

November 2000

Web Security for E-Commerce

Robert J. Boncella

Washburn University, zzbonc@washburn.edu

Follow this and additional works at: <https://aisel.aisnet.org/cais>

Recommended Citation

Boncella, Robert J. (2000) "Web Security for E-Commerce," *Communications of the Association for Information Systems*: Vol. 4 , Article 11.

DOI: 10.17705/1CAIS.00411

Available at: <https://aisel.aisnet.org/cais/vol4/iss1/11>

This material is brought to you by the AIS Journals at AIS Electronic Library (AISeL). It has been accepted for inclusion in Communications of the Association for Information Systems by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.



Communications of the **I**nsformation **S**ystems
Association for **I**nformation **S**ystems

Volume 4, Article 11
November 2000

WEB SECURITY FOR E-COMMERCE

Robert J. Boncella
Computer Information Science Department
and
School of Business
Washburn University
zzbonc@washburn.edu

TUTORIAL

WEB SECURITY FOR E-COMMERCE

Robert J. Boncella
Computer Information Science Department
and
School of Business
Washburn University
zzbonc@washburn.edu

ABSTRACT

This tutorial presents an overview of the major categories of Web site attacks, their effects, and possible countermeasures. The focus is the Web security necessary for a reasonable guarantee of secure e-commerce.

The tutorial is intended for those who have little or no knowledge of Web security and its importance to e-commerce. It provides a basic understanding of the issues, the techniques, and the nomenclature used. An annotated bibliography points the reader to additional sources on specific topics.

Keywords: Web security, e-commerce, digital certificates, digital envelope, digital signature, encryption, private/public key, SET - secure electronic transaction, and SSL – secure socket layer.

Note: Web security involves a large amount of nomenclature. To aid the reader, these new terms are shown in **bold** type. Table 1 lists the terms in alphabetic order and shows where they are defined.

Table 1
Web Security Terms

TERM	SEC.	TERM	SEC.	TERM	SEC.
Access control	4	HTTP	1	Proxy servers	3
Access security	3	HTTPD	1	Public/private key	3
Application layer	1	Information security	3	Remote host	1
Application level proxies	3	Intermediate node	1	Request/response	1
Asymmetric key	3	Internet	1	Router	3
Authentication	3	Intranet	1	Secret key	3
Circuit level proxies	3	IP spoofing	4	Secure channels	3
Client	1	ISP	1	Segment	1
Confidentiality	3	Kerberos	4	Server	1
Cryptography	3	Local host	1	Session layer	1
Data link layer	1	Message digest	3	SET- Secure electronic transaction	3
Datagram	1	Message integrity	3	Signed certificate	3
DCE	4	Negotiated secure session	3	Smart card type	4
Denial of service	4	Network layer	1	SSL- Secure socket layer	3
Digital certificates	3	Non-repudiation	3	Symmetric key	3
Digital envelope	3	One time pad	4	TCP/IP	1
Digital signature	3	One-way function	3	TCP/IP SYN attack	4
Distributed denial of service	4	OSI model	1	Transport layer	1
DNS spoofing	4	Package filtering	3	TSL- Transport layer security	3
Encryption	3	Packets	3	Two factor authentication	4
Extranet	1	Paranoid DNS Checking	4	UDP	1
Firewalls	3	Physical layer	1	URL flood	4
Fragmented	1	Ping of death	4	User authorization	4
Hash	3	Port numbers	1	Web browser	1
Header	1	Presentation layer	1	Web server	1
HTML	1	Protocol	1		

I. INTRODUCTION: WEB CONCEPTS FOR E-COMMERCE

CLIENTS AND SERVERS

The World Wide Web (WWW or Web) is implemented by means of an interconnection of networks of computer systems. This interconnection of computer systems provides information and services to users of the Web. Computer systems in this interconnection of networks that provide services and information to users of computer systems are called **Web Servers**. Computer systems that request services and information use software called **Web Browsers**. The communication channel between the Web browser (**client**) and Web server (**server**) may be provided by an Internet Service Provider (**ISP**) that allows access to the communication channel for both the server and client. The communication of the client with server follows a **request / response** paradigm. The client, via the communication channel makes a request to a server and the server responds to that request via a communication channel.

The Web may be viewed as a two- way network composed of three components:

- clients,
- servers, and a
- communication path connecting the servers and clients.

The devices that implement requests and services are both called **hosts** since these devices are "hosts" to the processes (computer programs) that implement the requests and services.

COMMUNICATION PATHS

The communication path between a server and a client can be classified in three ways:

- an **Internet**,
- an **intranet**,
- or an **extranet**.

An internet is an interconnection of networks of computers. However the **Internet** (with an upper case I) refers to a specific set of interconnected computer networks that allow public access.

An **intranet** is a set of interconnected computer networks belonging to an organization and is accessible only by the organization's employees or members. Access to an intranet is controlled.

An **extranet** uses the **Internet** to connect private computer networks or intranets. The networks connected together can be owned by one organization or several. At some point, communication between hosts in an extranet will use a communication path that allows public access.

For a request or response message to travel through a communication path, an agreed upon method for message creation and transmission is used. This method is referred to as a **protocol**. The de facto protocol of the Internet is the TCP/IP protocol. An understanding of the client/server request/response paradigm requires an overview the TCP/IP protocol. The TCP/IP protocol can best be understood in terms of the Open System Interconnection (OSI) Model for Data Communication.

THE OSI MODEL AND TCP/IP

The **Open System Interconnection Model** defined by the International Standards Organization (ISO) is a seven-layer model that specifies how a message is to be constructed in order for it to be delivered through a computer network communication channel. This model is idealized. In practice few communication protocol follow this design. Figure 1 provides a general description of each layer of the model. The sender of the message, either a request or a response message, provides input to the Application layer.

The Application Layer processes sender input and converts it to output to be used as input for the Presentation Layer. The Presentation Layer, in turn, processes this input to provide output to the Session Layer, which uses that output as input, and so on, until what emerges from the Physical Layer is a signal that can be transmitted through the communication channel to the intended

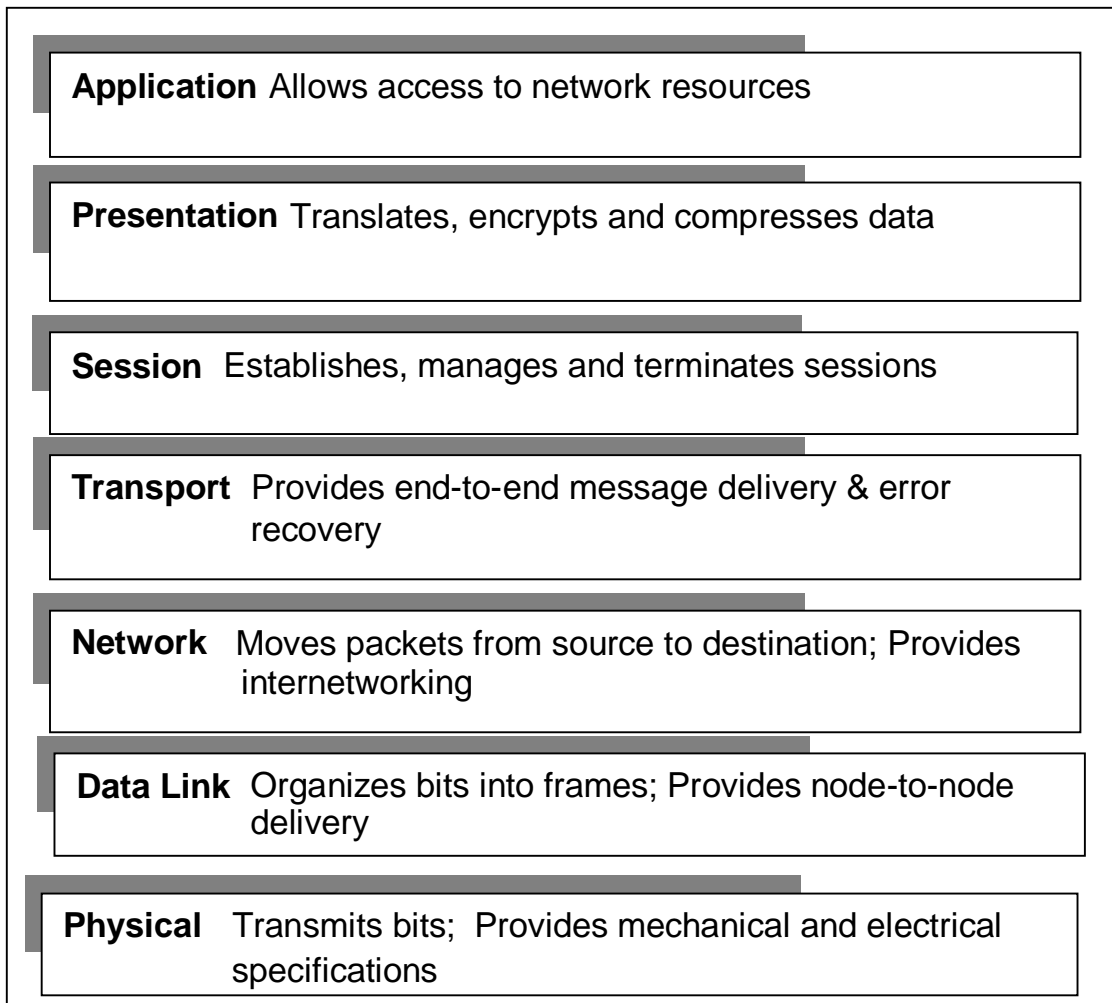


Figure 1. OSI Model

receiver of the message. The receiver's physical layer processes the signal to provide output to its Data Link Layer which uses that output as input and processes that input to provide output the receiver's Network Layer, and so on until that message is accepted by the receiver.

This process is depicted in Figure 2. Figure 2 also illustrates the signal (message) being relayed through the communication channel by means of **intermediate nodes**. An **intermediate node** is a host that provides a specific service whose purpose is to route a signal (message) efficiently to its intended destination.

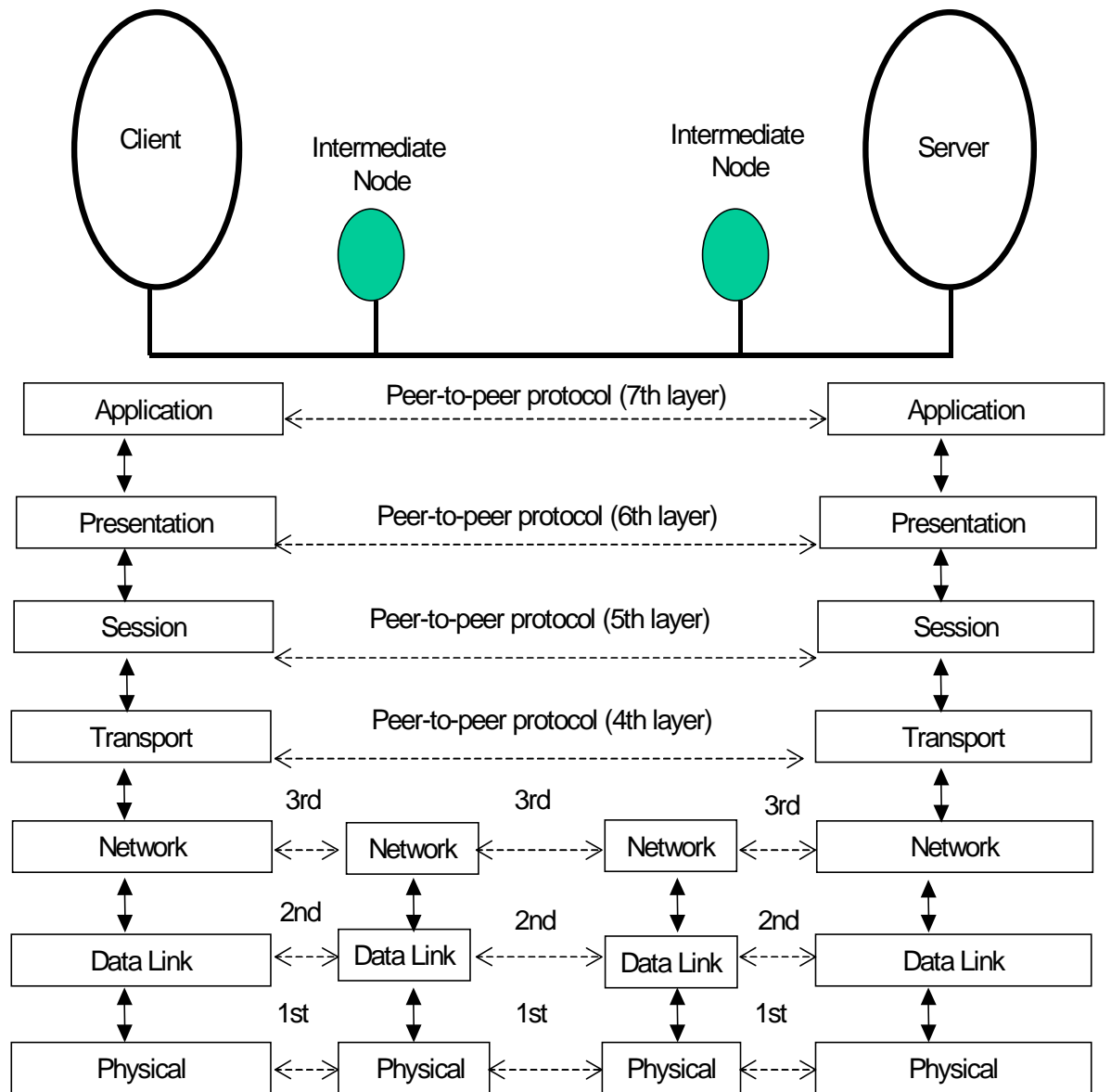
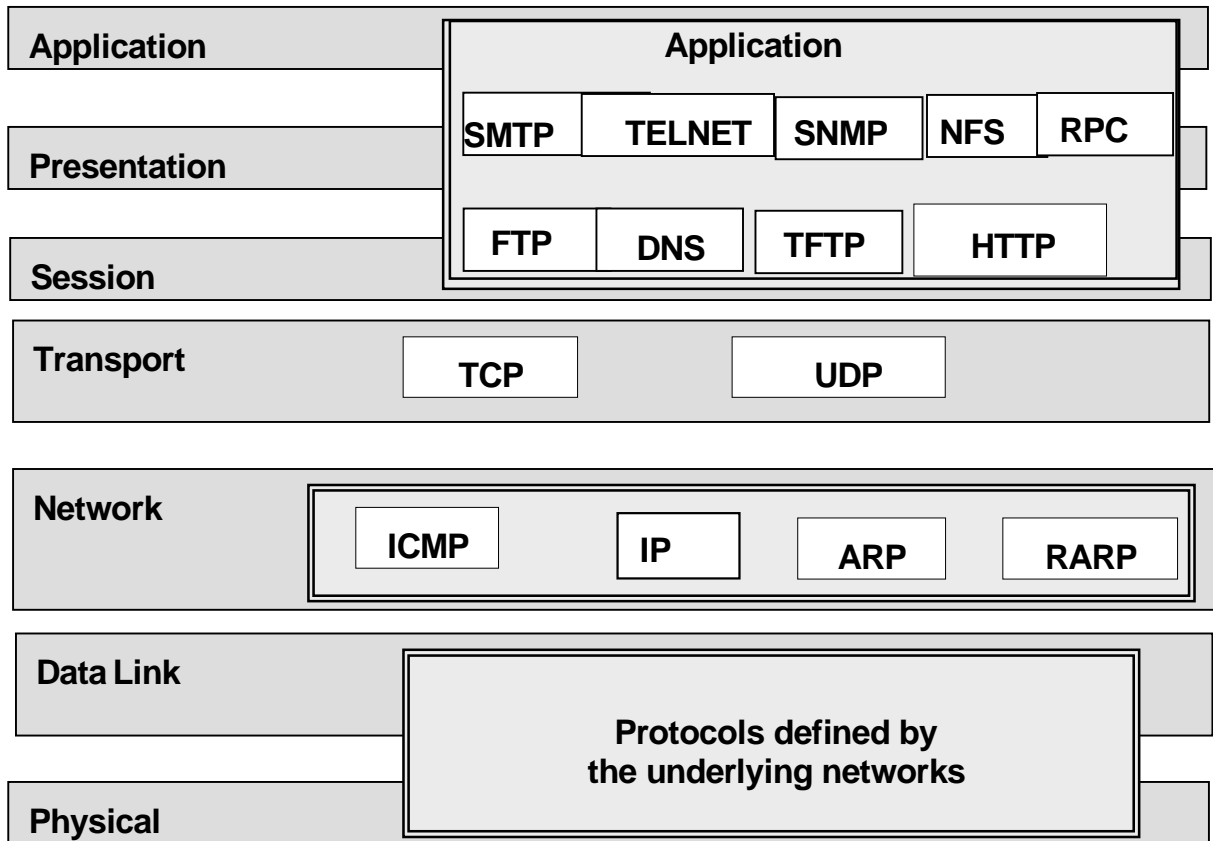


Figure 2. Messaging Delivery Using the OSI Model

Figure 3 depicts the TCP/IP protocol on the OSI Model (TCP/IP is an abbreviation for Transmission Control Protocol/Internet Protocol). For our purposes the **TCP/IP protocol** is made up of four layers. What follows is a brief overview of the TCP/IP protocol. For an introduction to the details of TCP/IP consult Forouzan [2000].



SMTP- Simple mail transfer protocol	TFTP- Trivial file transfer protocol
TELNET- Remote access program	HTTP- Hypertext transfer protocol
SNMP- Simple network management protocol	TCP- Transmission control protocol
NFS- Network file system	UDP- User datagram protocol
RPC- Remote procedure call	ICMP- Internet control message protocol
FTP- File transfer protocol	ARP - Address resolution protocol
DNS- Domain name system	RARP- Reverse address resolution protocol

Figure 3. The OSI Model and The TCP/IP Protocol

The Application Layer contains a number of applications that a user may use as a client process to request a service from a host. The client process is said to run on a **local host**. In most cases, the requested service will be provided by a **remote host**. In many cases there will be a similarly named application on the remote host that will provide the service. For example, the user may open a Web browser and request HTTP (Hyper Text Transfer Protocol) service from a remote host in order to copy an HTML (Hypertext Markup Language) formatted file into the user's Web browser. If the receiving host provides HTTP service, it

will have a process running named HTTPD that will provide a response to the client's request. Note that the users need to specify the host by some naming method and the service they desire from that host. This is taken care of by the use of a Universal Resource Locator (URL) (e.g. <http://www.wasburn.edu>). The Application Layer produces a message that will be processed by the Transport Layer.

The client's request will pass through the local host's Transport Layer. The responsibilities of the Transport Layer are to establish a connection with the process on the remote host that will provide the service requested. This client process to server process connection is implemented by means of **port numbers**. A port number is used to identify processes (programs in execution) uniquely. Unique identification is necessary since local hosts and remote hosts may be involved in a number of simultaneous request/response transactions. The hosts' local operating systems in concert with the TCP/IP protocol concept of port numbers can keep track of which of several responses corresponds to the correct client process request on that local host and which request corresponds to the correct service on the remote host.

The Transport Layer will cut the message into units that are suitable for network transport. In addition to the port numbers, the transport layer adds information that will allow the message to be reconstructed in the receiver's transport layer. Other information is added to these units that allow for flow control and error correction. The output from the transport layer is called a **segment**. The segment is composed of the data unit and a **header** containing the information described above. Figure 4 shows this process.

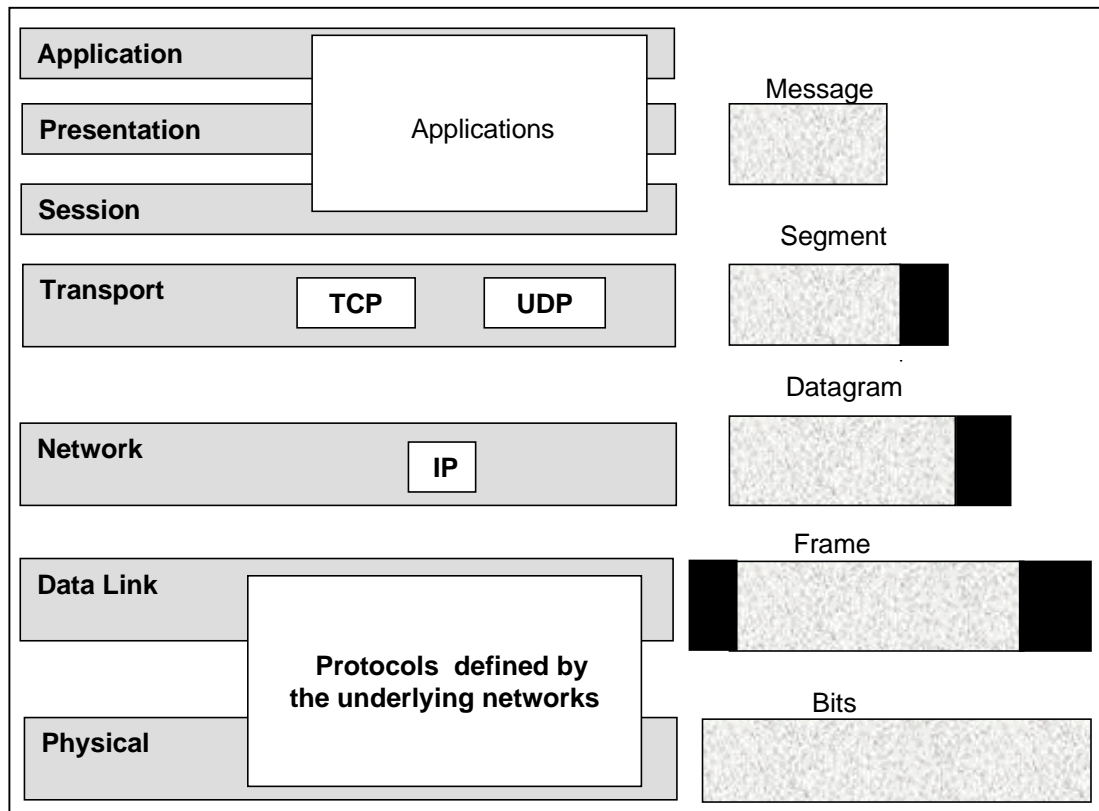


Figure 4. TCP/IP Message Delivery

The output of the transportation layer - a segment - is sent to the Network or IP layer. The responsibilities of the IP Layer include providing the Internet or IP address of the source (requesting) host and destination (response) host of the segment. One important part of the IP address is a specification of the network to which the host is attached. Depending on the underlying physical network, the segments may need to be **fragmented** into smaller data units. The information from the segment header is duplicated in each of these fragments as well as that the header information provide by the network or IP layer. The output of the IP layer is called a **datagram**.

The datagram is passed to the lowest layer where the physical addresses associated with the source and destination hosts IP addresses are added. The physical address of a host uniquely identifies the host on a network. It corresponds a unique number of the network interface card (NIC) installed in the host. An example is the 48-bit long Ethernet address provided by the manufacturer of an Ethernet card. When the TCP/IP protocol is installed on a

host, that host's physical address is associated with an IP address. The physical address allows a particular host to be independent of an IP address.

To understand Web security and e-commerce, we need to be aware of three concepts associated with the TCP/IP protocol. These are

- port address,
- IP addresses, and
- physical addresses.

These ideas allow the request/response message to be exchanged by the intended processes (as specified by port numbers.). Those processes are running on hosts attached to the intended networks (as specified by the IP addresses) and finally, running on the intended hosts (as specified by physical addresses.) Figure 5 depicts these address assignments and the layers responsible for their assignments.

II WEB SECURITY AND E-COMMERCE

Section I showed that we can view the Web as a two way network composed of three components (1) Web servers, (2) Web users and (3) a communication path connecting the servers and browsers. Web security requirements for such a network are more extensive than a single multi-user computer system or stand alone local area network since it includes a communication path with public access (the Internet) and at least two computing systems.

A general definition of Web security is:

"... (W)eb security is a set of procedures, practices, and technologies for protecting Web servers, Web users, and their surrounding organizations. Security protects you (the user) against unexpected behavior." [Garfinkel and Spafford 1997].

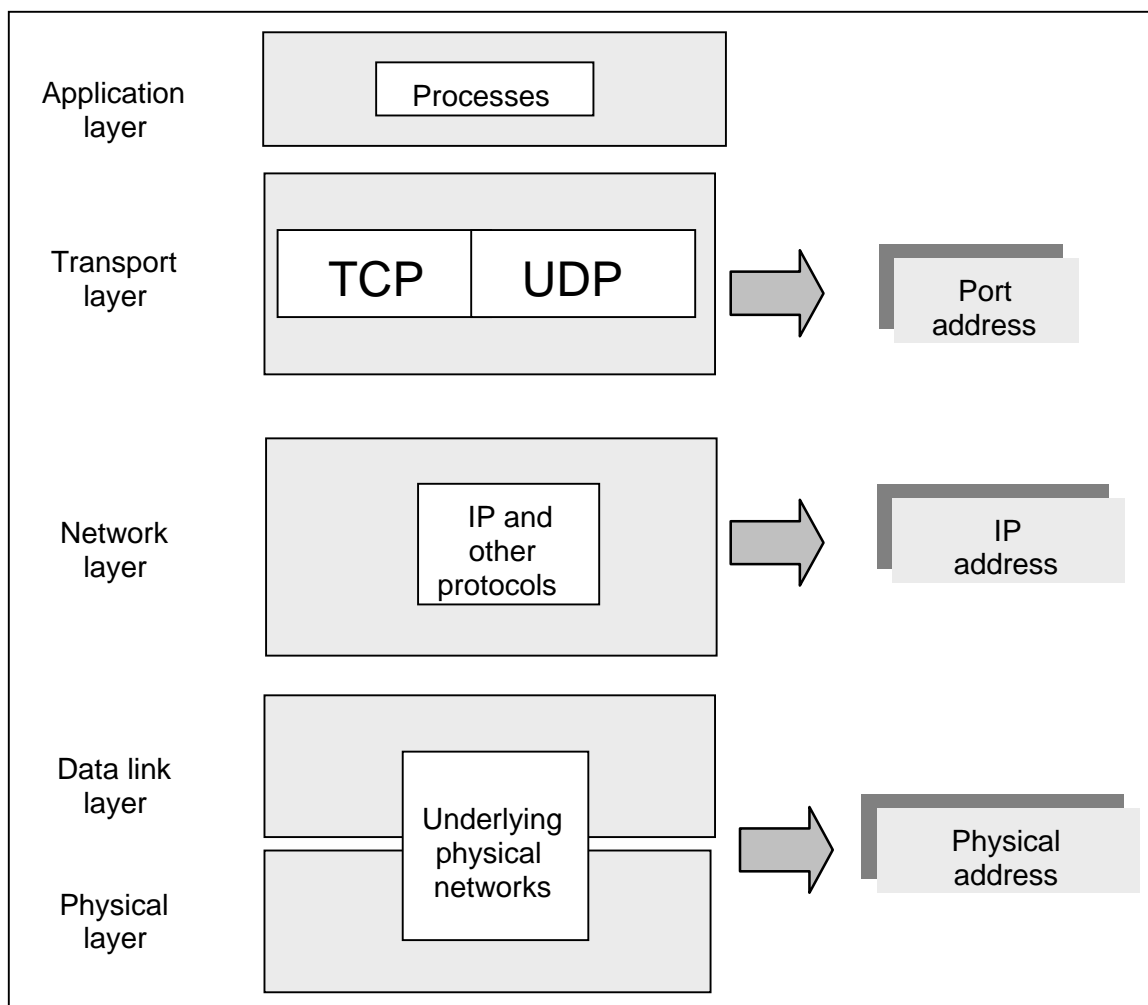


Figure 5. Address Types and Assignments in TCP/IP Protocol

Users and providers of Web services make a set of assumptions regarding expected behavior of the Web with regard to security.

User's perspective: From the users' perspective their expectation is that the service being provided is

- legitimate the services or information being supplied by the Web server is provided by the Web server the user expects to provide those services or information
- safe the services or information being provided will not contain computer viruses or content that will allow the user's computer system to be used for malicious purposes

private the provider of the requested information or services will not record or distribute any information the user may have sent to the provider in order to request information or services.

Server's perspective. From the server's perspective the expectation is that the requestor of the information or services is legitimate and responsible.

legitimate the user has been identified accurately;

responsible the user will not attempt to access restricted documents, crash the server, or use the server computing system as means of gaining illegal access to another computer system.

Joint perspective. From the perspective of both the server and the user, they have an expectation that their communications will be free from eavesdropping and reliable in that their transmissions will not be modified by a third party.

RISKS TO AVOID

The purpose of Web security is to meet the security expectations of users and providers. To that end, Web security is concerned with

- client-side security,
- server-side security, and
- secure transmission of information.

Client-side security is concerned with the techniques and practices that protect a user's privacy and the integrity of the user's computing system. The purpose of client-security is to prevent malicious destruction of a user's computer systems (e.g. by a virus that might format a user's fixed disk drive) and to prevent unauthorized use of a user's private information, such as use of a user's credit card number for fraudulent purposes.

Server-side security is concerned with the techniques and practices that protect the Web server software and its associated hardware from break-ins, Web site vandalism and denial of service attacks. The purpose of server-side security is to prevent modification of a Web site's contents, prevent use of the server's hardware, software, or databases for malicious purposes and to ensure

reasonable access to a Web site's services, i.e., to avoid or minimize denial of service attacks (Section IV).

Secure transmission is concerned with the techniques and practices that will guarantee protection from eavesdropping and intentional message modification. The purpose of these security measures is to maintain the confidentiality and integrity of user and server information as it is exchanged through the communication channel

With respect to e-commerce, Web security has as its main focus Web sever security and secure transmission. There is some concern with client-side security. However the client can be mostly assured that the client's security expectations will be met if the Web server and transmission channel are secure in the sense suggested above.

The reason for this focus is the nature of e-commerce. E-commerce can be simply defined as the exchange of goods and services for money. This exchange is transacted electronically, generally via the Web. Buyers (client processes) seek out reliable providers (server processes) of the goods and services they require.

Web server security is concerned with preventing attacks on Web sites. There are several ways to classify attacks on Web sites in order to understand their nature. The classification used in this tutorial partitions Web site attacks into two broad categories:

- attacks on Web site information and
- Web site accessibility.

Within attacks on Web site information, an overview of threats on a Web site's information integrity and confidentiality and their countermeasures will be given in Section III. Within attacks on a Web site's accessibility, an overview of denial of service and invalid authentication threats and their countermeasures will be presented in Section IV.

Web security is critical in conducting e-commerce transactions. As an example, Figures 6 and 7 depict a typical Business-to-Consumer (B2C) transaction and its possible security threats.

In Figure 7 security threats A to D can be handled by providing secure transmission using cryptographic methods while threat E and similar types can be managed managed by access control methods. Other types of security threats are are:

- illegal access of server computing system (Webjacking),
- illegal access of a client computing system, and
- unauthorized use of client information and denial of service.

III SECURITY THREATS AND THEIR COUNTERMEASURES

Broadly speaking of security threats fall into , two classes: (1) attacks against information in transit for which **information security** is needed and (2) attacks against site storing information for which **access security** is needed. Information security is discussed first.

INFORMATION SECURITY THREATS AND THEIR COUNTERMEASURES

To appreciate the countermeasures to information security threats on the Web an understanding of internet cryptography techniques is needed. These techniques are used in designing and implementing software whose function is to guarantee information integrity and confidentiality. Cryptography software can operate at the TCP/IP transport layer as well as the TCP/IP application layer. In addition, though not cryptography techniques, are methods that will monitor and control the traffic that enters and exits a Web site. In particular,

- sever proxies and
- firewalls .

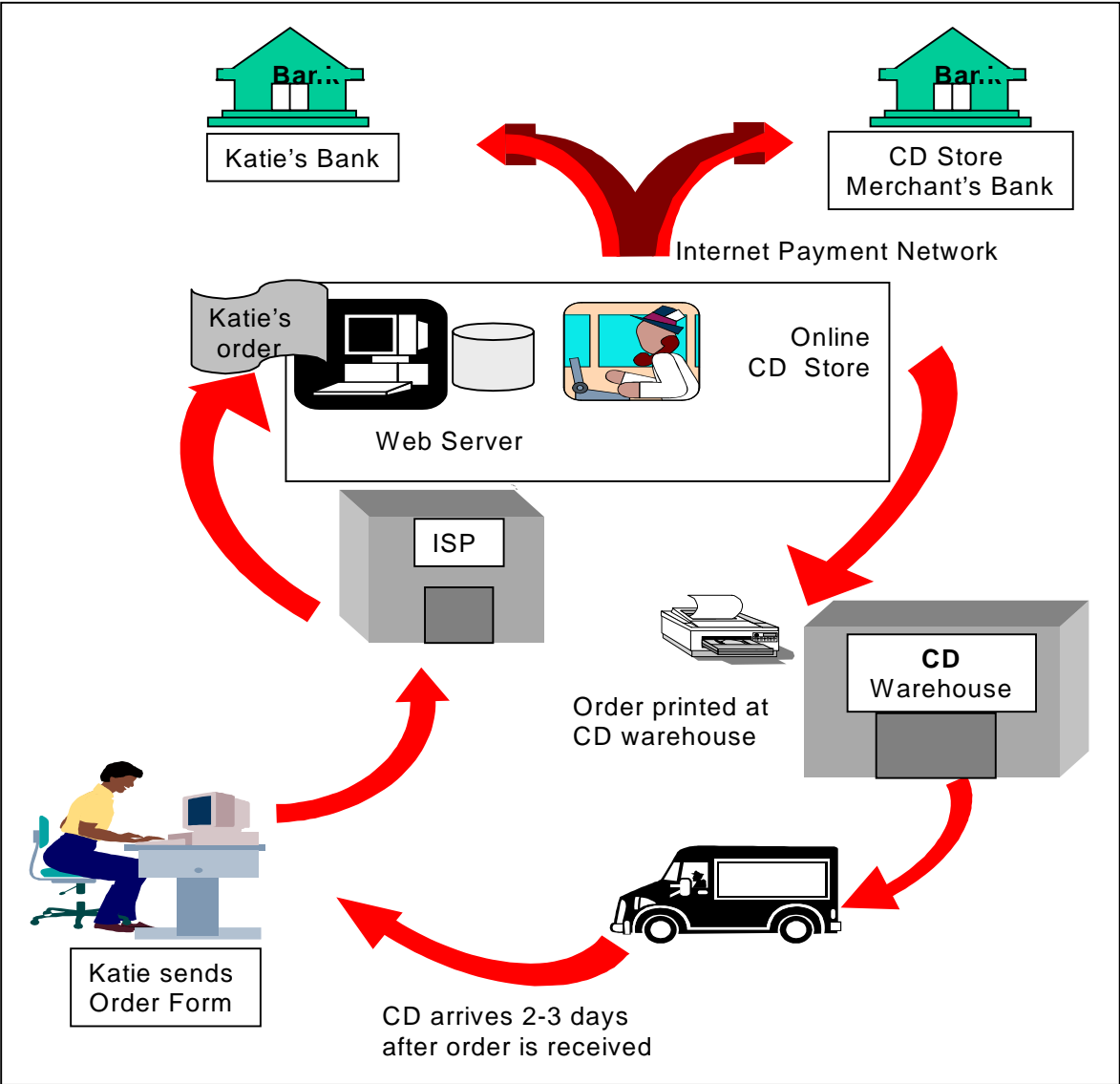


Figure 6. Typical E-Consumer to E-Business Transaction

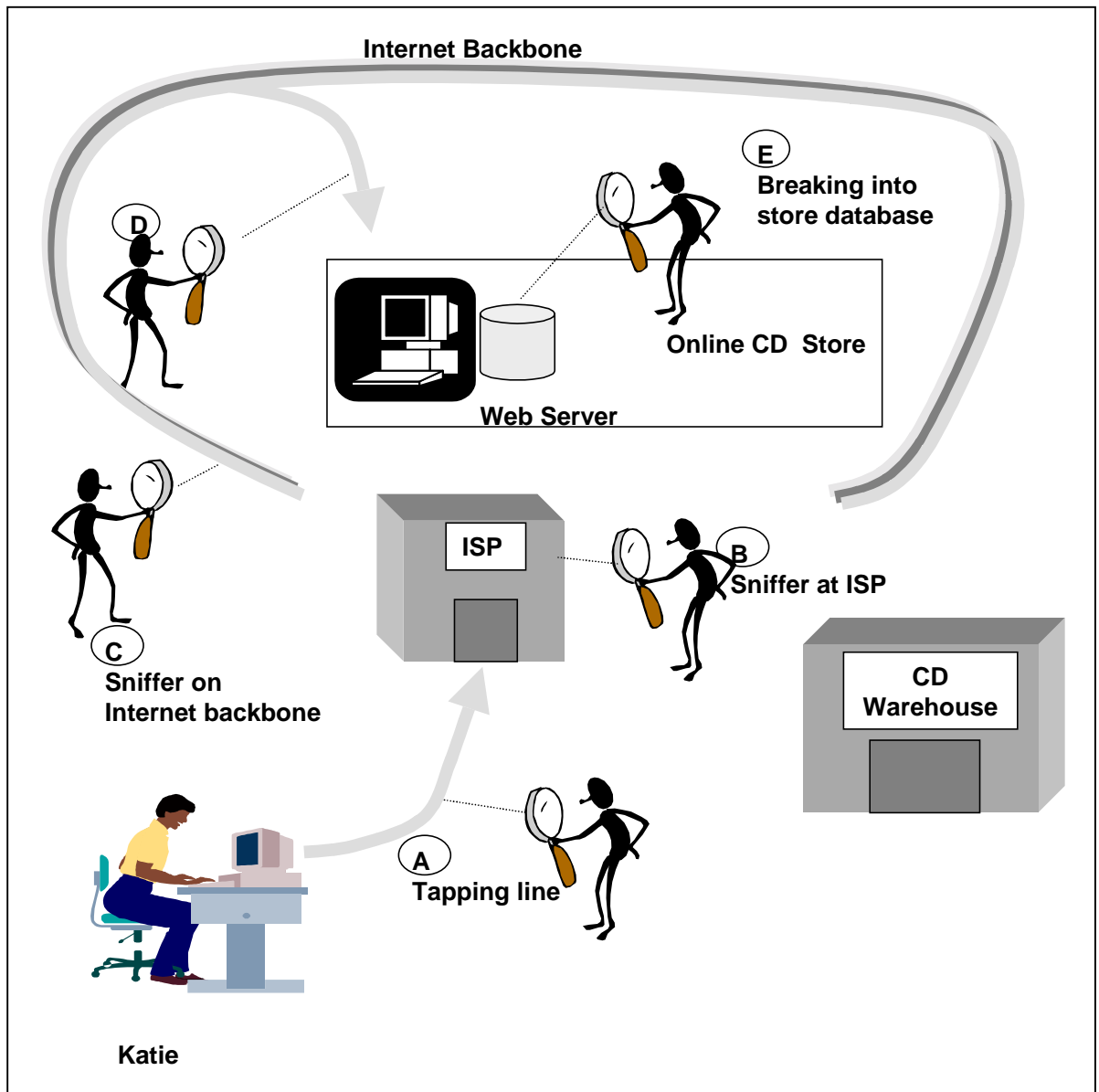


Figure 7. Security threats in a B2C transaction

Purpose of Cryptography

The purpose of cryptography is (1) to secure stored information - regardless if access is obtained and (2) to secure transmitted information - regardless if transmission is monitored.

Services Provided by Cryptography

The services provided by cryptography are:

confidentiality that provides privacy for messages and stored data by hiding;

message Integrity that provides assurance to all parties that a message remains unchanged;

non-repudiation that can prove a document came from X even if X denies it; and finally,

authentication that identifies the origin of a message and verifies the identity of person using a computer system.

Cryptography provides these services by means of encryption.

Encryption Overview

Using encryption techniques, the sender of a message converts the plain text to cipher text by use of an algorithm and key. The receiver of the cipher text then uses the appropriate algorithm and corresponding key to convert the cipher text to plain text. The algorithm is publicly known but the key is held private. For example, suppose the algorithm used is to offset a character by n positions in the ASCII table. The key value for this example is 4. This algorithm and key encrypts the string "CAT" as "GEX". It is obvious how to decrypt GEX if the key value is known.

Three types of encryption techniques are used for Web security. These are:

Secret key- where a single key is used to encrypt and decrypt information.

Public/private key- where two keys are used: one for encryption (public key) and one for decryption (private key).

A **one-way function-** where information is encrypted to produce a “digest” of the original information that can be used later to prove its authenticity.

Encryption Techniques

Secret key encryption is also known as **Symmetric Key** encryption. In this technique the sender and receiver have the same secret key that will encrypt and decrypt plain text. The strength of this encryption technique depends on key

length. The longer the key in bits the less likely it can be guessed by computing all 2^n possible combinations, where n is the number of bits in the key.

Some commonly used symmetrical algorithms are:

- the Data Encryption Standard (DES) with a 56 bit key;
- Triple DES, DESX, GDES, RDES with a 168 bit key;
- the RC series - RC2, RC4, and RC5 with variable key length up to 2048 bits;
- the IDEA algorithm, which is the basis of PGP using 128 bit key; and
- Blowfish, which has keys of variable length up to 448 bits.

More details and references about these symmetric algorithms can be found in Stein [1998].

Public/Private key encryption is also known as **Asymmetric Key Encryption**. In this encryption technique the user X has a pair of keys - one public and one private. The public key is used to encrypt a message. This public key can be used by anyone wishing to send a secure message to X . In order to decrypt the encrypted message used, X uses the private key that “matches” the public key used to encrypt the message. Only the private key of a public/private key pair can decrypt a message encrypted with the public key of the pair. Also, only the public key of a public/private key pair can decrypt a message encrypted with the private key of the pair.

The most common asymmetric algorithm is the RSA (Rivest Shamir Adelman) algorithm with key lengths ranging from 512 to 1024 bits. More details and references about asymmetric algorithms can be found in Stein [1998].

A **One-Way Function** is a non-reversible “quick” encryption that produces a fixed length value (number) called a **hash** or **message digest**. This value can be used to authenticate the contents of a message.

Common message digest functions are: the MD4 and MD5, which produce 128 bit hashes and the SHA produces 160 bit hashes. More details and references about these one-way algorithms can be found in [Stein 1998].

These encryption techniques provide for a number of cryptographic services:

These services are:

- **Digital Signatures**, which are used to "sign" messages to validate source and integrity of the contents
- **Digital Envelopes**, which are used to secure delivery of secret keys
- **Message Digests**, which are a short bit string hash of a message;
- **Digital Certificates** or **Digital Ids** which are used to authenticate: users, Web sites, public keys of public/private pair, and information in general
- **Secure Channels**, which establish a secure connection over private or public networks.

Digital Signature

To create a digital signature the sender encrypts the sender's identity string - something like "Hello I am Dr. Bob" - using the sender's private key. Then, the sender concatenates the encrypted text and the identity string together. This message is encrypted with the receiver's public key to create the encrypted message to be transmitted. The receiver decrypts the encrypted text with the receiver's private key. This yields two strings, a plain text of the sender's identity string and the sender's encrypted identity string. The encrypted text portion of the message is decrypted with the sender's public key. Note, the sender's public key is the only key that can decrypt a message encrypted with the sender's private key. This decrypted text can be compared with the transmitted plain text. If the plain text and the decrypted text are identical, the sender is validated since only a legitimate sender should possess the sender's private key.

Digital Envelope

Public/Private key encryption / decryption is useful for internet transactions because in most Internet transactions many users are sending information to one receiver. If symmetric key encryption were to be used, the receiver would need to keep a database of keys to be used with each sender/receiver transaction.

However, there are limitations to this technique. The asymmetric encryption/decryption technique is computationally slow compared to the symmetric technique. As a result it is not reasonable for large documents.

To solve this problem, a combination of symmetric and asymmetric methods can be used. This process creates what is called a **session key**. The sender creates and uses a symmetric (session) key to create cipher text. The sender uses the receiver's public key to encrypt the symmetric key, which is the digital envelope. The sender transmits both cipher text and a digital envelope to the receiver. The receiver "opens the digital envelope" by decrypting it with the receiver's private key. The contents - the symmetric key - is used to decrypt the additional transmitted encrypted text.

Message Digests

To create and use a message digest the sender uses a message as input to the digest function. The sender "signs" (encrypts) the output (hash value) with the sender's private key. The sender sends the signed hash and original message, in plain text or encrypted using the receiver's public key, to the receiver. The receiver decrypts the signed hash with sender's public key to obtain the hash value. The receiver then, if necessary, decrypts the original message and runs the plain text message through the digest function to obtain a hash value. If the receiver's decrypted hash value and computed hash value match, then the message has not been altered in transit.

Digital Certificates

Digital certificates are digital identifications and information whose legitimacy is guaranteed by a certification authority (CA). Digital certificates can be used to distribute the public key of a public/private pair. This guarantees the validity of the public key and also verifies the credentials of the entity associated with the public key. Certification authorities distribute these certificates.

Some CAs are:

- Versign - <http://www.versign.com>;
- U.S. Post Office - <http://www.ups.gov> and

- CommerceNet - <http://www.commerce.net>.

A digital certificate contains at least: a public key, e-mail address and the full name of certificate holder. Digital certificates are secure and cannot be forged or modified.

To create a Digital Certificate a user generates a public/private key pair. The user creates and sends a certificate request that contains identifying information and the user's public key. The CA verifies this information. The CA creates a certificate containing user's public key and information. In addition, the CA creates a message digest from the certificate and signs it with CA's private key. This message digest is called a **signed certificate**.

The process of using a digital certificate is straightforward as well. Before sending a secure message, the sender requests a signed certificate from the receiver. Upon receiving a response, the sender decrypts the signed certificate with the CA's known public key to obtain the message digest of the information and public key provided to the CA by the receiver. The sender then creates a message digest of public key and information provided by the receiver for the sender to use. The sender compares the message digests. If they match, then both the receiver and the public key are validated.

There are several types of digital certificates. Some are: site certificates that are used to authenticate Web servers; a personal certificate that is used to authenticate individual users; software publishers certificates that are used to authenticate executables; and finally, CA certificates, which are used to authenticate CA's public keys. The CA public keys are needed to decrypt a digital certificate sent in response to a request. Furthermore, all certificates have the common format standard of X.509v3.

Secure Channels

Encrypted traffic may use Symmetric Key or Public/Private Key encryption techniques. A secure channel through which transmission can take place is desired. In practice, this channel is set up before transmission begins. This approach is referred to as a **negotiated secure session**. The two dominant

ways of doing this are **Secure Socket Layer (SSL)** and **Transport Layer Security (TLS)**. Either service provides:

- authentication users and servers
- encryption to hide transmitted data - using symmetric or asymmetric techniques and
- message Integrity to provide assurance that data has not been altered during transmission.

Both require certificates be issued by a CA.

Furthermore, secure channels can be used in establishing a virtual network circuit across the Internet between specified private remote networks or hosts. The use of an encrypting router that automatically encrypts all traffic that traverses the links of the virtual circuit is one method. In addition, Tunneling Protocols may be used to setup a virtual private network. Some examples of tunneling are: PPTP by Microsoft - <http://www.microsoft.com>; Layer 2 Forwarding (L2F) by Cisco - <http://www.cisco.com>; and L2TP (combines PPTP and L2F) - <http://www.ietf.com>.

Secure Sockets Layer (SSL)

To setup a secure session between them, a client and server may use the SSL protocol. SSL is a competitor to S-HTTP (Secure Hyper Text Transfer Protocol). S-HTTP is an extension of HTTP. It is a general purpose encryption system using symmetric encryption. However, S-HTTP only encrypts Web protocols and therefore is not useful for non HTTP transfer. Figure 3 in Section I shows other types of TCP/IP applications.

The three versions are: SSL - v1.0, v2.0 and v3.0. SSL v3.0 is implemented in Netscape ver. 3.0 and higher and Internet Explorer version 3.0 and higher. SSL v3.0 supports Diffie-Hellman anonymous key exchange and the Fortezza smart card. For details see Stein [1998].

The characteristics of SSL are:

- it operates at the TCP/IP transport layer
- it encrypts (decrypts) input (output) from the application (network) layer (recall the OSI - TCP/IP model in Figure 3).

Further, any program using TCP can be modified to use SSL connections. In particular, an SSL connection uses a dedicated TCP/IP socket (e.g. port 443 for https or port 465 for ssmtp.) As a result, the code for the TCP/IP application must reflect this change.

SSL is flexible in choice of which symmetric encryption, message digest, and authentication algorithms can be used. When an SSL client makes contact with an SSL server, they try to pick strongest encryption methods they have in common. Also, SSL provides built in data compression. Data compression must be done before encryption.

When an SSL connection is established, browser-to-server and server-to-browser communications are encrypted. Encryption includes:

- URL of requested document
- Contents of the document
- Contents of browser forms
- Cookies sent from browser to server

Cookies sent from server to browser

Contents of HTTP header, but NOT particular browser to particular server.

In particular, socket addresses - IP address and port number - are not encrypted, but a proxy server can be used if this type of privacy is required.

Connection Process: The connection process is shown in Figure 8. To establish an SSL Connection, the client (browser) opens a connection to a server port. The browser sends a “client hello” message. A client hello message contains: the version number of SSL the browser uses and the ciphers and data compression methods it supports.

The server responds with a “server hello” message. The server hello message contains a session id and the chosen versions for ciphers and data compression methods the client and server have in common. The server sends its digital certificate, which is used to authenticate the server to the client.

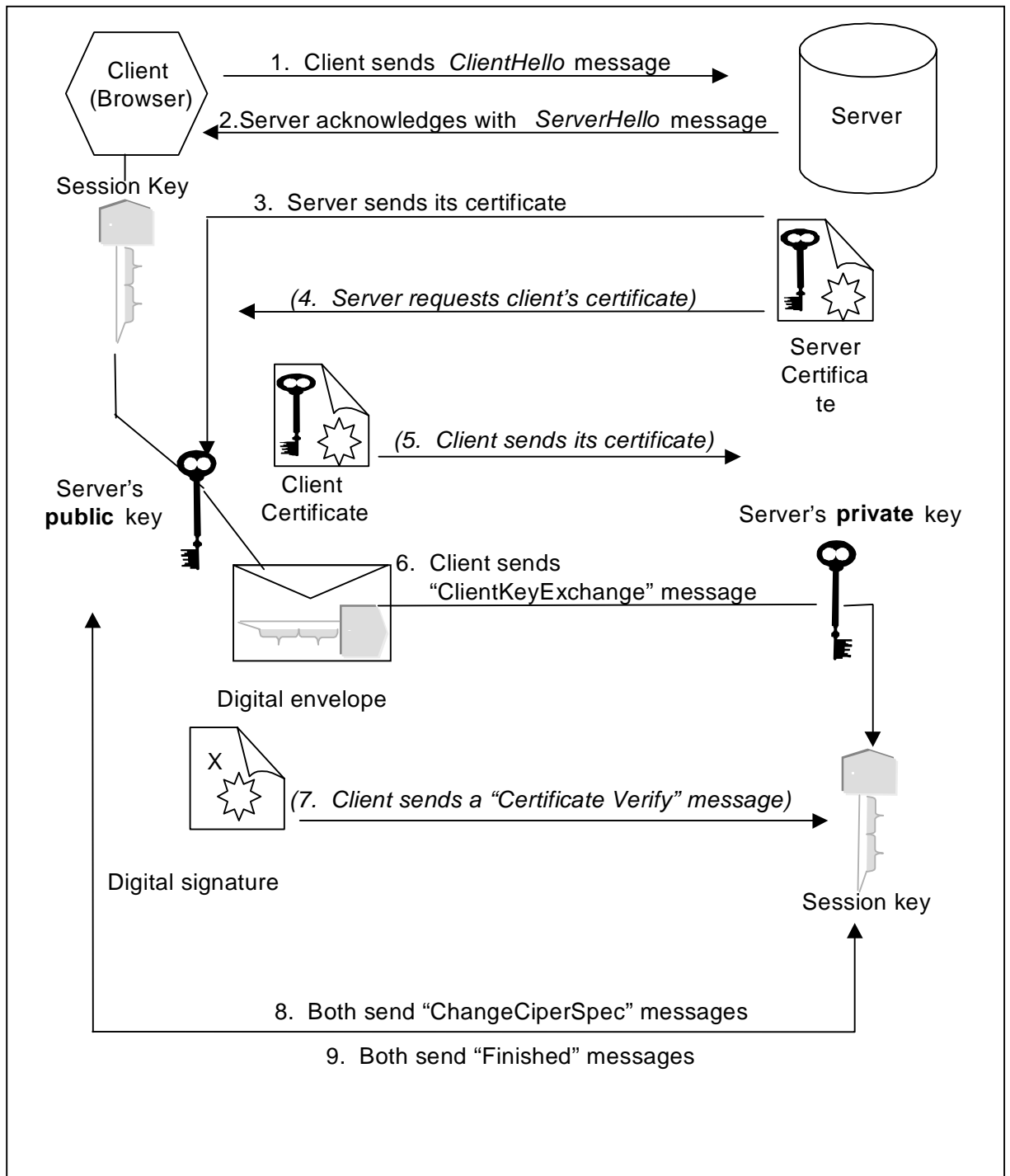


Figure 8 Process to Set Up an SSL Connection

Optionally, the server may request client's certificate. If requested, the client will send its certificate of authentication. If the client has no certificate, then connection failure results. Assuming a successful connection, the client sends a "ClientKeyExchange" message. This message is a digital envelope created using server's public key and contains the session key chosen by the client. Optionally, if client authentication is used, the client will send a certificate verify message. The server and client send "ChangeCipherSpec" message indicating they are ready to begin encrypted transmission. The client and server send finished messages to each other. The finished messages are message digests of their entire conversation up to this point. If the digests match, then messages were exchanged without interference.

Transport Layer Security (TLS)

An alternative to the SSL Protocol is the TLS protocol. This protocol is put forth as the IETF (Internet Engineering Task Force) Standard for a secure internet connection. It is a derivative of SSLv3.0 using different digest functions and different set of encryption algorithms. For more details on this protocol browse the TLS URL: <http://www.consensus.com/ietf-tls/>.

Also for more depth and details about SSL visit the following sites:

home.netscape.com/newsref/std/SSL.html and
home.netscape.com/ref/internet-security.html

Application Layer Security

Application layer security is provided by a number of user applications whose function is to guarantee secure communication. Some of these are:

- Secure Electronic Transactions (SET);
- Digital Payment Systems like First Virtual, CyberCash, DigiCash, Millicent, and
- Pretty Good Privacy (PGP), which is used to secure e-mail.

Once again, these are the applications that senders and receivers use to guarantee secure communications.

Secure Electronic Transaction (SET)

SET is cryptographic protocol developed by Visa, Mastercard, Netscape and Microsoft. It is used for credit card transactions on the Web. It provides:

Authentication of all parties in transaction;
Confidentiality: a transaction is encrypted to foil eavesdroppers;
Message integrity: not possible to alter account number or transaction amount; and

Linkage: attachments can only be read by third party if necessary.

In addition, the SET protocol supports all features of a credit card system, which are: cardholder registration, merchant registration, purchase requests, payment authorizations, funds transfer (payment capture), chargebacks (refunds), credits, credit reversals, debit card transactions. Further, SET can manage real-time & batch transactions and installment payments. Figure 9 illustrates the use of the SET protocol.

SECURING PRIVATE NETWORKS

A security concern lies in between guaranteeing secure message transmission and secure access to a Web server. To carry out electronic commerce a private network needs to allow access to network resources by users and processes over which the network has no control. Clearly, unlimited and unsupervised access is unwise. What must take place is limited external access, from both the outside to inside networks and the inside to the outside networks.

Networks need to be able to control external access. This can be done by means of **firewalls** and **proxy servers**. Firewalls provide a secure interface between an “inner” trusted network and “outer” untrusted network. Every packet to and from an inner and an outer network is “processed” in order to verify that the packet is legitimate.

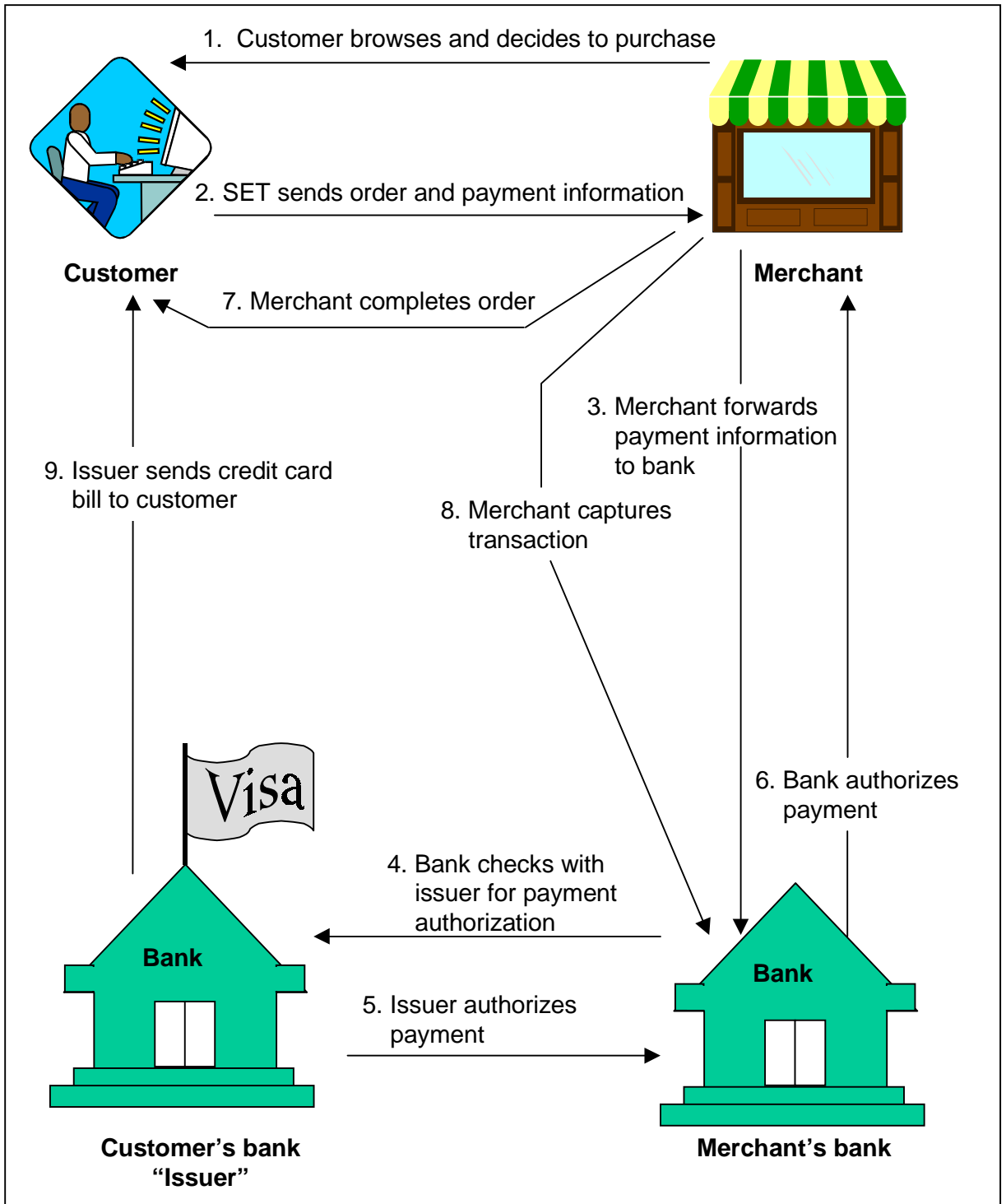


Figure 9. Secure Electronic Transaction Process

Firewalls require hardware and software to implement. The three main hardware architectures are: a dual-homed host, a screened host and a screened subnet host. These hosts are referred to as gateways or **bastions**, as in a defensive construction for castles.

Dual-homed Host

A dual-homed host is a computing system that has two network interface connections. One interface is attached to the secure private network and the other network interface is connected to the unsecure network - generally the Internet. The idea is that every frame going in or out of the private network passes through the dual-homed host. As a result, each frame can be processed at some level of the TCP/IP stack protocol. Figure 10 depicts this hardware architecture.

Screened Gateway Host

A drawback to using a dual-home host is that it requires all traffic on each network to be examined by the proxies running on the host. This task can be overwhelming. To minimize the amount of traffic the host needs to examine, it is possible to screen traffic and only allow certain frames in or out. Screening is done by connecting a networking device called a **router** to the dual-homed host.

The purpose of a router is to forward packets intended for a specific network. If a packet from the external network (e.g. the Internet) is not intended for the private network, then that packet is not sent to the dual-homed host and visa versa. Hence, the router screens packets for the dual-home host. Figure 11 depicts this hardware architecture. Note that the router has four network connections - two to the host (gateway) and one for each network.

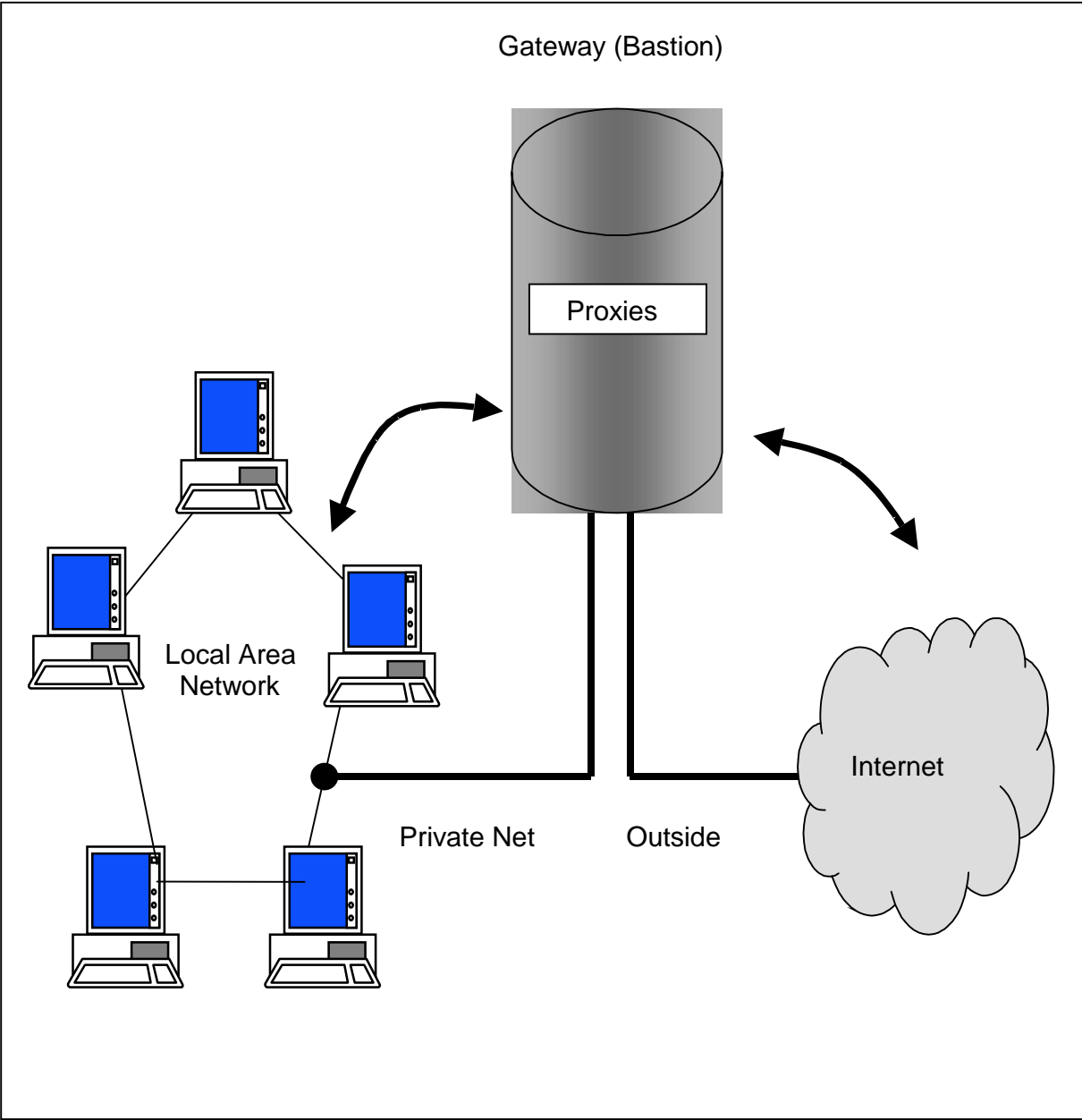


Figure 10. A Dual Homed Host

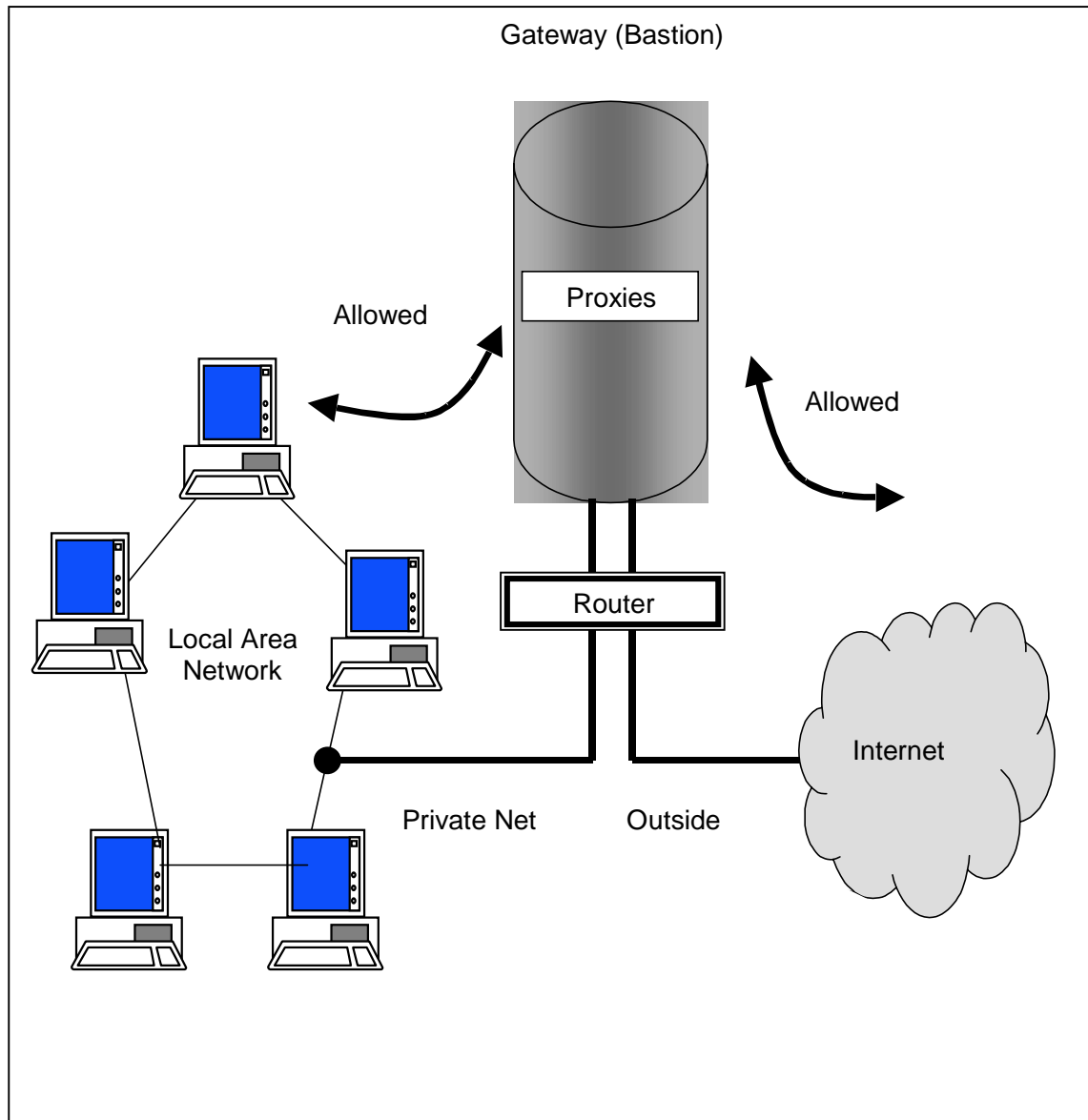


Figure 11. A Screened Gateway Host

Screened Subnet Gateway

In some situations access to certain private network resources needs to be more open than a dual-homed host allows. Openness can be accomplished by the use of a screened subnet gateway. For example, a public Web server needs to be outside of a highly secure private network, yet, at the same time, it does need some protection of limited access afforded by a router. Figure 12 shows this hardware architecture.

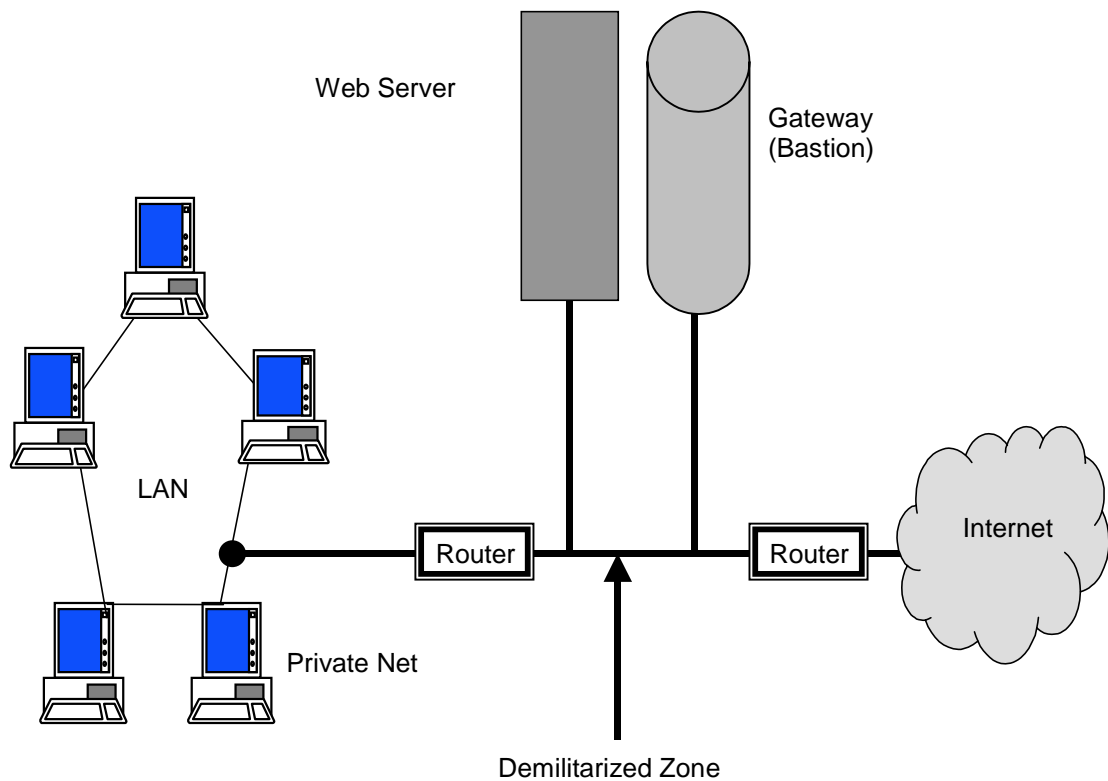


Figure 12. A Screened Subnet Gateway

In conjunction with the hardware used for firewalls, software is used to manage the network traffic in and out of the secure and non-secure networks. The software programs used are called proxies and filters. This software either allows or denies network traffic access to either network. The two types of proxy programs available are called **application-level proxies** and **circuit-level proxies**. Software that filters network traffic employs a technique called **packet filtering**.

Application level proxies are written for each particular protocol, e.g. HTTP or FTP or SMTP. Regardless of protocol, the proxy's function is to forward or not forward **messages** across the firewall. This decision is based on TCP/IP

information, for example, the source and destination ports and IP addresses. In addition, these proxies can decide based on the content of a message. For example, do not forward messages containing VB executable or ActiveX components, which can be sources of viruses or at least misbehaving programs.

Circuit level proxy software decides whether or not to forward **packets** across the firewall. The decision is made on the basis of header information in the packet; that is, by source and destination IP addresses and port numbers. They cannot peek into packets and determine the content of a message. The advantage of circuit level proxies is that they are very fast (less computation required) and very general (handle many protocols.) Examples of circuit level proxies are SOCKS (a freeware circuit level proxy) and SMLI - Stateful Multilayer Inspection Gateway that correlates incoming and outgoing packets.

Packet filtering techniques are technically not software. They are used with screen host or screened subnet host architecture. They use a router's routing table to decide which packets to forward or not forward. If the gateway (bastion) does not have proxy for a given service (e.g. TFTP), then the packet filter can be configured to bypass the firewall and allow that service.

The foregoing is a very broad and general overview of techniques for information security when using the Web for e-commerce. For details and complete explanations the reader is urged to consult the references in the annotated bibliography at the end of this article.

IV. ACCESS SECURITY THREATS AND THEIR COUNTERMEASURES

In addition to affording secure transmission, Web security is concerned with controlling access to the hosts that provide services to Web users. Two forms of access security threats of interest exist when using e-commerce. These are **access control** and **denial of service**.

Inadequate or lack of access control can lead to Webjacking, which can be considered Web site vandalism. Of more concern is access to user's data

such information as names, addresses, social security numbers, credit card numbers, and past purchasing activity. The countermeasures to these threats are:

- user authentication and
- user authorization.

Another access security concern is denial of service, which is discussed after access control. During a denial of service attack (DOS) users are unable to use server resources. If a system is used for an e-commerce activity, then that activity is no longer available. The countermeasures to the several types of denial of service attacks are limited. The countermeasures depend on firewalls and careful server system configuration.

ACCESS SECURITY THREATS – ACCESS CONTROL

Countermeasures to Access Security Threats

One countermeasure is user authentication. User authentication is a process used to identify a user who accesses a Web server to determine if the user is legitimate. This method is generally referred to as **access control**.

Additional access control can be gained by employing the idea of **user authorization**. User authorization is part of the user authentication process. Once the user is authenticated, the process also specifies what server resources that user may access. These resources would include files, scripts, directories, and hardware.

Access Control Methods

Access control can be achieved by using one of the following:

- The IP address, which validates a Web browser based on its host's IP address.
- The host's domain name, which validates a Web browser based on what domain to which the host belongs.
- The familiar user name and password technique where the user of a

browser is validated on the basis of user ID and its associated password.

- Client certificates where a remote user is issued a secure certificate to use as a digital signature to allow access to a Web server.
- Access control techniques based on network security protocols.

These access control methods solve validation problems associated with accessing a remote host via LANs and WANs. Example of these protocols are **Kerberos** and **DCE**

Authentication Based on Host IP Address and/or Domain Name

This method is used to grant access to clients based on their source IP address, Domain Name, network name, or subnetwork name. The advantages are ease of set up and the unlikely chance of being incorrectly configured.

The disadvantage is the difficulty in granting access to users who move from one computer to another. Additional overhead is required to set up the DHCP protocol and to use Web proxies. And finally, there are the security issues of **DNS spoofing** and **IP spoofing**. The first two disadvantages are overcome with additional computer system and network administration. The last disadvantage requires more effort to guard against illegitimate access.

DNS Spoofing is a method in which an attacker assumes control of the DNS host/name lookup system. The effect is that the attacker's host machine can appear to be coming from a legitimate domain. This attack can be countered by a technique called **Paranoid DNS Checking**. When using paranoid DNS to check, a server (upon receiving a packet from a client) uses the source IP address to make two DNS requests. The first resolves the client's IP address to get a domain name for that IP address. The returned domain name is then used to find an IP address using DNS service. If the IP address returned from this second request correlates with the first IP address of the client's host, then it is a legitimate remote host.

Another countermeasure is to use a firewall's DNS lookup methods. Since the firewall is under the Web site's administrative control, it is unlikely to have been compromised by an attacker.

IP spoofing requires a fair amount of technical expertise but the intent is to have an attacker's host take on the identity of a legitimate host. One technique of IP spoofing uses the source routing protocol in TCP/IP. The idea is to make it appear as if a request originates from the host within a secure LAN. This request can be used to insert CGI script or modify the properties of the server's operating system. Configuring routers and firewalls to reject connections using the source routing protocol can prevent this attack as well as configuring the server's operating system to reject connections using source routing.

Authentication Based on User ID and Password

This familiar technique requires a user to provide protected information in order to be authenticated. The advantages of this technique are:

1. The technique authenticates users not hosts.
2. Users can migrate from host to host.
3. No additional overhead in setting up Web proxies or DHCP.

One major disadvantage with passwords is user error. Users share passwords, forget passwords, do not keep passwords private, or choose poor passwords. In addition, passwords can be "sniffed" if transmitted over a network.

Countermeasures to these disadvantages exist. The user's lack of security about passwords can be helped by educating the user about the need for secure passwords. A useful technique is to instruct in choosing difficult-to-guess, but easy-to-remember passwords. A difficult-to-guess password is one that is not in any dictionary and contains a mixture of upper and lower case alphanumeric symbols.

A simple technique for choosing a difficult-to-guess but easy-to-remember password is the use of letters in a specific position in each word of a familiar phrase or title. For example - The Old Man and the Sea - can be used to generate the password - TOMATS or HLANHE.

A countermeasure to passwords being “sniffed” if transmitted over a network is to use a protocol that encrypts this kind of message. For example, using HTTP/1.0 basic authentication is carried out in plain text but coded in Base 64 MIME. This packet can be intercepted and decoded. Since the HTTP protocol is stateless, every access to a protected resource needs to be authenticated every time that resource is accessed, even if it is requested by the same Web browser. As a result, the basic authentication process occurs frequently and, hence, allows more opportunity for the packet to be sniffed. To prevent this problem,

1. use secure transmissions such as HTTP/1.1 which uses a digest authentication process or
2. setup a secure communication channel, for example an SSL connection

Authentication Using Client Base Certificate System

The process of authentication using client certificates is straightforward. When a user logs on (presents a certificate), the authentication server verifies the certificate is validated by opening it with the certifying authority's public key. The certificate contains the user's public key and personal identifying information - a signature. The server then sends a challenge to the user - a one-time value - which the user signs (encrypts) with the user's private key and returns it to the server. The server then signs (decrypts) the same value with its copy of the user's public key. If the signatures match, then the user is authenticated.

Other Forms of Access Control

The Kerberos authentication model uses a secure “key server.” Once users are authenticated, they are free to use any resource of the system. Moreover, as part of the logon process, the user is provided a session key and, as a result, all subsequent transmissions are encrypted.

A similar authentication technique is the **Distributed Computing Environment (DCE)**. DCE is designed by Open Software Foundation and is similar in design to the Kerberos authentication model.

Additional methods are:

- **Two Factor Authentication** where the user needs something, e.g. ATM card, and needs to know something, e.g. a PIN number.
- There is a **Smart Card Type** where a token access device that has information that is in sync with server information (e.g. counter, time, random number generator, etc.).
- A **One Time Pad** of user name and password. This pad is essentially a list of user name/password pairs good only for one logon per pair.

ACCESS SECURITY THREAT - DENIAL OF SERVICE

The intent of the denial service attack is to tie up the server by forcing the server to respond to an overwhelming number of bogus requests for service or make the server wait for a response that will never come. Some types of attacks are the **TCP/IP SYN attack, Ping of Death** and **URL flood**.

The TCP/IP SYN attack exploits the protocol by which a connection is first established between a client and server. To setup a TCP/IP connection a three step "handshake" protocol is used. First, the client requests a connection. Second, the server acknowledges the request and waits for the client to acknowledge with a SYN value set. If no client acknowledgements are ever received and there are many client requests, the server waits - effectively locked up.

Another DOS attack is the PING of Death attack. In this attack many clients continually "ping" the server. The ping is a request by a client to the server to let the client know the server is available. The server needs to respond to each ping.

A third type of DOS is to flood the server with URL requests. This can be done either by one client or by many clients in parallel. If many clients are used

in parallel, the type of attack is called a **Distributed Denial of Service** attack (DDOS).

The countermeasures available to apply to DOS attacks are minimal once the attack has started. In general, DOS attacks require client(s) to carry requests. The countermeasure is to locate source(s) of requests and terminate those processes or not accept network traffic from those sources.

Some countermeasures are available prior to attack. One way to prevent attacks is to make sure all hosts are going to be used legitimately. Of course, this approach requires securing all remote hosts, which is not likely.

V SUMMARY

This tutorial is an introduction to the issues when considering Web security for e-commerce. The material presented is by no means complete. This tutorial should be considered a starting point in a study of Web security. The bibliography is annotated to help guide a reader into a detailed study of this topic.

Editor's Note: This article was received on October 5, 2000. It was with the author 1 week for 1 revision and was published on November __, 2000. This article is based on a tutorial presented by the author at AMCIS 2000.

REFERENCES

- Forouzan, B.A. (2000) *TCP/IP Protocol Suite*, Boston, MA: McGraw-Hill.
- Garfinkel, S. and Spafford, G. (1997) *Web Security & Commerce*, Cambridge, MA: O'Reilly and Associates.
- Stein, Lincoln D. (1998) *Web Security: A Step-by-step Reference Guide*, Reading, MA: Addison-Wesley.

ANNOTATED BIBLIOGRAPHY

Atkins, D., Buis, P., Hare, C., Kelley, Nachenberg, C., Nelson, A.B., Phillips, P., Ritchey, T., Sheldon, T., and Snyder, J. (1997) *Internet Security Professional Reference Second Edition*, Indianapolis, IN: New Riders.

This book provides detailed information on how to set up and administer secure computing systems. Its focus is on UNIX based systems although there is some information on Windows NT and Java security as well. It is intended for a technically sophisticated reader.

Cheswick, W.R., Bellovin, S.M. (1994) *Firewalls and Internet Security: Repelling the Wily Hacker* Reading, MA.: Addison-Wesley.

This text contains everything you need to know about setting up and maintaining a firewall on the Internet. It is limited in its scope since its focus is on systems using the UNIX operating system.

Denning, D., and Denning, P.J., (1998) *Internet Besieged Countering Cyberspace Scofflaws*, New York, NY: ACM Press.

This book is a collection of papers over a variety of internet security issues. The issues range from Web security, cryptography, secure electronic commerce, and Law, privacy, and education. The papers are intended for readers with a broad technical background. This book is excellent for a novice to obtain an overview of issues associated with Internet security.

Forouzan, B.A. (2000) *TCP/IP Protocol Suite*, Boston, MA: McGraw-Hill.

An excellent text that provides an introductory but detailed overview of the TCP/IP protocol and its uses.

Garfinkel, S. and Spafford, G. (1997) *Web Security & Commerce*, Cambridge, MA: O'Reilly and Associates.

An excellent source of information concerning Web security and e-commerce. A readable book for those with very little technical expertise in this area.

McClure, S., Scambry, J., and Kurtz, G., (1999) *Hacking Exposed: Network Security Secrets and Solutions*, Berkley, Ca.: Osborne/McGraw-Hill .

This text provides detailed explanations on various hacking techniques. An excellent source for details of attacks and their countermeasures. A fair amount of technical background required.

Smith, R.E., (1997) *Internet Cryptography* Reading, MA,: Addison-Wesley.

Outstanding introduction to the topic of cryptography for the Web. Presentation of complex material well done so that it is accessible for readers of all backgrounds.

Stallings, W., (2000) *Network Security Essentials: Applications and Standards*, Upper Saddle River, NJ: Prentice-Hall.

Comprehensive in its treatment of its subject matter. However only a portion of the text is devoted to Internet security. Not intended for the technically timid reader.

Stein, Lincoln D. (1998) *Web Security: A Step-by-step Reference Guide*, Reading, MA: Addison-Wesley.

One of the better texts in this area for both the technical and non-technical reader. It is a good mix of concepts and practice. It is a bit more accessible by a wider audience than the Garfinkel and Spafford book listed above.

ABOUT THE AUTHOR

Robert J. Boncella (<http://www.washburn.edu/cas/cis/boncella>) is Professor of Computer Information Science at Washburn University, Topeka, KS.. Dr. Boncella has a half time appointment in the Computer Information Sciences Department where he conducts classes in Data Communications and Computer Networks. He also has a half time appointment with Washburn University's School of Business where he offers instruction on Computer Based Information Systems in the school's MBA program.

He holds a Ph.D. and Masters degrees in Computer Science from the University of Kansas. In addition Dr. Boncella holds a Master of Arts in Philosophy from The Cleveland State University. He is a member of ACM, AIS, AAI, and IEEE.

Dr. Boncella current areas of interest are Web Based Information Systems, Intelligent Agents and Decision Making Under Uncertainty.

Copyright ©2000, by the [Association for Information Systems](#). Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than the [Association for Information Systems](#) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or fee. Request permission to publish from: AIS Administrative Office, P.O. Box 2712 Atlanta, GA, 30301-2712 Attn: Reprints or via e-mail from ais@gsu.edu



EDITOR
Paul Gray

Claremont Graduate University

AIS SENIOR EDITORIAL BOARD

Henry C. Lucas, Jr. Editor-in-Chief New York University	Paul Gray Editor, CAIS Claremont Graduate University	Phillip Ein-Dor Editor, JAIS Tel-Aviv University
Edward A. Stohr Editor-at-Large New York University	Blake Ives Editor, Electronic Publications Louisiana State University	Reagan Ramsower Editor, ISWorld Net Baylor University

CAIS ADVISORY BOARD

Gordon Davis University of Minnesota	Ken Kraemer University of California at Irvine	Richard Mason Southern Methodist University
Jay Nunamaker University of Arizona	Henk Sol Delft University	Ralph Sprague University of Hawaii

CAIS EDITORIAL BOARD

Steve Alter University of San Francisco	Tung Bui University of Hawaii	Christer Carlsson Abo Academy, Finland	H. Michael Chung California State University
Omar El Sawy University of Southern California	Jane Fedorowicz Bentley College	Brent Gallupe Queens University, Canada	Sy Goodman Georgia Institute of Technology
Ruth Guthrie California State University	Chris Holland Manchester Business School, UK	Jaak Jurison Fordham University	George Kasper Virginia Commonwealth University
Jerry Luftman Stevens Institute of Technology	Munir Mandviwalla Temple University	M. Lynne Markus Claremont Graduate University	Don McCubbrey University of Denver
Michael Myers University of Auckland, New Zealand	Seev Neumann Tel Aviv University, Israel	Hung Kook Park Sangmyung University, Korea	Dan Power University of Northern Iowa
Maung Sein Agder College, Norway	Margaret Tan National University of Singapore, Singapore	Robert E. Umbaugh Carlisle Consulting Group	Doug Vogel City University of Hong Kong, China
Hugh Watson University of Georgia	Dick Welke Georgia State University	Rolf Wigand Syracuse University	Phil Yetton University of New South Wales, Australia

ADMINISTRATIVE PERSONNEL

Eph McLean AIS, Executive Director Georgia State University	Jennifer Davis Subscriptions Manager Georgia State University	Reagan Ramsower Publisher, CAIS Baylor University
---	---	---