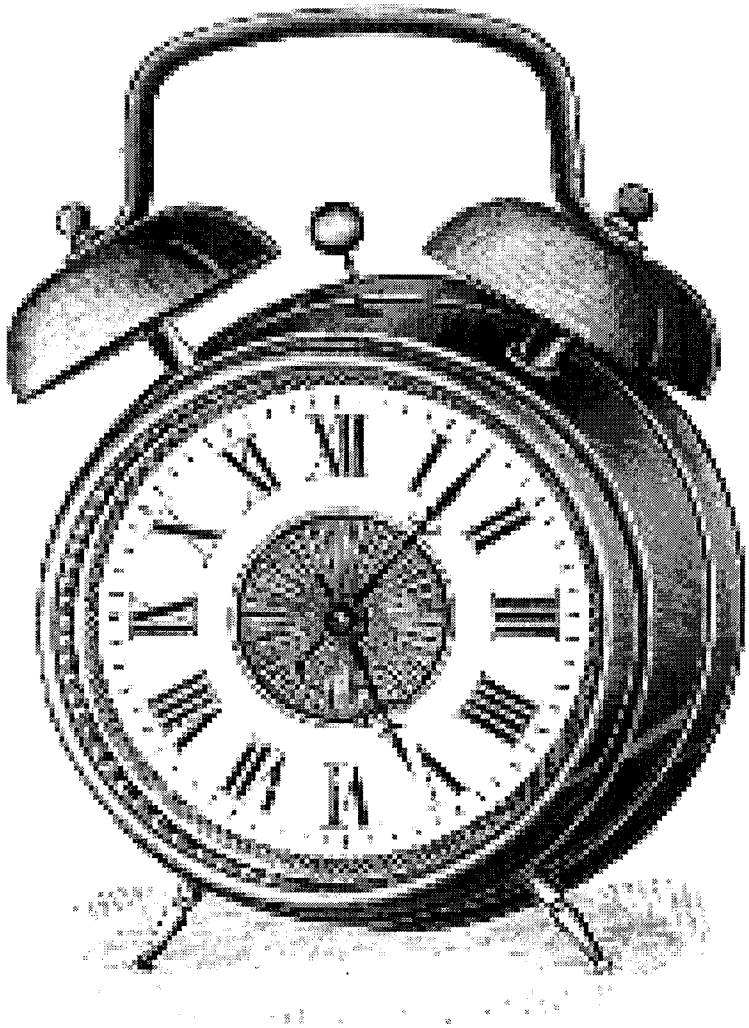
The Security of Commerce over the



Internet

By Tyrone Grandison

ABSTRACT

Security of Commerce over the Internet

Tyrone Grandison

Internet commerce is believed to be a booming industry, though the major fear of many interested people is its security. This paper seeks to explain what is the Internet, how Internet commerce works and the ways in which Internet commerce can be compromised. It then identifies the current methods being used to make Internet commerce more secure and then outlines what I consider to be the necessary environment for truly secure Internet commerce.

Table of Contents

Chapter 1
Acknowledgements

First and foremost, I must make it abundantly clear that prior to embarking upon this research, I was neither an expert in Security nor an expert in Internet commerce. I merely set upon a journey to discover the issues that have, are and may affect the world of Security as it relates to Internet commerce.

In the process of doing this, I have incorporated ideas from numerous sources in the respective fields. I take this opportunity to openly thank them for their input. Irrespective of the nature of the input, that is whether direct or indirect, I am truly grateful to them all for providing a diverse and vast intellectual pool.

I must say that I am especially appreciative of the works provided by Byte magazine, Internet World magazine, Phrack magazine, Risks-Forum Digest, PC Magazine, IEE Computer magazine, Oracle magazine, Charles Cresson Wood and Anup K. Ghosh. Your publications made this topic so much more understandable. I hope I have done justice to all of your ideas and materials and not corrupted the main intent of your works.

A special word of thanks to the lecturers who advised me, the people who helped proofread this piece and the numerous persons who offered support and encouragement in my many hours of darkness. With their assistance, I think that my initial goal of trying to relay the issues involved in the security of commerce over the Internet in an understandable manner has been achieved.

Finally, to all the people that I have not explicitly mentioned, but played a part in the production of this document, I thank you wholeheartedly for your efforts.

Chapter 2
Foreword

This paper is essentially a guide for people who are interested in the security issues that are being faced by people who wish to enter into the world of Internet commerce. Prior exposure to Internet commerce is not a requirement for any reader of this piece. It is also intended to make managers and Internet commerce web-site designers aware of the risks that they will be faced with.

I will try to make the intricacies of Internet commerce and the potential attack methods as clear as I possibly can to the reader. However, I will not go into too much detail on the topics that I deem too vast in scope to be covered entirely by this paper. Nevertheless, I will point the avid reader in the direction in which he or she may find supplementary information. I have taken it as my challenge to explain as simply as possible the solutions that have been used thus far to reduce the security problems being faced by companies worldwide and the merits and inadequacies of these solutions.

It seems to me that the majority of people believe that the notion of having a totally secure environment is an impossibility in today's world. I tend to partially subscribe to this viewpoint, but it is my firm belief that it is possible to make *system compromise* an extremely difficult and painstaking task. So painstaking that even the most motivated and knowledgeable villain will prefer to get a legitimate job than to try to *compromise* your system.

My reasons for partially agreeing with the notion that an absolutely secure environment will never exist are based in pragmatism. Firstly, security is not an absolute concept. No one can say that a bank or even a piece of software was, is and will always be secure. One can only speak of something being secure at a particular time. Thus, we see that security is a dynamic. An entity is considered safe because flaws have not yet been found. Such flaws may be uncovered some

time in the future. Secondly, the notion of an absolutely secure environment would require that all aspects of that environment are secure. A weakness in any area of the environment would mean that a security breach would be possible. I believe that most humans either do not believe in or do not implement such a holistic approach to security. They always seem to view a particular area as being insignificant and thus neglect or forget its security.

It is hoped that this paper will enlighten Internet commerce professionals and consumers.

Chapter 3
Introduction

The importance and impact of computers

As we move towards a global society, it is evident that computers will become an increasingly important and integral part of our lives. As this integration proceeds, computers will impact our lives even more greatly than they have ever done before. People will be able to do most tasks (if not every task) from their computer, thus making life more comfortable and a lot easier for the majority of us.

The influence of the Internet

For a long time now, the industry has seen that there is great potential for computers in the world of commerce, but has always been hindered from fully exploiting this potential by some factor or the other. With the technological strides that have been taken in the past ten or so years, specifically the success of the Internet, another dream can soon be realized.

Why will the Internet will change business?

The Internet is seen as a global community. Now, this community provides an avenue of untapped revenue for most businesses. It is these two facts that make the Internet the medium that will effectively change the landscape of modern business practices and strategies.

The rationale for indulging in Internet commerce

More and more companies are coming to the realization that offering their products and services over the Internet may be their best course of action. It is essential that they take advantage of this medium for two main reasons. The first reason is that the market that these companies are now dealing with is the WORLD. Firms are always looking for a larger target audience and the Internet seems to be the medium to make them a mouse click away from any Internet user in the world. The second reason is that people are, in my estimation, always looking for the easy way out. The average human prefers to

click on some mouse to order their goods and have them delivered, than to actually go to the store and buy it themselves. Although some may argue that there is a certain allure associated with being able to physically examine one's goods, I believe that the perceived benefits accrued from staying in Jamaica and buying a watch from Switzerland will eventually outweigh one's need for prior physical examination.

The right time for Internet commerce

Even though the time is right for commerce over the Internet, caution has to be taken when one is delving into this field. Internet commerce is viewed with suspicion by most businesses because they perceive the Internet as being inherently insecure. This is a very valid concern, as the initial Internet design was not terribly concerned with security.

Overview of this paper

This paper takes the following approach in its examination of the security of Internet commerce:

1. An introduction of the Internet and Internet commerce is given
2. Ways in which Internet commerce participants can be attacked are described.
3. The consequences of these attack techniques are examined.
4. The current solutions for these attack techniques are described.
5. A framework is proposed which will seek to address the threats to Internet commerce.
6. A look at the future of secure Internet commerce is taken.

It should be noted that this paper reflects the Internet world as it stands in 1998. It is inevitable that new attack methods and new solutions will emerge, so the framework that will be proposed must be flexible.

Chapter 4
An Introduction of Internet commerce

4.1 What is Internet commerce?

"Internet commerce is the process of using the World Wide Web by companies to transact their services"
- Anonymous

"Internet commerce is the buying and selling of goods and services, using the Internet as one's communication medium"
- Tyrone Grandison

Traditional commerce versus Internet commerce

Commerce is essentially the buying and selling of goods and services. These products are offered by business entities and are consumed by people who want these goods and services. Internet commerce is basically the same thing with one exception. This exception is that the medium used to offer these products and to conduct one's transactions is the Internet. The services being offered via this medium can be as varied and wide in scope as the services currently being provided in the traditional commercial setting.

The diversity of Internet commerce

Normally, when one thinks of Internet commerce, one immediately envisions a company offering services to a consumer, where a consumer is seen as a computer user in the comfort of his or her home. However, it should be made clear that this consumer can also refer to another business, which is interested in the company's products and services. I feel this distinction is necessary because many people have limited their view of Internet commerce and, in so doing, mistakenly underestimate the usefulness and impact that Internet commerce will have on their lives. This is potentially very dangerous.

The impact of Internet commerce

Internet commerce not only defines a process, but also defines a new business paradigm. In essence, it requires a different way of thinking and thus defines a new way of conducting business. Many firms hoping to enter the field will need to address this issue. How will Internet commerce impact on current business operating conditions and organizational structure and dynamics? This question needs to be given careful consideration before one commits to an Internet commerce strategy.

4.2 What is this "Internet" thing?

"The Internet is a large network of networks, in which all the networks can communicate with each other."
- Tyrone Grandison

"The Internet is a distributed network, which is built on standard networking protocols"[1].

Humble beginnings

The Internet was designed to provide easy and inexpensive access to databases and expensive hardware resources[2]. It had its beginnings as a research project for the U.S. Department of Defense (DOD) Advanced Research Projects Agency (ARPA) and was essentially an experimental Wide Area Network (WAN). It was then known as ARPANET. Kleinrock, Lynch, Postel, Roberts and Wolff wrote an excellent article on the history of the Internet in the February 1997 issue of Communications of the ACM.

The language of the Internet

Around the mid-1970s, the ARPANET community developed a machine-independent communications protocol called Transfer Communications Protocol/Internet Protocol or TCP/IP. This protocol serves as the basis for the Internet as we know it. The phrase *TCP/IP* is normally used generically, and refers to the whole family of Internet protocols. This family of protocols includes:

[1] "The Internet: An overnight Star?" ORACLE Magazine, Jan./Feb. 1996, Vol. X, No. 1, 41
[2] "The Internet: An Overnight Star?" ORACLE Magazine, Jan./Feb. 1996, Vol. X, No. 1, 42.

- IP - Internet Protocol,
- TCP - Transfer Control Protocol,
- UDP - User Datagram Protocol,
- ICMP - Internet Control Message Protocol,
- ROSE - Remote Operations Service Element,
- ACSE - Association Control Service Element,
- CMIP - Control Management Information Protocol,
- OSI - Open Systems Interconnection,
- ARP - Address Resolution Protocol,
- Ethernet for LANs (Local Area Networks),
- And many more

To understand how communication takes place on the Internet and by extension how a transaction occurs over the Internet, section 4.3 will examine the following protocols: IP, TCP, UDP, ICMP, ARP and Ethernet for LANs.

2. How does the Internet work?

The purpose of this section

As we already know, the Internet is a large group of networks. All these networks can communicate with each other and they possess the ability to work together towards a common goal. How are these possibly heterogeneous systems able to do this? By using a standard vocabulary with consistent semantics, which are defined in a language that is agreed upon by entire community. This section will give you insight into this language (TCP/IP); allowing you to see how communication takes place over the Internet. This will be done by examining the specification of a TCP/IP node and how packets get sent from one computer to another computer. However, there are a few things that must be cleared up before we proceed.

Defining the terms IP node, IP network and Ethernet

There is some terminology that has to be dispensed with before we go on. A computer that is connected to the Internet is called an Internet node or IP node (whichever you prefer). A network that is connected to the Internet is referred to as an IP network. Ethernet refers to a particular physical medium that is used to connect computers in a

particular network.

The triviality of the physical medium

Most networks are configured using Ethernet cable and reference is made throughout this paper of local nodes that are on an Ethernet. The physical medium used for a network is trivial; the concept is universal (that is medium-independent).

Defining a router

Networks are connected by what are called routers. A router is simply a node that when given a packet, passes it on to an appropriate node. Thus, routers determine the path that packets must travel to reach their final destination.

What is an IP address?

Each IP node has a unique IP address. These IP addresses are assigned by the network manager according to the IP network the computer is attached to. Each IP address is a four-tuple. A four-tuple is essentially four numbers (each number is referred to as an octet) separated by a period, where each number should be in the range from 0 to 255 inclusive. Each digit is referred to as an octet, because 8 bits are required to store it. For example, the IP address for the *minotaur* server at the University of the West Indies is 196.3.0.2.

The use of domain names over IP addresses

Users normally refer to computers by domain names rather than IP addresses. These domain names are translated to IP addresses by a distributed database known as the Domain Name System (DNS). Now, that we have dispensed with the terminology required to understand the rest of this text, we can speak about how two nodes speak to each other.

Simple explanation of communication between two nodes

Data flow over the Internet is conceptually a very simple issue. A network application sends information through a series of layers. These layers send it over the physical connections and it is received by the destination machine, where it is sent up a few layers and given to the receiving network application. To understand how one can be

attacked on the Internet, one has to understand this seemingly easy process.

Describing a typical TCP/IP node

The layers that we just mentioned are what we refer to as the TCP/IP suite (in the strictest sense, we should actually call them the Internet protocol suite). Each node on the Internet must conform to a particular architecture, as stated in the TCP/IP specification. This architecture is shown in Figure 1.

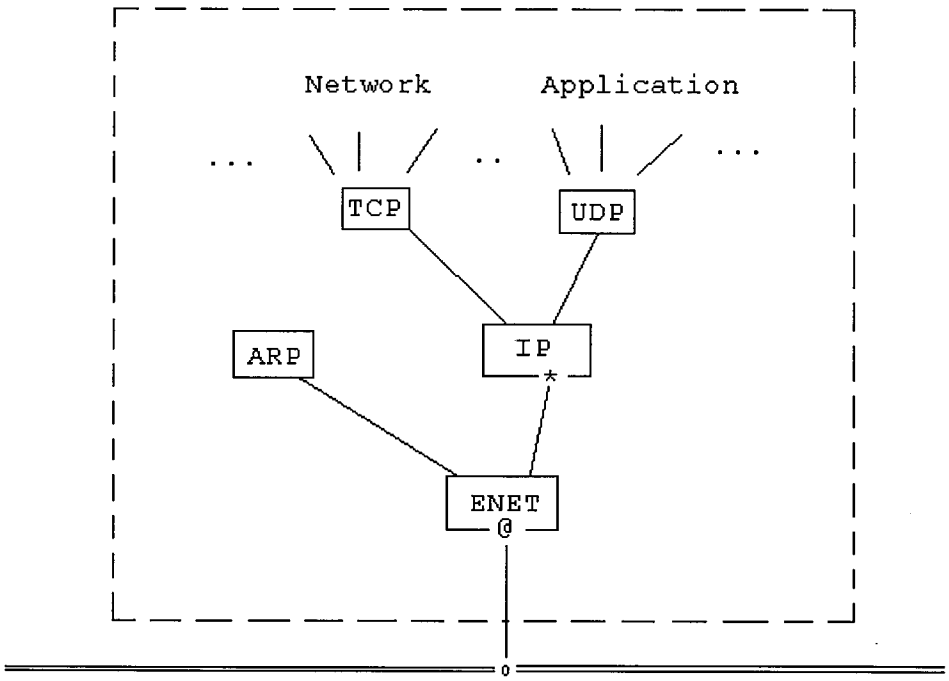


Figure 1. Basic TCP/IP Network Node

Key for Figure 1

ENET	Ethernet
*	IP address
o	Transceiver
@	Ethernet

Address

Explaining the finer details of Figure 1

A running computer must have at least one IP address and one Ethernet address. The double line at the bottom of Figure 1 represents

the Ethernet cable. I will not go too in depth into the inner workings of the basic elements of a TCP/IP node, but bear in mind that some level of detail is required.

Overview of the flow of data through a typical TCP/IP node

Now, a network application message can take one of two paths when it is being sent over the Internet. It may be sent to the TCP module, which passes it to the IP module, which sends it to ENET module, which in turn puts in on the physical medium. The alternate route is sending the message to the UDP module, which sends it to the IP module, which passes it to the ENET module, which passes it to the physical connection. What determines which path is taken? The answer is simple, the network software. The network software puts a header (whether UDP or TCP) on the front of your data and this states the module it should go to. Now we will examine each module in our TCP/IP node in order to give us a clearer picture of how communication takes place.

Examination of TCP

TCP is a connection-oriented, reliable transport protocol. Connection-oriented protocol implies that two hosts participating in a discussion must first establish a connection before data may be exchanged. This is done with the three-way handshake process. Reliability can be provided in a number of ways; the two of interest to us are data sequencing and acknowledgement. TCP assigns sequence numbers to every byte in every segment and acknowledges all data bytes received from the other end. TCP puts a header at the front of each datagram. This header contains at least 20 octets, but the most significant ones are a source and destination port number and a sequence number.

Basic purpose of TCP

Basically, TCP is responsible for making sure that the commands get through to the other end. It keeps track of what is sent, and retransmits anything that did not get through. If any messages are too large for one datagram, TCP will split it into several datagrams, and make sure that they all arrive correctly.

Examining a few features of TCP

Let us examine more closely TCP sequence numbers and the three-way handshake process.

TCP sequence numbers

TCP sequence numbers can simply be thought of as 32-bit counters. They range from 0 to 4,294,967,295. Every byte of data exchanged across a TCP connection (along with certain flags) is sequenced. The sequence number field in the TCP header will contain the sequence number of the first byte of data in the TCP segment. The TCP control flags are: SYN (Synchronize Sequence Numbers), ACK (Acknowledgement), RST (Reset), URG (Urgent), PSH (Push) and FIN (Finish). For more information on these flags, please read *Internetworking with TCP/IP Vol. 1 by Douglas E. Comer.*

The three-way handshake

The three-way handshake process is illustrated in Figure 2.

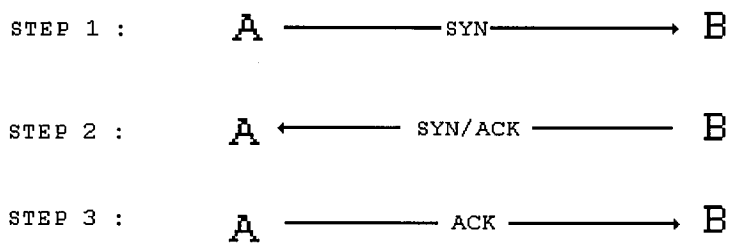


Figure 2
The three-way handshake

[3]At step 1, A tells B that it wants a connection. This is the SYN flag's only purpose. A tells B that the sequence number field is valid, and should be checked. A will set the sequence number field in the TCP header to it's ISN (initial sequence number). B, upon receiving this segment (at Step 2) will respond with it's own ISN (therefore the SYN flag is on) and an acknowledgement of A's first segment (which is the client's ISN+1). A then ACK's the server's ISN (3). Now data transfer can take place.

How does TCP allow many users to access it at the same time?

[3] "Project Hades", Phrack Magazine, Vol. 7, Issue 49.

To grant simultaneous access to the TCP module, TCP provides a user interface called a port. Ports are used by the operating system kernel to identify network processes. They are strictly transport layer entities. Together with an IP address, a TCP port provides an endpoint for network communications. In fact, at any given moment all Internet connections can be described by 4 numbers: the source IP address and source port and the destination IP address and destination port.

Examining UDP

UDP is a connectionless datagram delivery service that does not guarantee delivery. UDP is considered an alternative to TCP. It is designed for applications in which the sequencing of datagrams together is not important. It fits into the system much like TCP. UDP sends the data to IP, which adds the IP header, putting UDP's protocol number in the protocol field instead of the TCP protocol number.

UDP versus TCP

UDP does not do as much as TCP. It does not split data into multiple datagrams and it does not keep track of what it has sent so it can re-send if necessary. However, UDP provides port numbers so that several programs can use UDP at once. UDP is primarily used when one query is normally simple (that is, it can fit in one datagram). In this case the complexity of TCP is not required and would in fact add unnecessary overhead. For example, if computer A does not know the Ethernet address of computer B, it may use UDP to send a small message requesting that computer B send the information.

A look into the IP module

IP is a connectionless, unreliable network protocol. It has 32-bit header fields to hold address information. IP's job is to route packets around the network. It uses its routing table to determine which node to send the packet to next. Details of the routing mechanism used will not be discussed here. For more information on routing, please peruse "*An introduction to Internet Protocols*", *Phrack Magazine, Vol. 3 Issue 29*. IP provides no mechanism for reliability or accountability. It leaves this to the layers higher in the hierarchy. IP simply sends out datagrams and hopes they make it intact. If they don't, IP can try to

send an ICMP error message back to the source, however this packet can get lost as well. Since IP is connectionless, it does not maintain any connection state information. Each IP datagram is sent out without regard to the last one or the next one. This, along with the fact that it is trivial to modify the IP stack to allow an arbitrarily chosen IP address in the source and destination fields make IP easily subvertable.

An examination of ICMP

ICMP is similar to UDP in that it handles messages that fit in one datagram. However it is even simpler than UDP. It does not even have port numbers in its header. Since all ICMP messages are interpreted by the network software itself, no port numbers are needed to say where it should go. ICMP is used for relaying error messages. For example, you might try to connect to a system and get back a message saying "Host not found".

To complete our discussion about the components of a TCP/IP network node, let's turn our attention to ENET and ARP.

Looking at ENET

ENET (Ethernet) is similar to a party line (this is where everyone is connected to a shared line, and everyone is able to listen in on a conversation, even though the conversation may be addressed to a particular person). When packets are sent out on it, every host on the Ethernet sees them. The Ethernet is not very secure; it is possible to have packets being read by two or more places.

Clearing up the Ethernet-Internet connection

Please be advised that there is no *connection* between the Ethernet and the Internet. A host that is on both the Ethernet and Internet has to have both an Ethernet connection and an Internet server. Again, let me state that the Internet supports a myriad of physical media and as an issue of preference I have chosen to use Ethernet in this paper.

An examination of ARP

ARP translates the IP address into an Ethernet address. A conversion table is used (this is called an ARP table) to convert the addresses.

This allows one to specify an IP address and connect to one's Ethernet (because ARP does the conversion for you).

Now that we have examined all the pieces that comprise the TCP/IP node, we now have a better picture of how an Internet commerce transaction takes place. It is also prudent to say that each TCP/IP node has a set of basic services that can be done on it.

The basic services that an TCP/IP node should possess

Any system on the Internet is expected to possess a minimal set of network applications. Applications must be provided to allow the following:

- Mail
  Allows one to send messages to other users on other computers.
- File Transfer
  Allows a user on any computer to get files from another computer, or send file to another computer. A program called FTP (File Transfer Protocol) is normally used to this.
- Remote login
  Allows a user to log in on any other computer on the network. A program named telnet accomplishes this task.

The reason for it all

This completes our discussion on how the Internet works. Please note that details concerning routing, domain name servers, OSI layers etc not been included. This would only add to this document's complexity and not significantly affect the comprehension of the topics left to be discussed. The information provided in this section will be the basis of the attack methods that will be discussed in Chapter 5.

"A mesh of interconnected servers and clients that use the same standard format for creating documents (HTML) and accessing documents (HTTP). The mesh of links, both from server to server and from document to document, is metaphorically called the Web."
- http://www.glossary.com

"A collection of resources (Gopher, FTP, http, telnet, Usenet, WAIS and others) which can be accessed via a web browser."
- Anonymous

"A collection of hypertext files available on web servers."
- Anonymous

"A set of specifications (protocols) that allows the transmission of web pages over the Internet."
- Anonymous

A brief history

One can envision the World Wide Web as a worldwide collection of text and multimedia files and other network services interconnected via a system of hypertext documents. HTTP (Hypertext Transfer Protocol) was created in 1990, at CERN, the European Center for Nuclear Physics in Geneva, Switzerland, as a means for sharing scientific data internationally, instantly, and inexpensively.

The function of hypertext

Hypertext allows a word or phrase to contain a link to other text. To achieve this, they developed a programming language called HTML (Hypertext Markup Language), that allows you to easily link you to other pages or network services on the Web.

The basic elements of the WWW

The basic elements of the World Wide Web are:

- HTTP (Hypertext Transfer Protocol) - the set of standards used by computers to communicate and share files with each other.
- URL's (Uniform Resource Locator) - the "address" of a resource (file or directory) on the Web.
- HTML (Hypertext Markup Language) - the programming "tags" added to text documents that turn them into hypertext documents.

The infrastructure upon which the Web was built is the Internet. Thus, each node on the Web must conform to the TCP/IP node specification.

4.5 The components of an Internet commerce transaction?

The four critical components

Every Internet commerce transaction involves four critical components[4], namely:

- The software being run by the customer (client-side software),
- The transaction protocols,
- The Commerce Server and
- The operating system software.

The even more critical, but less controllable component

I will venture to suggest that humans should be included as one of these critical components. Though, they pose a security risk that is beyond reasonable control, their influence on the process of Internet commerce is very important.

A description of the four critical elements

The client-side software is basically the browser being run by the client. Transaction protocols are the standards used to carry out a transaction. The Commerce Server offers the company server's to the customers and the operating system software is the foundation upon which all software must run.

4.6 How does Internet commerce work?

The essentials of Internet commerce

There are two things that are essential to Internet commerce; namely commerce servers and clients. A firm's commerce server is the machine that acts as the company's interface to customers and processes all on-line transactions for all services offered the server. In the case of a retail company, their commerce server may have a storefront (a virtual view of the company) or an ordering system (to allow clients to order goods on-line).

A look at the commerce server

There is no standard as to what the structure of the commerce server

[4] Ghosh, Anup K "E-Commerce Security: Weak Links, Best Defenses", 1998, Pg. xii.

should look like. Since the server is connected to the Internet, it should operate like any other TCP/IP node.

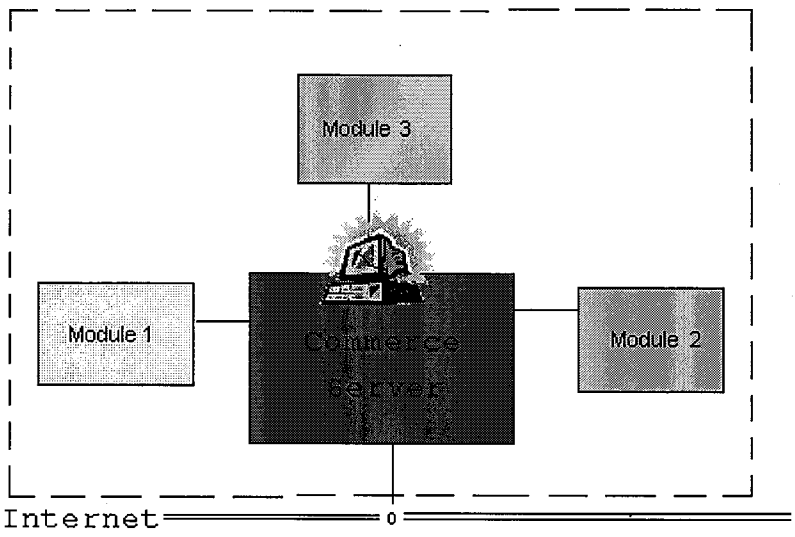


Figure 3: Typical Internet Commerce Environment

The basic model for a commerce server is illustrated in Figure 3.

The server is connected to the company's intranet (Modules 1, 2 and 3 are a part of the company's intranet), as well as being connected to the Internet. Normally, safeguards are put in place to prevent access to the company's intranet via the commerce server.

The operation of the commerce server

Internet commerce operations are easily described. A firm offers services on its commerce server. A client uses these services. The operation mechanism of a transaction is normally governed by a transaction protocol, for example SET (Secure Electronic Transaction), but the field is too young for any definitive discussions to take place. Essentially, all we need to know is that since these two entities are IP nodes, a data transmission in its simplified form is just two TCP/IP nodes communicating.

The underestimation of Internet security

The architects of the Internet had good intentions when they included openness and interoperability in their design philosophy. However, as art tries to mimic life, the Internet is reflecting the diversity that exists in real life. People that are willing to exploit other people and people willing to assist anyone in need are common to our everyday lives, and such they will be common in the world of the Internet. Security is an integral part of human life, and as such, more thought should have gone into making security an integral part of the Internet.

Public reaction to Internet commerce

Internet commerce has been received with skepticism by many people, because of over-glamorizing by the media of acts of computer theft and fraud. It is indeed true that a few years ago, not much consideration was paid to security of transactions on the Internet, but now advances are being made. Nonetheless, we cannot afford to be complacent. As the Internet and its attackers evolve, so must the security.

Chapter 5
How Internet commerce can be disturbed?

"If you're looking for computer security then the Internet is not the place to be"[5]
- Mudge, member of "The Loft" (a hacker think-tank)

"Through an Internet connection, an experienced computer user can easily control any other computer on the Net"[6]
- Nahum Goldman, CEO Computer Security Canada Inc.

Overview

In this chapter, I will focus on the ways in which an attacker may try to tamper with business-consumer relations. I will only be presenting a generalized list of attack methods. As this chapter is being written there are many more system-specific and system-independent attack methods being formulated and documented. The important thing to remember is that an Internet commerce transaction can be compromised if any of the four critical components of an Internet commerce transaction is breached. The attack methods that we will be discussing in this chapter are: malicious programs, software flaws, spoofing, jamming, sniffing, hijacking, server attacks, routing attacks, message alteration, replay and delay, TEMPEST, poor internal security measures and unauthorized access.

5.1 Malicious Programs

General Description

Malicious software (otherwise called malware) is software that is engineered to have destructive effects on one's computer system. Malicious programs may be used to compromise the client-side software, the commerce server or the operating system software. There are many ways in which malicious code behave and even more

[5] Associated Press, Tuesday, May 19, 1998.
[6] Mitton, Jon-Paul, "Web Security Certification Prevents Rash of Banking Security Breaches".

ways in which they can be spread. Some programs are placed on a target machine and execute periodically, while others start executing once they have been triggered. In this section, I will discuss some of the ways in which a malicious program can be transported and some of the various types of malicious programs that exist today.

The transport of malicious programs

In today's society, there are a myriad of ways in which malicious programs can be spread. Among them are: using steganography, using push technology, using dynamic web pages, using worms and using cookies.

Steganography

Steganography is essentially the hiding of a message in a larger message. Its primary purpose was to be an efficient way to send confidential messages. However, in recent times, a darker side of steganography has been uncovered. As well as, being able to send confidential messages, one can also send malicious code. The great danger with steganography is that if applied properly, the hidden code or message can be undetectable. Steganography allows someone to hide any program in images, photographs, audio clips, etc.

Push Technology

Push technology, in spite of its merits, can be manipulated to be a transportation agent for malicious programs. The premise of push technology is that the user gives a site the right to periodically place (push) information on to their machine. Once the initial permission has been given, the site may place any program onto the user's machine (malicious or otherwise) and the user does not (and many times will not) be prompted when files are being pushed or even told what files are being pushed. A prime example of the application of push technology is in getting the latest versions of the McAfee anti-virus program from the McAfee web site. One can ask the administrators of the McAfee web-site to download the latest upgrades directly to your computer (this is perceived to be far better than actually periodically checking the web-site oneself). The danger with this is that one gives the site's administrators the rights to one's computer, so they are free to download any program they want to

one's machine (even malicious code).

Dynamic Web Pages

Dynamic web pages, that is web pages that do more that just display information, have led to yet another technology that can be exploited to transfer malicious code. Dynamic web pages are said to contain executable content (also called *active content*). Currently, any web page that contains active content may potentially contain a malicious program waiting to be downloaded to a machine. Web users should be very careful of active content. Any web page that utilizes recent Internet programming advancements such as Java applets, ActiveX controls, JavaScript programs, VBScript programs and even plug-ins (which are special-purpose interpreters that execute audio files, video files, etc.) may potentially contain malicious code.

Worms

Worms are programs that are used as transportation mechanisms for other programs. They use a network to spread programs from one system to another. Normally, worms utilize a flaw in a network operation to get its package from one system to the next. A worm has three basic processes:

- Initially, it searches for a target system,
- Then it establishes a connection to this system
- And finally it transports its program to the target system and execute its program.

This executing program may be another worm or it may be malicious code.

An infamous example of a worm is the program that was released on the Internet in November 1988. This program caused most of the network to come to a standstill. Another example is the case of a Russian hacker who used a worm program to steal $8 million dollars from a firm.

Cookies

A cookie is a file that stores information on a user's operations during a web session. Cookies were conceived to solve the issue of the statelessness of the HTTP protocol. Statelessness refers to the fact

that every web request to a web server has no knowledge of the last request sent to that server. Since cookies are stored on the user's machine, it may be possible for a web server to modify files on the user's machine (by going through the cookie) and execute his own code (which may be malicious). Having studied how malicious program can get on one's machine; an examination of some of the various types of malware is appropriate.

The types of malicious programs

The types of malicious programs are: viruses, logic bombs, parasites and Trojan horses. Please note that these malicious programs are by no means independent. It is indeed possible for logic bombs to contain viruses or even for viruses to contain parasites.

Viruses

A *virus* is a program that infects another program by replicating itself into the program. A virus has to go through three phases:

- The infection phase, where a program is infected from an existing virus
- The activation phase, where this newly infected program is signaled to find another program to infect.
- The replication phase, where the virus finds a suitable program and copies itself to it.

The majority of viruses are destructive, however there are some that only replicate themselves. These types of viruses (often referred as bacteria or rabbits) merely consume system resources and may degrade the performance of one's system.

Logic bombs

A *logic bomb* is a program that remains dormant on a computer system until it is activated. Logic bombs can be activated by anything that the computer system can detect. Normally, activation of these logic bombs is time-based (a *time bomb)* or based on the absence or presence or some piece of data. For example, a programmer may build a software package with a logic bomb that is activated if there is no evidence on the computer system that the program has been fully paid for. When activated, the logic bomb will fulfill its function, whether it is to erase all the data on someone's machine or it is to send all your

financial information back to the programmer.

Parasites

A *parasite* is a piece of code that is added to an existing program and draws information from the original program. It is primarily used to gather information for an attacker, but it may also be used to trigger the execution of a malicious program.

Trojan horses

A *Trojan horse* is a software entity that appears to do something normal but which also does some unwanted operation. A perfect example of a Trojan horses is a program that mimics your ordinary login program and also stores your password for later perusal by an attacker. Another example is a program that claims to be the new enhanced version of the game Command & Conquer, but is actually the original game with a program that steals files from your computer system and may even delete the entire file system.

5.2 Software Flaws

The misguided trust of software buyers

Software flaws may exist in the client-side software, the commerce server software or even the operating system software. There has traditionally been a trust relationship between software vendors and their customers. Customers assumed that the software being provided contains no flaws. This is never the case. Current software engineering techniques used to produce commercial software can only detect the presence of an error, they cannot test for the absence of an error. Thus, a company has to be aware of the software that it is using. Take for example, the Pentagon. They recently tested 15,000 of their systems. The Information Warfare Division of the Defense Information Systems Agency of the U.S. Department of Defense found that 90% of the systems were vulnerable to common intrusion techniques[7]. This is not an isolated example. Errors are being discovered in the UNIX operating system at a rate of one error per day. Many errors are being uncovered each day in existing software packages.

[7] Johnson, Anna "Companies Losing Millions over Rising Computer Crime".

An example of the exploitation of a software flaw

Consider, a demonstration given by the German Chaos Club of the security problems with ActiveX controls. On 28 Jan 1997, a Berlin Television broadcast showed them performing a demonstration of how they could electronically transfer funds from someone's account without needing to steal the PIN (Personal Identification Number). A victim visits a web site that downloads an ActiveX application automatically. The application starts and checks to see if Quicken (a popular financial software package) is loaded on the user's hard disk. If so, Quicken is given a transfer order, which is saved by Quicken in its pile of pending transfer orders. The next time the victim sends off the pending transfer orders to the bank (and enters in a valid PIN for that) all the orders are executed and the money transferred without the user knowing.

5.3 Spoofing

General Description

Spoofing is essentially electronic impersonation. In a spoofing attack, someone fraudulently represents themselves as some other person or organization. A successful spoofing attack may call into question the trust relationship that should exist between clients and producers. A smart impersonator may easily corrupt this relationship for his own benefit. This research has lead me to believe that there are four major spoofing techniques, namely:

- Web Spoofing
- Address Masquerading
- Address Spoofing (TCP Spoofing)
- DNS Spoofing

5.3.1 Web Spoofing

General Description

In web spoofing, the attacker creates a "shadow copy" of the entire World Wide Web[8]. All access to this "shadow" web must go through the attacker's machine, allowing the attacker to access any data going

[8] Felten, Edward W., Dirk Balfanz, Drew Dean, and Dan S. Wallach "Web Spoofing: An Internet Con Game", Technical Report 540-96, Department of Computer Science, Princeton University.

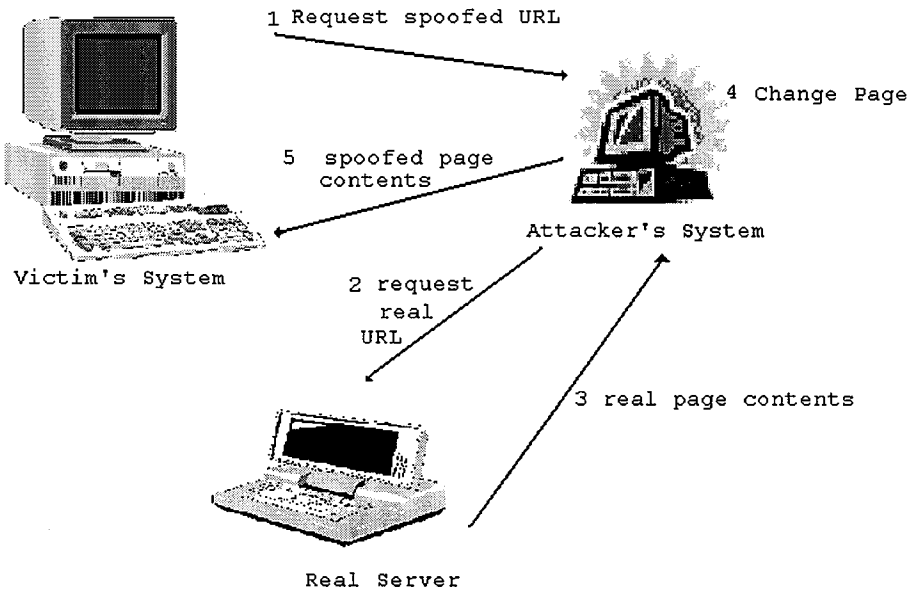
into or coming out of his target's system. This attack method uses a little technique called URL rewriting to make an Internet commerce transaction totally transparent to an attacker.

The requirements needed for Web spoofing to work

For this attack to work, the attacker's Web server must be the "middle man". He must be between the victim and the rest of the Web. How is this done? There are many ways in which this can be done and the more creative and knowledgeable the attacker, the more ways he will come up with. For example, he could lure his victim into his "false" Web by putting a link to the "false" Web in a popular Web page or even simply by emailing the target a pointer to this "false" Web.

The operation of the attack

After the victim has been lured into the attacker's world. The attacker rewrites all of the URLs any web page requested by the victim so that they point to the attacker's server rather than to the real server. To further illustrate this, let us assume that the attacker's server is on the machine www.evil.com. The attacker will rewrite a URL by adding http://www.evil.com to the front of the URL. For example, http://www.yahoo.com becomes http://www.evil.com/http://www.yahoo.com. Figure 3 shows what happens when the victim requests a page.



The steps are as follows:

1. The victim's browser requests the page from the attacker's server
2. The attacker's server requests the page from the real server
3. The real server provides the page to the attacker's server
4. The attacker's server rewrites the page
5. The attacker's server provides the rewritten page to the victim.

Since all the URLs in the rewritten page refer to the www.evil.com; if the victim follows a link on the new page, then he will still be getting pages via the attacker. The victim is more or less trapped in the attacker's false Web.

5.3.2 Address Masquerading

General Description

In address masquerading, an attacker configures his network interface with the address intended for another system (the target). Normally, when the target shuts down for routine maintenance or is forced to shut down due to the attacker, an attacker can masquerade as the target by configuring his system with the target's network address.

5.3.3 Address Spoofing (TCP Spoofing)

General Description

This is also called the TCP sequence number attack. Let's call the target system A and the attacker's system B. The attack proceeds as follows:

- B knows that A trusts system C enough to allow him open access to his system.
- Thus, B tries to impersonate C.
- B will keep C busy while he creates IP datagrams that appear to be coming from C and attempts to predict the TCP sequence number that will be produced next (Recall the three-way handshake).

TCP sequence number prediction attacks can be very effect on unprotected networks (that is networks that do not generate TCP sequence numbers in an unpredictable fashion)

5.3.4 DNS Spoofing

General Description

In a DNS spoof attack, the attacker forges information about which machine names correspond to which network addresses. The attacker then assumes the DNS name of another system by either corrupting the name service cache of a victim system, or by compromising a domain name server for a valid domain.

The operation of a type of DNS spoofing attack

The attack that corrupts a domain name server proceeds as follows :

- The attacker advertises two IP addresses for itself, its real IP address and the IP address of the target machine.
- The attacker keeps the target occupied.
- Now, when anyone refers to the target the DNS entry will point to the attacker and he will get all the target's packets.

Additional spoofing attacks

There are many more spoofing methods that are system-specific. For example, TTY spoofing, which is specific to UNIX systems, involves the theft of login accounts by the attacker and the act of pretending to be these victims (for more on TTY spoofing, please read Phrack Magazine Vol. 4, Issue 49).

5.4 Jamming

General Description

Jamming (also called "Denial of Service" or "Message Denial") is an attack on devices and networks. This attack cripples the aforementioned resources and in effect denies external users access to these resources[9]. In a nutshell, jammers use software programs to tie up a Web site's server. An attacker needs only to run a program that will generate enough traffic to one's site that it denies service to the site's legitimate users.

Types of jamming attacks

There are three types of Denial of Service (hereafter referred to as

[9] Downey, Jeff "Denial of Service Attacks: Can't Say No", PC Magazine, April 21, 1998, Vol. 17 No. 8, 203-204

DoS) attacks:

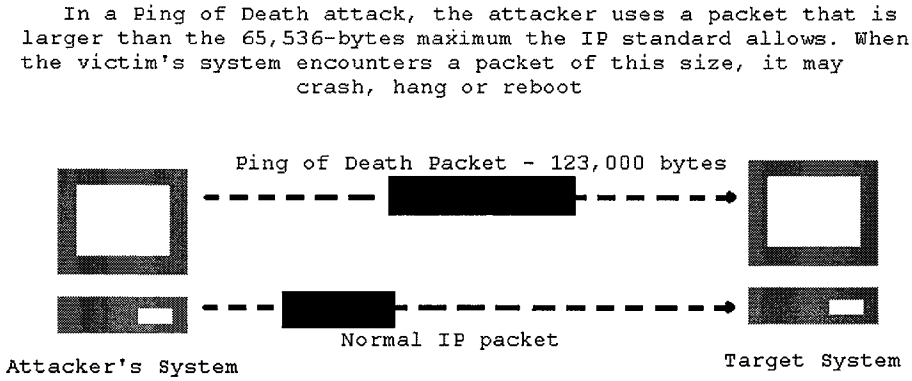
- Those that exploit a bug in a TCP/IP implementation,
- Those that exploit a shortcoming in the TCP/IP specification and
- Brute-force attacks that congest your network with so much useless traffic that no other traffic can get in or out.

Examples of jamming attack methods

A few of the more popular attacks are: the Ping of Death attack, the Teardrop attack, the SYN attack, the Land attack, the Smurf attack and the UDP Flood attack. The Ping of Death and Teardrop attacks exploit bugs in TCP/IP implementation. The SYN and Land attacks exploit weaknesses in the TCP/IP specification. The Smurf and UDP Flood attacks are brute-force attacks. Let us now focus our attention on each of these attacks.

The Ping of Death attack

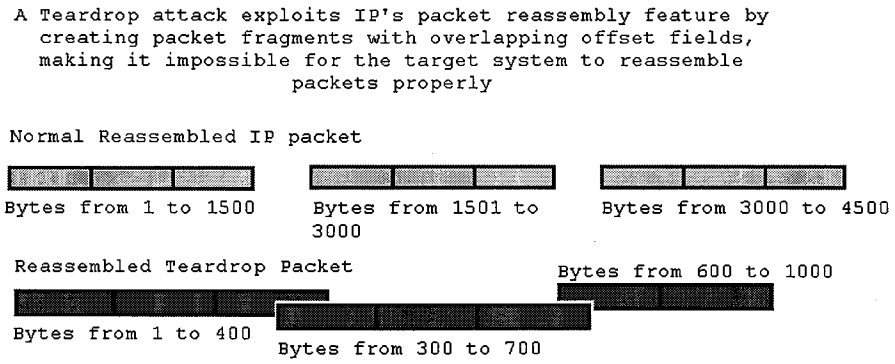
The Ping of Death attack uses a program called *ping* (which can be found on most network operating systems) to create an IP packet that exceeds the maximum 65,536 bytes of data allowed by the IP specification. This packet is then sent to the victim and when he receives it, his system may hang, reboot or crash his computer (depending on the operating system he is using).



The Teardrop attack

The Teardrop attack takes advantage of a weakness in the reassembly of IP packet fragments. As stated before, as IP packets proceed along their journey, they may have to be broken up into smaller chunks; each called a fragment. Each fragment looks like the original IP packet except it contains an offset field that says, for instance "This

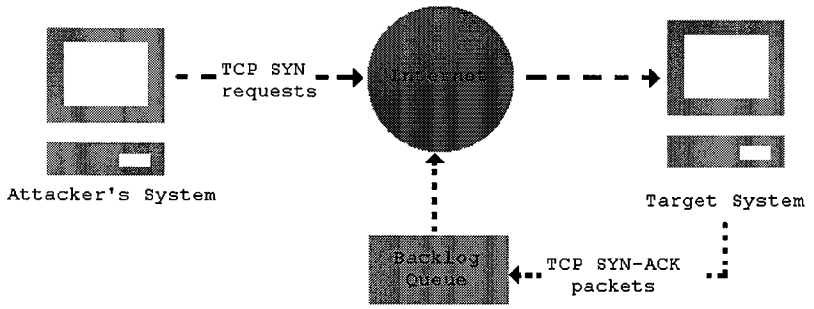
fragment is carrying bytes 300 through 700 of the original IP packet". The Teardrop program creates a series of IP fragments with overlapping offset fields. When these fragments are reassembled at the destination host, the machine will either crash, hang or reboot.



The SYN attack

A SYN attack exploits a weakness in the TCP/IP specification. Normally, an application that initiates a session sends a TCP SYN synchronization packet to the receiving application. The receiver sends back a TCP SYN-ACK acknowledgement packet and then the initiator responds with an ACK acknowledgement (this is the three-way handshaking discussed earlier). A SYN attack floods a targeted system with a series of TCP SYN packets. Each packet in the flood has a bad source IP address. Each packet causes the targeted system to issue a SYN-ACK response. While the target waits for the ACK that follows the SYN-ACK it queues up all outstanding SYN-ACK responses on what is known as a backlog queue. This backlog queue has a finite length and is quite small. Once the queue is full, the system will ignore all incoming SYN requests. SYN-ACKs are moved off the queue only when an ACK comes back or when an internal timer, which is set at relatively long intervals, terminates the three way handshake. Since, all responses are sent to the source IP address. But this source IP address either does not exist or is down, therefore the ACK that should follow a SYN-ACK response will never come back. This creates a backlog queue that is always full, making it nearly impossible for legitimate TCP SYN requests to get into the system.

A SYN attack exploits IP's three-way handshake model. By only initiating the handshake and not responding to the second portion, the attack forces the target system to store acknowledgement packets in its finite-size backlog queue until the queue overflows and will not acknowledge any more requests.



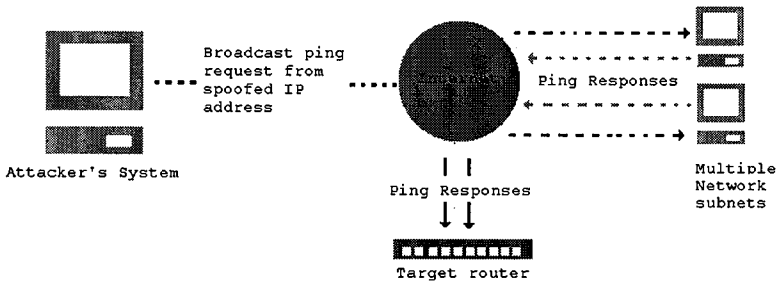
The Land Attack

A Land attack is a variation of the SYN attack. In a Land attack, SYN packets are flooded into a network with a spoofed source IP address of the targeted system.

The Smurf attack

The Smurf attack is a brute-force attack targeted at a feature in the IP specification known as direct broadcasting addressing. A Smurf attacker floods your router with Internet Control Message Protocol (ICMP) echo request packets. Since the destination IP address of each packet is the broadcast address of your network, your router will broadcast the ICMP echo request packet to all hosts on the network. If you have numerous hosts, this will create a large amount of ICMP echo request and response traffic. If an attacker chooses to spoof the source IP address of the ICMP echo request packet, the resulting ICMP traffic will not only clog up your network (the intermediary), but will also congest the network of the spoofed source IP address (the victim).

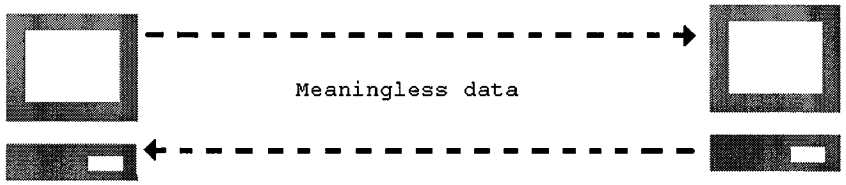
A smurf attack happens when an attacker spoofs the target system's IP address and then broadcasts a ping request to several subnets. The resulting flood of ping responses ties up the victim's system as well as the various network subnets pinged.



The UDP Flood attack

The UDP Flood attack links two unsuspecting systems. By spoofing, the UDP Flood attack hooks up one system's UDP chargen service, which for testing purposes generates a series of characters for each packet it receives, with another system's UDP echo service, which echoes any character it receives in an attempt to test network programs. As a result, a nonstop flood of useless data passes between the two systems.

An attacker creates a UDP Flood by connecting an unsuspecting system's UDP character-generating service, chargen, to another unsuspecting system's UDP echo service. Once the link is made, the two systems are tied up exchanging a flood of meaningless data



Other jamming details

The details on the jamming methods discussed above have been taken from Jeff Downey's article on "Denial of Service Attacks" in the April 1998 issue of PC Magazine. Jamming is a very scary reality for anyone with a commerce server. It has the potential to allow firms to lose vast sums of money. Though, it has not been a common attack strategy used by hackers, it is gaining in popularity in recent years[10].

2.6 Sniffing

General Description

Sniffing, otherwise called snooping or sniffering, is essentially electronic eavesdropping. Originally, sniffers were used to debug application and network problems. Sniffers are network monitoring programs that are secretly installed on network hosts by attackers[11]. A

[10]Downey, Jeff "Denial of Service Attacks: Can't Say No", PC Magazine, April 21, 1998, Vol. 17 No. 8, 204.

[11] Update on Network "Sniffing" Security Vulnerabilities, NASA Automated Systems Incident Response Capability, #94-10.

sniffer runs quietly on the target system recording any information, from all incoming and outgoing traffic to the target, that the attacker finds interesting. The sniffer writes this information to a hidden file that is retrieved later by the attacker.

Initial questions to be resolved

Two questions immediately come to mind. How does the attacker gain access to the target in the first place? And why are these sniffer programs so hard to detect? The answer to the first question is simple. He exploits the knowledge he has about an individual's system. He could know that the password file is world-reachable by certain programs (for example, TFTP). He could know that the local file system can be easily mounted on to his system (world-mountable). He could have simply gotten a login that was captured by another sniffer. Once he has logged on to the target he can exploit some known operating system flaw or use a sniffed "system administrator" password to install his sniffing program. Now why are these sniffers so hard to detect? As part of the installation, one of the system's critical files will be replaced with a Trojan horse to hide the sniffer. On a UNIX system, he may replace */usr/login* or */usr/kvm/ps* or */usr/etc/in.telnetd*. On a Windows NT system, he may replace *winlogon.exe*, *services.exe* or even *ntoskrn1.exe*.

The many uses of sniffers

Sniffers can also be used in many diverse ways. They could be used to steal credit card numbers from Web vendors. They could be used as the first or intermediary step for other attacks, for example DNS spoofing. They could be used to inject Trojan horses into downloads and they could be could be used to track people's activities by viewing their cookie files.

General Description

In a hijack, the intruder uses privileges, attained through less than legitimate channels, to get into someone's system (the target) and take control of the target's resources. Hijacking comes in many flavors. Most resources that are used by an Internet user can be hijacked, from

his phone connection to a datagram he sent to his telnet connection. The most popular kind of hijack is called session hijacking.

Session Hijacking

In session hijacking, the intruder taps into a system's software that accesses or controls the behavior of the local TCP module. A successful hijacker will be able to use the open connection of his target for his own purposes. If the target has already been authenticated, then the system assumes that any information sent by the hijacker is actually sent by the target. This has grave implications for both the client and the producer. Its effect on Internet commerce is similar to that achieved by spoofing.

5.7 Server Attacks

General Description

In this context, a server attack is an attack that is focused on the commerce server. A commerce server can be attacked from anyone of four angles. An attack may exploit the:

- Server operating system software
- Front-end server software,
- Back-end databases or
- Interface software

Let us now take a look at each of these potential points of attack.

Server operating system software

This is the operating system that is run on the server. If there are any ways to compromise the operating system then it is possible for someone to bypass all the security checks in the software running on the operating system. This means that a flaw in the operating system of the server will lead to the entire server being insecure, in spite of all the security measures placed on the other software on the system.

Front-end server software

Front-end server software is essentially that section of the system that people will be working with. It implements the screens and messages that are displayed. It allows users to perform transactions and

administrators to monitor and maintain the system. The complexity of front-end server software is an indication of the number of flaws present in it. The more complex the software, the more errors it is likely to have. Attackers normally exploit flaws that arise as a result of the incorrect installation and or configuration of the software.

Back-end databases

Back-end databases store the transaction data for the firm and as such their security is of the utmost importance. However, they can lack the correct access controls needed to prevent unauthorized persons from browsing the database or, even worse, they can be accessible to anyone on the Web (as was the case in 1997, when the U.S. Social Security Administration made the Personal Earnings and Benefits Estimate Statements of U.S. citizens available for Web requests).

Interface software

The interface software are those programs that allow one to retrieve information from Web forms, update back-end databases and perform on-line Web site searches. Being that interface software is written in a specialized language, such as CGI (Common Gateway Interface), there may be problems that appear due to the nature of the language. For example, CGI has properties that can be used to extract information from one's system. A user of CGI scripts can be used to execute system commands that might perform operations that may have negative effects.

5.8 Routing Attacks

General Description

IP supports both dynamic and source routing. Routing attacks exploit IP's source routing option. A given route between two points is dynamically determined based on factors such as link availability, distance and speed between routers and so on. However, a sending system can override a dynamic route by specifying a source-route. It is possible for an attacker to construct bogus route updates that hosts and other routers will believe and obey. The attacker can modify a route so that all packets going to or leaving from the target must go through his system. After hosts begin routing packets to an address

chosen by the attacker, every packet that travels the route is endangered. In the former case, the attacker is free to tamper with any information sent to or received from the target.

5.9 Message Alteration, Replay and Delay

Message alteration

Any message that travels along the Internet is at risk of having data added, removed or changed while it is on its journey. This is called *message alteration*. Remember that there is no mechanism for message integrity at the IP layer. An attacker can modify the header information in a datagram and the recipient will be unable to detect it.

Message Replay

An attacker may also make duplicate of a message and send copies of it to whomever he pleases. This is called *message replay*. A perfect example of where message replay may be used is when one is making an on-line purchase. If one is buying a small gift from Gifts-R-US, then it is possible for Gifts-R-US to hire a hacker to send (replay) the message to other Gifts-R-US associated companies. This would leave one with a seriously large bill.

Message Delay

An attacker may hold a message and not deliver it to its correct destination until he deems it fit. This is known as *message delay*.

5.10 TEMPEST

General Description

TEMPEST stands for Transient Electromagnetic Pulse Emanation Standard. It is a US government code word for a classified set of standards to limit electromagnetic radiation from electronic equipment. Given that electronic equipment such as computers, printers, etc. give off electromagnetic emanations, an attacker can use equipment to monitor and retrieve information being processed by any particular user. All of this is done without the user knowing that his actions are being watched.

The plausibility of a TEMPEST attack

Even though this attack seems to be taken straight from the pages of some spy novel, it is a realistic option for the attackers that must resort to extreme means to get the information they need. It is not at terribly expensive attack to implement. For example, to get the equipment to view the emanations of a monitor may cost in the region of five hundred US dollars. The fact that this attack is possible should not send people into a state of hysteria or paranoia, rather it should help them realize that there are dangers all around us. However, it is rumored that the risk of TEMPEST attack is a lot less than it used to be.

5.4.1 Poor Internal Security Measures

The threat from within

Most companies overlook or trivialize the threat from within. In a November 1997 report, the U.S. Senate Sub-Committee study revealed that internal users were responsible for nearly half of all break-ins. The Sub-Committee estimated losses of around $800 million in 1995 due to break-ins. The same is true for Australia, a study carried out by Deloitte company in 1998 revealed that 90% of the companies polled (three hundred Australian companies) had traced the source of a breach to person within the firm. This is definitely not a trivial figure.

The general feeling

One of the easiest ways to compromise a company's information system is to collaborate with someone with a wealth of knowledge about the firm (employees and ex-employees are excellent candidates for this job). Companies feel that they can protect themselves from inside attack and normally place a lot of effort on securing the system from external attack.

Issues that must be addressed

Among the many internal security problems are:

- Poor password management.
- Inadequate security facilities to protect hardware and software resources.

- Limited or no access control mechanisms for databases with sensitive information.
- Poor configuration of software systems.
- Lack of formal security policies.

Any employee, with knowledge of a firm's resources, will more effectively know how to hack into and easily get out of the firm's system. The employee, like everyone else, can use any of the other attack methods spoken about. Normally, it tends to be easier to compromise a system from the inside than it is to do so from the outside. The reason for this is that there is normally less emphasis placed on internal security than is placed on external security

5.12 Final Thoughts

In this chapter, several attack techniques have been outlined. None of these techniques are stand-alone entities. Several of them can be incorporated into an attack plan. It is hoped that the reader has gained some insight into these attack methods. Please bear in mind that the information presented represents the situation as of early 1998. And as such, it is wise to be up-to-date when it comes to these techniques, because new attacks are being tried every day and people are constantly working on nullifying the effects of the current ones.

Chapter 6
Internet commerce: Consequences of Security Breaches

Overview

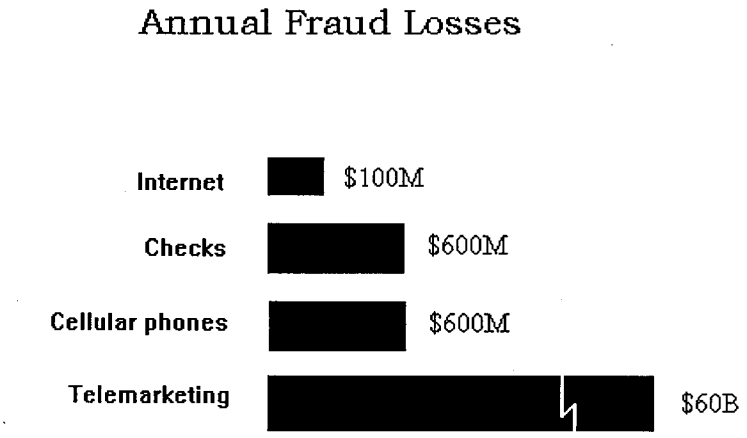
Any of the techniques presented in the previous chapter (or any combination of them) can be used to hurt a firm's reputation and or income. Security breaches are more prevalent than we would like to think and there is an ample supply of cases that demonstrate how a compromise can adversely affect a business. In this chapter, the various consequences that a firm can face will be discussed and real-life examples will be provided to show the frequency and randomness of these attacks. Please note that most of the examples discussed have been taken from the Risks-Forum Digest.

6.1 Fraud and Embezzlement

Life on the Internet models real life. It is therefore, not surprising to know that fraud is prevalent on the Internet. I have heard a line of argument being purported by E-Commerce over-enthusiasts that fraud on the Internet is very small in comparison to the fraud in other sectors. Their arguments are supported by the figures in Figure 4. However, according to Alexander Baugh, senior vice president of professional indemnity at AIG Europe, "Computer fraud is growing at a rate of 500 percent a year". Even more startling is a recent study by Deloitte & Touche on behalf of the European Commission, which reported that businesses in the European Union have lost from 6 billion to 60 billion EC units - a major share of this is due to fraud over the Internet. The study notes that much of the fraud is as a result of criminals setting up Web sites that seem like legitimate businesses. These facts indicate to me that Internet fraud and embezzlement could easily get out of control if not properly addressed.

There are many institutions and personnel that share my view that fraud is an issue that should be taken very seriously. According to statistics from the U.S. National Centre for Computer Crime, fraud

makes up 44 percent of computer crime. A study done by the U.K. Association of British Insurers in 1996 estimates that the cost of computer fraud amounted to 250 million pounds. They also claimed that this was only 20 percent of the actual losses. These are not figures that one can simply disregard as being comparatively insignificant. Several industry professionals have also been seeing the very real threat that exists. Take for example, Mike Carlton, senior manager of the Fraud Investigation Unit of Ernst and Young. He says the cost of computer fraud to UK businesses could be as high as 10 billion pounds per year. Fraud is big business, which needs to be taken a whole lot more seriously by everyone involved in Internet commerce.



Source: National Fraud information Center, Washington; first Union Bank, Charlotte, NC; Cellular Telecommunications Industry Association, Washingston.

Figure 4

A prime example of the effects of Internet fraud is the Burns National Bank, an American bank based in Colorado. This company was forced to cancel its debit-card program and reportedly lost an estimated US$300,000. Hackers counterfeited plastic cards and acquired the account number sequences from the bank's software. These account numbers were then encoded in the magnetic strip on the back of the cards. The bank detected the fraudulent scheme and equipped all subsequent cards with an extra security feature. However, within a month, the accounts were penetrated again.

6.2 Service Interruption or Degradation

Service Degradation or Interruption is possibly the most feared consequence of an attack. Every firm fears that someone can, with very little ingenuity, cut them off totally from their client-base. Many horror stories have been told about companies being attacked using denial-of-service attacks, and left with nothing to do but wait until the attacker has had his fun. The most horrifying reality is that attacks that cause service interruption or degradation are relatively easy to launch and are almost impossible to guard against. Here is a prime example of what can happen when such an attack is launched.

In 1996, PANIX (Public Access Networks), a New York-based Internet Service Provider, was attacked by a group of crackers who used the SYN attack against them. This attack left them unable to provide Internet service to any of their companies and literally crippled them.

6.3 Data Integrity Violation

Though, data integrity attacks as it pertains to the Internet are not a popular topic, these attacks can have severe financial repercussions on a firm. Most data integrity violations tend to be incidental and unintentional, but the possibility always exists that someone may maliciously alter your data for their benefit. Here is an example that shows the effects that data integrity violations can have on a firm.

[12]"In January 1997, a program error in discount brokerage Charles Schwab's Telebroker system resulted in incorrect information being conveyed to investors that queried account information over the phone. The program error resulted in a number of mutual funds being excluded from the accounting of investors' assets. Therefore, when investors used Telebroker to determine the value of their holdings in Schwab, the response from Telebroker was a value less than their actual holdings, if they held one of the excluded mutual funds. The result was panic among scores of investors who believed that the

[12] Ghosh, Anup K "E-Commerce Security: Weak Links, Best Defenses", 1998, 19.

market value of their holdings must have dropped significantly in a very short period of time. According to Schwab, a system upgrade or modification caused the resulting error."

6.4 Confidential Information Disclosure

Everyone fears that confidential information may get into the hands of the wrong people. How can one truly safeguard against their confidential data (which is either travelling over the Internet or stored in someone's server somewhere) being stolen and distributed to the wrong people? It is a question we have grappled with since the advent of government systems. How does one know that the government is not using the confidential information they have gathered on people for its own benefit? This question has to be answered in the non-Internet environment before the can be adequately addressed in the on-line world. Needless to say, the Internet and the openness of computer systems that contain your data make it just a bit easier for your data to be disclosed. Take for example, the case described below.

In 1997, the computer records of a bank in Japan were penetrated. Sensitive customer information was taken. The bank claimed that the problem occurred because of a software upgrade. During the upgrade, an employee from the software vendor allegedly swiped the information and offered it to a mailing company.

6.5 Privacy Violation

When carrying out a transaction over the Internet, most people assume that their transaction details will remain private. They assume that no one can see the transaction and that no one can attain information about the transaction. There are several ways someone can access someone's private data. We have seen the effectiveness and ease of sniffing. We have discussed the results of *active content* in web applications. We know that software bugs may accidentally give out your data. It is also possible to use the HTTP protocol to ascertain transaction details. Needless to say, data privacy is not guaranteed. It may be hard to achieve, but it is not an impossibility. Let us look at

two scenarios.

In November 1996, an error occurred in an E-commerce product called SoftCart, which resulted in consumers' credit card numbers (which were collected for purchase orders) being exposed to the Internet. That is, anyone on the Internet was able to access them. Mercantec, the software vendor, explained that the problem was due to human error. Mercantec purports that when the merchant installed SoftCart, the file with the credit card numbers was improperly placed in a directory completely accessible to the Web.

In the November 18, 1997 edition of the Swedish publication Svenska Dagbladet it was reported that U.S. software maker, Lotus, deposits encryption keys shipped with Lotus Notes with the United States Government. This is indeed the case. The result of this is that U.S. authorities are able to look at any sensitive information transmitted over the Internet by foreign companies that use the Lotus suite of programs.

2.0 Sabotage

Sabotage has been well documented and has been in existence long before the advent of the Internet. The Internet merely makes it easier and a lot safer for the knowledgeable perpetrator to carry out such a task. The Internet was designed on a faulty premise. It was designed on the premise that humans are all honest, ethical, trustworthy individuals. As such it is a saboteur's dream, virtually unlimited access to everything and everywhere (note the use of virtually, some places will require a bit of effort to get into). Sabotage is normally viewed as being performed by disgruntled ex-employees and industrial spies, but it is a lot easier to do these days, given the body of information that exists and how easily it can be accessed. A few of the more exotic cases of sabotage that have gotten significant media coverage are discussed below.

In November 1997, federal prosecutors in New York City charged a former employee of Forbes magazine for breaking into the company's computers, sabotaging the systems and causing an estimated

US$100,000 in damage. Although he denied the charges, George Parente allegedly broke into Forbes' computers after he was dismissed on April 21, 1997 and caused the systems to crash and important data to be lost.

Also in November 1997, Senal Arabaci, a Manhattan computer programmer, was charged with sabotaging a computer system at Art Assets LLC by deleting and modifying files after a billing dispute with the company.

6.7 Vandalism

Vandalism (or web defacing) is the process of rewriting someone's web page (illegally) and displaying a message of one's choice. The messages left can range from politically motivated to ego-inspired. Vandalism may seem trivial to some, but it can have devastating effects on a company's public relations. Such attacks can reduce the public's confidence in a firm and this may mean more business for the firm's competition. Also, one is never quite sure of the extent of the damage done by a vandal. Is the web page the only thing the vandal tampered with? There have been a few well-known cases of vandalism in recent times.

In September of 1996, the CIA web site was cracked by a group of Swedish hackers who were against a particular Swish court case against a group of youths arrested for computer security crimes. The hackers rewrote the page to read "Welcome to the Central Stupidity Agency." They also provided links to the PlayBoy web site, hacker web sites and also left a few broken links.

In March of 1997, NASA's web page was vandalized. The message left on the page made reference to the Internet Liberation Front (ILF) and promised to continue where the ILF left off.

6.8 Unauthorized Access

Armed with the knowledge of how to access computer systems and take control of resources, attackers can create terrible inconveniences

for the general public. Inconveniences that could lead to an increase in one's utility bill (phone and electricity bills are favorite targets) or even lead to the modification of one's personal data (many honest, hardworking persons may just overnight be recorded as hardened felons). These risks are very real. Identification theft is an all too common incident at the U.S. Department of Motor Vehicles (DMV). People can attain one's social security number, through illegal means, and essentially steal one's identity. Many such cases are currently being investigated. Less exotic events can also occur. Take for example, the 1997 incident at the U.S. Federal Trade Commission.

In November 1997, the U.S. Federal Trade Commission announced that more than 38,000 consumers would receive credits or refunds totaling $2.74 million. These credits or refunds applied to telephone charges they unknowingly incurred when their computer modems were allegedly hijacked and routed to expensive international numbers.

6.8 Final Thoughts

If you have doubts believing that extent to which computer crime has pervaded our society, examine a few of these statistics.

In a study done in 1998 by the Computer Security Institute, it was found that of the 540 specialists (from U.S. corporations, financial institutions, government agencies and universities) polled 64 % of them reported computer security breaches within the last 12 months. The study said "This represents a dramatic increase of 16 percent over the 1997 survey results." The survey further reveals that many victims of computer crime were unable to actually quantify their loss. However, 241 organizations tabulated their figures and this led to a total of $136 million. This is an increase of 36 percent since 1997. The other stalk reality is that most organizations refuse to report security breaches, because of the repercussions that they fear it will have on their business. Thus, this figure is an gross under-estimate[13]. "You're never going to know the total cost because you can't measure

[13] Johnson, Anna "Companies Losing Millions over Rising Computer Crime".

what you haven't discovered."[14], Mike Carlton, senior manager in Ernst & Young's Fraud Investigation Unit.

A similar study done by the accounting firm Deloitte & Touche on 300 Australian companies found that two in five had experienced compromise in 1997. Twenty per cent of the firms suffered 6 or more incidents that year.

In a conference in Ottawa, Canada early in 1997, well-known American security specialist Winn Schwartau estimated that the U.S. economy loses more than US$100 billion per annum through industrial espionage and that this has been growing at a rate of 500 percent per annum since 1992. Similarly, in the UK, the 1996 NCC Information Security Breaches Survey identified a 200% increase in computer crime from 1995 to 1996.

These are clear indicators that security is an issue that should be taken very seriously. The effects of a system compromise can range from mildly annoying to maliciously destructive, but the bottom line is that there will be an effect.

[14] Grayson, Ian "It takes a Digital Detective to track down today's computerised criminals", 1997.

Chapter 7
Current Security Solutions

"The best strategy in war is to win without fighting"
- Sun Tzu, *The Art of War.*

The current approach to Internet commerce security

It is believed that for commerce over the Internet to be considered viable and secure, there are four issues that must be addressed, namely: confidentiality, authentication, message integrity and non-repudiation. In addressing these issues, it is hoped that one will effectively render some of the attack methods discussed earlier useless. For those methods that are unaffected, various tools and methods have been developed to deal with them. There is also the issue of authorization, but this has little bearing on our discussion, thus only an introduction of it will be given.

Authorization versus Authentication

Authorization is proving that one has permission to use a particular facility. It allows people to anonymously do transactions. There is very vibrant debate ensuing concerning the use of authentication versus authorization. The popular approach has been to use authentication in situations where authorization may be more appropriate. For example, using authentication techniques to restrict access to a facility may not always be a wise decision. Though research in authorization techniques is vibrant, there is some skepticism surrounding how readily the players in the Internet commerce arena will embrace these techniques. The facts that the potential for abuse by criminal elements is great and that traceability will be difficult (probably even impossible) contribute towards the non-use of these techniques. Having discussed authorization, it is now time to examine the other issues.

Confidentiality

Confidentiality refers to "the ability to protect information from

disclosure"[15]. Many customers will not want information about themselves being liberally distributed by the people they do business with. Thus, sending sensitive information over insecure lines or having this information stored in locations that can be easily compromised is out of the question. Traditionally, confidentiality has been achieved by either using encryption or steganography. These techniques will ensure that sniffing attacks are futile.

Authentication

Authentication is proving one's identity. A properly implemented authentication technique will ensure that spoofing attacks are ineffective. A further discussion of authentication will be given in *section 7.3*.

Message Integrity

Message integrity refers to the fact that one must not assume that the information one receives is totally, or even partially, correct. Messages may be modified as they travel over the Internet. This implies that a mechanism has to be in place to check if the message has been corrupted. If one is buying a house over the Internet, then one wants to be sure that someone does not increase the amount you are should be paid. Ensuring message integrity means that message alteration attacks will not be successful.

Non-repudiation

The principle of non-repudiation is pretty simple. For any transaction, we require that none of the entities involved can in the future claim that the transaction never took place. This is a perfectly logical principle. It is just mirroring the commerce framework as it exists in the non-computing world. This is exactly what one hopes Internet commerce will do. The effects of not having non-repudiation are obvious: widespread fraud. Non-repudiation is not the solution to any particular attack technique. It simply ensures that a trust relationship can be established and maintained between the customer and merchant.

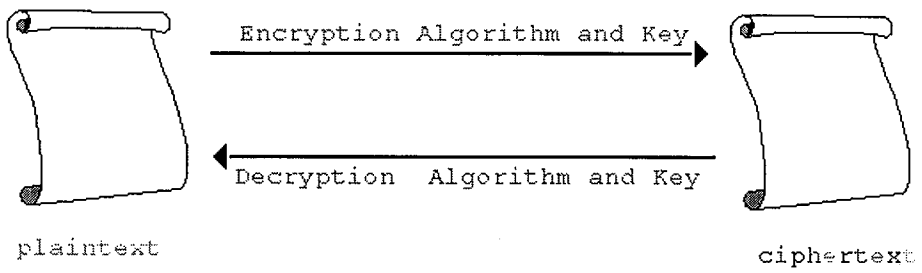
---

[15] Sdrs, Camillo "Encryption and Strong Authentication for Electronic Commerce", Abstract, Helsinki University of Technology.

Summary

The four issues discussed above ensure that sniffing, spoofing and message alteration attacks do not affect the Internet commerce transaction. However, there are several other attack methods that need to be addressed. For these methods, other schemes have been devised. We will now elaborate on some of the solutions being used to nullify the effects of these attack techniques.

7.1 Encryption



"Encryption is the conversion of data into some unintelligible form"
- Anonymous

General Description

The primary purpose of encryption as it relates to Internet commerce has been to make the data transmitted from one user to another user, incomprehensible to a third party attack. That is, no one should be able to listen in on other people's conversations. Formally, encryption is the application of an algorithm, together with what is called an encryption key, to scramble information in order to make it unreadable. The initial data is referred to as plaintext and the end-result (the scrambled data) is called ciphertext. The reverse of encryption is called decryption. This decryption process can only occur when the right decryption algorithm and key are used.

Types of encryption algorithms

Encryption comes in two flavors. There are private-key or symmetric encryption algorithms and there are public-key or asymmetric encryption algorithms. Let us now discuss each in detail.

7.1.1 Symmetric Encryption Algorithms

Overview

This set of algorithms embodies what is considered the traditional or the conventional view of encryption. In these algorithms both the sender and the receiver of the message must know and use a common secret key. The sender encrypts the message with this key and the receiver decrypts it using the same key. Symmetric algorithms can be categorized as either stream or block algorithm (or cipher). A stream cipher operates on a very small unit of data and a block cipher operates on fixed-sized blocks. For more information on these ciphers please read Camillo Sdrs's article on "Encryption and Strong Authentication for Electronic Commerce". Implementations of symmetric algorithms tend to take a short time to encrypt and decrypt data. Some of the more popular private-key encryption algorithms are DES, IDEA, PKZIP, Skipjack and Enigma. The fact that only one key is used leads to a few complications. We will identify and discuss these complications and take a deeper look into DES.

The security of key generation

When parties at different locations must trust an unsafe communication medium, the task of secret key generation becomes susceptible to compromise by an onlooker. If this happens then this onlooker can eavesdrop on the parties' conversation.

Key management in open systems

Also, in open systems with a large number of users, symmetric encryption systems have problems providing secure key management (where key management refers to the generation, transmission and storage of keys). We do not want a third party to easily steal a set of keys and use them at will.

7.1.1.1 DES - Data Encryption Standard

Overview

This is one of the oldest encryption algorithms. It was developed at IBM in the mid 1970s and later adopted by the US Department of Defense. DES is a block-cipher algorithm. It encrypts blocks of 64 bits of binary data at a time. DES uses a key, whose length is 64 bits (in

actuality, only 56 bits is used as what is classically considered a key and the other 8 bits are used for error checking). Since, the key length is 56 bits; there are about 72 quadrillion (2^{56}) possible values that the key can take. Let us now look on the operation of DES.

The operation of DES

Initially, one must choose a key and convert the plaintext into its binary form. The conversion is very simple to do. Since there are 26 letters in the English alphabet, each letter can be represented by a 5 bit binary number. For each 64-bit block, the leftmost bit is known as the first bit and the rightmost bit is known as the sixty-fourth bit. The DES algorithm specifies that one must change the order of the bits in each block. The order is changed according to a permutation table (that is, there is a table that states that the 23rd bit should be the 1st bit, the 64th bit should be the 2nd bit and so on).

The result of this permutation is split into two halves. The 1st thirty-two being called L_0 and the last thirty-two bits being called R_0. The data then undergoes a transformation sixteen times (this transformation is explained in detail in the Federal Information Processing Standards publication on DES). The purpose of this transformation is that to interweave the message and key. After each transformation each half is renamed. For example, after the first iteration, R_0 becomes R_1 and L_0 becomes L_1; after the second iteration R_1 becomes R_2 and L_1 becomes L_2 and so on. After the sixteenth iteration, one will have L_{16} and R_{16}. These strings are combined into one string by making R_{16} the 1st thirty-two bits and making L_{16} the last thirty-two bits. The 64-bit string is then entered into the inverse of the initial permutation function and the result is the ciphertext. Decryption is done by running this process backwards. Having seen how DES operates, let us speak about how secure DES is.

The security of DES

DES has remained secure for the past 20 years, in the face of phenomenal advances in encryption, mathematical and hardware technology. To date, the most effective way to attack DES is by brute-force (that is, to search the key space). DES's biggest vice is that its key length (56-bit) is becoming more insignificant to break as

the months go by. The US government has claimed that DES provides excellent security. However, on Wednesday, July 15, 1998, the Electronic Frontier Foundation (EFF) proved how unsafe DES is. In the RSA Laboratory's "DES Challenge II" contest, it took their machine (the EFF DES Cracker) less than 3 days to complete the challenge, shattering the previous record of 39 days set by a massive network of tens of thousands of computers. As this is being typed, research is being undertaken to formulate a new Advanced Encryption Standard (AES). This AES is being viewed as the replacement for DES.

7.1.2 Asymmetric Encryption Algorithms

General Description

In 1976, Whitfield Diffie and Martin Hellman first introduced the world to the concept of asymmetric encryption. It was an attempt to solve the key management problem faced in symmetric algorithms. In public-key encryption, each user has two keys, a private key and a public key. Both these keys are linked mathematically. The public key is freely distributed to everyone and the private key is kept secret. Let's assume that Jerry has a public key called A and a private key called B. Anyone wishing to send a secure message to Jerry can use the key A to encrypt the private message, but only Jerry can decrypt the message with the key B. Under this scheme, private keys are never transmitted, thus offering increased key security. However, public-key encryption suffers from a key management problem of a different kind. The problem is that a large enough repository for all the public keys of everyone that one may interact with cannot be constructed. However, in recent times there have been creative solutions to this problem; making this less of an issue.

The vulnerability of asymmetric algorithms

Asymmetric algorithms are often used to produce digital signatures for authentication and checks for message integrity. We will look at these later. These algorithms are founded on mathematical problems that are considered "hard". Since these problems are considered "hard" and no one has been able to conclusively prove that they are "hard" or not, asymmetric algorithms are sensitive to breakthroughs in mathematics. By far, the most popular public-key encryption

algorithm is RSA.

7.1.2.1 RSA – Rivest, Shamir and Adleman

Introduction

RSA was developed in 1977 at Massachusetts Institute of Technology by Ron Rivest, Adi Shamir and Leonard Adleman. It is a public-key algorithm that can be used for both encryption and authentication. The essence of this algorithm is modular arithmetic. As the algorithm is explained, the term "modular arithmetic" will be explained.

The operation of RSA

The algorithm works as stated:

- Two large prime numbers are chosen, let us call them p and q.
- Their product is found, let us call that $n = pq$; this n is called the modulus.
- Another number, e, is chosen. The number e is less than n and relatively prime to $(p-1)(q-1)$ [which means that the numbers e and $(p-1)(q-1)$ have no common factors except 1].
- Another number d is found such that $(ed-1)$ is divisible by $(p-1)(q-1)$.

The values of e and d are the public and private exponents, respectively. The public key is the pair (n,e) and the private key is the pair (n,d). It is of utmost importance that the factors p and q be kept secret.

The security of RSA

It is assumed that it would be difficult to get the private key from the public key (n,e). However, if one can factor n into p and q, then one can get the private key (this is known as a "factoring" attack). Therefore, RSA's security hinges on the assumption that factoring is difficult. Though several methods have been theoretically posed to break RSA, the only one that has been achieved any notable success is a brute-force factoring attack.

7.1.3 Thoughts

General Problems

For all encryption algorithms, the key length and the strength of the key management protocols (as well as the strength of implementations

of these protocols) are critical to the strength of an algorithm. Current advances in hardware technology have dictated that key length increase every few months to ensure that algorithms cannot be easily broken. Weak implementations of key management protocols can allow for what are normally considered to be strong cryptosystems to be cracked very easily. Thus, two important factors in the security of any algorithm are key length and key management protocols. Let us now, compare private-key and public-key encryption.

Symmetric versus Asymmetric

Symmetric algorithms have to address a key generation problem, but they tend to be a lot faster than asymmetric algorithms. On the other hand, asymmetric algorithms offer increased key security, as private keys never have to be transmitted or revealed. Asymmetric algorithms can also provide a way to provide authentication. Using asymmetric algorithms, one would require either the sharing of some secret or the inclusion of some third party in order to provide authentication. The pros and cons of each class of algorithms normally lead us to use a hybrid algorithm, which incorporates the benefits of each. Normally, an asymmetric algorithm is used to securely transmit a symmetric key to the correct recipient and to provide authentication and integrity and the much faster symmetric algorithm is used to encrypt/decrypt the actual message.

The use of encryption

Unfortunately, encryption is only widely used to protect data, which is in transit, from third party attack. Many people looking at securing Internet commerce tend to focus primarily on developing stronger encryption. It is my belief that encryption is not the silver bullet solution for Internet commerce security and that such a perception should be discouraged. The security of data while not in transit (that is, while it is on the client or server machine) must also be considered and be given widespread attention.

[redacted]

Background

Currently, in the United States, there is a big debate brewing over

export laws on encryption. The government claims that for law enforcement to effectively do their job, they should be provided with the keys for any U.S. product that contains cryptographic code. They have also sought to allow only inferior cryptographic products to leave the country. Aside from hinting of unethical behavior, this is counter-productive and anti-"Internet commerce growth". In an effort to make this debate inconsequential, Ron Rivest, in March of 1998, proposed a technique that allows one to transmit data securely without encrypting the data first. This technique is called "chaffing and winnowing". This is still an idea in development, and as such there are some adjustments to be made to make this technique more efficient. However, it holds a great deal of promise and one can possibly foresee this technique being a viable alternative to encryption. Let us examine the entire concept of "Chafing and Winnowing".

The concept of "Chaffing and Winnowing"

The principle is deceptively simple. One simply adds chaff (useless data) to a particular message at the sending end and at the receiving end one will winnow (eliminate the useless data from) the data received. The technique does not use an encryption key, thus the argument for the rendering of keys to the government no longer holds water. As proposed in Rivest's paper, *"Chaffing and Winnowing: Confidentiality without Encryption"*, the process should proceed as follows:

- The sender and receiver agree on a secret authentication key (SAK). *This SAK is used to authenticate the origin and contents of the packets sent between them.*
- The sender breaks the message into packets, each with its own sequence number, and authenticates it using this secret authentication key. *This break-up of the message is already done at the TCP layer.*
- The sender appends to each packet a message authentication code (MAC). *This is computed from the packet contents, its sequence number and the SAK.*
- Then chaff is added to this message. *That is, we introduce counterfeit packets into the stream. These packets have the correct overall format, have reasonable serial numbers and reasonable message contents, but have MACs that are not valid. The chaff*

packets may be randomly intermingled with the good (wheat) packets to form the transmitted packet sequence.

- The receiver, armed with knowledge of the SAK, recomputes the MAC and compares it to the received MAC and all the packets that fail this test are considered to be chaff.

Let us take a look at an example that shows how this procedure might work. Given the following sequence:

(1,"Bob Smith, my assistant programmer, can always be found", 121)
(2,"wasting company time talking to colleagues. Bob never",34332)
(3,"finishes given assignments on time. Often Bob takes extended",3)
(4,"breaks. Bob is a dedicated individual who has absolutely no",34)
(5,"knowledge in his field. I firmly believe that Bob can be",1111)
(6,"dispensed with. Consequently, I duly recommend that Bob be",2)
(7,"executed as soon as possible.", 76989)

The order in each 3-tuple is sequence number, message and MAC. If the chaffing algorithm is sophisticated, one may produce the following stream:

(1, "Bob Smith, my assistant programmer, can always be found",121)
(1, "hard at work in his cubicle. Bob works independently, without",8)
(2,"wasting company time talking to colleagues. Bob never",34332)
(2,"thinks twice about assisting fellow employees, and he always", 42)
(3,"finishes given assignments on time. Often Bob takes extended",3)
(3,"measures to complete his work, sometimes skipping coffee", 656)
(4,"breaks. Bob is a dedicated individual who has absolutely no",34)
(4,"vanity in spite of his high accomplishments and profound", 5609)
(5,"knowledge in his field. I firmly believe that Bob can be", 1111)
(5,"classed as a high-caliber employee, the type which cannot be", 33)
(6,"dispensed with. Consequently, I duly recommend that Bob be",2)
(7,"promoted to executive management, and a proposal will be", 8901)
(7,"executed as soon as possible.", 76989)

At the receiving end, all the 3-tuples with an invalid MAC will be discarded. The stream that is sent along the Internet totally hides the

intent of the message and anyone who attempts to look at it while it is in transit will be viewing "rubbish". However, it is currently unrealistic for chaffing algorithms to produce output streams that always coherent.

The effectiveness of "Chaffing and Winnowing"

This technique has the potential to be just as secure as encryption. However, there are a few areas that need to be addressed before this can happen. The effectiveness of this technique will depend on the MAC algorithm, on how the original message is broken into packets, and on how the chaffing is done. For more information on these issues, please go to http://theory.lcs.mit.edu/~rivest/chaffing.txt

7.3 Authentication

Definition

In essence, authentication is the process of identifying oneself. The process is simple. An individual asserts an identity and then is asked to verify this assertion by providing some other piece of evidence. The combination of these assertion and verification processes is what is call authentication.

Overview

In the physical world, use all our senses to identify someone, but modern-day computers do not have these natural cues available to it. Thus, one is faced with the task of differentiating between people and verifying someone's honesty when they assert an identity. Miller purports that there are three factors that can be used in the authentication process:

- Knowledge - something that an individual knows.
- Possession - something that an individual owns.
- Characteristic - something that an individual is.

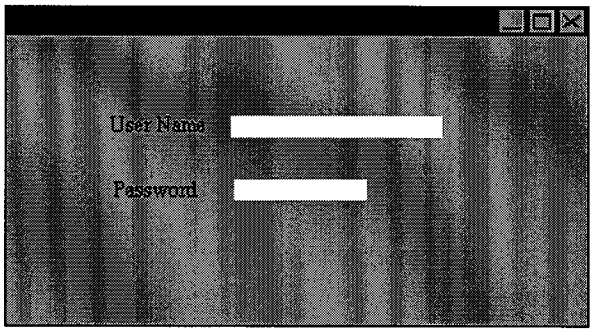
Many of us are used to the basic authentication scheme of entering a user name and password. However, this is not a particularly strong technique, as anyone who steals your user name and password can impersonate you. The more popular authentication techniques make use of the knowledge and possession factors. Both these factors have

lead to id-password systems, digital certificates and digital signatures (Currently, both digital signatures and digital certificates are implemented using public-key encryption). The study of the characteristic property has spawned the field of biometrics. Let us now look at the four basic authentication methods currently being used, namely: basic authentication, digital signatures, digital certificates and biometric authentication.

7.3.1 Basic Authentication

General Description

Basic authentication techniques rely on two-factor authentication (that is, they require two different sources of identification). They require us to state our identity and give proof of this identity. For example, there are many systems that ask us for a user name and a password; some of us are also familiar with the ATM card and PIN (Personal Identification Number) situation.



Incorporating security into basic authentication

Because one does not want someone to steal one's authentication information while it is in transit, one generally tries not to send it over unsafe communication lines. In this case, making the data unintelligible (currently realized through encryption) solves the problem. However, if this information is frequently sent over these lines, the probability that the information will be sniffed increases. A possible solution for this problem is to have the client send his authentication information only once (at the beginning of the transaction), and then send the client either a security token, ticket or

cookie. These devices are files, sent to the client, that allow the server to uniquely identify each client. The client will not need to send his authentication information once the token, ticket or cookie has been accepted. However, there are risks associated with each of these devices. Primarily, each may be the carrier of some malicious code.

7.3.2 Digital Signatures

Definition

A digital signature is a sequence of numbers, which is computed as a function of a message and a user's private key. Digital signatures and hand-written signatures are similar, but whereas hand-written signatures are constant, regardless of the document being signed, a digital signature varies with the data. For example, three different messages signed by the same user will generate three different signatures.

The message digest

Because public-key encryption is slow, one often signs a condensed version of a message, called a message digest. Message digests are generated by hash functions. "A hash function is a keyless transformation function that, given a variable-sized message as input, produces a fixed-sized representation of the message as output (i.e., the message digest)"[16]. For example, a hash function may condense a message of any length into a 128 or 160-bit digest.

The hash function

For a hash function to be considered secure it must possess two properties. It should be *1-way* and *collisionless*. *1-way* means that given a digest and the hash function, it is not computationally feasible to find the message that produced the digest. *Collisionless* means that it is impossible to find two messages that hash to the same digest. A secure hash function guarantees that signing a message digest provides the same security services as signing the message itself.

The process of creating a digital signature

The digital signature generation/verification process is as follows.

- Initially, one assumes that the two users (let us call them *User A*

[16] http://www-08.nist.gov/nistpubs/800-7/node211.html

and *User B*) have agreed upon a hash function and a signature algorithm for the signature verification process. For our purposes, we will also assume that *User A* needs to send a signed message to *User B*.

- *User A* types the message, and then generates a digest for it.
- The digital signature is computed as a function of the digest and *User A*'s private key.
- Then the message, along with the signature, is sent to *User B*.
- On receiving the message, *User B* generates a digest using the received message.
- He then uses this digest, *User A*'s public key, and the received signature as input to a signature verification process.
- If the verification of the signature is successful, then *User B* is guaranteed that the message was not modified. *If for any reason, even one bit of the original message was changed, the digest generated using the received message would cause the signature verification process to fail.*

The process not only guarantees *User B* that the message was not modified, but it also guarantees that User A sent the message.

The other uses of a digital signature

A digital signature cannot only provide integrity and authentication, but it can also provide 'non-repudiation with proof of origin'. The concept of 'non-repudiation with proof of origin' is similar to authentication, but stronger in that the proof can be provided to a third party. To provide 'non-repudiation with proof of origin' using a digital signature, the sender signs a message (or digest) using his private key. Since only the sender can access the private key, the signature is unforgeable evidence that the sender generated the message. In contrast, 'non-repudiation with proof of origin' cannot inherently be provided in a conventional cryptosystem. Since, both parties involved in a communication share a secret key, both parties can deny sending a message, claiming that the other party is the sender.

Some of the more popular digital signature algorithms are DSA (Digital Signature Algorithm) and RSA (Rivest, Shamir, Adleman).

For more information on these algorithms, please read the Federal Information Processing Standards Publication 186 - DIGITAL SIGNATURE STANDARD (DSS), May 19, 1994.

7.3.3 Digital Certificates

General description

A digital certificate is a block of data that contains the owner's public key and the digital signature of a certificate authority (CA). The purpose of this certificate is to associate an authentic identity with a given key pair. The need for such a mechanism is obvious. Since every user on the Internet has a public key/private key pair and each key pair has no inherent association with any identity, a way had to be developed to eliminate the possibility of fraud. The absence of such a system would mean that it would be difficult for people to differentiate legitimate business entities from crooks.

The creation of a digital certificate

An individual generates his key pair and asks that it be certified by a trusted certification authority. The CA checks the user's details and if they are indeed true, the CA uses its private key to sign an electronic document that binds the identity of an individual (or organization) to the individual's public key (this document is known as the certificate). Users can check the authenticity of another user's public-key by verifying the CA's signature on the certificate using the CA's public key. A digital certificate is similar to a passport. It is intended to prevent individuals from generating key pairs and falsely claiming to be someone else.

7.3.4 Biometric Authentication

General Description

Biometrics is the study of the measurement of physiological and behavioral traits. Biometric authentication is the identification of humans based on these traits. Physiological traits include one's fingerprints, the size and shape of one's hands and fingers and one's retinal patterns. Behavioral traits include one's signature, one's voice, one's speech patterns or even the speed or pattern of one's typing. Physiological measurements tend to be more inconvenient for

customers than behavioral measurements, but behavioral measurements are subject to a greater degree of variability. For example, customers would not be very accommodating if every time you wanted to do a credit card transaction, you had to do a DNA test (physiological measurement). However, although one's signature is expected to change over time, a customer will view the provision of a signature as less obtrusive than the DNA test. Having gotten an idea of what biometrics is all about, let us turn our attention how biometric systems can be examined.

Examining biometric authentication

Biometric authentication methods may fail in two ways: they may mistakenly confirm an identity when it is the wrong person (a false acceptance, this is normally expressed by a number that represents the false acceptance rate) or they may mistakenly reject the identity of the right person (a false rejection, this is normally represented by a number that indicates the false rejection rate). Each biometric method has a particular value for each of these two rates. The biometric technique selected must suit the needs of the firm seeking to employ it and the needs of the area in which it is to be used. In choosing a biometric technique, the cost of the technique, the reliability of the technique, the speed of the technique and the satisfaction and convenience of the customer will need to be given serious consideration. A biometric technique will only be commercially viable if it is inexpensive, reliable, quick, convenient and customer-centric.

The requirements of biometric systems for Internet commerce

Biometric techniques for Internet commerce will need to possess low false rejection rates. This is due primarily to the fact that a firm cannot afford to incorrectly reject the identity of a customer. However, the implementation of these techniques will require that firms will need to provide the supporting infrastructure, and that the consumers will be required to put in place the necessary hardware and software components.

The current situation

Currently, most biometric methods are too expensive to be used in

Internet commerce, but in recent times companies have been researching and developing biometric authentication systems, particularly systems that detect retinal patterns.

7.3.5 Thoughts

Problems

The main problem with authentication techniques is the lack of standardization. The diversity of the techniques will not bode well for the future of Internet commerce. Having a diverse pool of (possibly proprietary) authentication techniques will mean that systems may not be able to communicate with each other. For the future of Internet commerce to look even remotely attractive, this diversity cannot be allowed to happen.

The future of authentication

Current authentication methods exist to ensure that a particular user can be identified. Though authentication has been primarily focused on verifying identities, I believe that techniques can be developed to verify that a legitimate entity cannot execute malicious code on a user's machine. Thus, taking authentication to the next level. Allowing authentication to not only embrace verification, non-repudiation and message integrity, but also message intent.

7.4 Protocols

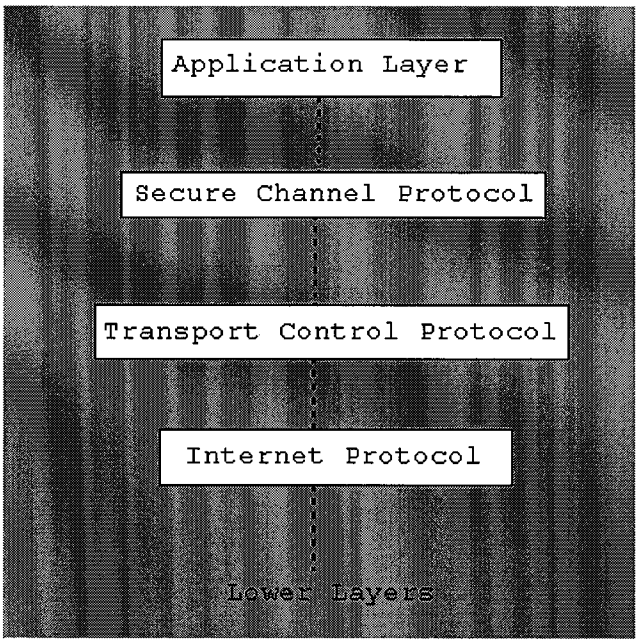
Definition

A protocol is a formal description of the messages to be exchanged and rules to be followed in order for two or more systems to exchange information. Protocols are standard languages which entities can use to communicate. They also offer a standard way to ensure that secure properties are included in applications. As stated previously, the Internet is based on a protocol called TCP/IP; however TCP/IP provides no services for securing a data transmission. Thus, it becomes necessary to create protocols that ensure that our Internet transactions are secure. There are two strains of protocols that relate to Internet commerce, namely: secure channel protocols and secure payment protocols.

7.4.1 Secure Channel Protocols

General Description

A secure channel protocol simply ensures that a message transmitted from an originator (a source) to a destination (a sink) is safe from third party attack. In effect, a secure channel protocol is used to secure Web sessions. Given, the fact that the Internet was designed for open communication between heterogeneous platforms, security was not defined as a part of the TCP/IP protocol. A secure channel layer is normally placed between the TCP layer and the application layer and it ensures that all communications are secure. Currently, secure communications is guaranteed by encrypting data at this secure channel layer and sending the data, in its encrypted form, to the lower layers.



The generality of secure channel protocols

As good an idea as secure channel protocols are, their focus on securing any Web-based session does not specifically address the transferring of payment between client and merchant and this is viewed as a big drawback. Examples of secure channel protocols are: SSL and S-HTTP. Let us take a closer look at these two channel protocols.

7.4.1.1 SSL - Secure Session Layer

Background

SSL was developed by Netscape Communications in 1994. And as Netscape browsers are currently the most popular, SSL can be considered the most dominant secure channel protocol around. We are already familiar with the concepts behind the operation of SSL (we discussed in the section on encryption).

Communicating using SSL

SSL provides facilities to do two things: authenticate the Web server and/or client and encrypt the communication channel. SSL specifies that initially the Web server and client must agree on the parameters of the secure session. This is done by a series of handshakes (that is, special messages between the server and client) After the secure connection has been established, public-key encryption is used to authenticate the server and/or client and to exchange the private session key between the Web server and client. A private session key is generated and this is used to encrypt the message. Then, the private key and the encrypted message are encrypted with the user's public key. This packet is then sent. The security of SSL depends on the strength of its components, that is, the strength of the public-key and private-key encryption used. There have actually been a few cases of people breaking SSL.

7.4.1.2 S-HTTP - Secure HyperText Transfer Protocol

Background

S-HTTP was developed by Enterprise Integration Technologies and commercialized by Terisa Systems. It is essentially an extension of the HTTP protocol. Unlike SSL, S-HTTP runs at the application layer, rather than below it. S-HTTP supports a number of different secure technologies, ranging from symmetric encryption to message digests.

Communicating using S-HTTP

During the initialization segment of the connection process between the client and server, the secure properties for a session are negotiated. This is done using the security negotiation header that S-HTTP defines

for packets sent during the Web session. These headers can contain information on "the type of secure technologies to use, the algorithms that the party will support, the direction in which the party desires the property to be enforced and the mode in which the property is requested (required, optional, refuse)."[17] After the session's secure properties have been agreed upon, the session is secured by encapsulating the data in a secure envelope.

"The secure envelope supports confidentiality of Web session contents, message integrity and authentication of clients and servers"[18]. Let us look at the components of this secure envelope. To ensure that all these above-named features are supported, the sender generates a (random) secret key (using symmetric encryption) and uses it to encrypt the message, he then encrypts the secret key with the recipient's public key and then generates a digest from the message and encrypts it with his private key. Thus, our secure envelope contains the encrypted message, the encrypted secret key and the encrypted digest.

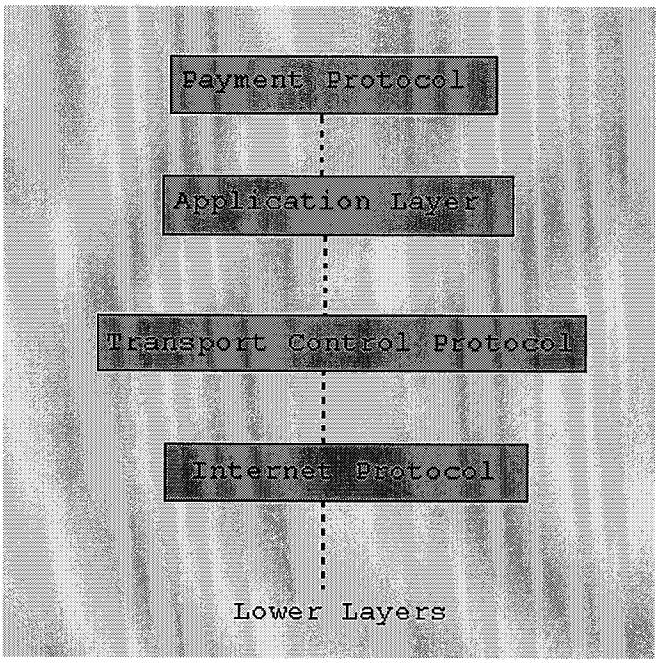
7.4.2 Secure Payment Protocols

General Description

A secure payment protocol is one that not only ensures that the data in a transmission is safe from interceptors, but also defines a set of rules that dictate how a payment transaction is to take place. The key difference between secure channel protocols and secure payment protocols is that secure payment protocols provide a method to assure merchants of payment while providing consumers assurance of information (especially credit card number) confidentiality. Secure payment protocols are placed in a layer above the application layer. Thus, theoretically, they can use all the functions below them; this includes the functions in the secure channel layer. To the consumer, secure payment protocols may be preferred to secure channel protocols. For example, using a secure channel protocol (like SSL) ensures that information gets safely to the merchant, but makes no guarantee on the security of the information once it is on the merchant's machine.

---

[17]Ghosh, Anup K "E-Commerce Security: Weak Links, Best Defenses", 1998, 121.
[18]Ghosh, Anup K "E-Commerce Security: Weak Links, Best Defenses", 1998, 121.



Problems being faced by secure payment protocols

Since, the realization that this type of protocol was important to Internet commerce, several companies have put forward their proposals for payment protocols. A few problems have arisen from this effort. Firstly, most of the solutions tend to be proprietary, which will not work in the Internet environment. Secondly, some of the emerging protocols are pure software solutions whereas some payment systems may consist of secure hardware components (e.g. smart cards). And finally, some protocols work with specific payment instruments (for example, credit cards) and cannot be extended to other models (e.g. checks). The solutions to these problems are still the focus of research projects around the world.

The types of payment protocols

A taxonomy of payment protocols has evolved. These protocols are either designed for stored-account systems or designed for stored-value systems.

7.4.2.1 Protocols for stored-account systems

Overview

Stored-account systems are "modeled after existing electronic payment systems such as credit and debit card transactions in which a corresponding account of credit (or deposit) is maintained"[19]. These systems represent new ways of using conventional banking services to transfer funds over the Internet. The actual money involved in a transaction never leaves the bank; it is accounted for at some time in the future. Stored-account systems are characterized by a high level of accountability and a high degree of traceability. With stored-account systems, it is possible for the client's purchase history to be compiled and a personal profile of the client's spending habits to be established. Stored-account systems also require that on-line verification be done. This implies that the cost of the transaction will increase and a delay in the approval of the transaction may be introduced. Examples of protocols for stored-account systems are: CyberCash's Secure Internet Payment System, Secure Electronic Transaction (SET), First Virtual's Internet Payment System. Let us talk about SET.

7.4.2.1.1 SET – Secure Electronic Transaction

Background

SET is a recent standard for secure payments over the Internet being developed by MasterCard, Visa, GTE, IBM, Microsoft, Netscape, SAIC, Terisa Systems and Verisign. The SET specification is very lengthy, thus I will be giving an overview of its operation. For flexibility, SET has no specifications for the following: the shopping or ordering process for Internet goods, the payment method selected and the platform or secure properties necessary for securing SET client and server machines. However, under the SET specification confidentiality, data integrity, client authentication and merchant authentication must be provided.

Communication using SET

A typical SET transaction has the following steps:

- The consumer sends a request for a transaction to the merchant.
- The merchant acknowledges the request.
- The consumer digitally signs a message digest of the order and encrypts the credit card number.
- The merchant sends the purchase amount to be approved and the

[19]Ghosh, Anup K "E-Commerce Security: Weak Links, Best Defenses", 1998, 97.

credit card number to the merchant's bank.

- The approval or denial is sent back to the merchant.
- The merchant confirms the purchase with the customer.
- The consumer sends a status inquiry to the merchant.
- The merchant responds to the purchase status inquiry.
- The merchant requests payment to the bank.
- The back sends confirmation of payment.

The information on the operation of the SET protocol was taken from the pages of "E-Commerce Security" by Anup K. Ghosh. Given the tremendous corporate support that SET has been getting, it is likely that it may become the de facto industry standard.

7.4.2.1.2 iKP – Internet Keyed Payment Protocols

Background

iKP is an architecture that was developed at IBM's T.J. Watson Research Center and Zurich Research Laboratory. It is designed to be used in secure payments involving three or more parties. It is also purported to be a general protocol, but in my estimation it seems more of a stored-account system protocol. The first implementation of iKP was for a credit-card payment system.

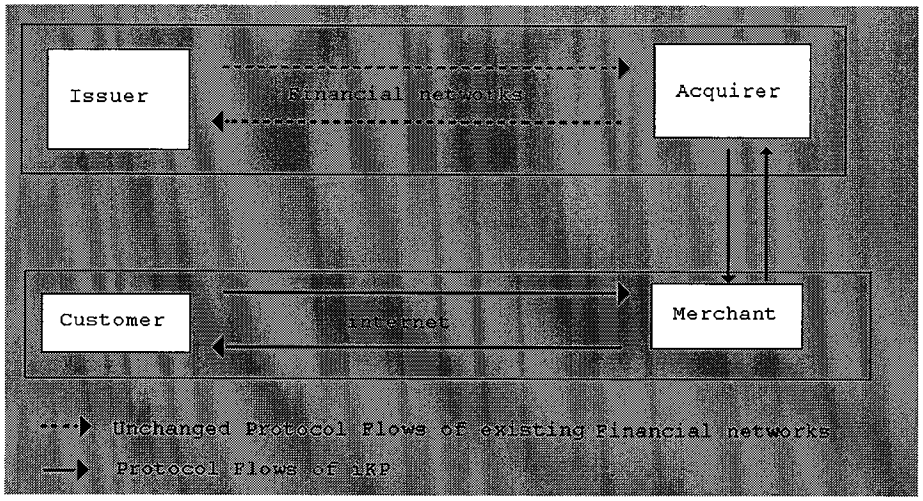
Communicating using iKP

A typical iKP transaction involves six flows:

- INITIATE - in which the customer sends to the merchant to begin the transaction
- INVOICE - the merchant's response to the client's request. This response may optionally contain the seller's signature of the transaction data.
- PAYMENT - the customer's response, which contains a payment "slip" including the buyer's account number and possibly a PIN, encrypted with the acquirer's public key, and optionally the customer's signature on the transaction data
- AUTH-REQUEST - which the merchant sends to the acquirer, containing the encrypted payment slip.
- AUTH-RESPONSE - the acquirer's response to the merchant, containing the acquirer's signature on the transaction data.
- CONFIRM - a confirmation from the merchant to the customer

that the transaction has been authorized.

This protocol is claimed to allow customers to order goods, services, or information over the Internet, while relying on existing financial networks to implement the necessary payments. This is also demonstrated in the figure below.



7.4.2.2 Protocols for stored-value systems

General description

Stored-value systems are an attempt to replace cash with its electronic equivalent, e-cash. However, stored-value systems extend the concept of cash as we know it. Cash is normally exchanged between two persons that are in close proximity to each other. E-cash can be exchanged between any two individuals, irrespective of the individuals' locations. With cash, there is a significant delay in the transfer of large sums of money, e-cash reduces this problem. However, whereas the flow of cash can be traced (due to their serial numbers), such traceability controls need not be present in e-cash systems.

The benefits of e-cash systems

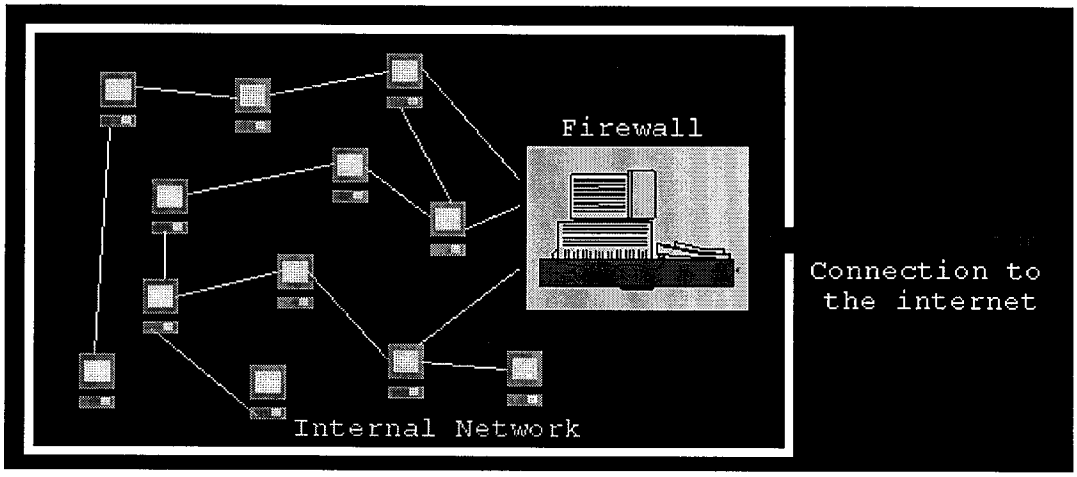
E-cash is a bearer certificate, which is normally stored on some device (for example, personal computer or smart card) and is transferred between the parties, just as cash changes hands. With e-cash systems the transfer of value is instantaneous. This is in stark contrast to

stored-account systems, which require bank approval and the debiting and crediting of the necessary accounts. E-cash systems also offer a certain level of anonymity. The absence of control and auditing implies that counterfeiting could go undetected. This security concern is the primary reason why stored-value systems are currently used for small-value transactions. However, a balance has been struck between anonymity and traceability, stored-value systems are expected to take off. Examples of stored-value systems are: DigiCash's e-cash, CyberCash's CyberCoin, Mondex, CAFE and VisaCash.

7.5 Firewalls

General description

A firewall is essentially a combination of hardware and software that provide an entry point to a trusted network for users of an untrusted network, which is in this case the Internet. Firewalls are a means of specifying what traffic leaves and enters a firm's network. They can also be used to specify that certain sections of a private network can be restricted to particular employees.



The composition of a firewall

Most firewalls consist of a screening router and an application gateway (also referred to as a bastion host). The screening router, as the name suggests, can be configured to block packets based on particular characteristics, for example, blocking all packets from a

particular IP address or a particular TCP port etc. The bastion host is used to provide an extra layer of protection to certain network applications. Firewall technology is still young and the search for the perfect firewall is still being fervently being undertaken.

7.6 Intrusion Detection Systems

An Intrusion Detection System (IDS) is based on the principle that an attack will be noticeably different from normal activity. Intruders to a system are very likely to exhibit a different pattern of behavior from that of a legitimate user. The IDS is expected to detect these abnormal patterns by analyzing numerous sources of information that are provided by the existing system. If the IDS identifies a possible attack, then a program is called that recommends a course of action that should be taken against the attacker. For more information on IDS, please read "Intrusion Detection Systems (IDS): A survey of existing systems and a proposed distributed IDS architecture" by Steven R. Snapp and others.

7.7 Automated Auditing Systems

Most systems provide one with the ability to keep an audit trail on the system's users. That is, records can be kept of the activity of individual users. Automated Auditing Systems use these audit trail records and present *information of interest* in an user-friendly form. There are two problems that these systems must address. The first problem is to decide what kind of activity should be audited (i.e. what would one consider *information of interest* or auditable events). The second problem is what to do with the vast amount of information that is present in the audit trails. Both of these issues are being researched as this is being written. It should also be noted that audit trails were originally designed for accounting not for security, thus many of the desirable auditable events may be unavailable. Automated Auditing Systems allow us to identify what may be considered as suspicious activity.

7.8 Host and Network Audit IDS Systems

Each program has what are called date and time stamps. These stamps may refer to the last date the program was modified or even the date the program was created. Malicious programs, such as viruses, will change these stamps as they move from program to program. Time and Date Stamp Analysis Systems are systems that have a record of original date and time stamps for the files and uses this data to check to see if any files have been tampered with.

7.9 Virus Detection and Elimination Systems

These programs have become very popular in today's world. They remain resident on one's machine and warn users if any of the viruses known to it trying to execute on the computer system. One of the popular virus detection programs is Vsafe.

7.10 Quarantine Systems

This is a system on which new software is placed for a period of time. This new software is checked for viruses and for proper software behavior. This system can be used to detect anything from spoofed programs to Trojan horses. The size and checksum information for a particular software package (this can be usually obtained from a software vendor) and anti-virus programs are normally used to help with the checking and validation.

7.11 Backup and Recovery Systems

These systems ensure that the least amount of effort is needed to restore a system that has been brought down by an attacker. A backup and recovery system periodically takes a snapshot of one's system and dumps this information to an appropriate medium. These systems do not try to prevent an attack from taking place, they merely offer an easy way to recover from attacks.

7.12 Hardware Solutions

Certain hardware equipment has been designed to be tamper-proof. Tamper-proof simply means that no one can in any way alter the

equipment so that it can be used in a devious manner. A prime example of tampering with a device can be shown with an e-cash card. Imagine getting an e-cash card and using it until the balance is near to zero. Then using some electromagnetic device to reset the counter to the initial balance on the card. The cost of implementing a hardware attack is decreasing every year. Various device manufacturers, for example smart-card producers, have been claimed that their devices are tamper-proof.

7.1.3 Final Thoughts

The solutions discussed in this chapter represent only a small fraction of the solutions being used today. However, the set that we have examined should give us a clear indication of the kind of things that people have used to increase the security of their system.

Chapter 8
Required Framework for Secure Internet commerce

Background

The 'required framework for secure Internet commerce' should be an environment that allows both customers and merchants to feel secure when they are doing business on-line. I have tried to make this proposed framework realistically attainable and easy to implement. Continuing on the theme of a holistic approach to Internet security, this framework seeks to tackle the entire spectrum of problems that contribute to Internet commerce not achieving its maximum potential. The popular approach has been to think that there is a technological solution for every problem we face. This technology-driven approach cannot be used to solve all of our problems, as not all our problems have technological foundations. It should also be noted that technology cannot be used to address issues of trust and moral responsibility. With this realization, we have to look at the entire picture and see how best each problem can be solved, whether by technological means or otherwise. However, before delving into the details of this proposed framework, there are a few issues that should be addressed.

Who owns the Internet?

Though the Internet can be viewed as a truly collaborative effort, it cannot be disputed that the United States government has played a pivotal part in its development. This affords them a privileged position. Theoretically, they can dictate the direction and scope of the Internet of the present and future. Given that the Internet has evolved into a global entity, that cannot be classified as the property of any one individual, firm or government, it would be imprudent, at best, to think that any particular government can lay claim to it. Thus, the Internet belongs to everyone and as such its development is everyone's responsibility. Having established that all the users of the Internet will direct its future, one now has to focus on answering the question "How

much government intervention is necessary in the on-line commercial world?" In this context, government is used generically to refer to any arbitrary government of the world.

How much government intervention is necessary?

In my opinion, the answer is obvious. The success of the Internet has been made possible through government and private funding and through government's non-regulation of the Internet's operation, development and growth. Therefore, governments should be the providers of a user-friendly infrastructure and leave the intervention to a bare minimum. The development of Internet commerce should be spearheaded by the private sector and be under the supervision of non-governmental organizations like the Internet Society (ISOC). After the governments have laid the foundation, the question now becomes "How do we protect ourselves from attack?"

Protecting ourselves

Attack techniques were designed to take advantage of several well-known facts. They exploit software design and implementation flaws, make use of software configuration mistakes, exploit shortcomings in hardware design and implementation and uncover ways to maliciously use software and hardware products. These techniques are fueled by deficiencies in protocol specifications and implementations and remain viable due to public ignorance and lethargy and corporate cowardice. Apart from destroying the fabric that binds together a merchant and a client (i.e. trust), these methods may have damaging effects on a consumer's psyche and on a merchant's finances. The framework that I am suggesting seeks to minimize these effects and address all the above-mentioned areas in an effort to make Internet commerce security comparable to that of traditional commerce. The framework consists of three general components, namely: the technology component, the legislative component and the social component.

5.1 Technology component

The importance of this component

This component is by far the most important and the most susceptible to change. Its importance lies in the fact that it directly addresses

attack implementations. That is, if a program is written to flood a server with malformed packets, then this component should include a solution for this problem. This component's susceptibility to change is due to the dynamic nature of the computer industry. Thus, what is an effective attack today may be useless tomorrow or what is an effective solution today may be ineffective tomorrow.

The constituents of this component

This component explicitly addresses the four essential constituents of an Internet commerce transaction (these were discussed in Chapter 4), while the other components of this framework address Internet commerce security from a slightly different perspective, a perspective that is more subtle. For those who have forgotten, the four essential constituents of an Internet commerce transaction are the client-side software, the transaction protocols, the commerce server and the operating system software. The technology component consists of three sub-components: the hardware sub-component, the software sub-component and the standards sub-component.

8.1.1 Hardware

The alternatives

Hardware in this context refers specifically to computers (such as computers used by the clients and merchants), network equipment (such as routers and gateways) and Internet commerce devices (such as smart cards). There are a series of attacks that are geared towards hardware devices. For each hardware device, it is important for us to ascertain how it can be attacked. This will require the help of both the private and public sector. Realizing the faults of a particular piece of hardware is the first step. The next step is formulating ways to handle these faults. In my opinion, hardware should either be designed so that they are tamper-resistant and impervious to the misuse of their properties or protective devices should designed to strengthen the security of currently insecure hardware devices.

Alternative One

With the former alternative, this tamper-resistance and imperviousness to the misuse of the hardware's properties would have to be an integral part of the hardware engineering process. The inclusion of this

constraint on the hardware design process may add some cost, time and complexity to hardware design. The design process would require that a hardware device is rigorously tested and all flaws eliminated before it is considered for mass manufacture. This could be achieved by making intelligent CAD systems a vital part of the design process. The problem now lies in developing CAD systems that are sophisticated enough to be used in this manner and in convincing management that migrating to this new design methodology will be beneficial to them. It is my opinion that management will pose the bigger problem. Hopefully, with the current trend in management ranks to have an appreciation for the long-term benefits of technology, this concern may be less grave than I imagine. Another fact that may hinder this approach from being universally adopted is that there are millions of machines already in use that were not manufactured using this strategy. This implies that we are either asking people to change their old hardware. Although, this may not seriously affect people who have leased machines, it will not be an easy venture for most people. This fact may be the major contributor to the non-acceptance of this approach to hardware design.

Alternative Two

With the latter alternative, production of hardware systems would proceed as it does today. However, it would be the responsibility of the manufacturer to see to it that additional equipment, whether hardware or software, is developed to deal with the hardware properties that can be exploited. I feel that this alternative will appeal to a wider cross-section of people, because it does not require any radical change in a maturing industry and it is more cost-effective.

My proposed strategy

Initially in this component I think a hybrid strategy should be used. We should examine the flaws of current hardware and produce devices can protect people from attack. And at the same time, we look at designing any future hardware devices using flaw-free development techniques. If we finally get to the stage where flaw-free hardware production is a reality, then there will no longer be a need to include the second alternative in this sub-component. However, I see human intellect as being far superior to computational power and as such

humans may see flaws in hardware devices that could never be detected by a software package. Therefore, I expect this hybrid strategy to be used for a long time to come.

8.1.2 Software

The purpose

The software aspect involves securing the client software, the commerce server software, the operating system software and software implementations of Internet protocols. The issue of securing Internet commerce software requires that we take a look at the security of software products in general. Due to the fact that bugs are so common in software, the public has grown to expect a bug or two to surface every now and then. However, the impact of these bugs seems not to be truly understood. These bugs may potentially lead to the demise of one's computer system.

The current scenario

The widespread acceptance of bugs is an indication of how little importance we place on fault-free software development. The demand for software development times to decrease only aggravates the problem of bugs. The majority of the attack techniques that are emerging are as a result of poor use of programming language constructs, logical errors, software configuration mistakes and poor implementation of user specifications. Software engineering is still very much a young discipline in relation to other fields of engineering, and as such we expect some teething pains. However, the continuation of the status quo may put the entire profession in a state that will be near impossible to escape.

The prognosis

I feel that it is best to remedy an already bad situation before the problem grows to gargantuan proportions. It seems much wiser to take a preventive approach, rather than a prescriptive approach in a field that so greatly impacts the lives of millions of people. It now becomes clear that a shift in the current method of software development is required. I feel that changes will be needed in the software development process, in the area of programming languages and technology and in the area of software tools.

8.1.2.1 Software Development Process

The problems

The basic problem with the software development process concerns the development models used. None of the more popular models adequately address the issues of quality assurance and risk management. Being a young discipline, software engineering should learn from the other more established engineering disciplines. They have experienced and grown to appreciate:

- the effectiveness of simulation before construction,
- the virtues of re-use over construction from first principles,
- the use of quality checks at regular interval in the development process,
- the value of risk management,
- and the importance of the synergy between functionality and presentation.

Though some software engineering practitioners have recently embraced some of these concepts, it is still abundantly clear that these concepts have not yet been seen as a necessary addition to the profession. Let us now focus our attention on the other problems being faced by the two most popular software development models being used today, namely: the waterfall model and rapid prototyping.

The waterfall model

The waterfall model takes a long time, is sequential in nature and any errors in the initial stages will have disastrous on the remaining stages. What is important to us is the fact that the waterfall model leads to an incorrect solution if the user requirements are not correctly specified. What concerns us is the fact that these errors may not be noticeably in the implementation until a hacker discovers them and uses them to his advantage. In order to solve this problem, a way is needed to verify that the implementation matches the specifications correctly. This can be achieved by using formal methods to specify the requirements. The only problem we are now faced with is the inclusion of bugs due to sloppy programming or misuse of programming features. This topic will be dealt with in section 8.1.2.2.

Rapid prototyping

In rapid prototyping, the user has a lot of input in the design of the system. This is to the user's advantage and, most of the times, to the detriment of the system's structure. As users request changes to be made to the prototype, the system will change so much so that the initial structure and intent of the prototype may cease to exist. This coupled with the outdated documentation will no doubt have disastrous effects in the long run. The effects may range from integration problems, as the different segments of the system may have been warped during development, to instability, as there may be some leftover code from a previous review, which occasionally affects the program. The end-result is a package whose structure is unknown and for which the documentation is poor.

Hackers can use this to their advantage. If they are able to obtain the source code, then any errors or loopholes that they can find may be used as a part of an attack technique. The solution to the rapid prototyping problem is to incorporate a rigorous documentation regime. This regime would stipulate that the particulars of the system be detailed before the next evaluation. At this point, it would be wise to also clean up the code. It would ensure that segments are well documented and reflect the true structure (however, ad-hoc it may be). Thus, the software can now be easily checked for any possible exploitations beforehand. Before this discussion on the software development process ends, there is a matter that must be addressed

Though not a problem with the software development process itself, the sacrificing of quality and sometimes even the functionality of the software package to meet deadline is an issue that needs to be discussed. There is no excuse for doing this act and neither is there a clear-cut solution. It is essentially a human problem. Theoretically, there are two possible ways to decrease the effects of this deadline phenomenon. In actuality, each project leader must grapple with his own solution, but the imperative is that it must be put to rest.

The need to improve the software development process

In improving the software development process, it is hoped that less bugs will be included in software products and by extension there will

be less ways in which someone can exploit your system.

8.1.2.2 Programming Languages and Technology

The languages used to build current systems support features that, if not properly used, may allow an attacker to execute a potentially dangerous program on one's machine. Currently, programmers are not taught how to properly use potentially unsafe features. This cannot continue if we want secure Internet commerce tools to be written. Most of these potentially unsafe features form the foundation of some of today's more powerful language. In fact these features are the reason why these languages are so powerful. Among these potentially perilous constructs are array management, pointers, dynamic memory allocation and recursion. A prime example of the misuse of one of these features is the Internet worm of 1988. The programmer utilized the fact that C allows one to overwrite memory locations after the end of an array (this is called a stack overflow attack). To lessen this problem, we either have to educate programmers on the security risks posed by certain constructs in various languages or move to safer languages, that reduce the threat of these features. However, it should be noted that even the safer languages may possess constructs that may be misused. Again, programmers must be fully aware of these constructs and how they can make trying to misuse them futile.

8.1.2.1 Software Tools

In this context, software tools refer to tools that assist in the management of security features in an Internet commerce system. For our discussion, any software package created to combat or lessen the effects or after-effects of a particular attack method or a group of attack methods is considered a software tool. In today's world, there are many software packages that have insecure features and many more software risks caused by careless configuration. There are also several tools that lessen the risk associated with unsafe software. Among these products are: cookie management systems, password management systems, intrusion detection systems, automated auditing systems, key management systems, anti-virus systems, information content filters, backup and recovery systems, configuration management systems and network management systems. I foresee a whole lot more these tools popping up as new and innovative ways are

found to misuse software. However, these software tools may also pose security risk themselves. If they are not properly implemented or set up, they can be just as vulnerable as the other software packages. Thus, care has to be taken in the construction of these tools and in their use.

8.1.3 Standards

Introduction

In today's world of Internet commerce, there is very little standardization in the areas of commerce server technology, payment protocols, authentication techniques and encryption. This is simply unacceptable. Without standardization, Internet commerce systems will not be able to work together properly. This will not only serve to hamper the growth of Internet commerce. In order to guarantee secure Internet commerce, standards need to be open, consistent and complete.

Openness

In today's world, Internet standards tend to be very open (that is non-proprietary). All indications point to the fact that this trend will continue. And so it should. Releasing a standard for criticism to the entire world is possibly the most effective way to find out if it can stand up to harshest of environments. If there is no one is the world who can refute a standard, then that instills in us a feeling of confidence when we are using it.

Consistency

As the Internet is a universal entity, standards must also be consistent across physical borders. Having one encryption standard for the United States and a weaker standard for the rest of the world is simply unacceptable if we want Internet commerce to be globally viable. The universality of the Internet dictates that any technological advancement that is made should be universally applicable. Thus, an Internet must be universal (i.e. consistent across all geographical borders).

Completeness

There are some attack techniques that exist simply because different

people had different interpretations of what should be implemented in the gray areas not addressed by a particular standard. For example, the Ping of Death attack exploits the facts that there is a maximum size for a TCP/IP packet and that the TCP/IP specification does not address what should be done when an over-sized packet is sent. If this was specified, then this attack technique would not have been ever formulated. Therefore, one now sees why standards need to be complete.

8.2 Legislative component

The need for legislature

For Internet commerce to truly be as safe as traditional commerce there has to be a legal environment in which Internet commerce can take place (similar to the one that exists in traditional commerce). As evidenced by the fact that Internet commerce is so prevalent today, in a legislature-free on-line commercial world, it is clear that the legal framework is by no means a deterrent for Internet commerce. However, it may be a limiting factor. A legislative framework is vital to consumer and business confidence. Lack of such an environment will cause people to be tentative about indulging into Internet commerce. On the other hand, with legislature comes bureaucracy. This component should seek to instill confidence in Internet commerce market, while making sure that the associated bureaucracy is kept to a minimum.

The viability of Internet commerce without legislation

Despite the fact that mechanisms to ensure non-repudiation exist and can be easily incorporated into Internet commerce systems, these mechanisms are rendered useless if they are not legally binding. In an Internet commerce world devoid of a legal framework, the possibility still exists that people can be swindled on-line by firms who claim to be ignorant about transactions that have occurred between them. In some countries, this scenario is a reality. This brings to the fore the need for a uniform legislative framework, which means that any legal framework that is established has to be an international collaborative effort.

How to create this legislative framework

Creating such a framework is a very attainable task. In recent times, similar situations have been faced and conquered, for example the collaboration of worldwide entities to plan the future of the TCP/IP protocol. The approach used was to create an objective, unbiased and independent body to perform a specific task and place a non-governmental supervisory board in control of this body. Take for example, the Internet Engineering Task Force (IETF) and the Internet Activities (IAB). The IETF is the body that is responsible for solving short-term engineering needs on the Internet and it is under the direct supervision of the IAB *(the IAB oversees the development of the Internet protocols)*. I think that a similar approach should be used in the creation of this framework. I suggest that a body called the Internet commerce Consortium (ICC) be formed. This body will oversee all activities dealing with the development of Internet commerce. Another body called the Internet commerce Legislative Team (ICLT) would develop the legal framework and be under the supervision of the ICC. For coordination purposes, the ICC should be placed under the supervision of the Internet Society (ISOC), the body that oversees Internet development. The members of the ICLT would be drawn from experts in the fields of Internet technology, traditional commerce, Internet commerce and international law. The character of the ICLT should be beyond reproach. Their primary focus will be to protect both customers and merchants, irrespective of their native country. To prevent the ICLT from ever evolving into a vehicle for someone's personal agenda, guidelines need to be outlined to prevent compromise of its primary function from ever being contemplated.

Issues to consider in this framework's development

In formulating this framework, it would be wise to make it flexible enough to incorporate technological advances and broad enough to facilitate the legal needs of the countries of the world, while not bogging it down with the details of any country's legal system. The aim is to develop a minimal set of guidelines that will encourage a fair, free and just Internet marketplace. This legal environment should act as a catalyst for Internet commerce and it should address some of the areas that are dealt with in traditional commerce, as well as tackle Internet-specific issues. Among the issues that must be dealt with are:

merchant responsibility, consumer rights, data security, intellectual property laws, privacy laws, taxation, international contract laws, bank secrecy laws, data encryption methods and mandatory record keeping.

Implementing this framework

After developing this framework, the ICLT may ask the various countries of the world to agree to implement this Internet commerce framework. Those nations who refuse to cooperate would have the companies of their nationals blacklisted. Appropriate ways would have to be devised to alert customers, who are transacting business on-line, that the company they are transacting business with will not uphold the legal framework. The rejection of this framework by a government raises a few very important issues. Let us say that government X rejects the framework and company Y is owned by a national of the country ruled by government X. If company Y is willing to work under the tenets of the framework, then some mechanism should exist to allow company Y to do so. The legal framework should properly address any side issues, such as the one just mentioned.

Post-implementation

After the implementation of the framework, the brunt of the work is left to the individual governments of the world. The responsibility of seeing that companies adhere to the framework is in their hands. Also, they will handle breaches of this framework. The effective duty of the ICLT would then become to interpret the framework for individual countries and to make amendments to it where necessary. Maintenance of this framework should be done with the utmost care, as one does not want to ratify a change that may have disastrous long-term implications.

5.3 Social component

The make-up of the social component

For secure Internet commerce, it is necessary for us to have educated Internet commerce participants. They should be educated not only about the technological advances being made by hackers, but also educated about their rights and responsibilities. Education empowers us and a greater level of education will foster a new level of awareness

in the Internet commerce society. This level of awareness will inevitably allow the effects of attack techniques to be minimized and may even allow for more constructive input into the development of Internet commerce. Thus, one can see that the thrust of this component is education. For far too long has the technique of not volunteering information been practiced. In the Internet commerce arena this strategy will only encourage consumers to be more skeptical.

The benefits of education

The adage "knowledge is power" is one of those truisms that is often said, but never truly appreciated. This is demonstrated by the lack of emphasis on educating consumers. It appears that the effects of knowledge on the general public have either been not examined or they have been deemed too costly. However, the long-run benefits of educating the participants prove to save costs rather than to increase them. A well-informed public is better able to protect itself and has a better perception of the current state of affairs. Customers should be aware of potential security threats and the various ways of nullifying the effects of these threats. Also, merchants should know what would-be attackers may potentially do and how to stop them. Access to this information about the attack techniques, their associated solutions and the locations of patches or fixes should be distributed to everyone.

Nature of the information

We must also consider that even though we want to educate the public in order to allow them to properly protect themselves, we do not want to provide them with enough information to allow them to become attackers. Thus, descriptions of both attack techniques and solutions should be general enough not to allow the public to try and simulate these attacks or circumvent the solutions. The important thing is that they know of the presence of these attacks and have the means to defend themselves. However, it should be noted that it will be necessary to describe how an attack can be performed. In doing this, there is always a possibility that a cunning, knowledgeable attacker may be able to simulate this attack, in spite of the fact that the description was sketchy. Thus, the issue of generality needs to be taken seriously.

The feasibility of distributing technical information

Some people may be concerned that some of the information about attack methods may be too technical for the general public, but I think otherwise. When the Internet and World Wide Web began to gain prominence in the eyes of the public, many researchers thought that the language of domain names and IP addresses would be too technical for the public. These researchers saw two possible solutions. Either the naming system needed to become simpler or the public would have to adapt to the technical scheme. They thought the public would never go for the former option, but they were surprised. So, for the proponents of the argument that the information may be too technical for the public, I say to you "never underestimate the capabilities of the public". Given that this information can be understood by the public, the question of how to get this information to them now needs to be addressed.

How to distribute this information

There are two possible ways to get the information to the general public. The first way is to place either the information or links to the information at spots on the Internet that have a high likelihood of being visited. The second way is to have an independent trusted authority distribute this information and administer changes to the particular software programs (if necessary). In both instances a way to authenticate the information is necessary. This can be found in one of two ways; either an organization can describe and implement such a scheme and everyone adheres to it or we entrust the entire verification process to a specific body.

Way One: posting the information for viewing

The first method of getting the information to the participants is very simple to understand. The person or persons who have information that they wish to share with the public will have to send this information to a body. This body has the task of verifying the information and ensuring that the information is placed on the more popular sites. The inclusion of this body adds some structure to our scheme and eliminates the possibility of a false individual, with contacts at a verification firm, from posting bogus information. The

technique's biggest asset is its simplicity. However, one needs to realize that the general public is wary of clicking on flashy links that shout out "New Windows NT bug". Also, more experienced users tend to focus on the quest that they are currently pursuing and usually refuse to divert from it.

Way Two: sending the information to the clients

The other way of getting the information to the public is to ask that they subscribe to an independent trusted third party. This means that the third party will update the information on the person's computer periodically and make the necessary changes. This third party will receive information on the attack techniques and their solutions and will verify it, and make the updates. This can be seen as an application of push technology. The benefits are obvious; customers do not have to waste time searching for information. However, giving an organization the right to put things onto your computer is never a good thing to do, because they may possibly place malicious programs on your system if they so desire.

The strategy to be used

Either way can be effectively used. One has to keep in mind that each has its pitfalls. I would however extend the first way and place the information on all possible media available.

8.4 Thoughts

Despite the fact that this framework seeks to institute what may be considered an ideal, I think that it is a realistic ideal. Firms will know when it is time for them to cooperate for the better good of their individual businesses. I know that sooner or later they will recognize that such a framework is indeed necessary. I must also state that I doubt that the social component will be given the treatment it deserves, but I hope that it is addressed in some respect. It is better to be partially included than not included at all.

Chapter 9
Secure Internet commerce: The Future

"Prediction is difficult, especially of the future"
- Neils Bohr

The factors

I believe that the future of the Internet will be dictated by four factors, namely:

- The evolution of the Internet.
- The advances in the hardware and software fields.
- The level of ignorance of the public.
- Private sector cooperation.

The evolution of the Internet

To see the Internet of the future, one only needs to look at the Next Generation Internet (NGI) and Internet2 projects. The Internet of the future will no doubt solve some of the problems that we are facing today and of course it will introduce new ones. A lot of care should be taken in this effort to transform the Internet. It would be prudent to actually conceive the possible problems that may arise and envision their solutions or else we might be in a worse situation that what currently exists.

Advances

Advances in hardware and software may offer both good and bad. These advances may offer faster, cheaper and more durable products, but at what cost. Normally, the cost is that the product becomes very complex in nature. With this complexity, there will be a new set of problems to be tackled. The magnitude of the effect of these advances will be determined by how thoroughly planned out the design of these products was. It should be noted that product design should include a bit of forethought. One should think of possible abuses and misuses of the product and seek to find solutions.

Public ignorance

The level of ignorance of the public will continue to be a sore point for a while to come. The public in this context refers to management, consumers, business owners and regular employees. Keeping the public in ignorance will only aid the efforts of the elements who wish to attack systems forever.

Private sector cooperation

The willingness of the private sector to cooperate on things like standardization efforts and legislature will determine how quickly Internet commerce will grow. It is true that every firm is looking for a comparative (or even an absolute) advantage, but to look for it in such a far-reaching industry will lead to nothing short of chaos. I can see a gradual move towards a secure Internet commerce environment. A move that can be thought of in terms of the short, medium and long term.

The short and medium term

In the short to medium term, I can see the general *miseducation* of the public continuing. This will have a severe impact on Internet commerce security. People will remain incognizant of the real dangers of the net and the methods that can be used to protect themselves. However, it has been brought to my attention that in spite of the skepticism that is present, it is said that commerce over Internet is booming. This means that people are either people have been lulled into a false sense of security or they just don't know the real threats. The fact that usage of Internet commerce tools is expected to more than triple by the year 2001 is not only exciting, but also scary. The probability that a lot of firms may go bankrupt is very plausible. This period will lead to people realizing that to maximize their individual gains, it is sometimes wiser to initially invest in the collective.

The long term

In the long term, however, I see that firms will truly realize the need for cooperation, non-proprietary standards and education. This will mean that a framework similar to the one that exists in traditional commerce would have to be established. My only concern is how long it will take for firms to come this realization.

Chapter 10
Conclusion

No one can dispute that Internet commerce has immense potential. It is also an indisputable fact that we need to have secure Internet commerce. My stance is that this can only be truly achieved by using a holistic approach to securing Internet commerce. It cannot be emphasized enough that this approach should be dynamic in order to keep up with the changes of the fast-paced world around us. The framework that I propose seeks to address:

- The changes in hardware and software technologies that are needed.
- The need for hardware and software tools that will help ensure security.
- The empowerment of the public. *This will foster the trust that is required in any business environment.*
- The legal power of an Internet transaction and legal responsibilities of Internet commerce participants.

People may want to believe that currently Internet commerce is as safe as traditional commerce. I disagree with this stance. My contention originates from the facts that Internet commerce transactions are not universally legally binding and that the general buying public does not yet fully trust on-line business transactions. It is also very important for senior management, the people who make the decisions and shape the business landscape, to get rid of their archaic notions of security.

In 1997, the publication Information Week surveyed 1,271 U.S. system/network managers. Only 22% believed that their own senior managers regarded information security as "extremely important" It is this underestimation of the significance of security that will also seriously impair the growth of Internet commerce. Anna Johnson purports (and I agree with her) that several myths seem to be perpetuated at the senior management level and that these myths will damaging effects on Internet commerce security. These myths are:

- The "it won't happen to us" myth
  "No one would be interested in stealing from us, or penetrating, damaging, destroying or otherwise tampering with our network."
- The "we run the best systems so they must be secure" myth
  "Our hardware, operating systems and software are made by a first-class vendor" or "we have the most recent version of this product"
- The "our vendor will look after us" myth
  "Our vendor will tell us if a vulnerability is found in our system"
- The "we don't need to test our systems" myth
  "We've taken all the precautions that need to be taken so penetration tests are not needed"
- The "it's the I.T. department's job" myth
  "They look after the systems so they look after security as well" (without giving them the resources, time and money to do so)
- The "our sub-contracted I.T. company will take care of security" myth
- The "we can't afford it" myth
  "Security is a luxury that for which don't have the budget"

This just highlights the fact that education is essentially to the success of securing Internet commerce. Senior managers, like most people, tend to place too much blind faith their software and hardware manufacturers, believing that everyone is looking out for the managers' best interest. This is not always the case. One needs to come to the realization that different participants have different motivations. And for Internet commerce to be secure, motivations need to be derailed for a period of time and the greater good and greater profit need to be considered.

"Look not to fleeting immediate gain, but towards the greater long-term profits"
- Tyrone Grandison

Appendix I: The lure and benefits of Internet Commerce

Put simply, the Internet is conceptually an enormous marketplace with a phenomenal growth rate. This is the primary fact that drives most companies to venture into Internet Commerce. According to the research firm Input Inc., the Internet population is expected to reach 60 million by 1999 and there are currently more than 10,000 companies now providing Web servers. This implies that the Internet is an enormous conduit for communicating with customers. They also say that in 1994, Internet-related electronic commerce accounted for less than 1 percent of the total business exchanged electronically and by 1999, this share is expected to grow to 20 percent. These statements bring out the fact that the marketplace provided by the Internet is very appealing to businesses. This is further supported by research from Piper Jaffray Inc. They estimate that businesses are expected to fork out US$31 billion on electronic commerce-related software, hardware, products and services by the year 2001, while Internet-based purchases will exceed US$228 billion. Thus, we can see that the main impetus for firms is the enormous earnings that can be gained from the global marketplace. No doubt the fact that current estimates indicate that the number of users of the Internet worldwide grows by 1 percent each month also helps to shape these companies decisions. However, most companies make this business decision without first getting a complete picture of the Internet Commerce landscape. Nonetheless, many firms bypass the feasibility study and look focus solely on the benefits.

The benefits of Internet Commerce are well known. In this appendix, they are presented by category. First the benefits to the customer and then the benefits to the firm. The benefits to the customer are:

- Product information and company services can be easily and quickly accessible to any customer from any part of the world.
- Customers can access details and transact from the comfort of their homes or offices.
- Customers are able to conduct transactions at any time of the day.
- Given the extremely competitive market that Internet Commerce will spawn, the quality of services will improve.

The benefits to the business are :

- Businesses can reduce the time it takes to send and receive purchase orders, invoices, and product information.
- Better communications can be established and maintained between suppliers and customers.
- Businesses can provide instant access to new product and service details.
- No money wasted in sending paper prospectuses, email brochures with file attachments, etc.
- No handling of paper money - everything is digital. Hence, transactions are settled on-line.

Appendix II: Supposedly Secure Non-electronic transactions

Here are examples of common, everyday credit card transactions The aim of this exercise to show that credit card transactions may be considered as on the same security level irrespective of the fact the transaction is done on-line or not. These examples were taken from documents provided by Wilcox Enterprises.

A Typical In-Store Credit Card Transaction

A card is presented to a cashier at the checkout area of the store. The cashier either uses electronic approval and prints a receipt, or the card is imprinted on a receipt. After the signature is in place the merchant copy is placed safely in the cash register. In most cases, the card is in plain sight the entire time and there is little or no danger of the number or card being stolen, right? Wrong. The clerk could reprint the receipt after you leave the store, or may have embossed two receipts simultaneously, or may even make a handwritten copy of the receipt from the store copy. The credit card companies do everything in their power to ensure that the merchant can be trusted with your card, however your card number can easily fall into the hands of a dishonest employee. We don't really want to think about the bank employees do we? Is this a secure transaction? Not really, but we are used to it, and accustomed to the risk, if we ever think of it.

A Typical Restaurant Credit Card Transaction

A restaurant is probably one of the riskiest places to use a credit card, yet we do it all the time. The server drops the check on our table and we put our credit card on the little tray or in the fake leather folder and wait for the server to take it away. Where do they take it? What do they do with it? We don't know. They come back with it a few minutes later with some of receipt, which we sign and leave on the table. Who picks the receipt up after we leave? This transaction is incredibly risky and insecure. Restaurant transactions suffer all of the problems of an in-store transaction except we let a total stranger take our card to some other part of the restaurant, do whatever they want with it, then we leave the merchant copy of the receipt sitting on the table where any passerby, including restaurant employees and patrons,

could pick it up and either take it or copy it. This transaction is one of the least secure of the types we normally do, and is made more risky by our blind acceptance of it.

A Telephone Order

The primary risk is that you have no idea who you are talking to. You give your credit card number to a total stranger who you are assuming is the merchant you think it is. There is some small risk that someone could also be eavesdropping on the conversation and pick off your number that way. The same risks as with the regular in-store transaction then take effect. This is more secure than restaurant transactions in that you have limited the potential exposure somewhat. However, the merchant has no way to ensure that you are the rightful cardholder either.

Appendix III: Recent Security Concerns

The following are stories taken from the Risks-Forum Digest. They show just how significant a threat exists to computer systems worldwide and the effects of the exploitation of the weaknesses of these systems.

The stories are taken verbatim from the Digest and are placed in one of three categories: Hacks, Software Problems and Miscellaneous. Below is a table of contents for the various stories.

Date: Wed, 22 Apr 98 9:45:17 PDT
From: "Peter G. Neumann" <neumann@csl.sri.com>
Subject: Hackers claim major U.S. defense system cracked

A Reuters article by Andrew Quinn in today's print and electronic media notes that a group calling itself Masters of Downloading (a new MOD, including members in the U.S., Britain, and Russia) claims that it has been able to obtain secret files from a computer system used to control military satellites, via the Defense Information Systems Network (DISN). The files include the DISN Equipment Manager (DEM), which controls the U.S. network of military Global Positioning System (GPS) satellites. MOD members apparently informed John Vranesevich (who runs the computer security website AntiOnline <www.antionline.com>) of their exploits.

From honig@206.40.207.40 Thu May 28 15:35:58 1998
Date: Wed, 04 Mar 1998 17:10:23 -0800
From: David Honig <honig@206.40.207.40>
To: cypherpunks@toad.com
Subject: Win95/NT attack in CNN news

Hacker attack crashes Windows systems coast-to-coast

March 4, 1998 Web posted at: 10:05 a.m. EST (1505 GMT)

SAN DIEGO (AP) -- Computer security experts blame hackers for an Internet attack that caused computers running Microsoft's Windows NT software to crash from coast to coast, mostly in government and university offices.

While no real harm was done, it was too early to gauge the full extent of the attack.
Experts said the far-flung glitches could only have been the result of a deliberate act, The San Diego Union-Tribune reported Wednesday.
The crash Monday night affected computers running Windows NT --

the operating system for larger computers and networks -- and Windows 95.

Problems were reported at the Massachusetts Institute of Technology, Northwestern University, the University of Minnesota and University of California campuses in Berkeley, Irvine, Los Angeles and San Diego. Unclassified Navy computers connected to the Internet also crashed on Point Loma and in Charleston, South Carolina, Norfolk, Virginia, and elsewhere.

"It happened so fast," said Craig Huckabee, a research associate in the Computer Systems Laboratory at the University of Wisconsin. "In our department, I would have to say about 90 percent of the machines were affected."

Despite the coordination of the attack, the computers that crashed could be restarted without losing information, computer security experts said.

The attackers used the Internet to broadly distribute a snippet of deliberately malformed data, said Ron Broersma, a civilian computer security expert at the Navy labs on Point Loma. The prank exploits a glitch in the Windows NT program by instructing the computer to devote excessive memory resources to solve a problem that can't be solved.

Microsoft security manager Ed Muth said the company is working on a software patch that fixes the vulnerability in Windows NT programs. An unidentified Microsoft executive told the Union-Tribune it was unknown if the attack was related to Microsoft Chairman Bill Gates' appearance Tuesday at a Senate hearing where he defended his company against allegations of antitrust violations.

Date: Fri, 17 Jul 1998 03:31:45 -0400
From: Matt Blaze <mab@crypto.com>
Subject: ``Better DES challenge'' solved by John Gilmore and ``Deep Crack''

On June 23 1997, I offered a prize of 56 bits ($7.00) for finding a DES key with a certain interesting property. In particular, I wanted a DES key such that some ciphertext block of the form <XXXXXXXX> decrypts to a plaintext block of the form <YYYYYYYY>, where X and Y represent any fixed eight-bit byte value repeated across each of the eight bytes of the 64 bit DES codebook block.

Finding a key of this form would require either computational effort approximately equal to searching the DES keyspace or discovering a new cryptanalytic technique against DES. Knowing such a key would therefore demonstrate that it is feasible to mount an exhaustive search against the DES keyspace or that there is some weakness in DES that allows keys to be found analytically. This challenge, then, has the desirable property that a result ``speaks for itself'' in demonstrating the weakness of DES, without the need for an ``honest broker'' who must safeguard the solution. The solution keys could not be known to any people who haven't themselves searched the keyspace or found some other weakness. It would be just as much of an accomplishment for me to claim the prize as it would be for anyone else.

I am pleased to announce that the prize has been claimed. On July 2, 1998, John Gilmore, of the Electronic Frontier Foundation, informed me that:

With a (parity-padded) key of 0E 32 92 32 EA 6D 0D 73, the plaintext of 8787878787878787 becomes the ciphertext 0000000000000000

According to John, this solution was found after several days of work with the EFF ``Deep Crack'' hardware, a specialized parallel processor optimized for DES key search. Information on Deep Crack can be found at <http://www.eff.org/descracker>. I am especially gratified that this DES challenge problem was chosen as the first application of the Deep Crack hardware, and that the challenge has revealed data that might, perhaps, yield some additional analytic clues about the structure of the DES algorithm.

A number of individuals and organizations generously pledged additional bits to supplement my original (quite modest) 56 bit prize,

for a total over 10000 bits ($1250.00). I will be contacting these individuals privately to inform them that their pledges have come due.

Note that although the prize has been claimed and the contest is now officially closed, there may be other solution keys (in fact, we'd expect to find about 255 more, if DES emulates a random permutation). I encourage the community to continue looking for solution keys. Indeed, it would be interesting to know how many such keys actually do exist in DES.

Software Problems

Date: Thu, 30 Jul 1998 17:06:13 -0400
From: Edupage Editors <educause@educause.unc.edu>
Subject: Software flaw exposes e-mail programs ...

A security flaw found in several of the most widely used e-mail programs (Microsoft Outlook Express, Microsoft Outlook 98, and Netscape Mail) could be used by malicious persons to send computers using those programs a virus that could destroy or steal data and could cause those computers to crash. The flaw, which is known as a buffer overflow error, occurs when a program fails to check the length of each character string. This failure means that a string too large to fit into an allotted memory location will lock up the program and fool the operating system into running attacker code in its place. Whereas new languages such as Java have built-in safeguards to prevent this kind of programmer error, older languages such as C and C++ do not. Computer security specialist Steven Bellovin says, "C makes it too easy to slice your fingers off, and programmers all over the world are doing so with great regularity." (*The New York Times*, 30 Jul 1998; Edupage, 30 July 1998. This is the Finnish find.)

Date: Mon, 17 Aug 1998 13:13:00 -0700
From: David.Ratner@software.com (David Ratner)
Subject: Computer flaw makes water undrinkable

A computer glitch in Lewiston, Maine, shut down the chlorination system and caused the chlorine content of the city water to drop below the safety threshold, affecting 40,000 residents. This occurred during the night, and was not discovered until a routine check 14 hours later. Notices were then sent out to 9,000 homes advising people ``to boil the water before drinking." It took 30 hours to solve the problem. The city has now installed an automatic system to notify an on-call supervisor in case this recurs. [Source: *USA Today*, Tech Report, Glitches of the Week, updated 17 Aug 1998, http://www.usatoday.com/life/cyber/nb/nb1.htm>; PGN Abstracting]

How many other computer systems that require 24x7 service don't have 24x7 monitoring?

Date: Wed, 11 Mar 1998 13:46:02 -0500
From: "Edelson, Doneel" <doneeledelson@aciins.com>
Subject: Radar blip lost Air Force One

The Federal Aviation Administration is investigating whether an air traffic tracking system went out amid reports that Air Force One vanished from radar screens for 24 seconds. Broadcast reports said the airplane disappeared from radar screens Tuesday morning as President Clinton traveled to Connecticut. ... The long-range radar system at the center has a history of going off and momentary blips are a frequent occurrence, DiPalmo said.

Miscellaneous

Date: Tue, 23 Jun 1998 10:36:26 ECT
From: Paul van Keep <pvk@acm.org>
Subject: ISP security fiasco

WorldOnline, one of the large dutch ISP has suffered a number of security failures recently. These were mainly attributable to human error and weak OS level security measures. The most prominent mistake was to assign passwords to users by using a combination of the first four letters of their userid and a 4 digit code. I even doubt that

the 4 digit code is randomly chosen but even if it is, cracking an account with this knowledge is pretty easy and straightforward. In an attempt at damage control, WorldOnline last week stated that it's system is secure and that users should not worry, although they do not feel responsible for breakins on websites that they host. To prove their point and to get some positive publicity, they even launched a competition with a prize of $7400 for the first reproducible crack. The prize was claimed within a few days by a cracker who managed to extract thousands of private e-mails from a mail server. Another team cried foul because the system they had hacked into (running the internal helpdesk) had been abruptly switched off in an attempt to stop the crackers. The dutch provider association (NLIP) has denounced the competition as a cheap publicity stunt.

Date: Mon, 22 Jun 1998 01:32:39 +0300
From: Mikko Hypponen <Mikko.Hypponen@DataFellows.com>
Subject: New virus posts user documents to public newsgroups

A Word macro virus called WM/PolyPoster was recently found. As the number of macro viruses is soon reaching 3000, there's nothing special about this. However, under the right conditions, this virus sends copies of a victim's Word documents to 23 different Usenet newsgroups under subject lines like "New Virus Alert!," "Important Princess Diana Info" and "How to find child pornography."

Risks are obvious and three-fold:
1. Private and confidential data is disclosed to the world.
2. When unsuspecting fellow users download and read these documents, they get infected themselves.
3. The user's name get's archived to services like DejaNews as posting messages related to software pirates or child porn.

More details at:
http://www.DataFellows.com/news/pr/eng/fsav/19980618.htm
http://www.DataFellows.com/v-descs/agent.htm

This virus is not known to be widespread at this time.

Mikko Hermanni Hypponen - Mikko.Hypponen@DataFellows.com
Tel +358 9 859 900
Data Fellows Group, PL 24, FIN-02231 Espoo, Finland
http://www.DataFellows.com/

Date: Mon, 27 Jul 1998 18:03:55 -0400
From: Eddie Sullivan <eddie@merl.com>
Subject: A 4-digit PIN truncation

I've discovered an interesting problem with bank teller machines. At least with my BankBoston ATM card, only the first four digits of the personal identification number are relevant. I have a five-digit PIN, and I've tried typing just the first four, and I've also tried the first four plus an incorrect fifth digit. In both cases, the machine was more than happy to fork over money. I've tried it in BankBoston and USTrust ATM's.

Glossary

ACSE

Association Control Service Element. The method used in OSI for establishing a call between two applications. Checks the identities and contexts of the application entities, and could apply an authentication security check.

active content

Refers to web applications that contain executable statements.

address resolution

A means for mapping Network Layer addresses onto media-specific addresses. See ARP.

API

Application Program Interface. A set of calling conventions defining how a service is invoked through a software package.

ARP

Address Resolution Protocol. The Internet protocol used to dynamically map Internet addresses to physical (hardware) addresses on local area networks. Limited to networks that support hardware broadcast.

ARPA

Advanced Research Projects Agency. Now called DARPA, the U.S. government agency that funded the ARPANET.

ARPANET

The network built by ARPA. A packet switched network developed in the early 1970s. The "grandfather" of today's Internet. ARPANET was decommissioned in June 1990.

authentication

Guaranteeing that a message really has come from the person who claims to have sent it.

bridge

A device that connects two or more physical networks and forwards packets between them. Bridges can usually be made to filter packets, that is, to forward only certain traffic. Related devices are: repeaters which simply forward electrical signals from one cable to another, and full-fledged routers which make routing decisions based on several criteria. In OSI terminology, a bridge is a Data Link Layer intermediate system. See repeater and router.

broadcast

A packet delivery system where a copy of a given packet is given to all hosts attached to the network. Example: Ethernet.

bomb

A program that performs some destructive action that a particular instance. Either when a particular time is reached (time bomb) or some event occurs (logic bomb).

CERN

The European Center for Nuclear Physics

CGI

Common Gateway Interface. The language used to write Web interface programs.

CMIP

Control Management Information Protocol. The OSI network management protocol.

connectionless

The model of interconnection in which communication takes place without first establishing a connection. Sometimes (imprecisely) called datagram. Examples: LANs, Internet IP and OSI CLNP, UDP, ordinary postcards.

connection-oriented

The model of interconnection in which communication proceeds through three well-defined phases: connection establishment, data transfer, connection release. Examples: X.25, Internet TCP and OSI TP4, ordinary telephone calls.

confidentiality

Making sure that the contents of the message have not been disclosed to third parties.

CSMA/CD

Carrier Sense Multiple Access with Collision Detection. The access method used by local area networking technologies such as Ethernet.

DARPA

Defense Advanced Research Projects Agency. The U.S. government agency that funded the ARPANET.

DDN

Defense Data Network. It is comprised of MILNET and several other
U.S. Department of Defense networks.

DoD

U.S. Department of Defense.

DNS

Domain Name Server. The distributed name/address mechanism used in the Internet.

domain

In the Internet, a part of a naming hierarchy. Syntactically, an Internet domain name consists of a sequence of names (labels) separated by periods (dots), e.g., "tundra.mpk.ca.us." In OSI, "domain" is generally used as an administrative partition of a complex distributed system, as in MHS Private Management Domain (PRMD), and Directory Management Domain (DMD).

dotted decimal notation

The syntactic representation for a 32-bit integer that consists of four 8-bit numbers written in base 10 with periods (dots) separating them. Used to represent IP addresses in the Internet as in: 192.67.67.20.

encapsulation

The technique used by layered protocols in which a layer adds header information to the protocol data unit (PDU) from the layer above. As an example, in Internet terminology, a packet would contain a header from the physical layer, followed by a header from the network layer (IP), followed by a header from the transport layer (TCP), followed by the application protocol data.

encryption

Transforming the message to a ciphertext such that an adversary who overhears the ciphertext can not determine the message sent. The legitimate receiver possesses a secret decryption key that allows him to reverse the encryption transformation and retrieve the message. The sender may have used the same key to encrypt the message (with symmetric encryption schemes) or used a different, but related key (with public-key schemes). DES and RSA are familiar examples of encryption schemes.

fragmentation

The process in which an IP datagram is broken into smaller pieces to fit the requirements of a given physical network. The reverse process is termed reassembly.

FTP

File Transfer Protocol. The Internet protocol (and program) used to transfer files between hosts.

gateway

The original Internet term for what is now called router or more precisely, IP router. In modern usage, the terms "gateway" and "application gateway" refer to systems which do translation from some native format to another. See router.

hijacking

The theft on someone's resources.

HTML

Hypertext Markup Language. The language used to write Web pages.

HTTP

Hypertext Transport Protocol. The protocol used by the World Wide Web.

IAB

Internet Activities Board. The technical body that oversees the development of the Internet suite of protocols (commonly referred to as "TCP/IP"). It has two task forces (the IRTF and the IETF) each charged with investigating a particular area.

ICMP

Internet Control Message Protocol. The protocol used to handle errors and control messages at the IP layer. ICMP is actually part of the IP protocol.

IESG

Internet Engineering Steering Group. The executive committee of the IETF.

IETF

Internet Engineering Task Force. One of the task forces of the IAB. The IETF is responsible for solving short-term engineering needs of the Internet. It has over 40 Working Groups.

intranet

A company's private network.

Internet

The worldwide internet based on the TCP/IP protocol.

internet

A collection of networks interconnected by a set of routers, which allow them to function as a single, large virtual network.

internet address

A 32-bit address assigned to hosts using TCP/IP. See dotted decimal notation.

IP address

The internet address for a machine. It consists of four digits,

e.g. 196.7.8.90

IP datagram

The fundamental unit of information passed across the Internet. Contains source and destination addresses along with data and a number of fields which define such things as the length of the datagram, the header checksum, and flags to say whether the datagram can be (or has been) fragmented.

IP network

Any network connected to the Internet

IP node

Any node connected to the Internet.

IRTF

Internet Research Task Force. One of the task forces of the IAB. The group responsible for research and development of the Internet protocol suite.

message integrity

Making sure that the message contents have not been altered, deliberately or accidentally, during transmission

MILNET

MILitary NETwork. Originally part of the ARPANET, MILNET was partitioned in 1984 to make it possible for military installations to have reliable network service, while the ARPANET continued to be used for research.

multicast

A special form of broadcast where copies of the packet are delivered to only a subset of all possible destinations. See broadcast.

name resolution

The process of mapping a name into the corresponding address. See DNS.

NetBIOS

Network Basic Input Output System. The standard interface to networks on IBM PC and compatible systems.

NCP

Network Control Protocol. The protocol used by ARPANET.

non-repudiation

Making sure that a message, sent between two parties, cannot later be said to not occur by any party involved.

octet

A single digit of the IP address.

OSI

Open Systems Interconnection. An international standardization program to facilitate communications among computers from different manufacturers.

PCI

Protocol Control Information. The protocol information added by an OSI entity to the service data unit passed down from the layer above, all together forming a Protocol Data Unit (PDU).

PDU

Protocol Data Unit. This is OSI terminology for "packet." A PDU is a data object exchanged by protocol machines (entities) within a given layer. PDUs consist of both Protocol Control Information (PCI) and user data.

PSN

Packet Switched Node. The modern term used for nodes in the ARPANET and MILNET. These used to be called IMPs (Interface Message Processors).

plug-in

A special-purpose interpreter that allows audio files, video files etc. to be executed. A plug-in works as follows: when the Web browser starts to download a file of the plug-in's format, the plug-in will execute the instructions in the file.

ping

Packet internet groper. A program used to test reachability of destinations by sending them an ICMP echo request and waiting for a reply. The term is used as a verb: "Ping host X to see if it is up!"

port

The abstraction used by Internet transport protocols to distinguish among multiple simultaneous connections to a single destination host. See selector.

protocol

A formal description of messages to be exchanged and rules to be followed for two or more systems to exchange information.

proxy

The mechanism whereby one system "fronts for" another system in responding to protocol requests. Proxy systems are used in network management to avoid having to implement full protocol stacks in simple devices, such as modems.

RARP

Reverse Address Resolution Protocol. The Internet protocol a diskless host uses to find its Internet address at startup. RARP maps a physical (hardware) address to an Internet address. See ARP.

repeater

A device which propagates electrical signals from one cable to another without making routing decisions or providing packet filtering. In OSI terminology, a repeater is a Physical Layer intermediate system. See bridge and router.

ROSE

Remote Operations Service Element. A lightweight RPC protocol used in OSI Message Handling, Directory, and Network Management application protocols.

router

A system responsible for making decisions about which of several paths network (or Internet) traffic will follow. To do this it uses a routing protocol to gain information about the network, and

algorithms to choose the best route based on several criteria known as "routing metrics." In OSI terminology, a router is a Network Layer intermediate system. See gateway, bridge and repeater.

RPC

Remote Procedure Call. An easy and popular paradigm for implementing the client-server model of distributed computing. A request is sent to a remote system to execute a designated procedure, using arguments supplied, and the result returned to the caller. There are many variations and subtleties, resulting in a variety of different RPC protocols.

rsh

A program that allow you to log in to a remote host computer.

S-HTTP

a connectionless protocol that wraps messages in a secure digital envelope.

sniffing

Electronic eavesdropping.

spoofing

Electronic impersonation.

steganography

The art of hiding a secret message within a larger one in such a way that the adversary can not discern the presence or contents of the hidden message. For example, a message might be hidden within a picture by changing the low-order pixel bits to be the message bits.

TCP

Transmission Control Protocol. The major transport protocol in the Internet suite of protocols providing reliable, connection-oriented, full-duplex streams. Uses IP for delivery.

TCP/IP

Transfer Control Protocol/Internet Protocol. The language used by Internet nodes

telnet

The virtual terminal protocol in the Internet suite of protocols. Allows users of one host to log into a remote host and interact as normal terminal users of that host.

three-way-handshake

The process whereby two protocol entities synchronize during connection establishment.

transceiver

Transmitter-receiver. The physical device that connects a host interface to a local area network, such as Ethernet. Ethernet transceivers contain electronics that apply signals to the cable and sense collisions.

trojan horse

A program whose purpose is well defined, but also contains a backdoor or attack program.

UDP

User Datagram Protocol. A transport protocol in the Internet suite of protocols. UDP, like TCP, uses IP for delivery; however, unlike TCP, UDP provides for exchange of datagrams without acknowledgements or guaranteed delivery.

URL

Uniform Resource Locator. The address of a file or directory on the Web.

worm

A program which periodically performs a malicious and possibly destructive activity.

Bibliography

- **"A TCP/IP Tutorial: Behind the Internet (Part One of Two)"**, Phrack Magazine, Vol. 3, Issue 33.
- **"A TCP/IP Tutorial: Behind the Internet (Part Two of Two)"**, Phrack Magazine, Vol. 3, Issue 34.
- **"An introduction to Internet Protocols (Part One of Two)"**, Phrack Magazine, Vol. 3, Issue 28.
- **"An introduction to Internet Protocols (Part Two of Two)"**, Phrack Magazine, Vol. 3, Issue 29.
- **"An introduction to Packet Switched Networks"**, Phrack Magazine, Vol. 2, Issue 18.
- Arnold, N. **"Unix Security: A practical Tutorial"**, McGraw-Hill Inc., 1993.
- Bellovin, Steven M **"Trends in Internet Security"**, AT&T Bell Laboratories.
- Berners-Lee, Tim **"World Wide Computer"**, Communications of the ACM, February 1997, Vol. 40, No. 2, 57-58.
- Braun, David **"Encryption Stalemate threatens E-Commerce, National Security"**.
- Byte Magazine, June 1997.
- Comer, Douglas **"Internetworking with TCP/IP Vol. 1: Principles, Protocols and Architecture"**
- Cavalli, Dr. Alexander **"Electronic Commerce and the Internet: Building a New Paradigm for Business"**.
- Cavalli, Dr. Alexander **"Electronic Commerce over the Internet and the Increasing Need for Security"**, Dec. 8, 1995.
- **"CGI Security Holes"**, Phrack Magazine, Vol. 7, Issue 49.
- Denning, Dorothy **"Crime and Crypto on the Information Highway"**, Journal of Criminal Justice Education, Spring 1995.
- Denning, Dorothy **"Data Security and Cryptography"**.
- Denning, Dorothy **"Encryption Policy and Market Trends"**, May 1997.
- Denning, Dorothy **"The United States vs. Craig Neidorf : A debate on electronic publishing, constitutional rights and**

hacking?", Communications of the ACM, March 1991, Vol. 34, No. 3, 24-32.

- **"Doing Business on the Internet"**, ORACLE Magazine, Jan./Feb. 1996, Vol. X, No. 1, 13-15.
- **"Don't let the Web pass you by"**, Computerworld, Sept. 1, 1997, Vol. 31, No. 35, 8.
- Downey, Jeff, **"Denial of Service Attacks: Can't Say No"**, PC Magazine, April 21, 1998, Vol. 17 No. 8, 203-204.
- **"E-Commerce brings applause, some fears"**, Computerworld, Sept. 1, 1997, Vol. 31, No. 35, 28.
- **"E-Commerce for the 21st Century"**, IEEE Computer, Vol. 30, No. 5, May 1997, Pg. 44-47.
- **"Electronic Commerce"**, ORACLE Magazine, July/August 1996, Vol. X, No. 6, 46-47.
- Easton, Jaclyn **"Is it safe to shop On-line"**.
- Farmer, Dan **"COPS and Robbers: UN*X System Security"**, 1989.
- Felten, Edward W., Dirk Balfanz, Drew Dean, and Dan S. Wallach **"Web Spoofing: An Internet Con Game"**, Feb 1997, Technical Report 540-96, Department of Computer Science, Princeton University.
- **"File Descriptor Hijacking"**, Phrack Magazine, Vol. 7, Issue 51.
- Finn, Bob **"Embezzlement in Small to Medium Sized Agencies"**, April 2, 1997.
- Federal Information Processing Standards Publication 186 - DIGITAL SIGNATURE STANDARD (DSS), May 19, 1994, http://bilbo.isu.edu/security/isl/fips186.html.
- Fulton, Sean **"Taking Stock of your Web-server software"**, InternetWorld, Nov. 1997, 64-70.
- Ghosh, Anup K. **"E-Commerce Security: Weak Links, Best Defenses"**, Published by John Wiley & Sons, Inc.1998
- Grayson, Ian **"It takes a Digital Detective to track down today's Computerised Criminals"**, September 1997.
- Heath, Jim **"How electronic encryption works and how it will change your business"**
- http://www.eff.org/descracker/
- http://www.rsa.com
- http://www-08.nist.gov/nistpubs/800-7/node211.html
- **"IP-spoofing Demystified"**, Phrack Magazine, Vol. 7, Issue 48.

- Irving, Larry **"The Risks and Rewards of Electronic Commerce (Information technology expands business)".**
- Jamison, John and Randy Nicklas, Greg Miller, Kevin Thompson, Rick Wilder, Laura Cunningham, Chuck Song **"vBNS: not your father's Internet"**, IEEE Spectrum, July 1998, Pg. 38-46.
- Janson, Phil **"Internet keyed Payment Protocol"**, June 1995.
- Jeffsmith, H. **"Privacy Policies and Practices : Inside the Organizational Maze"**, Communications of the ACM, Vol. 36, No. 12, 105-121.
- Johnson, Anna **"Companies Losing Millions over Rising Computer Crime".**
- Jones, Susan B. and Cedelia d'Oliveira **"Computer Security breached".**
- Lawton, George **"The Internet's Challenge to Privacy"**, IEEE Computer, Vol. 31, No. 6, June 1998, Pg. 16-18.
- Lodin, Steven W., Christoph L. Schuba **"Firewalls fend off invasions from the Net"**, IEE Spectrum, Vol. 35, No. 2, February 1998, Pg. 26-34.
- Kleinrock, Leonard and Daniel C. Lynch, Jon Postel, Lawrence G. Roberts, Stephen S. Wolff **"The Past and Future History of the INTERNET: the science of future technology"** Communications of the ACM, Vol. 40, No. 2, February 1997.
- Mehler, Mark **"Electronic Commerce: A marketing Revolution. The Case for Electronic Commerce".**
- Mitton, Jon-Paul **"Web Security Certification prevents rash of Banking Security Breaches"**, http://www.CSCI.ca/
- Mueller, Robert **"Electronic Commerce: A marketing Revolution *Risks and Rewards*".** http://www.CSCI.ca/
- Murry, David **"Internet Banking and Commerce: Security".**
- ORACLE Magazine, Vol. XII, No. 1, Jan./Feb. 1998.
- **"PC Application Level Security"**, Phrack Magazine, Vol. 7, Issue 50.
- **"Project Hades"**, Phrack Magazine, Vol. 7, Issue 49.
- **"Risks of ActiveX"**, The Inside Running, Feb. 1997.
- Risks-Forum Digest, April 1998, Vol. 19, Issue 66.
- Risks-Forum Digest, Dec 24, 1997, Vol. 19, Issue 52.
- Risks-Forum Digest, Dec. 9, 1997, Vol. 19, Issue 49.
- Risks-Forum Digest, Nov. 26, 1997, Vol. 19, Issue 47.

- Risks-Forum Digest, Nov. 17, 1997, Vol. 19, Issue 46.
- Risks-Forum Digest, Apr. 4, 1997, Vol. 19, Issue 04.
- Risks-Forum Digest, Nov. 11, 1997, Vol. 19, Issue 45.
- Risks-Forum Digest, Nov. 1, 1997, Vol. 19, Issue 02.
- Risks-Forum Digest, Jan. 6, 1998, Vol. 19, Issue 53.
- Risks-Forum Digest, April 2, 1998, Vol. 19, Issue 65.
- Rivest, Ronald L. **"Chaffing and Winnowing: Confidentiality without Encryption"**, April 24, 1998. http://theory.lcs.mit.edu/~rivest/chaffing.txt
- Samuelson, Pamela **"Is information property?"**, Communications of the ACM, March 1991, Vol. 34, No. 3, 15-18.
- Smith, Danny **"Enhancing Security of UNIX systems"**.
- Snapp, Steven R. and James Brentano, Gihan V. Dias, Terrance L. Goan, L. Todd Heberlein, Che-lin Ho, Karl N. Levitt, Biswanath Mukherjee, Tim Grance, Douglass L. Mansur, Kenneth L. Pon, Stephen E. Smaha **"Intrusion Detection Systems (IDS): A survey of existing systems and a proposed distributed IDS architecture"**, July 1991.
- **"Spin Your Own Web"**, ORACLE Magazine, Jan./Feb. 1996, Vol. X, No. 1, 44-45.
- Sdrs, Camillo **"Encryption and Strong Authentication for Electronic Commerce"**, Abstract, Helsinki University of Technology.
- Stevens **Unix Network Programming**.
- Stoll, Clifford **"Stalking the Wily Hacker"**, Communications of the ACM, Vol. 31, No. 5, 484-497.
- Synder, Joel **"The trouble with Java"**, InternetWorld, Nov. 1997, 36-38.
- **"Technical Guide to Digital Certification"**, Phrack Magazine, Vol., Issue 52, January 26, 1998.
- **"The Internet: An Overnight Star?"** ORACLE Magazine, Jan./Feb. 1996, Vol. X, No. 1, 41-42.
- **"The Internet's Explosive Growth"**, PC Magazine, Vol. 14, No. 9, May 16, 1995.
- **"TTY Spoofing"**, Phrack Magazine, Vol. 4, Issue 41.
- Update on Network "Sniffing" Security Vulnerabilities, NASIRC Bulletin #94-10.
- Van Vleck, Tom **"The Risks of Electronic Communication"**.

- Walker, Sophie **"Web sites open companies to computer fraud risk"**, October 30, 1997.
- Walrand, Jean **"Communications Networks: A First Course"**, 1991.
- Wilcox Enterprises **"Security on the Web"**.
- Wilson, Janet **"The IETF: Laying the Net's Asphalt"**, Computer, August 1998, 116-117.