Security Issues in Networks with Internet Access

CARL E. LANDWEHR, MEMBER, IEEE, AND DAVID M. GOLDSCHLAG [1]

Invited Paper

This paper describes the basic principles of designing and administering a relatively secure network. The principles are illustrated by describing the security issues a hypothetical company faces as the networks that support its operations evolve from strictly private, through a mix of Internet and private nets, to a final state in which the Internet is fully integrated into its operations, and the company participates in international electronic commerce. At each stage, the vulnerabilities and threats that the company faces, the countermeasures that it considers, and the residual risk the company accepts are noted. Network security policy and services are discussed, and a description of Internet architecture and vulnerabilities provides additional technical detail underlying the scenario. Finally, a number of building blocks for secure networks are presented that can mitigate some of the vulnerabilities.

Keywords: *computer network security, internet, cryptography, authentication*

I. INTRODUCTION

Historically, people developed networks so that they could share expensive computing resources. Today, people buy cheap computers in order to gain access to networks. The efficiency of e-mail for conducting internal business drives companies to build extensive local area networks, and demand for access to the Internet soon follows. In many cases, access to the Internet may already have been arranged for particular machines that are not simultaneously attached to the in-house network. But company managers read the newspapers and are cautious about connecting. Should they be? More generally, what security issues need to be considered when one network is connected to another?

Securing a system always means considering vulnerabilities, threats, countermeasures, and acceptable risk. No system will be perfectly secure, particularly if security is deemed to include continuity of service. Simply shutting down a system in the face of an attack, particularly in the case of a network, is rarely a viable approach from the system owner's point of view, but maintaining the availability of data that have been corrupted or compromised is also of dubious value. The security engineer seeks system designs that yield reasonably secure operation in the anticipated threat environment.

We first outline some principles for designing and administering a relatively secure network. Next, we describe the evolution of a hypothetical company's networks together with the vulnerabilities, threats, countermeasures, and risks the company faces and the choices it makes. In Section 4, we discuss network security policies and services more broadly and relate them to the example. For those wishing more technical detail, we summarize Internet architecture, vulnerabilities, and building blocks that support both connectivity and security in Sections 5 and 6. Section 7 summarizes and concludes the paper.

II. PRINCIPLES FOR SECURE NETWORKING

We define a computer network to include the network infrastructure as well as attached devices. These devices include computers, file servers, printers, scanners, modems, and more. In this context, there are several sets of questions to consider:

1) What are we trying to protect? Is it the secrecy or integrity of data accessible over the network, the use of those network resources, or the availability of data and network resources? Different applications will have different concerns. In this paper, we focus most strongly on protecting the confidentiality and integrity of data dispersed throughout the network, because there are usually already strong commercial incentives for keeping the system available.
2) What operations are authorized? Authorization policies may be motivated both by the trust placed in the source of an operation and the consequences of unauthorized operations. A trusted source, such as a bank teller working at a console, may be authorized to invoke many operations. Untrusted sources, perhaps customers at automated teller machines, may have fewer operations available.

[1] Authors addresses: Carl E. Landwehr, Naval Research Laboratory, Code 5542, Washington, DC 20375-5337, Voice: (202)767-3381, Fax:(202)404-7942, E-mail: Landwehr@itd.nrl.navy.mil. David M. Goldschlag, Divx, 570 Herndon Parkway, Herndon, VA 20170, Voice:(703)708-4028, Fax: (703)708-4088, E-mail: David.Goldschlag@divx.com. This work was performed while both authors were with the Naval Research Laboratory.

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

3) What is the security perimeter? Often, a boundary of trust is placed between machines inside a corporate network and those outside. Machines inside the corporate network are trusted, and machines outside the network are not. Firewalls implement this sort of perimeter security and are a prudent component of network security solutions.

The technical approach we advocate is to control incoming connections to trusted machines and protect the data on those machines. Incoming data must be interpreted within a restricted environment. Arriving e-mail, for example, should not be able to invoke operating system (OS) operations outside of mail delivery. And incoming data should not be trusted until some estimate of its trustworthiness is obtained (e.g., by running executables through a virus checking program). Data must be protected too, while stored on disk, while in transit over the network, and while being manipulated as part of a computation on a user's machine. Different kinds of protection may be applied in different places: data moving over the Internet between corporate sites may be protected cryptographically, while data moving over wires within a building may be protected by pressurized conduit.

Experience has revealed a number of principles that should guide designers and administrators when building a network that is to meet defined security requirements. These principles play a role both in planning and operating the network. Successful security may be boring --- its consequence is the absence of security problems. Yet a successful penetration, from the attacker's view, is one that goes undetected. So it is important to operate the system with vigilance.

· *Know the configuration of the network's hardware and software*. What hardware (computers, cables, bridges, routers, switches, modems, printers, etc.) and software is running on the network? What communication protocols are being used (AppleTalk, Ethernet, TCP/IP, IPX, SNA, DECNET, NFS, Kerberos, etc.) Who is using the system and from where? What is defined to be within the security perimeter and what is outside? Without knowing the particular configuration of the network, security administrators cannot identify the network's vulnerabilities, will not know which security patches to look for on an ongoing basis, and will be unable to detect unauthorized changes that may affect the network's security.

  Know the vulnerabilities of the configuration. The vulnerabilities represent weak points in the system that could be exploited, accidentally or intentionally, and cause security to be compromised. Determining the network's vulnerabilities requires considering both the network's normal and exceptional behavior. Although many vulnerabilities have been widely publicized ([ChBe 94], [FeMo 97]), detailed vulnerability information on newly discovered vulnerabilities is often closely guarded. Manufacturers and incident response centers like CERT [CERT], CIAC [CIAC], AUSCERT [AUSCERT], and other members of FIRST [FIRST] distribute vulnerability information regularly. Other public Web sites ([COAST], [W3C], [Stein], [NIH]) also provide helpful information and tools.

· *Know the threat and consider it in relation to the vulnerabilities to assess the risk*. The threat environment determines the kinds of attacks expected, and these usually depend strongly on the value of the resources contained in the system. When those resources are information, rather than material goods, their value can fluctuate with time and with external events, so the threat environment may change from day to day, and any analysis needs to recognize this fact. Is the company concerned about attacks from insiders or outsiders? Are attacks likely to come from vandals, disgruntled employees, thieves, or industrial spies seeking commercial gain? What kinds of users have access to the system? What information do they have access to? Assuming the attacker knows the system's vulnerabilities, where are attacks likely to occur? This information helps determine what countermeasures are needed. Knowledge of the vulnerabilities, threats, and countermeasures permits an assessment of the residual risk present in the network, which must be carefully studied to see if appropriate trade-offs among cost, function, and security have been made.

· *Authenticate and authorize access to network resources*. The network must authenticate a user before deciding whether the user is authorized to access a resource. These resources may include OS calls and filesystem access. The level of authentication may differ across applications: in a Web access, authentication may only identify from what Internet Protocol (IP) address a connection is being made. In a login, the user is authenticated also. Authentication may itself introduce vulnerabilities (if, for example passwords are relied on but the user chooses a weak password or the password file is stolen). Sessions may be authenticated as well (using, for example, Kerberos [SNS 88] [KPS 95], described in Section 5.3).

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85*, No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

- *Maintain audit logs*. The combination of authentication and audit helps a security administrator determine whether a security breach occurred, and where it came from. Logs must be carefully controlled, however: intruders may attempt to modify the logs to hide evidence; and logs may contain private information [GrMo 84].
- *Have a plan to detect and respond to security incidents*. This plan should include who should be informed, whether certain services should be (temporarily) shut down, how lost data is to be restored, how normal operation can be resumed, and so on. The plan should also consider how to coordinate with other security administrators, to help trace the source of attacks.
- *Ensure individual accountability*. A single individual should have overall responsibility for analysis, design, and implementation of security infrastructure (this is the individual, whom we will refer to as the security manager, who should be concerned about all of the previous principles in this list). Routine security should be made easy for individual users, but all users should also understand that they will be held individually accountable for the resources they control.

The next section shows how one hypothetical company might apply these principles and approaches as its networks grow from completely internal and private to fully integrated with the Internet.

III. The Networks of the TwoHands Corporation

The TwoHands Corporation develops, produces, and markets specialized gloves used in waste disposal and other safety-related applications. In addition to its headquarters, located in a large city, it operates a research laboratory near a large university, which develops new glove designs and coatings, and several factories and warehouses in different parts of the country. The gloves are sold through wholesale and retail outlets.

Automation at TwoHands began with the introduction of computers into the purchasing, accounting and payroll systems in the administrative headquarters building, years ago. Separately, word processing systems began to replace typewriters on desks throughout the organization. These systems typically shared a single printer or two among several workstations, and cables were run among offices to support this. Over time, PCs connected by an Ethernet replaced the specialized word processing system. Demand grew for direct access to the purchasing and accounting systems, and the mainframes that supported those functions were replaced by servers that communicated with clients that could be run from the many PCs.

Large orders for TwoHands products are taken by the sales force and transmitted to the headquarters, where they are checked against inventories and production schedules and dispatched to appropriate factories. TwoHands also receives orders by mail from a catalog it distributes.

Orders are filled from inventory and also contribute to decisions on which particular gloves to produce, and in what quantity. Production schedules in turn drive purchases of raw materials.

In the research laboratory, computers were introduced to perform scientific calculations. The company obtained this service first through a timesharing service bureau, accessed through dial-up telephone lines. Scientists initially used small computers to control experimental equipment. As these machines became more capable, scientists used them for scientific calculations and eventually for word processing as well, and the timesharing service was discontinued. As in the headquarters building, a communications cable plant, separate from the telephone system, was gradually put in place. With this scene in mind, we identify five stages of development in the TwoHands networks and consider the operations, threats, vulnerabilities, countermeasures, and risks in each. The following descriptions illustrate and represent the security considerations in the TwoHands environment; they are not intended to be comprehensive and complete.

A. Stage One: Private Networks Only

1) Mode of Operation

Customer orders arrive by telephone, fax, or postal mail and are entered into the order processing system. Office operations and word processing are handled with in-house servers and desktop clients (PCs) connected by an in-house LAN. A private network of lines leased from the telephone company connects the headquarters to the research laboratory, the factories, and the warehouses. Apart from these lines, there are no external connections to the LAN (or at least there are not supposed to be any). The headquarters, laboratories, and factories use E-mail over the private network to coordinate their activities and for other purposes, both formal and informal. This network is illustrated in

Figure 1.

2) Vulnerabilities

a) Unauthorized configuration of authorized software: An authorized user or administrator can configure authorized software incorrectly resulting in misdelivery of

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

messages or other problems. A user can configure the network interface on his computer so that all traffic that passes by on the LAN is available for browsing (packet sniffers). Passwords or sensitive corporate or personal information could be obtained in this way.

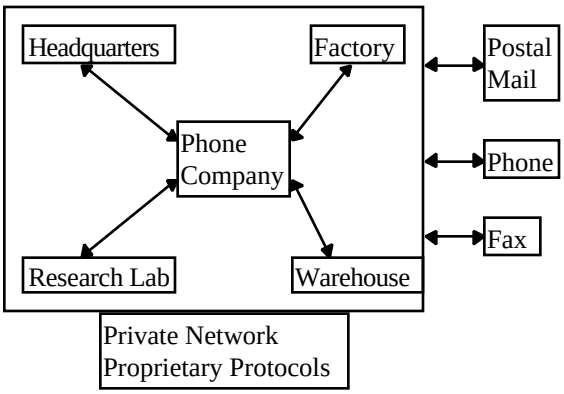

Figure 1: Stage One: Private Networks Only

a) Insertion of malicious software: In the TwoHands environment, client workstations have diskette drives, and users can load arbitrary software with them. Authorized users may inadvertently introduce malicious software (viruses or Trojan horses) through diskettes brought from home and used on the company system.

b) Unauthorized external connections: Users can attach modems to their PCs and hook them to the telephones on their desks. This opens a back door to the private LAN, if the computer is left running and the modem can answer the phone. Even if this is not the case, this connection provides another path for the user to download and install arbitrary software.

c) Unauthorized access to private network: An intruder might gain physical access to the private network lines and tap them or eavesdrop on microwave links that form part of the private net.

3) Threats

Authorized users (insiders) may do something foolish or risky, either because they don't understand the risk they are taking, or they decide that the risk is justified because taking it will help them do their job, or they don't care. A disgruntled insider might take such actions intentionally. Unauthorized users (outsiders) should not, in theory, be a problem, if the system has no external connections, but a determined opponent might gain physical access to the plant or search for unintended external connections. An outsider may also use "social engineering" to deceive insiders into taking insecure actions. Although the network perimeter here is in theory well defined, the trust placed in internal machines and users may not be appropriate. If an insider has attached a modem to a PC, an outsider who discovers its phone number may be able to masquerade as an insider.

4) Countermeasures

Passwords can be used to control access to individual clients and servers and provide some degree of accountability. Virus detection software can be installed on clients and servers. There are also packages available that will monitor changes to software configurations on a group of systems attached to a LAN. These usually depend on a program running on the client to communicate changes to a monitor running on the server. It is possible to disable the diskette drives on the clients and to install tamper-resistant hardware to prevent changes to the client (and server) software and hardware configurations. Security awareness training for the authorized users, so they understand how their actions can affect system security, is essential; the security manager needs to arrange the training.

5) Risks

Permitting diskette drives and off-the-shelf hardware and software leaves the systems open to the risks of users (or any personnel with physical access to the machines) installing additional, possibly malicious software or hardware.

6) What Happened

Trusting its (newly security-aware) employees and not expecting significant outside attacks, TwoHands decided not to disable diskette drives or install tamper-resistant hardware. In order to provide some degree of protection against the installation of unlicensed copies of commercial programs, it installed a configuration monitoring program. Some users did in fact install games and other software, and eventually several clients were infected with a virus. After eradicating the virus with some difficulty, TwoHands purchased virus checking software and required its installation on all clients. It also reassessed its backup procedures.

B. Stage Two: Private Networks Plus Read-Only Web Server

TwoHands decides to increase the visibility of its products by putting descriptions, pictures, and order information for its most popular products on the World Wide Web. Orders will still be accepted in the same ways as before (postal mail, phone, fax).

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

1) Mode of Operation

Operations are unchanged, except that a Web site, not connected to the internal networks, is established with a commercial Internet Service Provider (ISP). Customers can review information on the Web site, including product specifications, new product announcements, company telephone numbers, and so on. A system administrator is placed in charge of the Web site; the administrator is responsible for installing new information, reporting on sources of queries and so on. This network is illustrated in Figure 2:

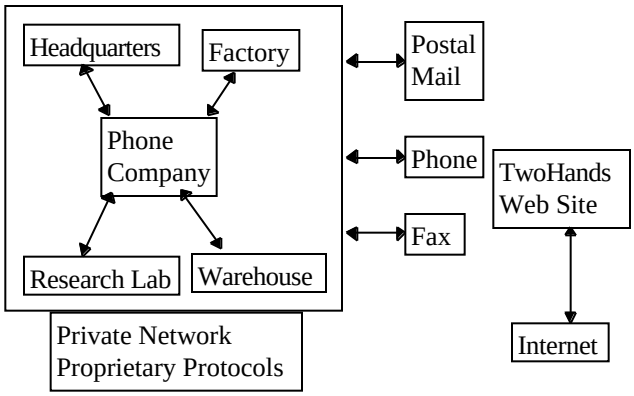

Figure 2: Stage Two: Private Networks Plus Read-Only Web Server

2) Vulnerabilities

a) Internet-based denial of service: The Web server, as long at it is only providing catalog information, is vulnerable to denial of service attacks that originate in the Internet (e.g. an Internet administrator from outside the company who misconfigures a router so that traffic destined for TwoHands' Web site is sent to a "black hole").

b) Unauthorized changes to data on Web server: If there is a bug in the web server that can be invoked by Internet users (e.g., by sending URLs so long that they overflow a buffer and corrupt internal software), or if the protections of the ISP's Web server are penetrated in other ways, unauthorized users may be able to modify TwoHands catalog information. The results are unpredictable; wrong information could be placed in the catalog, or pointers could be inserted that lead to bogus Web pages.

c) Web address spoofing: Some vulnerabilities come from operating in the World Wide Web environment that are not technical in the same sense but are nonetheless real. For example, competitors or vandals could set up Web sites with similar names (e.g., in addition to the correct website, `http://www.TwoHands.com`, they could establish `http://www.TwoHands.org` or `http://www.ToHands.com`) and place bogus information on them.

3) Threats

To the internal systems, the threats are unchanged. The new Web site, however, is accessible to anyone on the Internet, including people who make mistakes, as well as vandals (e.g., individuals who intentionally saturate TwoHands' Internet Service Provider with bogus traffic and prevent access by legitimate users).

4) Countermeasures

The primary countermeasures are to run a reliable Web server package and keep abreast of advisories provided by organizations such as the Computer Emergency Response Team (CERT) [CERT], AUSCERT [AUSCERT], or other members of the Forum of Incident Response Teams (FIRST [FIRST]) as new flaws or attacks are discovered and distributed. Access to the server should be limited to the administrator, and the server host should provide only those capabilities it requires to act as a Web server. It should not provide a general execution environment, and any development activities should be conducted off-line. Backups of all information on the server should also be maintained off-line. A good system administrator, (i.e., one who carefully maintains the system configuration, understands its vulnerabilities, monitors it for possible intrusions, and installs patches and updates as they are distributed) is a key countermeasure.

5) Risks

TwoHands' internal systems are no more at risk than in the previous mode.

It runs the risk of denial of service or vandalism to its Web-based system. The information the server holds is by definition public, so confidentiality is not an issue. Novel attacks seem to surface in the Internet from time to time that even the best system administrators have not have anticipated or are powerless to repel. An example is the SYN-flooding denial of service attack that was mounted several times in 1996 before effective countermeasures were developed ([CA21- 96], [SKKSSZ 97]). If the Web site becomes the company's primary method for attracting sales, this could nevertheless be a significant risk. A company in a highly seasonal business (e.g., a florist whose site is down for a day in the week before Valentine's Day) might be more concerned.

6) What Happened

Initially, the Web site attracted no attention, but TwoHands began to advertise the site in print and to direct

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

queries to it, particularly from new customers, and the site's use grew over time. The ISP proved generally reliable, but on one occasion, for no apparent reason, the ISP was attacked by outsiders and service was unavailable for several hours. Several times data on the Web server were modified, but TwoHands' administrator could not detect the means of attack. Correct data values were reloaded from backups.

C. Stage Three: Private Nets Plus Forms-Based WWW Server

Finding that the on-line catalog is attracting some attention, TwoHands decides to begin taking orders via HTML forms. This entails implementing CGI scripts on the existing Web server and providing a path for the orders placed to enter the private network.

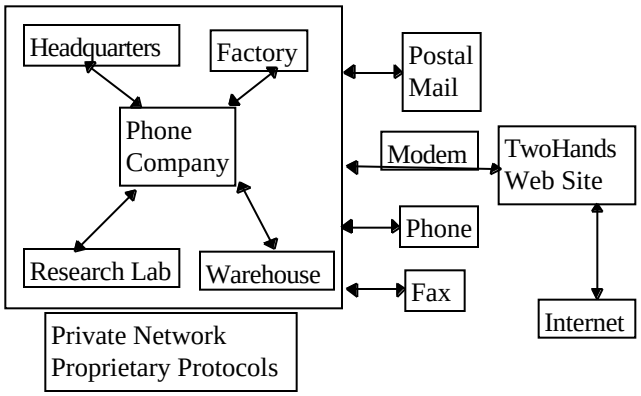

Figure 3: Stage Three: Private Nets Plus Forms-Based WWW Server

1) Mode of Operation

The existing Web site is expanded substantially to support the anticipated additional processing required to handle forms-based requests. Scripts are developed, tested, and installed. Each form submitted creates an e-mail message, which is posted to a separate account with the commercial ISP. The order-taking staff periodically reads the e-mail, accessing the ISP via modems attached to their desk-top clients. This network is illustrated in Figure 3.

2) Vulnerabilities

a) Compromise of order information: Unless specific measures are implemented to authenticate customers and protect their communications with the server and the e-mail that the server sends to the order-taking staff, the order information may be compromised. If this information is sensitive (e.g., it includes credit card numbers), this vulnerability is significant.

b) Spoofing of order information: Outsiders may be able to forge e-mail that seems to have come from the Web site but did not (and thereby, for example, flood the order-taking office with false requests).

c) Loss of control of Web site: Valid, but unsafe scripts [CGI] can permit an outsider to "take over" a Web site and thereby alter the information it provides, generate false requests, close it down, or initiate other malicious actions. Since TwoHands is using a commercial ISP, they may be vulnerable to weaknesses not only in their own scripts but also to those in scripts provided by other clients of the same ISP. This was in fact a vulnerability in stage two as well, but the consequences were less severe. If the e-mail generated by CGI scripts is delivered to a server different from the web site, the vulnerabilities of that site are also a concern.

3) Threats

In addition to the threats already listed, outsiders may eavesdrop on the orders placed, may place bogus orders, or may gain control of the web server.

4) Countermeasures

A secure Web server (i.e. one that implements SSL [FKK 96] or SHTTP [ReSc 97] protocols [IETF]) can be used, and orders can be restricted to those originating from Web clients that converse in a secure mode (i.e., information passed over the link is encrypted). Encryption is not a panacea, but its use here would increase the level of effort required for an outside snooper to eavesdrop successfully. Before scripts are installed, they must be checked by a knowledgeable system administrator to be sure they will not make the server vulnerable to well-known attacks. Vigilance on the part of the order-taking staff is also a useful countermeasure; they should be alert for unusual patterns of orders (or a lack of orders) that might indicate tampering with the Web server or phony orders being placed. Depending on the size and volume of orders the company has, it may be reasonable to simply call or send e-mail back to the individual placing the order for confirmation. This makes it harder for a spoofer to succeed, since in addition to generating the initial false order, she would have to intercept the confirmation message and reply to it.

5) Risks

The increased dependence on Web technology brings some increased risks. Establishing the Web site and assuring the scripts are safe requires some expertise. As the scripts and the system on which they are executed change, good system administration is needed to assure that flaws are not introduced and that any detected flaws are

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

removed. There is consequently some risk that occasional problems will occur. For TwoHands, compromise of the identities of their customers or even of order information may not be a major concern, but loss of, say, a file containing customer credit card numbers might well damage TwoHands reputation significantly. If a commercial ISP is used for the Web server and for e-mail, the ISP itself is trusted not to misbehave (just as the telephone company is trusted in the pre-Internet operations). If customers grow to depend on the Web-based order process and a denial of service attack is mounted, business may be lost. One fundamental issue here is that TwoHands is storing sensitive customer information (credit card numbers and orders) on a machine that TwoHands itself would not trust to store company proprietary information.

6) What Happened

TwoHands designed forms that permitted taking complete orders, including credit card numbers, so it moved to a Web server that supported SSL. It took care with its forms and scripts, and the system worked well. E-mail was delivered to the same site that hosted the Web pages. However, another company that maintained Web pages on the same server provided by the ISP was less careful and was vulnerable to an attack through a script. When that attack succeeded, the outsiders who initiated it gained access to all information on the server, including that collected by the TwoHands forms. The credit card numbers (which, although they were encrypted for transmission, were stored unencrypted on the server) could have been stolen, though it could not be determined whether this actually occurred. To protect its customers, TwoHands had to notify them of the problem.

D.Stage Four: Internet Service through Firewall with Forms-Based WWW Server

As TwoHands' communications with the outside world change, their internal networks, though still isolated, keep pace. The increasing popularity and decreasing cost of TCP/IP networks leads TwoHands to organize its leased lines as a corporate intranet to ease the integration of its office and laboratory systems, to make internal and external communication more uniform, and to reduce costs. After the incident with the ISP provider, TwoHands decides to establish its own connectivity to the Internet to gain better control of its sensitive information. A commercial firewall is purchased to support connectivity to the Internet from the corporate intranet, and the Web server is configured on a separate machine outside the firewall.

1) Mode of Operation

The processing of orders in the new architecture works much as it did before, except that now the orders are e-mailed directly to order-takers' accounts on servers attached to the corporate intranet instead of to an external system. It is no longer necessary for these desktop clients to have modems or phone lines attached, since they can send e-mail to Internet addresses directly from the local network. Internally, Web servers are used to front-end the inventory control systems so order takers can check on stocks and place orders using a forms-based interface similar to that used by customers. In addition, TwoHands' executives begin to use Internet e-mail to coordinate relationships with other companies. This is illustrated in Figure 4:

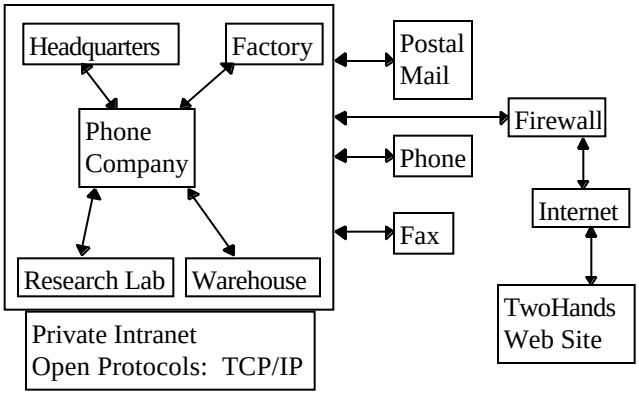

Figure 4: Stage Four: Internet Service through Firewall with Forms-Based WWW Server

2) Vulnerabilities

a) Attacks on internal systems via misconfigured firewalls: Although it is supposed to be a security-enforcing box, the firewall also presents new vulnerabilities. Without the firewall, the internal networks are isolated from the Internet. If the firewall is carefully configured, it can provide good protection from outsiders, but if not, it may only provide a false sense of security, and may permit outsiders to attack internal systems. The basic mechanisms used by firewalls vary, but they range from filtering packets (i.e., simply refusing to deliver packets that are suspicious based only on their address information), to parsing the contents of packet sequences to try to determine whether they conform to expected behavior for a given service, to supporting proxy services for applications so that only well-understood commands, rather than raw data, traverse the firewall. Configuring a firewall correctly for a given installation requires a clear understanding of a good deal of Internet infrastructure (see Section 5). An inadequately configured firewall can, for

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

example, make internal hosts visible to the outside world and therefore vulnerable to attack, it may pass traffic from untrusted hosts and ports that was supposed to be blocked, or it may provide an incorrect proxy server that lets malicious traffic into the internal network.

b) Importing of malicious software or exporting of sensitive information via properly configured firewall: The firewall, even if properly configured, provides an avenue through which insiders (or malicious software inadvertently invoked by insiders) can leak information or import malicious code [FeMo 97][Bell 97]. In fact, "fire-door" might be a more apt name for these devices, since their purpose is to be at least part-way open most of the time. The firewall also provides a path for leakage of information through covert channels [Lamp 73]: malicious software inside the trust boundary could, for example vary the spacing within e-mail messages to communicate TwoHands' projected earnings for the next quarter to a devious investor.

c) Compromise of identity of correspondents: Even if order information is encrypted, it will be possible for observers in the network to detect that traffic is flowing between TwoHands and its customers. This is not ordinarily a concern for TwoHands, but the e-mail traffic with executives of other companies might raise this issue if, for example, TwoHands were forming a strategic alliance with another company for business purposes and it did not want this fact revealed prematurely.

d) Back doors into internal networks: Frequently system administrators themselves find it convenient to install modems directly on host systems inside the firewall, so that they can work remotely. Users may make similar arrangements if they find traversing the firewall too cumbersome. Any such links can render the entire internal network vulnerable to outside attacks.

3) Threats

As in stage three, but now outsiders have the opportunity to try to penetrate the firewall and gain access to the corporate intranet. The threat to the corporate Web site is reduced in that it is no longer shared with other companies, but TwoHands also faces a greater responsibility for managing it correctly and securely. In addition, if the firewall is configured with HTTP proxies, inside users will have much greater online access to information from outside the company, including software and documents (and therefore viruses). Even if all orders and e-mail messages are encrypted enroute and handled successfully at the Web site, it will still be possible for outsiders to observe which sites are exchanging messages with TwoHands.

4) Countermeasures

The firewall itself is a countermeasure; without it, connection to the Internet would be even more risky. Vigilant management of the firewall and the Web site are necessary as well. Users of the internal networks must also be vigilant, since each could inadvertently damage the security of the system, either by importing dangerous materials or by creating a back door connection to outside networks.

5) Risks

As noted, there is a risk that insiders may release sensitive information to the Internet or import malicious software that might damage internal operations or itself cause sensitive information to be released. In addition to the risk that the firewall is misconfigured so that an outsider could gain control of it, flaws may be discovered in the firewall itself, and these could be exploited by an outsider. Insiders may also set up "tunnels" through the firewall for legitimate purposes but that damage security. One of the ways a firewall protects systems from the Internet is to filter out certain kinds of packets. A tunnel encapsulates a message from one protocol in a second protocol and uses the facilities of the second protocol to traverse some part of the network. The encapsulation is stripped off at the destination and the original packet reappears. This facility is useful in many contexts, but it can also be used to defeat the protection a firewall attempts to provide. See Section 5 for further discussion. This stage in TwoHands' evolution decreases perimeter protection by connecting internal and external machines without changing the behavior of internal machines and users. This approach weakens one part of its security posture without compensating in another.

6) What Happened

TwoHands made the transition to the new form of operation successfully and operated without incident for some time. After a time, management began to find unaccounted for software on many desktop clients and grew concerned that unlicensed software and in some cases undesired software was running on corporate assets and could cause legal problems. Much of the software appeared to have been downloaded from Web sites outside the company. An incident in which a "macro" virus (i.e., one that operates in the context of an application program, such as Microsoft Word) caused significant lost time brought matters to a head, and TwoHands developed a new corporate policy on configuration management and the use of the World Wide Web.

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

E.Stage Five: Virtual Private Networks

After successfully managing its own Internet connection for the corporate Web site and unifying some of the communication protocols used for both intra-company and Internet access, TwoHands decides to use the Internet as a Virtual Private Network (VPN) to reduce leased line charges and make its network connectivity more flexible. The VPN consists of an Internet-connected firewall at each corporate site that encrypts traffic destined for any of the other sites and delivers it, using the Internet as the transmission medium. The network is not private, in the sense that it contains no leased private lines, but it is virtually private in that only the headers containing routing information are exposed to public view. At the same time, TwoHands begins to explore more effective use of electronic commerce.

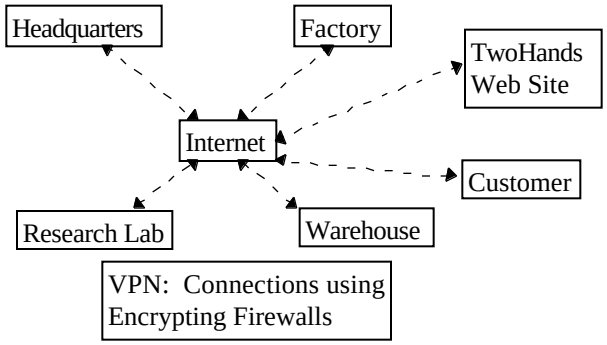

Figure 5: Stage Five: Virtual Private Networks

1) Mode of Operation

Rather than leasing lines to carry data between TwoHands corporate sites, each site manages its own Internet connection and runs an encrypting firewall. Communication between corporate sites is now carried over the Internet and is encrypted between these firewalls to keep the communication confidential. To save on long distance phone charges, faxes sent between corporate sites are sent as data messages over the VPN and then printed at the remote fax machine.

TwoHands also begins to unify the applications visible to customers on its Web site with its internal applications. Inventory changes, new product introductions, and discontinued products are automatically replicated to the Web-based catalog. The inventory systems of large customers and retailers are integrated with TwoHands' ordering system to place orders automatically in response to low inventories. This requires, among other things, compatible cryptographic systems at TwoHands and these customers. This network is illustrated in Figure 5.

2) Vulnerabilities

a) Contamination of internal databases via actions of cooperating companies: The integration of customers' inventory systems with TwoHands' ordering system exposes internal processes to the outside.

b) Service delays/denial for intranet traffic due to Internet/ISP problems: Although communication over the Internet is less costly than leased lines, it does not guarantee service or bandwidth. Depending on the ISP and the larger Internet backbone, delays may occur in delivering important messages.

c) Compromise or delay of intranet traffic due to crypto mismanagement: Both the encryption algorithms and key management techniques used in the encrypting firewalls present possible vulnerabilities to the VPN. The firewalls require manual entry of keys, and mistakes in rekeying can close down communications until the key is entered properly.

d) Compromise of sensitive internal information via interception and decryption of properly managed cryptography: The public Internet now carries confidential corporate information (though in encrypted form). Although in theory this is no different than using leased lines from the public phone system, the company must carefully select and configure the cryptography in the encrypting firewalls used in VPNs.

e) Compromise of internal traffic patterns: Even with encryption, outsiders may now infer when two of TwoHands' own corporate sites are communicating.

3) Threats

The threats are essentially the same as in stage four, but now more internal corporate information is flowing over the Internet, albeit in encrypted form, so somewhat more information is open to interception. A more significant threat might come from employees of other firms whose systems are now more tightly coupled with TwoHands' systems. Accidental or malicious acts on their part might trigger incorrect orders or production schedules to be created at TwoHands.

The automatic update of Web-based catalogs requires more communication between the public Web server and internal inventory systems and makes the Web server a more desirable target. Also, since large customers' needs now generate orders automatically, it is crucial that those systems function correctly.

4) Countermeasures

Additional training in the operation of the encrypting firewalls, particularly in dealing with their cryptographic

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

functions, can be provided to the system administrators at each facility. The encrypting firewall can be configured with strong cryptography, if all corporate sites are in a country that permits its use. International operations in this mode may not be feasible at present unless special export licenses are obtained. By using the same company as its ISP at all sites, TwoHands may be able to negotiate a guarantee of adequate capacity for communication within its network.

5) Risks

First, relying on the Internet for communication and on a single ISP is, at present, riskier than relying on a single phone company. Second, even the strongest cryptography is vulnerable to attack, and long term keys must be carefully protected.

Finally, even though the shared keys in TwoHands' and customers' ordering systems adequately authenticate each company and protect the confidentiality and integrity of their communications, a software malfunction of benign or malicious origin can still place incorrect orders or process orders incorrectly.

6) What Happened

Overall the restructured system works well and long distance telephone charges drop considerably. The integration of ordering, inventory, and production systems makes production more efficient. There are some snags, however. The single ISP supporting TwoHands VPN has several days of poor service as it upgrades its network. A large customer's ordering software malfunctions and automatically places a gigantic order that is only intercepted by an alert plant supervisor.

F. Summary: TwoHands Network Security Through the Stages

In stage one, virus checkers and configuration monitors help assure that undesired software is not present; password controls on desktop clients support the accountability and authorization needs. At stage two, outsiders are provided with legitimate access to a corporate system (the Web site) but that site is segregated from the internal systems and is operated as a read-only public site. Stage three introduces the storing of credit card numbers on the ISP-supported Web site, at which point a resource directly accessible to outsiders holds sensitive data. The accountability for securing CGI scripts on the ISP's server is unclear, and this contributes to the security failure. TwoHands addresses this problem in stage four by removing its dependence on the ISP, but it also introduces a connection between its internal networks and the Internet via a firewall. This makes it easier for employees to download software and in the end causes the company to attempt more stringent configuration control. It also makes the internal corporate networks vulnerable to Internet-based attacks for the first time, though the firewall blocks some attacks and significantly reduces the reduces the chance others will succeed. In stage five, additional responsibilities are placed on the administrators of the encrypting firewalls, since failures there may either deny service or disclose corporate data. In addition, the linking of the TwoHands production and inventory systems with those of its customers and suppliers introduces new dependencies, but without clear accountability, and almost leads to a disaster.

IV. Network Security Policy and Services

The description of TwoHands' networks above does not explicitly address security policy, but for any level of network security to be established and maintained, managers and users -- all affected parties -- must clearly understand what assets they need to protect against what threats and to what degree [Land 97]. While it should be easy to establish who owns a particular computer, it can be more difficult to determine who specifically "owns" (and hence is responsible for the security of) a network linking many machines, perhaps among several buildings, within a large company. Unless and until the company makes it clear who is in charge of the network, and who therefore determines what should be connected to it and how it is to be administered, its security will be uncertain. Of course establishing such a policy is necessary, not sufficient. In any system as dispersed as even a LAN within a single building, there will be opportunities for users to violate policy, either inadvertently or intentionally. Consequently, monitoring of the network connections and behavior will be needed to some greater or lesser extent, depending on the policy, the users, and the environment.

We cannot specify every possible security policy and its implementation, but we can identify the elements of security policies appropriate for networks. From these elements, an institution should be able to construct a security policy to meet its needs. Once the policy is constructed, means for enforcing the policy, which will be administrative as well as technical, must be decided.

The traditional elements of computer security policy aim to preserve the confidentiality, integrity, and availability of data processed or stored by the computer. Network security services, such as those specified in [OSI] concern the delivery of data between sending and receiving nodes. Properties desired may include confidentiality

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

(protection against eavesdropping), integrity (protection against data modification), authentication (assurance that the source of the data is accurately identified to the recipient), guaranteed delivery (assurance to the sender that the data will reach the recipient), and non-repudiation (inability of the sender to disavow data the recipient receives).

Sometimes more specialized policies are needed. For example, a sender may wish to hide from third parties the fact that he is communicating with a certain recipient. While the content of a data packet can be hidden by encrypting it, its source and destination are generally unprotected, so any eavesdropper can observe that information [SGR 97]. Alternatively, a sender may want to provide certain data anonymously in such a way that the recipient can assess the validity of the data without knowing its origin. This requirement arises in some electronic cash schemes, in which a customer may wish to pay as she does at a vending machine, with an anonymous but validatable coin or banknote. At the same time, such schemes may require some degree of traceability, so that large scale money-laundering can be prevented or at least detected [Wayn 96].

Accountability and authorization are two principles to apply when considering how to implement policies such as these. Accountability requires that actions can be traced to individuals. Authorization requires that access to system resources be limited to those individuals or agents explicitly authorized to use them. Authentication and audit mechanisms can be used to support these principles. In some circumstances, as noted above, there may be reasons to provide anonymity rather than accountability; these situations require special attention.

Returning to the TwoHands Corp., we can identify a number of security policy concerns. Customer order information, business records, and new technology under development in the research laboratory might all generate data that should be protected against unauthorized modification or disclosure and that need to be available without too much delay when required. In all stages, employees are trusted not to violate security policy intentionally, but some additional measures are taken to "keep honest people honest." Also, it is clear in each case who owns and maintains the nodes and links of the networks. TwoHands does not have any special requirements for supporting anonymous functions or for hiding the destinations of its network traffic.

V. Internet Architecture and Vulnerabilities

A. Internet Architecture

To understand where and how information flowing on the Internet can be disclosed, modified, or delayed, it is useful to understand something of its structure.

1) Links

The primary links of the Internet are provided by lines leased from common carriers in countries around the world. In other words, they are part of the public switched telephone networks (PSTNs), although lines used for Internet traffic are usually distinct from those used for voice traffic -- a line leased for Internet use will be dedicated to that purpose. The actual media that carry the traffic are thus as varied as the telephone networks themselves, ranging from copper wire to optical fiber and satellite links. Internet protocols have been used successfully over radio and cellular phone links as well.

2) Nodes

The primary nodes of the Internet are called *routers*. These are computers programmed to accept and forward packets of data (which today may represent real-time voice or video information as well as numbers or text) across the links. A router will be connected to two or more links. It examines each inbound packet for its destination address and, based on its routing tables, determines over which outbound link the packet should be forwarded.

In addition to the routers, there are host computers, which represent the sources and destinations of the packets that are routed through the network. Additional types of network nodes include gateways and firewalls; both of these can be considered types of routers. Gateways that connect networks using different protocols to the Internet will need to provide protocol translation services and addressing services. Firewalls can filter incoming and outgoing traffic, translate addresses and more.

Routing tables can be updated dynamically in order to permit the network to adapt both to outages and to new links. Although the original theory behind the Arpanet's switching scheme was to provide fully decentralized and dynamic routing, the tremendous size to which the Internet has grown has led to somewhat more hierarchical and static routing regimes in practice. Each packet in a series originating from a host attached to a local Internet Service Provider (ISP) and destined for a host attached to an ISP in another country is likely to traverse the same route, and that route is unlikely to include random ISPs in either country. Rather, the packets will be sent from a local ISP "upwards" in the source country to major routers and

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

circuits controlled by the large common carriers, "across" to the major carriers in the destination country, and "down" to the local ISP that connects the destination host.

3) Addressing

Internet Protocol (IP) packets contain both source and destination addresses, and software in the sending node is responsible for providing both of these. The next generation Internet Protocol (Version 6, or IPv6 [BrMa 96]) will expand the IP address space substantially, allowing 128 bits for source and destination addresses in contrast with the 32 bits provided by the current IPv4 scheme.

4) Domain Name System

Packets traverse the Internet using numerical addresses, but users and programs usually deal with more mnemonic addresses in the form of Domain Names. The Domain Name System (DNS) provides the infrastructure for translating domain names into IP addresses [RFC-1034, RFC-1035, RFC-2065].

DNS can provide more than a simple translation from an object name to an IP address. The resource records stored for a domain name that corresponds to a host, for example, may indicate what operating system and version number the host is running. Other resource records can designate a host that processes incoming mail for the specified domain, identify a name server for a domain, or map an alias to the real domain name for a host.

The size of the Internet dictates that the DNS database is distributed among many servers (called Name Servers), none of which has a complete copy. A name server knows the parts of the domain tree for which it has complete information; it is said to be an Authority for those parts of the space.

Although DNS is a critical operational part of the Internet infrastructure, it has no strong security mechanisms to assure data integrity or authentication. However, extensions to provide these services to security aware resolvers or applications through the use of cryptographic digital signatures are under review [RFC-2065]. These mechanisms are also being extended to the proposed dynamic update mechanisms [VTRB 96, East 97].

5) Protocols: IP, ICMP, UDP, and TCP

IP provides an unreliable, connectionless "best effort" delivery service that routes datagrams (packets) towards a specified IP address. IP also includes a protocol for reporting errors, the Internet Control Message Protocol (ICMP); routers use ICMP messages to report delivery failures, misroutings, congestion, and related problems to each other. Both Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are layered on top of IP. Application programs can use TCP or UDP to send messages to applications running on remote hosts [Come 91].

TCP uses IP services to implement a reliable, connection-oriented transport service. TCP tries to guarantee that messages it receives for transmission are delivered to the correct address uncorrupted, without duplication, and in order. User Datagram Protocol (UDP) uses IP services to provide connectionless datagram service. UDP packets may be lost, duplicated, or delivered out of order.

Both UDP and TCP incorporate the notion of *ports* to distinguish traffic sent to the same IP address but for different recipients. A port number in both protocols is a 16-bit integer. An application on one machine can send UDP datagrams to different processes (listening to different UDP ports) on a remote machine by addressing the datagrams to different ports. UDP queues traffic for different ports independently. For TCP, the connection is the fundamental abstraction, and a connection is specified by its two endpoints. Each endpoint is a pair of integers (*host, port*), where *host* is the host's IP address and *port* is a TCP port number on that host. This arrangement permits, for example, a program that accepts incoming mail to use only one local TCP port even though it may be communicating over many connections concurrently. TCP and UDP port numbers are independent, since each message specifies its protocol as well as its destination IP address and port number, and in both protocols certain protocol numbers are used, by convention, as the addresses for particular services. Such a reserved, and advertised, port number is called a *well-known port*. The well known port for e-mail delivery, for example, is TCP port 25; UDP port 53 provides access to the Domain Name Service (as does TCP port 53, but UDP is normally used for initial DNS queries).

A TCP connection is opened with a "three-way handshake": (1) the initiating host sends a "SYN" segment with its IP address and an arbitrary sequence number N, (2) the destination host replies by sending an acknowledgment and another arbitrary sequence number M, and (3) the initiator completes the protocol by acknowledging the second message. The sequence numbers are used subsequently to order traffic sent over the connection. The "SYN-flooding" attack referred to in Section 3.2 (Stage Two, Risks) represents an abuse of this protocol: the attacker sends repeated SYN segments, possibly from different (forged) IP addresses, but never replies to the

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

acknowledgments returned. This leaves the attacked host with many "half-open" connections, clogging the buffers allocated for this purpose. The continuing bogus requests prevent legitimate open requests from being served.

Closing connections is slightly more complicated, as there are various contingencies to consider -- for example, there may be segments en route at the time when one of the participants requests the that connection be closed

In UDP, of course, there are no connections, so there is no protocol to set them up or tear them down.

6) Application Layer Protocols: Telnet, FTP, SMTP, HTTP, SNMP

Most users access the Internet through programs that communicate using application layer protocols. These protocols support the transport of e-mail (SMTP) and files (FTP), the initiation of terminal sessions on remote hosts (Telnet), and the operations of World Wide Web Browsers (HTTP) and many other functions. These protocols may use TCP, UDP, or both to accomplish their functions. Typically, a host that supports a particular service, such as e-mail delivery, will have a process or processes that listen for requests phrased in the appropriate protocol (such as SMTP) over a particular port, as noted above.

B. Vulnerabilities

Any system that is connected to the Internet and uses the Internet to communicate with other systems inherits many well-known vulnerabilities. Some of these arise in the Public Switched Telephone Networks (PSTNs) that carry Internet traffic among routers and hosts. Others come from the routers and hosts themselves. Many have to do with the ways in which people use the Internet and come to rely on it as a means to access both information and processing resources. This section categorizes a number of such vulnerabilities by their location in the architecture.

1) Network

Sniffing. Since computer networks are shared communication channels, data passing over these channels can be intercepted. Such sniffing may be done by special hardware, or by configuring a machine to be in promiscuous mode, to accept all packets independent of their address.

Wardialing. Many networks can be accessed via modem connections, and many such modems are installed outside the control of the system administrator. A common method of identifying such connections is wardialing, in which an attacker programs a computer to connect to a series of telephone numbers (e.g. all numbers starting +1(212)555-). Any answers that are either a modem or a fax machine are logged for further investigation.

2) Protocol

Data Link Layer Security. Address Resolution Protocol (ARP), which is used to translate Ethernet addresses on a LAN to IP addresses, is open to manipulation. For instance, UNIX System V does not check whether received ARP packets are associated with an outstanding request. This could result in malicious responses to ARP requests and unsolicited updates to ARP tables. The most likely effect is denial of service, though "man-in-the-middle" attacks, in which the attacker manipulates addresses so that traffic between two hosts is routed through a compromised machine that masquerades as each host to the other, are also possible.

Network Layer Security. The implementation of IP is generally robust but can be manipulated. Routing is fairly open, which can lead to data not conforming to configured routing. IP packets can be injected directly onto the network, as well. ICMP has no authentication, which could permit manipulation of routing. For example, a malicious user could subvert local routing tables, or ICMP could permit unsolicited address mask reply packets. If an Internet Service Provider broadcasts incorrect routing tables to major backbone providers, it can cause large quantities of traffic to be routed to itself, creating "black hole" and denying service to large communities for significant periods of time [BVW 97].

Network security can be further compromised by the protocols used to manage the network routers. The Simple Network Management Protocol (SNMP) has poor authentication, and unless the routers are correctly configured, they are vulnerable to malicious reconfiguration.

Transport Layer Security. TCP has a weak mechanism for assuring consistent use of port numbers. UNIX systems assume that only privileged processes initiate connections from ports with numbers less than 1024, but there is no reason to assume that such processes are in fact trustworthy. In addition, TCP checksumming of IP packets is not strong, leading to a potential for forgery, injection of data and tailgating of packets. The randomness of TCP initial sequence numbers varies across UNIX systems, leading to a potential to inject packets into a connection between two users.

IP Origin Forgery. The origin of an IP message can be forged relatively easily. This in itself is not a serious vulnerability. However, as many higher level protocols use the IP origin as a form of identification, it becomes

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

serious. For instance, the r-commands (*rlogin*, *rsh*, etc.), which allow unauthenticated access between UNIX systems, use IP source as a primary authentication method.

3) Application

Many of the higher level protocols can be exploited to attack systems connected to the Internet. Many vulnerabilities are well known, such as those in versions of the sendmail system that allow an attacker to gain root privileges quickly. At this point, the attacker can stop audit of his actions, delete any previous audits, install malicious software, read, modify, or delete user applications or data, and then use the current system as a platform for launching an attack on further systems.

Although sendmail may contain the most infamous bugs, many other protocols and software components contain similar bugs and vulnerabilities. Many of these bugs originate from simple coding errors, such as a failure to check array bounds. Indeed, with the applications originating in university environments, and being written primarily with function rather than robustness in mind, it is not surprising that so many protocols and servers are open to attack.

As an example of vulnerabilities in newer protocols, consider the World Wide Web. Web servers provide information in response to requests from Web browsers. If a request exploits a bug in the Web server, the Web server's security mechanisms may be bypassed.

More importantly, however, the Web encourages the sharing of many different types of information, ranging from plain text to forms of executable content. Some, like Postscript, can alter the behavior of a printer. Others, like Java or ActiveX may allow access to resources and data on the machine that should be inaccessible. Java programs are precompiled and stored on a Web server. These mini-applications are known as Java applets and are downloaded to a network browser and executed locally. ActiveX is a technology for distributing software over the Internet. The ActiveX analog of a Java applet is called a *control.* An ActiveX control may be embedded in a Web page where it may be accessed via a network browser.

Java has been explicitly designed to address security issues through various mechanisms that essentially restrict the behavior of applets [McFe 97]. The most important mechanism is that Java programs are interpreted instead of being run in native code. The interpreter, the Java Virtual Machine (JVM) can prevent a program from executing operations that violate the security policy, or it may query the user before allowing such operations. Java itself is a type safe language; this type safety makes programs more robust against certain types of errors, especially pointer manipulation problems. For efficient interpretation, Java programs are compiled to an intermediate bytecode. Arbitrary programs in this bytecode may not satisfy Java's language safety rules, so bytecode is verified before each execution by the Java Bytecode Verifier.

Although Java's basic model is secure, each of these levels introduces room for bugs. The security policy may not be specified correctly. The JVM assumes that the bytecode verifier finds certain classes of errors, so bugs in the verifier may admit hostile programs to the JVM. The interpreter itself may not detect every possible attack. Finally, if Java is opened to ease access by Java programs to other parts of the OS (to make access to certain functions more efficient), Java security may be bypassed.

A number of implementation problems have been identified that allow Java applets to execute arbitrary machine instructions, interfere with other applets, and bypass the Java Security Manager.

ActiveX controls are not restricted in any way, so users must trust the software developer. In order to identify the developer, ActiveX controls may be digitally signed. If a browser accesses a control which is not signed, or the certificate is not recognized, then a dialog box is used to warn the user. This behavior is controlled by properties held in the browser. ActiveX controls have been published on the Internet that close down machines, format hard disks, and install viruses. Relying on users to determine which ActiveX controls are safe to use is a dangerous strategy. History teaches that users quickly become bored with "warning" dialog boxes that appear very frequently. Users either stop paying attention to them or find a way to prevent their appearance.

4) Above The Application Layer

Finally, it is worth considering a few kinds of vulnerabilities that occur above the application layer.

System access control and authorization data structures. A general vulnerability in systems connected to the Internet is the system password file. If a system password is compromised, then the system may be wide open to a variety of attacks. Recently, flaws in Windows NT and Windows 95 were reported that permit hostile web pages to unobtrusively extract a user's password and transmit it to the page's author, for example [Wayn 97].

Malicious software. A *Trojan horse* is a program that masquerades as a useful service but exploits rights of the program's user -- rights not possessed by the author of the Trojan horse -- in a way the user does not intend. A *virus* can be viewed as a Trojan horse that replicates itself by

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

copying its code into other program files [LBMC94]. Programs to detect and remove viruses has established a significant niche for itself in the software market. Malicious software can be a problem at many levels. A virus may take the form of object code that, when executed, copies itself to other executable files, for example, or it may be a set of macro commands for a word processor (e.g., Microsoft Word) that copies itself into all of the document files created by that word processor on the machine where the virus is executed. Malicious software executed on Internet-connected machines has the opportunity to use Internet protocols and facilities to propagate itself or to communicate with its author, for example, by mailing a copy of the password file.

Hidden application features: An application may store much more information in a file than is apparent when the file is opened by that application. It is often impossible to fully review the contents of a computer file, so that sensitive information can be exported from a system inadvertently. Microsoft Word's frequently used *fastsave* option provides an example. When text is deleted from a document and the document is then saved again, Word does not delete the text: it merely inserts a note not to display the deleted text. The text can be recovered using another program which disregards Word's instruction to itself.

VI. Security Building Blocks

A number of off-the-shelf technology products are available to mitigate the vulnerabilities outlined in the preceding section. These products can be grouped under the following broad headings: encryption, authentication mechanisms, intrusion detection tools, security management tools, and firewalls. Businesses throughout the world are starting to use such methods - judicious combinations of firewalls, encryption and authentication techniques, for example - to create corporate "Intranets" that are both connected to the Internet and reasonably protected from it. They recognize that their networks are not completely protected from the threats posed by connection to the Internet, but believe that on balance the net benefits outweigh the risks.

The Internet itself is an excellent source of information on the topics listed below. Web-based repositories such as [COAST], [NIH], and [Stein] make security-related tools and information easy to access.

1) Encryption: Privacy

How can the confidentiality of data be protected when it moves over public channels? One way is to encrypt the data [Schne 96]. Encryption uses an encryption algorithm and a key to change some input, called *plaintext*, to some output, called *ciphertext.* The ciphertext looks like a random string of bits to observers who do not know the appropriate decryption algorithm and key. Two classes of encryption/decryption algorithms exist: symmetric and asymmetric. Symmetric algorithms, likes DES (the Digital Encryption Standard), use the same algorithm and key for encryption and decryption. Imagine that Alice and Bob want to communicate privately using DES. They would agree to use a single key K. If Alice wanted to send a message M to Bob, she would encrypt M using the DES algorithm initialized with key K, which produces the ciphertext. For this example, call this ciphertext DES(K,M). To anyone who does not know K, DES(K,M) appears to be a random string. Bob, however, can decrypt the message by applying DES again to the ciphertext using key K. The result, DES(K,DES(K,M)) is the original plaintext, M. Bob can reply to Alice using DES with the same or another key they have agreed to use. Anybody who knows these keys can listen in on Alice and Bob's conversation, so these keys must be kept secret.

Symmetric encryption/decryption algorithms such as DES use an already existing shared secret (the key K) to efficiently share new secrets (the message M). How can Alice and Bob choose a key if they do not already share some secret? This challenge is called the key distribution problem. Until the early 1970s, when public key cryptography [DH], [RSA] was invented, keys were distributed by some other means (for example, a trusted courier).

Public key cryptography is also called asymmetric encryption because different keys are used for encryption and decryption (communicating parties do not share a key). Instead, everyone has two keys: a public key (P) , which is published widely, and a private (secret) key S, which the owner keeps secret to himself. In a public key scheme like RSA, if Alice wants to send a private message M to Bob, she looks up Bob's *public* key (in some trusted directory, for example), P_Bob, and encrypts M, using the public key algorithm RSA, to produce the ciphertext RSA(P_Bob,M). Bob can decrypt this ciphertext by applying RSA again, using his *private* key, S_Bob. The result, RSA(S_BOB,RSA(P_Bob,M)) is M. Notice that when using RSA, unlike DES, Alice can send Bob a private message without sharing a secret with Bob.

Alice does, however, need a reliable way to find out Bob's public key. In this context, a *digital certificate,* or simply *certificate,* is a data structure that binds Bob's name and public key, digitally signed (see next section) by a trusted agent, known as a *certification authority*. Alice may

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

request a certificate for Bob from a certification authority she trusts. If a private key is compromised, the certificate authority can announce this by distributing a *certificate revocation list* (CRL) to its subcribers. The set of mechanisms for operating a public key cryptosystem, including the maintenance of directories and certificates, is often referred to as a *public key infrastructure* (PKI).

Asymmetric crypto-systems are computationally expensive to use. It is not practical to encrypt and decrypt long messages using public key cryptography (one source estimates that RSA, the only widely used public key algorithm, is, at its fastest, about 1000 times slower than DES.) Therefore, public key cryptography is usually used by two parties to choose a new key that will be used later to encrypt messages using some efficient symmetric crypto-system. The process of key distribution and private communication is shown in Figure 6.

2) Encryption: Authentication

Both symmetric and asymmetric encryption techniques can be used to authenticate data as well. That is, messages can be cryptographically signed in such a way it proves that only parties that know the key could have signed the message. For example, if only data integrity, and not privacy, is important, a message can be sent as plaintext along with an encrypted *hash* of the message. A *cryptographic hash function*, like SHA or MD5, takes an arbitrarily long message and produces a short, fixed-size result called a hash. Even small changes in the long message result in apparently random changes in the output, so it would be virtually impossible for an attacker to replace the message with a different one, since the two messages would have different hashes. Thus, authenticating the hash is nearly as safe as signing the original message, but saves both computation time and communications bandwidth. If symmetric key encryption is used, the encrypted hash is often called a *message authentication code*, or a MAC. All parties participating in a communications session can produce and verify these MACs.

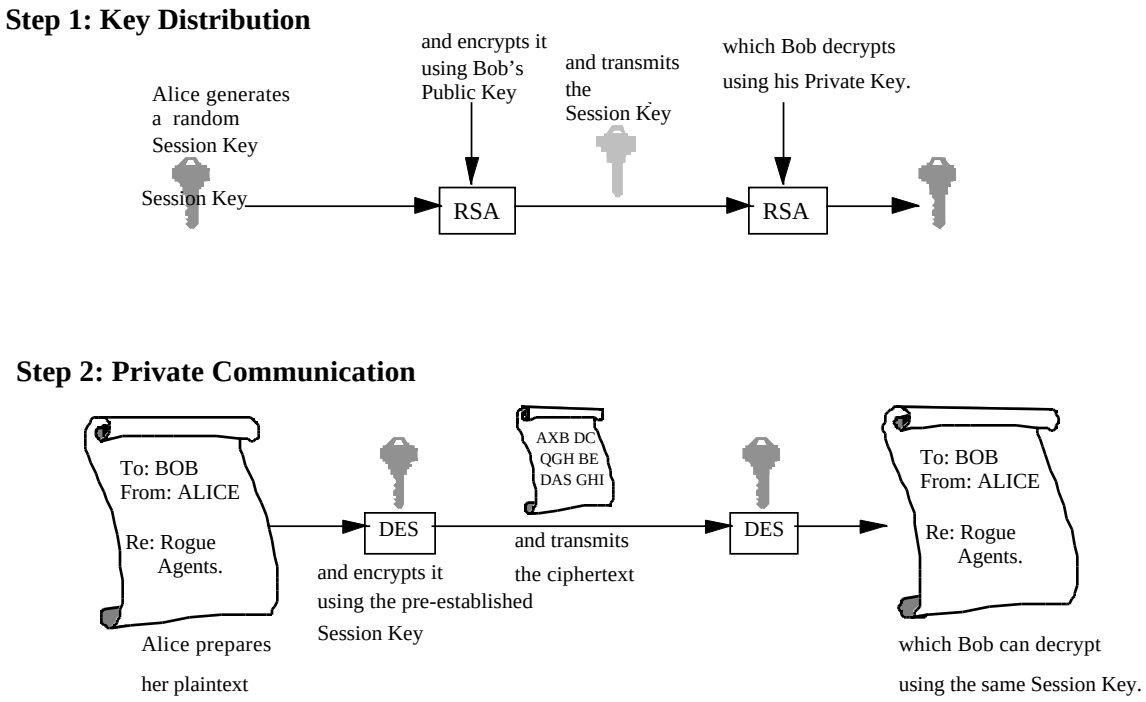

Figure 6: (a) Step 1: key distribution. (b) Step 2: private communication.

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

Public key crypto-systems can be used to identify the sender of a message. The process is like having the sender sign the message and is called a digital signature. RSA can produce digital signatures, as well as do encryption, because the public and private keys are interchangeable. The digital signatures can be used to authenticate the sender.

If Alice wants to package a message M so the recipient can verify that it is from her, she would encrypt M using RSA and her private key, S_Alice. The resulting ciphertext, RSA(S_Alice,M), is a digital signature that could only have been produced by Alice because only she knows her secret key, S_Alice. The signed packaged would include both M and RSA(S_Alice,M). If Bob wants to verify that M was really written by Alice, he "unsigns" the associated digital signature using Alice's public key, P_Alice. The result, RSA(P_Alice,RSA(S_Alice,M)), is M. If someone else tried to forge Alice's digital signature using some other private key, the unsigned message would not match M. (Since public-key cryptographic operations are computationally expensive, usually Alice would sign only the hash of the message.)

3) Encryption: Applications

Encryption [Schne 96] and digital signatures offer solutions to five important security requirements: confidentiality, access control, integrity, data origin authentication, and non-repudiation. Commercial encryption products (and some public domain implementations) are available that provide encryption services at different protocol layers, including the application layer, session layer, and the IP layer. There are also significant efforts underway to standardize access to encryption services by application programs (Crypto Application Program Interfaces, or CAPI), so that a variety of standards-based products can come to market (see [ICE 97] for further information).

Application layer encryption refers to encryption services applied to structured data, where different encryption and authentication procedures may be applied depending on the type of the data. For example, in e-mail, the body of the message may be encrypted for the recipient, and signed by the sender. An application layer encryption system based on public key certificates effectively involves three main components - a User Agent (UA) to which a user is authenticated, a repository of public keys for the potential recipients of encrypted data, and an associated encryption/digital signature capability. The UA acts on behalf of the authenticated user, so any digital signature created by the UA on his behalf can only be as strong as the original authentication between the user and the UA, the integrity of the UA, or the strength of the encryption algorithm. Encryption is performed by the system for specific recipients, which may include the originator. Several standards have been developed for the exchange of encrypted e-mail, including those for Privacy Enhanced Mail (PEM; see [Schn 96] p. 577 ff. for a summary), PGP [Zimm95], and Secure / Multipurpose Internet Mail Extensions (S/MIME), currently documented in Internet Drafts. Subsequently, commercial application level encryption systems such as “Entrust” from Entrust Technologies, Inc. have been developed. In its basic form, the Entrust system includes a Public Key Infrastructure and a tool for encrypting and signing or decrypting and reading signatures. Entrust also includes a toolkit for software vendors to allow them to integrate these capabilities into their products. At present there are already e-mail, e-form and word processing applications which are “Entrust aware”. So far no applications which have been made or are evolving to be “Entrust aware” are real-time applications (e.g. video-conferencing), although there appears to be no fundamental restriction against this use.

Session encryption acts in a similar way to an application encryption system, except that the authenticated entities will be the agents participating in the session. These may be, for example, a web browser and server. All data transmitted between the agents during a session will be encrypted but will be in the clear when presented to the applications. Typical session encryption would not authenticate users but rather applications on hosts. Examples of current session-based systems are SSL (Secure Sockets Layer), initially developed by Netscape Communications, Inc., and widely deployed in Web browsers and servers. SSL has two phases: in the first phase, the handshaking phase, the applications (typically a client and server) authenticate each other [2] and agree on a session key; in the second phase, applications transmit encrypted data using that key. It is typically easier to modify applications to use session encryption than it is to introduce application layer encryption, because session level encryption is basically another type of communications channel. Instead of opening of vanilla TCP connection, an SSL connection is opened. Also, the use of SSL is nearly transparent to the user of a browser because only the server needs a cryptographic certificate, while encrypted e-mail will remain more complex to use until it is more fully integrated into e-mail application

[2] In the first versions of SSL, the client authenticated the server, but not *vice versa*; mutual authentication was subsequently added.

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85*, No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

programs and until more users have cryptographic certificates.

IP layer encryption is used by some commercially available products that permit using the Internet as a substitute for a dedicated private line, although these tend to be relatively expensive and key management may be an issue. A pair of these products, each acting as a gateway between a private network and the Internet, will route all traffic to each other via Internet routers, which will see cleartext IP headers but encrypted payloads. The address of the actual destination system can be hidden in the encrypted payload, so that the Internet routers only see the addresses of the encrypting gateways. Firewalls that provide Virtual Private Networks (VPNs) use essentially this approach. VPNs are useful and they effectively protect communications between sites that are behind the encrypting firewalls; however, generalized web browser and e-mail delivery are not protected. The Internet Protocol is being extended with a security option, IPSEC, (see [RFC-1825, -1826, and -1827], which are under revision at this writing, and [Bell 96] for a description of vulnerabilities) and key distribution mechanisms that will support more flexible use of encryption at the IP layer.

4) Authentication Mechanisms

Passwords remain by far the most common form of user authentication. Users must choose, or be provided, unguessable passwords, and reusable passwords should never be passed over the Internet in the clear. There are both software and hardware technologies supporting one-time passwords [Hall 94]; disclosure of these passwords does not compromise the system. Encrypting the actual session prevents password sniffing and session hijacking.

One widely used authentication system is Kerberos [SNS 88], which is publicly available and can provide reliable authentication over open networks such as the Internet. Kerberos is a secret key authentication system that involves a central database keeping copies of the secret keys of all users. It uses DES or other algorithms for encryption and authentication, and allows entities to communicate over networks and to prove their identity to each other while thwarting eavesdropping or reply attacks. It also provides for data stream integrity and secrecy. When a user wishes to access a service, she authenticates herself to the *Key Distribution Center* (KDC) instead and asks for a *ticket* to communicate with the service. The ticket is only readable by the service, and it identifies the user and specifies the key the user will use when contacting the service. The KDC returns this ticket and the embedded key to the user. The user sends the ticket to the service. By using the embedded key, the user and service can authenticate themselves to each other, because each is confident that no one but themselves and the KDC know the key, and the KDC is trusted. Its dependence on a central database limits the ability of Kerberos to scale to very large user communities, but it has been used effectively in systems with thousands of users. And, of course, Kerberos is not without vulnerabilities, but new releases typically resolve known issues.

5) Intrusion Detection

Despite the best efforts of the protocol designers, implementers, and system administrators, it is prudent to assume that attacks will occur and some, unfortunately, will succeed. Therefore, it is vitally important that a means to detect and respond to these attacks is installed to protect critical information services. Both commercial and government developed intrusion detection products can detect known attacks and alert administrators.

Current intrusion detection systems are of two types: network based and host based. Network based intrusion detection systems pull each packet as it enters the monitored network and examine a series of them for patterns matching known attack methodologies. When a possible attack is discovered, the administrator is alerted and must take action to prevent further intrusions. These tools are similar to virus checkers, in that, as new attacks are discovered the tool must be modified to allow the discovery of these potential new attacks. Sample government products available are Network Security Monitor (NSM) and Network Intrusion Detection System (NIDS). Commercial products available are Net Ranger, Intruder Alert, CMDS and others. Tripwire [KiSp94] is a freely available and widely used UNIX utility that can alert a system administrator to changes in file systems that may signal an intrusion.

Host based intrusion detection systems are required to be run as background processes on each individual host within the monitored network. These systems detect probes of host ports, password guessing and other known attack methods, which are captured as part of the auditing features (if turned on) of the host. Intrusion detection systems can also build a profile of a host or user's typical behavior and alert an administrator about unusual activity.

There is some overlap between host based and network based pattern matching capabilities, and running the two in concert will sometimes yield two copies of the same detection alert. These techniques cannot detect all potential attacks and will provide some false alarms, however they can reduce the risk of undetected intrusions. It is difficult, however, to characterize precisely either the

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

absolute or relative effectiveness of alternative tools.

6) Security Management Tools

A number of tools exist that can be used by both attackers and system managers to test the security of a system. Widely available ones include:

SATAN . Security Analysis Tool for Auditing Networks (SATAN) is a network vulnerability toolkit, using a WWW front-end. In its simplest (and default) mode, it gathers as much information about remote hosts and networks as possible by examining such network services as finger, NFS, NIS, ftp and tftp, rexd, and other services. The information gathered includes the presence of various network information services as well as potential security flaws - usually in the form of incorrectly setup or configured network services, well-known bugs in system or network utilities, or poor or ignorant policy decisions. It can then either report on this data or use a simple rule-based system to investigate any potential security problems. Users can then examine, query, and analyze the output with an HTML browser, such as Mosaic, Netscape, or Lynx. While the program is primarily geared towards analyzing the security implications of the results, a great deal of general network information can be gained when using the tool, including network topology, network services running, and types of hardware and software being used on the network.

However, the real power of SATAN comes into play when it is used in exploratory mode. Based on the initial data collection and a user configurable rule set, it will examine the avenues of trust and dependency and iterate further data collection runs over secondary hosts. This not only allows the users to analyze their own network or hosts, but also to examine the real implications inherent in network trust and services and help them make reasonably educated decisions about the security level of the systems involved. SATAN should prove to be most useful when used by the system or security administrators who own or are responsible for the security of the systems involved.

Internet Security Scanner (ISS) can perform a scan on a host or network, to test for a common set of security flaws and errors in configuration. It is designed to carry out a simple vulnerability test on a network of computers in order to highlight systems that are vulnerable and may be used to gain access to more secure and important systems.

ISS highlights the risks of mixing secure and insecure machines on the same network. All hosts should be secured to the same level as the highest machine on the network. If a host is easy to penetrate then it will become the weak link in the security chain. An intruder can use this to exploit the configurations of other hosts. The secure hosts may not be insecurely set up but if they trust a host that isn't then they also become vulnerable

tiger. The tiger software suite, produced by the Texas A & M University, is a set of Bourne shell scripts, C programs and data files that are used to perform a security audit of UNIX systems. It has pre-defined configuration databases for AIX 3.x as well as many other systems.

The tiger suite has one primary goal: report ways in which root can be compromised. While checks are performed for other purposes, most of the checks are directed at this goal. The primary assumption made is that any UID other than 0 can be obtained and that any GID can be obtained by unauthorized persons.

The checks performed extend to cover other means of gaining root access, e.g. cron, inetd and setuid executables. Access through these methods is checked to see if any user, other than root, can alter any of the configuration files associated with these utilities. Specific checks are performed to see if anything in the root executable path can be modified by a normal user. All user accounts will have these checks performed but special attention is paid to the root account.

crack. crack is a dictionary based password guessing tool. To have a more complete understating of the implications of this kind of attack a brief description of the UNIX password encoding system is required.

UNIX passwords are not stored in the clear. A new password chosen by the user is encrypted using the DES algorithm. The encrypted password is then modified by deleting selected bits and re-encrypting it (for details on this, and other aspects of UNIX security, see [GaSp 96]). The result is a 14 character string from which the original password cannot be deduced. This string (which is now akin to a hash of the original password) is stored. When a user attempts to login, the computer checks the validity of the typed password by encrypting it in the same manner, and comparing the result against the stored string. If they are identical, then the password is valid.

snoop. Snoop is a promiscuous-mode IP packet sniffer. The present Sun operating system, Solaris, provides to administrators a command *snoop* for the capture and inspection of ethernet packets. By default it uses both the network interface and the streams buffer modules to capture packets, then displays a single line summary of each packet seen. It can also provide a far more detailed packet description, including information from the ethernet layer upwards to the top layer. This utility can also be used to monitor certain protocols and to sniff for authentication data, for example the rexec service.

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

7) Firewalls

A firewall is a system (one or more pieces of hardware and software) that acts as a barrier between two network segments, such as a protected enclave handling sensitive information and the Internet. A firewall can be considered the technical implementation of a security policy. It upholds the security policy of a network when connecting that network to a second network which has a less stringent security policy.

Firewalls are of two general types: packet filters and application gateways. A packet filtering firewall can be implemented on a router that screens packets before forwarding them between ports on the router. Filtering is controlled by a set of rules that make forwarding decisions based on the source and destination addresses in the IP packet, the packet's protocol, and the router port (identifying the physical network) the packet came from. Since all of this information is available for each packet, these decisions do not require the firewall to remember anything about prior traffic, so they are called *stateless*. Packet filtering firewalls are usually fast and cheap, and they can handle many different protocols. However, implementing a security policy as a set of filtering rules may be complicated, and some information that may be useful (or necessary) in making filtering decisions may not be available in a single packet (e.g., the identity of the source user). Although *stateful* packet filters (which keep some record of previous traffic) may be able to extract additional information from a series of IP packets, this may be done more easily via application gateways.

Application gateways, or proxy firewalls, filter application level data streams instead of IP packets. A client application, instead of making a connection directly to some server, makes a connection to an application specific proxy running on the firewall. Depending on the security policy, the proxy decides whether to make a connection on the client's behalf to the destination server. The proxy may require additional authentication of the client and may filter the connection based on content and context. Since the proxy is application specific, it may be hardened to trap protocol specific attacks (fields that are too long, for example). By hardening the proxy, one may avoid the need to harden individual applications on machines inside the firewall. Also, since connections to the outside always originate from the firewall, a proxy firewalls hides the internal structure of the network.

However, proxy firewalls are typically slower and more expensive than packet filtering firewalls. And, not all applications use connection based communication (e.g., audio and video is often broadcast using UDP).

SOCKS [SOCKS] is a proxy package with many extensions. The proxy server can run monitoring, auditing, and intrusion detection systems. SOCKS can authenticate clients using a variety of mechanisms including simple passwords, one-time passwords, and cryptographic challenge response. Just as client applications have to be proxy aware to use a proxy firewall, client applications have to be made SOCKS ready to communicate through a SOCKS server, and code libraries exist for making applications SOCKS ready. Unlike application proxies, the SOCKS server may not filter out application specific attacks, because it does not interpret the protocol data stream.

The security provided by firewalls is sometimes likened to the crunchy outer shell of a candy that is soft and chewy on the inside, in that once the firewall is penetrated (or circumvented) the intruder may find the internal systems easy to manipulate. The implication is that firewalls must not be seen as a panacea that reduces the need to administer the internal systems so that they remain as secure as possible.

VII. Summary and Conclusion

It is impossible to provide a comprehensive treatment of network security in a paper as short as this one must be, and any such treatment that goes into vulnerabilities and countermeasures in technical detail will soon be obsolete. Readers desiring a fuller understanding of both the problems and current solutions should consult the references and (particularly for current information) the Web sites listed.

We have focused on the security implications of attaching private networks to the Internet, but from the enterprise's view, security is a property of the system as a whole, not merely of its parts. If sensitive information is disclosed or modified improperly, or if critical services are denied to the enterprise or its customers, system security has been breached.

Designing a system that provides needed functions at acceptable risk and cost is an art that calls for careful balancing of these factors. Operating such a system calls for continuing vigilance, since, unless a denial of service attack is mounted, an intact, secure system may appear to operate just the same as a penetrated, insecure one. We have enumerated a number of principles that have proven useful in designing and operating systems that meet security requirements, and illustrated them through the account of the TwoHands Corporation's networks.

Connecting an internal network to the Internet increases

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85,* No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

the functions a system can provide. It can permit flexible communication with a wide range of individuals and enterprises, thereby enabling more convenient and efficient corporate operations. But it can also expose systems to a much wider variety of attacks than they previously faced. Further, systems are complicated, individual services have peculiar characteristics, and combinations are often unpredictable. Nevertheless, careful attention to the system design and operation can in most cases reduce the risks of Internet connection to an acceptable level.

Most network security controls now focus on the perimeter of the network, but the actions of individual users are crucial, too. Training users to behave responsibly is essential. The user's workstation is an extension of the user, and operates under his authority. Eventually, it must be secured as well.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the contributions of the members of Technical Panel 11, Secure Information Systems, of The Technical Cooperation Program to the development of Sections 5 and 6. In particular, thanks are due to Alex Scott of the United Kingdom, Vincent Taylor and Paul Devlin of Canada, Peter Drewer of Australia, and John McLean, Michael Harrison, and Dwayne Allain of the US. Michael Reed of NRL provided helpful review and comment. We also appreciate the comments of the anonymous referees.

REFERENCES

[AUSCERT] Australian Computer Emergency Response Team. http://www.auscert.org.au

[Bell 96] Bellovin, S.M. Problem Areas for the IP Security Protocols. *Proc. of the Sixth Usenix UNIX Security Symposium*, July, 1996. Available at URL: ftp://ftp.research.att.com/dist/smb/badesp.ps

[Bell 97] S. Bellovin. Network and Internet Security. In *The Computer Science and Engineering Handbook*, A. B. Tucker, Jr., editor, CRC Press, 1997, Chapter 92, pages 1949-1961, ISBN 0-8493-2909-4.

[BrMa 96] Bradner, Scott, and Allison Mankin, Eds. *IPng, Internet Protocol Next Generation*. Addison-Wesley, Massachusetts, 1996.

[BVW 97] Barret, R., S. Vonder Haar, and R. Whitestone. Routing Snafu causes Internet Outage. *Inter@ctive Week Online*, April 25, 1997, 2:55 p.m. PDT. http://www.zdnet.com/intweek

[CA21- 96] Computer Emergency Response Team. TCP SYN Flooding and IP Spoofing Attacks. CERT Advisory 96.21, September, 1996: ftp://info.cert.org/pub/cert_advisories/CA^~96.21.tcp_syn_flooding

[ChBe 94] Cheswick, W. R. and S. M. Bellovin. *Firewalls and Internet Security: Repelling the Wily Hacker*. Addison-Wesley, Massachusetts, 1994, ISBN 0-201-63357-4.

[CERT] Computer Emergency Response Team. http://www.cert.org

[CGI] CGI-security: http://www.cerf.net/~paulp/cgi-security/

[CIAC] Computer Incident Advisory Capability. http://ciac.llnl.gov/

[COAST] Computer Operations, Audit and Security Technology Homepage. URL: http://www.cs.purdue.edu/coast

[Come 91] Comer, Douglas E. *Internetworking with TCP/IP, Vol. I*. Prentice-Hall, New Jersey, 1991.

[DH] W. Diffie and M.E. Hellman. New Directions in Cryptography, *IEEE Transactions on Information Theory*, v IT-22, n. 6, Nov. 1976, pp. 644-654.

[East 97] Eastlake, D., 3rd. Secure Domain Name System Dynamic Update. Internet-Draft <draft-ietf-dnssec-update-04.txt>, 4 Feb 1997.

[FeMo 97 Ferbrache, D., and S. Mort. Malicious Software and Hacking. In *The Computer Science and Engineering Handbook*, A. B. Tucker, Jr., editor, CRC Press, 1997, Chapter 93, pages 1962-1979, ISBN 0-8493-2909-4.

[FIRST] Forum of Incident Response and Security Teams. http://www.first.org/

[FKK 96] Freier, A.O., P. Karlton, and P. Kocher. The SSL Protocol, Version 3.0. Internet Draft, November 18, 1996. <draft-ietf-tls-ssl-version3-00.txt> . Available at URL: http://ds.internic.net/internet-drafts/draft-ietf-tls-ssl-version3-00.txt

[FNC] Federal Networking Council Resolution: Definition of "Internet" Oct. 24, 1995. Available at URL: http://www.fnc.gov/Internet_res.html

[GaSp 96] Garfinkel, S. and E. Spafford. *Practical UNIX and Internet Security, Second Edition*. O'Reilly & Associates, Inc., 1996, ISBN: 1-56592-148-8.

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85*, No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

[GrMo 84] F. T. Grampp and R. H. Morris. UNIX Operating System Security. *AT&T Bell Laboratories Techincal Journal, 63*(8): pp.1649-1672, October 1984.

[Hall 94] N. M. Haller. The S/Key One-Time Password System. In *Proc. Internet Society Symp. on Network and Distributed System Security*, San Diego, CA, February 1994.

[ICE 97] International Cryptographic Experiment (ICE) Web page. URL: `http://www.tis.com/docsw/research/crypto/ice/index.html#WEBLINKS` This web page includes links to a wide variety of information sources on standards for cryptographic applications programming interfaces and standard interfaces for security services.

[IETF] Internet Engineering Task Force. `http://www.ietf.org/`

[KiSp 94] Kim, G.H. and E.H. Spafford. Writing, Supporting, and Evaluating Tripwire: A Publically Available Security Tool. Purdue Technical Report CSD-TR-94-019, Computer Science Department, Purdue University, 12 March 1994. This report and much relevant information is available at the COAST computer security archive, URL: `http://www.cs.purdue.edu/coast/`

[KPS 95] Kaufman, C., R. Perlman, and M. Speciner. *Network Security: Private Communications in a Public World*. Prentice-Hall, Englewood Cliffs, NJ, ISBN 0-13-061466-1.

[Lamp 73] B. Lampson. A Note on the Confinement Problem. *Communications of the ACM, Vol. 16*, No. 10, 1973.

[Land 97] Landwehr, C.E. Protection (Security) Models and Policy. In *The Computer Science and Engineering Handbook*, A. B. Tucker, Jr., editor, CRC Press, 1997, Chapter 90, pages 1914-1928, ISBN 0-8493-2909-4.

[LBMC 94] Landwehr, C. E., A.R. Bull, J. P. McDermott, and W. S. Choi. A taxonomy of computer program security flaws. *ACM Computing Surveys, Vol. 26*, No. 3, (Sept 1994) 211-254.

[McFe 97] McGraw, Gary and Ed Felten. *Java Security: Hostile Applets, Holes, and Antidotes*. John Wiley & Sons, 1997.

[NIH] `http://www.alw.nih.gov/Security/`

[OSI] Open Systems Interconnection -- Basic Reference Model -- Part 2: Security Architecture. ISO/IEC 7498-2.

[ReSc 97] Rescorla, E., and A. Schiffman. Secure Hypertext Transport Protocol. Internet Draft, March 1997. <draft-ietf-wts-shttp-04.txt> Available at URL `http://ds.internic.net/internet-drafts/draft-ietf-wts-shttp-04.txt`

[RSA] R.L. Rivest, A. Shamir, and L.M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Communications of the ACM*, v. 21, n. 2, Feb. 1978, pp. 120-126.

[RFC-1034] Mockapetris, P. Domain Names - Concepts and Facilities. RFC 1034, Nov. 1987.

[RFC-1035] Mockapetris, P. Domain Names - Implementation and Specification. RFC 1035, Nov. 1987.

[RFC-1825] Atkinson, R. Security Architecture for the Internet Protocol. RFC 1825, August, 1995.

[RFC-1826] Atkinson, R. IP Authentication Header. RFC-1826, August, 1995.

[RFC-1827] Atkinson, R. IP Encapsulating Security Payload. RFC-1827 August, 1995.

[RFC-2065] Eastlake, D., 3rd, C. Kaufman. Domain Name System Security Extensions. RFC-2065, Jan., 1997

[Schn 96] Schneier, B. *Applied Cryptography: Protocols, Algorithms and Source Code in C, Second Edition*, John Wiley and Sons, 1996, ISBN 0-471-11709-9.

[SKKSSZ 97] Schuba, C.L., I.V. Krsul, M.G. Kuhn, E.H. Spafford, A. Sundaram, D. Zamboni. Analysis of a Denial of Service Attack on TCP. *Proc 1997 IEEE Symp. on Security and Privacy*, pp. 208-223. ISBN 0-8186-7828-3.

[SOCKS] `http://www.socks.nec.com`

[SNS 88] J. Steiner, B. C. Neuman, and J. I. Schiller. Kerberos: An Authentication Service for Open Network Systems. *Proceedings of the Winter USENIX Conference*, Dallas, Texas, 1988.

[Stein] L. D. Stein. The WWW Security FAQ. URL: `http://www.genome.wi.mit.edu/WWW/faqs/www-security-faq.html`

[SGR 97] Syverson, P. F., D.M. Goldschlag, and M.G. Reed. Anonymous Connections and Onion Routing, *Proc. 1997 IEEE Symp. on Security and Privacy*, pp. 44-54. ISBN 0-8186-7828-3. See also URL: `http://www.itd.nrl.navy.mil/ITD/5540/projects/onion-routing/`

[VTRB 96] Vixie, P. (Ed.), S. Thomson, Y. Rekhter, J.

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85*, No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications

Bound. Dynamic Updates in the Domain Name System. Internet Draft, <draft-ietf-dnsind-dynDNS-11.txt>

[Wayn 96] Peter Wayner. *Digital Cash: Commerce on the Net*, AP Professional, Chestnut Hill, Mass., 1996

[Wayn 97] Peter Wayner. Security holes threaten Microsoft game plan. *New York Times* (CyberTimes section) March 28, 1997.

[W3C] W3C Security Resources. URL: `http://www.w3.org/pub/WWW/Security/`

[Zimm95] Philip R. Zimmerman. The Official PGP User's Guide. MIT Press, Cambridge, 1995.

Carl E. Landwehr (member, IEEE) received the B.S. degree from Yale University, New Haven, CT, and the Ph.D. degree from the University of Michigan, Ann Arbor.

He is the Head of the Computer Security section of the Center for High Assurance Computer Systems at the U.S. Naval Research Laboratory, where he has been since 1976. His research interests currently center on practical methods for designing and building systems of computers that can provide high assurance that critical properties will be enforced, and in particular on identifying simple components that can be combined with off-the-shelf systems to support system security. He recently served on a National Research Council panel investigating security and privacy issues in medical information systems, and he serves as an Expert Consultant to the North Atlantic Treaty Organization on computer security. He is an Associate Editor of the *High Integrity Systems Journal*. He also has been an Associate Editor of the *Journal of Computer Security*.

Dr. Landwehr is an Associate Editor of IEEE TRANSACTIONS ON SOFTWARE ENGINEERING. He edits Cipher, the electronic newsletter and Web site (http://www.itd.nrl.navy.mil/ITD/5540/ieee/cipher) of the IEEE Computer Society's Technical Committee on Security and Privacy. He received the International Federation for Information Processing Silver Core for his work as Founding Chair of its Working Group 11.3 on Database Security. He received the IEEE Computer Society Golden Core for his work on behalf of its Technical Committee on Security and Privacy.

David M. Goldschlag received thePh.D. degree in computer science from the University of Texas at Austin in 1992.

He conducted computer security research at the U.S. Naval Research Laboratory and the National Security Agency. He currently is with Divx, Herndon, VA, where he works on intellectual property protection. His research interests are in cryptography, security, automated theorem proving, and system design and analysis, particularly on how to structure systems so that critical function is isolated and easy to analyze and implement.

Landwehr, C.E. and D.M. Goldschlag. Security issues in networks with Internet access. *Proceedings of the IEEE, Vol. 85*, No. 12 (Dec. 1997)1034-2051. This version of the paper differs in citation formats, pagination, and other minor details from the published version. This version available from NRL Code 5540 Web site: http://www.itd.nrl.navy.mil/ITD/5540/publications