discuss the most common security schemes used on the Internet: SSL and IPSec. Section 6 explores the issue of denial of service attacks. Section 7 discusses how to create a secure network.

2. Attacks on TCP/IP

When TCP/IP was designed, security was not a primary concern. However its extensive usage has uncovered a number of weaknesses. Below are some well-known vulnerabilities of the TCP/IP protocol suite.

2.1 SYN-Flood, Smurf and UDP flood attacks

The TCP connection establishment process consists of a three-way handshake. Host A sends SYN message containing initial sequence number for A, Host B replies with SYN & ACK containing its own initial sequence number and acknowledging sequence number of A. Finally host A sends ACK to acknowledge the sequence number of B. To allow hosts with high latency to successfully make connections, host B waits for 75 seconds for the acknowledgement from A. During this time, it has to keep track of the partially opened connection in a listen queue. (Figure 1)

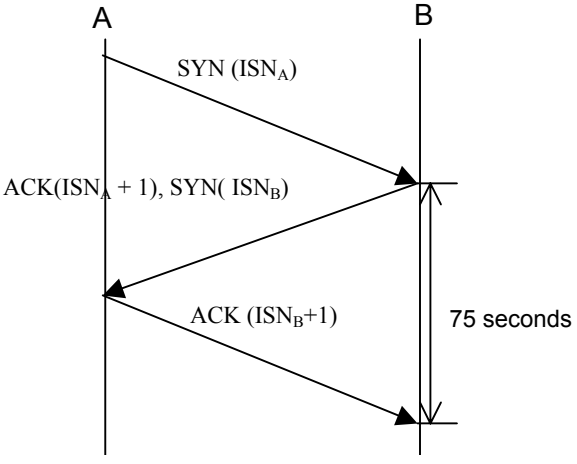

Figure 1: Regular TCP connection setup

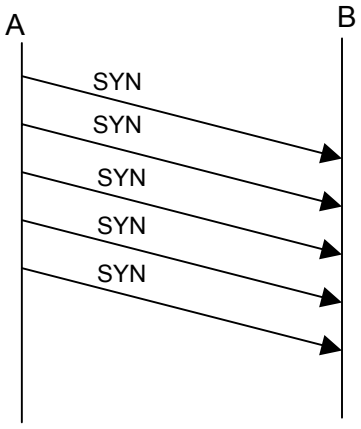

Figure 2: SYN Attack

A malicious host can exploit this to bombard a server with SYN messages so that the server runs space in the listen queue and can't accept any other connections. This attack is further used to implement other attacks ([2], [3]). A more potent version of this attack has been used lately, employing a large number of (possibly compromised) hosts to generate SYN requests. This is a common kind of Distributed Denial of Service attack. The attacker can easily avoid detection, by labeling incorrect source address in the IP header.

DDoS attacks have also been carried out using ICMP flooding (called smurf attacks). Here, the attacker sends an ICMP echo request packets to broadcast addresses with the source address being the IP address of the victim. Consequently, the victim receives a flood of echo replies. Similar attacks have been carried out using UDP.

2.2 IP spoofing

IP spoofing is an attack where the attacker pretends to be sending data from an IP address other than its own [1]. The IP layer does no other authentication and believes the source address in the packet. Many higher-level protocols also make this assumption. Thus simply by mislabeling the source address, the attacker can gain unauthorized access. However, communication in such a case is going to be one-way since the victim will continue to send packets to the source address (To prevent this host from responding, the attacker typically brings it down using DoS attacks). Also, the attacker needs to correctly guess the sequence number (32 bit) sent by the victim in the final ACK otherwise the connection wouldn't be established. However, both the problems can be overcome. In Berkeley systems, the initial sequence number (ISN) variable is incremented by a constant amount once per second, and by half that amount each time a connection is initiated. Thus, if one initiates a legitimate connection and observes the ISN used, one can calculate with a high degree of confidence, ISN used on the next connection attempt.

2.3 ICMP Misuse

ICMP is the basic network management protocol of the TCP/IP protocol suite and provides a number of methods that can be abused [1], [3]. ICMP may be used for targeted denial of service by sending Destination Unreachable and Time to Live Exceeded messages. These messages cause the receiving entity to drop the connection. Although the messages require the attacker to know local and remote port number.

ICMP redirect messages can be used to send packets through attacker's host (Here the attacker needs to be on local network). ICMP can also be used for port scanning and network mapping [5]. This information can be further used to carry out other attacks. Other attacks relying on ICMP include OS scanning, Inverse Mapping and Tunneling (for communicating behind firewalls).

2.4 Connection Hijacking

An interesting variant of IP spoofing can be used by a host to insert itself in the middle of a connection between two hosts. With IP spoofing, the attacker might face

difficulty in bypassing systems that do authentication using passwords etc. Connection hijacking can effectively be employed in such cases. The attacker can send a TCP reset message to the client (with source address labeled as that of the server) and carry on sending packets to the server. This requires the knowledge of IP addresses and ports being used at both ends in addition to sequence numbers. All this information is available if the attacker can see the traffic.

2.5 Snooping

The initial version of TCP/IP provided no encryption for payload. As a result most traffic today is exchanged in plaintext and is vulnerable to snooping en route to destination. Snooping can easily be deployed where the physical media is shared (e.g. Ethernet, Wireless etc.) as well as on routers and gateways.

3. Requirements

Protecting an abstract resource such as information is usually more difficult than providing physical security because information is elusive. Information security encompasses many aspects of protection:

Data Integrity: A secure system must protect information from unauthorized change. Connection Hijacking and IP Spoofing are examples of attacks on data integrity.
Data Availability: The system must guarantee that outsiders cannot prevent legitimate access to data. ICMP misuse and Denial of service are attacks on data availability.
Confidentiality: The system must prevent outsiders from making copies of data as it passes across a network. Packet snooping on transmission media and ICMP redirect attacks are examples where confidentiality is compromised.
Authentication: The system must allow the communicating entities to validate each other's identity. IP spoofing is an example where the basic authentication mechanism (IP address) is subverted.
Replay avoidance: To prevent outsiders from capturing copies of packet and using them later, the system must prevent a retransmitted copy of a packet from being accepted. Replay attacks are typically carried out when the packet contents are encrypted. In such cases one way to generate valid packets is by resending a copy of valid packet.

There are a wide variety of schemes each addressing one or more aspects of security. As we look at examples in the following sections, we will point out which particular aspects the scheme covers.

4. Secure Sockets Layer (SSL)

The SSL protocol runs between TCP/IP and application layer protocols. It provides authentication, confidentiality, integrity and replay protection on transport layer. The protocol can be divided in two stages - SSL handshake protocol and SSL record protocol. During the SSL handshake, the client sends to the server a random number along with its SSL version and cipher settings. The server also sends client its own SSL version, cipher settings, a random number and additionally its own certificate. The client

verifies the server's certificate and generates premaster secret for the session. It encrypts this premaster secret using the server's public key (inside the certificate) and sends the encrypted secret to the server. The server then decrypts this message using its private key and performs a series of steps on this to generate the master secret. Both client and server utilize this to arrive at a session key. Any further communication is carried out using symmetric key encryption with the session key. The SSL is based on digital certificates to authenticate and exchange session key.

Digital Certificates consist of server's distinguishing name, its public key, validity period of the key, Certification authority (CA) and certification authority's signature of the certificate. On getting this certificate, the client checks for the validity period. Then it sees whether the CA is one of its trusted CAs. In such a case, its public key is already available to the host (Internet browsers come with a list of CAs and their public keys). The client uses the public key to validate the CA's signature and is thus sure that the public key belongs to the entity mentioned in the certificate. Digital certificates thus use pre-established trust relationship with CAs to create new trust relationships. Digital certificates are issued by a certifying authority after verifying the identity of an entity. Verisign is an example of certifying authority.

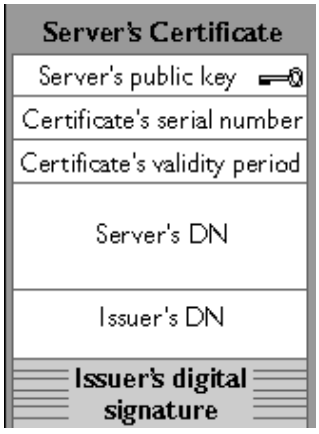

Figure 3: Digital Certificate

5. IP Security (IPSec)

The most ambitious of all the efforts to integrate security into the Internet has been directed at the IP layer. IPSec [7], as the architecture is called, is a framework (as opposed to a single protocol or system) for providing different security features discussed in Section 3. As an architecture, IPSec is highly flexible in that it provides three degrees of freedom. First, it allows user to select from amongst a wide variety of encryption algorithms and security protocols. Second, it allows users to select which aspects of security they want to address. Third, IPSec allows user to select the granularity with which the security services are applied. For example, IPSec can be used to protect both "narrow" streams (e.g., individual TCP connections) as well as "wide" streams (e.g., all packets flowing between a pair of gateways).

IPSec consists of two pieces. A pair of protocols that implement the available security services - Authentication Header (AH) [8], which provides data integrity, authentication, replay protection and Encapsulating Security Payload (ESP) [9], which provides confidentiality. These can be used together or individually to provide exact services the user desires. The second aspect is key management protocol: Internet Security Association and Key Management Protocol (ISAKMP) [10].

Two entities, which need security, establish a Security Association (SA). A security association defines all parameters (e.g., encryption algorithm, key generation technique, authentication mechanism, key, lifetime of key etc) required to execute the

security services of AH and ESP from one host to another (2 SAs are required one in each direction). ISAKMP's role is to define procedures and packet formats to establish, negotiate, modify and delete security associations. To avoid the overhead of exchanging this information repeatedly, the two entities exchange this information in the beginning and maintain the information corresponding to a Security Parameter Index (SPI). The SPI is inserted in AH or/and ESP.

5.1 IPSec Authentication Header

AH is inserted above the IP header. Its format is shown in Figure 4 below. AH replaces the Protocol field in IP header to 51. The original value in this field is recorded in the Next Header field of AH. The sequence number field starts from 0 and provides replay protection. Different algorithms including digital signature, symmetric key algorithms, HMACs may be used to generate authentication data. For example, HMAC with MD5 can be used to generate authentication data. Here a hash is created on the data and a user key. Note that authentication data ensures data integrity and authentication but the payload is still available in plaintext. As mentioned earlier, the choice of algorithms, key etc. is carried out using ISAKMP. The SPI field is used by the receiving host to identify the security association to which the packet belongs.

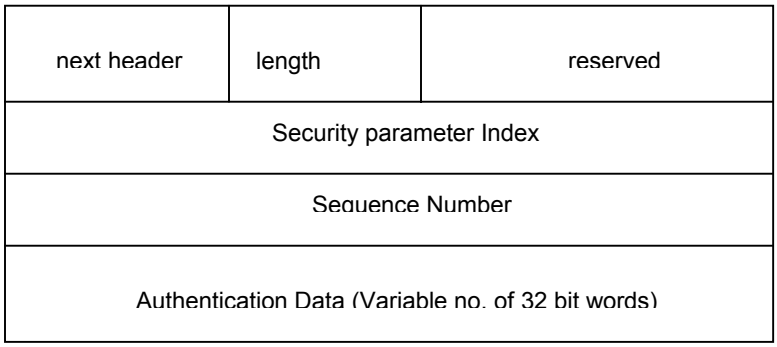

Figure 4: AH Format

5.2 IPSec Encapsulating Security Payload

ESP header is designed to provide a mix of security services. The ESP header (Figure 5) is inserted after the IP header and before the upper layer protocol header (when used between a pair of hosts), or before an encapsulated IP header (when used to tunnel between a pair of security gateways). SPI, sequence numbers and authentication data have the same usage as in AH. Payload data is encrypted using encryption algorithm and key defined in security association. One of the common choices for encryption algorithm is DES. DES is used to encrypt contents of the whole packet (excluding IP).

One of the most popular ways to use the ESP is to build an IPSec Tunnel between two gateways. For example, a corporation wanting to link two sites using Internet or shared links could configure a tunnel from a gateway at one site to the other. Such a tunnel provides data integrity, confidentiality and authentication.

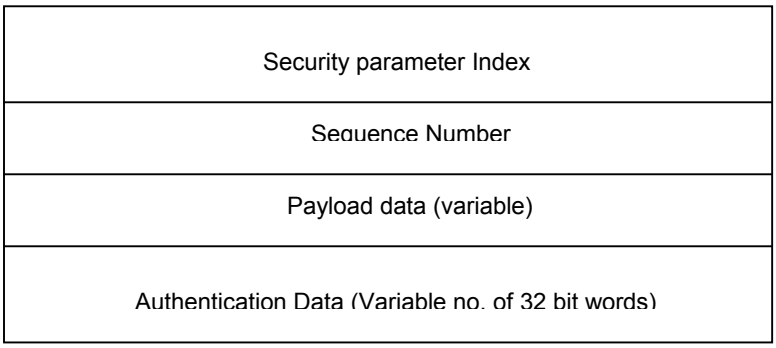

Figure 5: ESP Format

6. Countering denial of service attacks.

The basic issue behind denial of service attacks is that the protocol allows anonymity, which allows a node to engage in destructive behavior without fear of identification. Second, the protocol is susceptible to asymmetric resource usage. This is a fundamental weakness of the protocol which requires change in it. One way to address the first issue is for the edge routers to ensure that the source address in the packet is labeled correctly. This is an easy and efficient approach to ensure that a packet can be traced back based on the IP address and thus works by deterrence. However, this scheme is not widely implemented since there is no incentive for the originating edge router to do extra processing. Other ways to provide authentication, including digital signatures, are susceptible to denial of service attacks using replay of a valid packet. They also consume resources in checking signature and therefore cannot be employed. Another approach is using IP traceback to communicate with the routers closer to the source recursively until the originating router is reached. However, this scheme is slow because it operates across many autonomous systems and therefore different administrations. To address the second issue, the protocol should maintain minimal information at the server end. Instead the server can put the state (ISN etc.) in the packet when it communicates with the client and the client would have to return back this state in the ACK. This, however, prevents the server from retransmitting the SYNACK when the SYNACK gets lost for a valid client.

7. A secure communication network

A secure communication network has to account for all the different aspect of security discussed earlier. An easy way to implement such a network would be to provide a public key/private key pair to each IP Address along with a digital certificate, each node also has the public key of the certifying authority. The key pairs could be based on one of many public key cryptography schemes. Lets illustrate how secure communication can take place using Diffie-Hellman public key scheme. For host X to initiate communication with Y, it only needs to send its certificate. Y responds with its own certificate. This is all the information required to arrive at the implicit shared secret key (using Diffie-Hellman algorithm) at both nodes. The node X has a private key x and

a public key g^x mod p. Similarly, node Y has a private key y and a certified public key g^y mod p. The pair X and Y share the mutually authenticated secret g^{xy} mod p. This shared secret is implicit. It does not need to be communicated explicitly to either node. Each node can compute this secret based on knowledge of the other node's identity and public key (embedded in certificate). This mutually authenticated long-term secret is used to exchange a symmetric key K for the session. Each element in the network can communicate with any other element securely.

The initial processing cost is the sum of cost of validating certificate and calculating shared secret key which is 11 ms (1024 bit Diffie Hellman) [11]. The symmetric key based encryption method limits the throughput of the server to 35 Mbps (single Celeron 850MHz only performing 3DES encryption - 168 bit keys). Assuming average data transfer per session to be 100KB, the throughput of the system taking initialization into account would reduce the throughput to 26 Mbps. To avoid initial cost of calculation of the shared key used to exchange the symmetric key for the session, one could cache it along with the public index of the other node.

An important aspect of this scheme is assignment of the certificate to the IP address this way in case a node's private key is compromised, the node could be assigned a new IP address (and a new public key). All communication is based on name mapping which could be update in the DNS. Another important consideration in such a network would be that ingress filtering should be performed otherwise it will be easier to bring a node down with DoS since more resources are consumed in the connection setup.

References

[1] Steven M. Bellovin, "Security Problems in the TCP/IP Protocol Suite", Computer Communications Review, pp. 32-48, April 1989

[2] Gary C. Kessler, "Defenses Against Distributed Denial of Service Attacks", http://rr.sans.org/threats/DDoS.php

[3] Dave Dittrich, "Distributed Denial of Service (DDoS) Attacks / tools", http://staff.washington.edu/dittrich/misc/ddos/

[4] Chris Low, "ICMP Attacks Illustrated", http://rr.sans.org/threats/ICMP_attacks.php

[5] Mark Spencer, "Cheops", http://www.marko.net/cheops/

[6] Douglas E. Comer, "Internetworking with TCP/IP, Principles, Protocols, and Architectures - Volume 1", Section 32, 2000

[7] Stephen Kent et al, "Security Architecture for the Internet Protocol", RFC 2401, November 1998

[8] Stephen Kent et al, "IP Authentication Header", RFC 2402, November 1998

[9] Stephen Kent et al, "IP Encapsulating Security Payload", RFC 2406, November 1998

[10] Douglas Maughan et al, "Internet Security Association and Key Management Protocol", RFC 2408, November 1998

[11] Wei Dai, "Crypto++ Benchmarks", http://www.eskimo.com/~weidai/benchmarks.html