

Best Current Practices in Security

Compiled by Ken Camp – ken@ipadventures.com (802) 372-8860

Foreword

This paper is not to be construed as a unique or original work. This information was collected and compiled from a variety of widely recognized sources listed in the Acknowledgements at the end. Network security is a dynamic function, requiring constant attention. Security is a process, not a product. This is an organization of the best thoughts regarding the process of providing network security.

What Makes a Good Security Policy?

A security policy is a formal statement of the rules by which people that are given access to an organization's technology and information assets must abide.

The purpose of a security policy is to inform users, staff and managers of their requirements for protecting technology and information assets. The policy should specify the mechanisms through which these requirements can be met. Another purpose is to provide a baseline from which to acquire, configure and audit computer systems and networks for compliance with the policy. Therefore an attempt to use a set of security tools in the absence of at least an implied security policy is meaningless.

An Appropriate Use Policy (AUP) may also be part of a security policy. It should spell out what users shall and shall not do on the various components of the system, including the type of traffic allowed on the networks. The AUP should be as explicit as possible to avoid ambiguity or misunderstanding. For example, an AUP might list any prohibited USENET newsgroups. (Note: Appropriate Use Policy is referred to as Acceptable Use Policy by some sites.) The AUP also defines prohibited network services such as online chat (IRC, ICQ , etc.) and access to inappropriate web sites.

The characteristics of a good security policy are:

1. It must be achievable through system administration procedures, publishing of acceptable use guidelines, or other appropriate methods.
2. It must be enforceable with security tools, where appropriate, and with sanctions, where actual prevention is not technically feasible.
3. It must clearly define the areas of responsibility for the users, administrators, and management.

The components of a good security policy include:

1. Computer Technology Purchasing Guidelines, which specify required, or preferred, security features. These should supplement existing purchasing policies and guidelines.

2. A Privacy Policy which defines reasonable expectations of privacy regarding such issues as monitoring of electronic mail, logging of keystrokes, and access to users' files.
3. An Access Policy, which defines access rights and privileges to protect assets from loss or disclosure by specifying acceptable use guidelines for users, operations staff, and management. It should provide guidelines for external connections, data communications, connecting devices to a network, and adding new software to systems. It should also specify any required notification messages (e.g., connect messages should provide warnings about authorized usage and line monitoring, and not simply say "Welcome").
4. An Accountability Policy, which defines the responsibilities of users, operations staff, and management. It should specify an audit capability, and provide incident handling guidelines (i.e., what to do and who to contact if a possible intrusion is detected).
5. An Authentication Policy, which establishes trust through an effective password policy, and by setting guidelines for remote location authentication and the use of authentication devices (e.g., one-time passwords and the devices that generate them).
6. An Availability statement, which sets users' expectations for the availability of resources. It should address redundancy and recovery issues, as well as specify operating hours and maintenance downtime periods. It should also include contact information for reporting system and network failures.
7. An Information Technology System & Network Maintenance Policy, which describes how both internal and external maintenance people are allowed to handle and access technology. One important topic to be addressed here is whether remote maintenance is allowed and how such access is controlled. Another area for consideration here is outsourcing and how it is managed.
8. A Violations Reporting Policy that indicates which types of violations (e.g., privacy and security, internal and external) must be reported and to whom the reports are made. A non-threatening atmosphere and the possibility of anonymous reporting will result in a greater probability that a violation will be reported if it is detected.
9. Supporting Information which provides users, staff, and management with contact information for each type of policy violation; guidelines on how to handle outside queries about a security incident, or information which may be considered confidential or proprietary; and cross-references to security procedures and related information, such as company policies and governmental laws and regulations.

There may be regulatory requirements that affect some aspects of your security policy (e.g., line monitoring). The creators of the security policy should consider seeking legal assistance in the creation of the policy. At a minimum, legal counsel should review the policy.

Once the security policy has been established it should be clearly communicated to users, staff, and management. Having all personnel sign a statement indicating that they have read, understood, and agreed to abide by the policy is an important part of the process. Finally, the policy should be reviewed on a regular annual basis to see if it is successfully supporting your security needs.

Defining Security

Computer security means to protect information, a core business asset. It deals with the prevention and detection of unauthorized actions by users of a computer. This has been extended to include privacy, confidentiality, and integrity. For example:

- ?? Chinese Foreign Ministry spokesman Zhu Bangzao rejected allegations that China stole U.S. nuclear secrets, saying such claims are meant to undermine China-U.S. relations. Meanwhile, a CIA-led task force was assessing how much damage may have been done to U.S. national security after a Chinese scientist at the Los Alamos National Laboratory in New Mexico allegedly shared nuclear.
- ?? Two parties agree and seal their transaction using digital signatures. The signature cannot be ruled invalid by state legislature or other law making bodies because it uniquely identifies the individuals involved.
- ?? You visit a Web site and the site collects more personal information than you are willing to divulge or the site distributes data to outside parties. By doing this, it compromises your privacy and opens your world to other parties.

This definition implies that you have to know the information and the value of that information in order to develop protective measures. You also need to know to which individuals need unique identities and how much information may be divulged to the outside world. A rough classification of protective measures in computer security is as follows:

- ?? **Prevention**—Take measures that prevent your information from being damaged, altered, or stolen. Preventive measures can range from locking the server room door to setting up high-level security policies.
- ?? **Detection**—Take measures that allow you to detect when information has been damaged, altered, or stolen, how it has been damaged, altered, or stolen, and who has caused the damage. Various tools are available to help detect intrusions, damage or alterations, and viruses.
- ?? **Reaction**—Take measures that allow recovery of information, even if information is lost or damaged.

The above measures are all very well, but if you do not understand how information may be compromised, you cannot take measures to protect it. You must examine the components on how information can be compromised:

- ?? **Confidentiality**. The prevention of unauthorized disclosure of information. This can be the result of poor security measures or information leaks by personnel. An example of poor security measures would be to allow anonymous access to sensitive information.
- ?? **Integrity**. The prevention of erroneous modification of information. Authorized users are probably the biggest cause of errors and omissions and the alteration of data. Storing incorrect data within the system can be as damaging as losing data. Malicious attackers also can modify, delete, or corrupt information that is vital to the correct operation of business functions.
- ?? **Availability**. The prevention of unauthorized withholding of information or resources. This does not apply just to personnel withholding information. Information should be as freely available as possible to authorized users.
- ?? **Authentication**. The process of verifying that users are who they claim to be when logging onto a system. Generally, the use of user names and passwords accomplishes this. More sophisticated is the use of smart cards and retina scanning. The process of authentication does not grant the user access rights to resources—this is achieved through the authorization process.

- ?? **Authorization.** The process of allowing only authorized users access to sensitive information. An authorization process uses the appropriate security authority to determine whether a user should have access to resources.

Reviewing Current Policies

Establishing an effective set of security policies and controls requires using a strategy to determine the vulnerabilities that exist in computer systems and in the current security policies and controls that guard them. The current status of computer security policies can be determined by reviewing the list of documentation that follows. The review should take notice of areas where policies are lacking as well as examine documents that exist:

- ?? Physical computer security policies such as physical access controls.
- ?? Network security policies (for example, e-mail and Internet policies).
- ?? Data security policies (access control and integrity controls).
- ?? Contingency and disaster recovery plans and tests.
- ?? Computer security awareness and training.
- ?? Computer security management and coordination policies.
- ?? Other documents that contain sensitive information such as:
 - ~~///~~ Computer BIOS passwords.
 - ~~///~~ Router configuration passwords.
 - ~~///~~ Access control documents.
 - ~~///~~ Other device management passwords.

Protective Approaches

Deny all/ Allow all

There are two diametrically opposed underlying philosophies that can be adopted when defining a security plan. Both are legitimate models, and the choice between them will depend on the site and its needs for security.

The first option is to turn off all services and then selectively enable services on a case-by-case basis, as they are needed. This can be done at the host or network level as appropriate. This model, which will here after be referred to as the "deny all" model, is generally more secure than the other model described in the next paragraph. More work is required to successfully implement a "deny all" configuration as well as a better understanding of services. Allowing only known services provides for a better analysis of a particular service/protocol and the design of a security mechanism suited to the security level of the site. In current firewall philosophies, this is sometimes referred to as the "paranoid" approach.

The other model, which will here after be referred to as the "allow all" model, is much easier to implement, but is generally less secure than the "deny all" model. Simply turn on all services, usually the default at the host level, and allow all protocols to travel across network boundaries, usually the default at the router level. As security holes become apparent, they are restricted or patched at either the host or network level. In current firewall philosophies, this is sometimes referred to as the "permissive" approach.

Each of these models can be applied to different portions of the site, depending on functionality requirements, administrative control, site policy, etc. For example, the policy may be to use the "allow all" model when setting up workstations for general use, but adopt a "deny all" model

when setting up information servers, like an email hub. Likewise, an "allow all" policy may be adopted for traffic between LAN's internal to the site, but a "deny all" policy can be adopted between the site and the Internet.

Be careful when mixing philosophies. Many companies are willing to pay the cost of security for their external traffic and require strong security measures, but are unwilling or unable to provide similar protections internally. This works fine as long as the outer defenses are never breached and the internal users can be trusted. Once the perimeter or firewall is breached, subverting the internal network is trivial.

Protecting the Network

There are several problems to which networks are vulnerable. The classic problem is a "denial of service" attack. In this case, the network is brought to a state in which it can no longer carry legitimate users' data. Attacking the routers and flooding the network with extraneous traffic typically accomplish this. Please note that the term "router" is used as an example of a larger class of active network interconnection components that also includes components like firewalls, proxy-servers, etc.

An attack on the router is designed to cause it to stop forwarding packets, or to forward them improperly. The former case may be due to a misconfiguration, the injection of a spurious routing update, or a "flood attack" (i.e., the router is bombarded with unroutable packets, causing its performance to degrade). A flood attack on a network is similar to a flood attack on a router, except that the flood packets are usually broadcast. An ideal flood attack would be the injection of a single packet, which exploits some known flaw in the network nodes and causes them to retransmit the packet, or generate error packets, each of which is picked up and repeated by another host. A well-chosen attack packet can even generate an exponential explosion of transmissions.

Another classic problem is "spoofing." In this case, spurious routing updates are sent to one or more routers causing them to misroute packets. This differs from a denial of service attack only in the purpose behind the spurious route. In denial of service, the object is to make the router unusable; a state which will be quickly detected by network users. In spoofing, the spurious route will cause packets to be routed to a host from which an intruder may monitor the data in the packets. These packets are then re-routed to their correct destinations. However, the intruder may or may not have altered the contents of the packets.

Unfortunately, there is no adequate protection against a flooding attack, or a misbehaving host or router that is flooding the network. Fortunately, this type of attack is obvious when it occurs and can usually be terminated relatively quickly.

Protecting the Services

There are many types of services and each has its own security requirements. These requirements will vary based on the intended use of the service. For example, a service that should only be usable internally may require different protection mechanisms than a service provided for external use. It may be sufficient to protect the internal server from external access. However, a web server, which provides a home page intended for viewing by users anywhere on the Internet, requires built-in protection. That is, the service/protocol/server must provide whatever security may be required to prevent unauthorized access and modification of the Web database.

Internal services (i.e., services meant to be used only by users within a site) and external services (i.e., services deliberately made available to users outside a site) will, in general, have protection requirements that differ. It is therefore wise to isolate the internal services to one set of server host computers and the external services to another set of server host computers. That is, internal and external servers should not be co-located on the same host computer. In fact, many sites go so far as to have one set of subnets (or even different networks) that are accessible from the outside and another set that may be accessed only within the site. Of course, there is usually a firewall that connects these partitions. This is frequently referred to as a demilitarized zone (DMZ). Proper operation of the firewall is crucial to success in this configuration.

Intranets connect different parts of an organization (e.g., divisions of a company). While this document generally differentiates between external and internal (public and private), sites using intranets should be aware that they would need to consider three separations and take appropriate actions when designing and offering services. A service offered to an intranet would be neither public, nor as completely private as a service to a single organizational subunit. Therefore, the service would need its own supporting system, separated from both external and internal services and networks. Virtual private network technologies have been widely used to safely deploy intranets over the Internet infrastructure. In the past, these networks were often transported over a corporate frame relay network.

One form of external service deserves some special consideration, and that is anonymous, or guest, access. This may be either anonymous FTP or guest (unauthenticated) login. It is extremely important to ensure that anonymous FTP servers and guest login userids are carefully isolated from any hosts and file systems from which outside users should be kept. Another area to which special attention must be paid concerns anonymous, writable access. A site may be legally responsible for the content of publicly available information; so careful monitoring of the information deposited by anonymous users is advised.

Now we shall consider some of the most popular services: name service, password/key service, authentication/proxy service, electronic mail, WWW, and file transfer. Since these are the most frequently used services, they are the most obvious points of attack. Also, a successful attack on one of these services can produce disaster all out of proportion to the innocence of the basic service.

Electronic Mail

Electronic mail (email) systems have long been a source for intruder break-ins because email protocols are among the oldest and most widely deployed services. Also, an email server requires access to the outside world; most email servers accept input from any source. An email server generally consists of two parts: a receiving/sending agent and a processing agent. Since email is delivered to all users, and is usually private, the processing agent typically requires system privileges to deliver the mail. Most email implementations perform both portions of the service, which means the receiving agent also has system privileges. This opens several potential security gaps that this document will not describe. There are implementations available which allow a separation of the two agents. These implementations are generally considered more secure, but still require careful installation to avoid creating a security problem.

World Wide Web (WWW)

The Web is growing in popularity exponentially because of its ease of use and the powerful ability to concentrate information services. Most WWW servers accept some type of direction

and action from the persons accessing their services. The most common example is taking a request from a remote user and passing the provided information to a program running on the server to process the request. Some of these programs are not written with security in mind and can create security holes. If a Web server is available to the Internet community, it is especially important that confidential information not be co-located on the same host as that server. In fact, it is recommended that the server have a dedicated host that is not "trusted" by other internal hosts.

Many sites may want to co-locate FTP service with their WWW service. But this should only occur for anon-ftp servers that only provide information (ftp-get). Anon-ftp puts, in combination with WWW, might be dangerous (e.g., they could result in modifications to the information your site is publishing to the web) and in themselves make the security considerations for each service different.

File Transfer (FTP, TFTP)

FTP and TFTP both allow users to receive and send electronic files in a point-to-point manner. However, FTP requires authentication while TFTP requires none. For this reason, TFTP should be avoided as much as possible.

Improperly configured FTP servers can allow intruders to copy, replace and delete files at will, anywhere on a host, so it is very crucial to configure this service correctly. Access to encrypted passwords and proprietary data, and the introduction of Trojan horses are just a few of the potential security holes that can occur when the service is configured incorrectly. FTP servers should reside on their own host. Some sites choose to co-locate FTP with a Web server, since the two protocols share common security considerations. This practice is not recommended, especially when the FTP service allows the deposit of files (see section on WWW above). As previously stated, services offered internally should not be co-located with services offered externally. Each should have its own separate host.

TFTP does not support the same range of functions as FTP, and has no security whatsoever. This service should only be considered for internal use, and then it should be configured in a restricted way so that the server only has access to a set of predetermined files (instead of every world-readable file on the system). Probably the most common usage of TFTP is for downloading router configuration files to a router. TFTP should reside on its own host, and should not be installed on hosts supporting external FTP or Web access. TFTP generally remains a tool used solely within an IT group.

Protecting the Protection

It is amazing how often a site will overlook the most obvious weakness in its security by leaving the security server itself open to attack. Based on considerations previously discussed, it should be clear that: the security server should not be accessible from off-site; should offer minimum access, except for the authentication function, to users on-site; and should not be co-located with any other servers. Further, all access to the node, including access to the service itself, should be logged to provide a "paper trail" in the event of a security breach.

Password Assurance

While the need to eliminate the use of standard, reusable passwords cannot be overstated, it is recognized that some organizations may still be using them. While it's recommended that these

organizations transition to the use of better technology, in the mean time, we have the following advice to help with the selection and maintenance of traditional passwords. But remember, none of these measures provides protection against disclosure due to sniffer programs.

The importance of robust passwords - In most cases of system penetration, the intruder needs to gain access to an account on the system. One way to accomplish that is through guessing the password of a legitimate user. This is often performed by running an automated password-cracking program, which uses a very large dictionary, against the system's password file. The only way to guard against passwords being disclosed in this manner is through the careful selection of passwords that cannot be easily guessed (i.e., combinations of numbers, letters, and punctuation characters). Passwords should also be as long as the system supports and users can tolerate.

1. Changing default passwords - Many operating systems and application programs are installed with default accounts and passwords. These must be changed immediately to something that cannot be guessed or cracked.
2. Restricting access to the password file - In particular, a site wants to protect the encrypted password portion of the file so that would-be intruders don't have them available for cracking. One effective technique is to use shadow passwords where the password field of the standard file contains a dummy or false password. The file(s) containing the legitimate passwords are protected elsewhere on the system.
3. Password aging - When and how to expire passwords is still a subject of controversy among the security community. It is generally accepted that a password should not be maintained once an account is no longer in use, but it is hotly debated whether a user should be forced to change a good password that's in active use. The arguments for changing passwords relate to the prevention of the continued use of penetrated accounts. However, the opposition claims that frequent password changes lead to users writing down their passwords in visible areas (such as pasting them to a terminal), or to users selecting very simple passwords that are easy to guess. It should also be stated that an intruder would probably use a captured or guessed password sooner rather than later, in which case password aging provides little if any protection.

While there is no definitive answer to this dilemma, a password policy should directly address the issue and provide guidelines for how often a user should change the password. Monthly password changes are common business practice today. It is recommended that passwords be changed whenever a privileged account is compromised, there is a critical change in personnel (especially if it is an administrator), or when an account has been compromised. In addition, if a privileged account password is compromised, *all passwords on the system should be changed.*

4. Password/account blocking - Some sites find it useful to disable accounts after a predefined number of failed attempts to authenticate. If this approach is taken, the mechanism should not "advertise" itself. After disabling, even if the correct password is presented, the message displayed should remain that of a failed login attempt. Implementing this mechanism will require that legitimate users contact their system administrator to request that their account be reactivated.
5. A word about the finger daemon - By default, the finger daemon displays considerable system and user information. For example, it can display a list of all users currently using a system. This information can be used by potential intruders to identify legitimate usernames. It is recommended that sites consider modifying finger to restrict the information displayed or disable it entirely.

Auditing

This section covers the procedures for collecting data generated by network activity, which may be useful in analyzing the security of a network and responding to security incidents.

What to Collect

Audit data should include any attempt to achieve a different security level by any person, process, or other entity in the network. This includes login and logout, super user access (or the non-UNIX equivalent), ticket generation (for Kerberos, for example), and any other change of access or status. It is especially important to note "anonymous" or "guest" access to public servers.

The actual data to collect will differ for different sites and for different types of access changes within a site. In general, the information you want to collect includes: username and hostname, for login and logout; previous and new access rights, for a change of access rights; and a timestamp. Of course, there is much more information that might be gathered, depending on what the system makes available and how much space is available to store that information.

One very important note: do not gather passwords. This creates an enormous potential security breach if the audit records should be improperly accessed. Do not gather incorrect passwords either, as they often differ from valid passwords by only a single character or transposition.

Collection Process

The host or resource being accessed should enact the collection process. Depending on the importance of the data and the need to have it local in instances in which services are being denied, data could be kept local to the resource until needed or be transmitted to storage after each event.

There are basically three ways to store audit records: in a read/write file on a host, on a write-once/read-many device (e.g., a CD-ROM or a specially configured tape drive), or on a write-only device (e.g., a line printer). Each method has advantages and disadvantages.

File system logging is the least resource intensive of the three methods and the easiest to configure. It allows instant access to the records for analysis, which may be important if an attack is in progress. File system logging is also the least reliable method. If the logging host has been compromised, the file system is usually the first thing to go; an intruder could easily cover up traces of the intrusion.

Collecting audit data on a write-once device is slightly more effort to configure than a simple file, but it has the significant advantage of greatly increased security because an intruder could not alter the data showing that an intrusion has occurred. The disadvantage of this method is the need to maintain a supply of storage media and the cost of that media. Also, the data may not be instantly available.

Line printer logging is useful in system where permanent and immediate logs are required. A real time system is an example of this, where the exact point of a failure or attack must be recorded. A laser printer, or other device which buffers data (e.g., a print server), may suffer from lost data if buffers contain the needed data at a critical instant. The disadvantage of, literally, "paper trails" is the need to keep the printer fed and the need to scan records by hand. There is also the issue of where to store the, potentially, enormous volume of paper that may be generated.

For each of the logging methods described, there is also the issue of securing the path between the device generating the log and actual logging device (i.e., the file server, tape/CD-ROM drive, printer). If that path is compromised, logging can be stopped or spoofed or both. In an ideal world, the logging device would be directly

Ongoing Activities

1. At this point in time, your site has hopefully developed a complete security policy and has developed procedures to assist in the configuration and management of your technology in support of those policies. How nice it would be if you could sit back and relax at this point and know that you were finished with the job of security. Unfortunately, that isn't possible. Your systems and networks are not a static environment, so you will need to review policies and procedures on a regular basis. There are a number of steps you can take to help you keep up with the changes around you so that you can initiate corresponding actions to address those changes. The following is a starter set and you may add others as appropriate for your site.
2. Subscribe to advisories that are issued by various security incident response teams, like those of the CERT Coordination Center, and update your systems against those threats that apply to your site's technology.
3. Monitor security patches that are produced by the vendors of your equipment, and obtain and install all that apply.
4. Actively watch the configurations of your systems to identify any changes that may have occurred, and investigate all anomalies.
5. Review all security policies and procedures annually (at a minimum).
6. Read relevant mailing lists and USENET newsgroups to keep up to date with the latest information being shared by fellow administrators.
7. Regularly check for compliance with policies and procedures. Someone other than the people who define or implement the policies and procedures should perform this audit.

Best Practices Checklist

Companies that use the Internet for business purposes develop detailed security plans and implement comprehensive security policies to protect themselves from potential risks. However, you do not have to go that far in order to have a safe experience on the Internet. Microsoft recommends that you take a few minutes to think about how you use your computer and what ways you use the Internet.

When you think about computer security, think of a wall of protection. The wall is not built with one giant brick but is made up of many smaller bricks. When all of the bricks are used together, the security becomes stronger. In fact, Microsoft has published the Ten Immutable Laws of Security to help our customers understand the core elements of a security strategy.

Microsoft has provided the following best-practices checklist to help you identify the bricks that you can use to build your wall of protection.

<input type="checkbox"/>	Use Antivirus Software
<input type="checkbox"/>	Use Strong Passwords
<input type="checkbox"/>	Verify Your Software Security Settings
<input type="checkbox"/>	Product Security Updates

<input type="checkbox"/>	Personal Firewalls
<input type="checkbox"/>	Back Up Early and Often
<input type="checkbox"/>	Protect Against Power Surges and Loss

Appendix A - When Bad Things Happen

Proactive and Reactive Strategies

The security plans should include both *proactive* and *reactive* strategies.

The *proactive* or pre-attack strategy is a set of steps that helps to minimize existing security policy vulnerabilities and develop contingency plans. Determining the damage that an attack will cause on a system and the weaknesses and vulnerabilities exploited during this attack helps in developing the proactive strategy.

The *reactive* strategy or post-attack strategy helps security personnel to assess the damage caused by the attack, repair the damage or implement the contingency plan developed in the proactive strategy, document and learn from the experience, and get business functions running as soon as possible.

Testing

The last element of a security strategy, testing and reviewing the test outcomes, is carried out after the reactive and proactive strategies have been put into place. Performing simulation attacks on a test or lab system makes it possible to assess where the various vulnerabilities exist and adjust security policies and controls accordingly.

These tests should not be performed on a live production system because the outcome could be disastrous. Yet, the absence of labs and test computers due to budget restrictions might preclude simulating attacks. In order to secure the necessary funds for testing, it is important to make management aware of the risks and consequences of an attack as well as the security measures that can be taken to protect the system, including testing procedures. If possible, all attack scenarios should be physically tested and documented to determine the best possible security policies and controls to be implemented.

Reactive Strategy

A reactive strategy is implemented when the proactive strategy for the attack has failed. The reactive strategy defines the steps that must be taken after or during an attack. It helps to identify the damage that was caused and the vulnerabilities that were exploited in the attack, determine why it took place, repair the damage that was caused by it, and implement a contingency plan if one exists. Both the reactive and proactive strategies work together to develop security policies and controls to minimize attacks and the damage caused during them.

The incident response team should be included in the steps taken during or after the attack to help assess it and to document and learn from the event.

Assess the Damage

Determine the damage that was caused during the attack. This should be done as swiftly as possible so that restore operations can begin. If it is not possible to assess the damage in a timely manner, a contingency plan should be implemented so that normal business operations and productivity can continue.

Determine the Cause of the Damage

To determine the cause of the damage it is necessary to understand what resources the attack was aimed at and what vulnerabilities were exploited to gain access or disrupt services. Review system logs, audit logs, and audit trails. These reviews often help in discovering where the attack originated in the system and what other resources were affected.

Repair the Damage

It is very important that the damage be repaired as quickly as possible in order to restore normal business operations and any data lost during the attack. The organization's disaster recovery plans and procedures (discussed in "Security Planning") should cover the restore strategy. The incident response team should also be available to handle the restore and recovery process and to provide guidance on the recovery process.

Document and Learn

It is important that once the attack has taken place, it is documented. Documentation should cover all aspects of the attack that are known, including: the damage that is caused (hardware, software, data loss, loss in productivity), the vulnerabilities and weaknesses that were exploited during the attack, the amount of production time lost, and the procedures taken to repair the damage. Documentation will help to modify proactive strategies for preventing future attacks or minimizing damages.

Your goals will be largely determined by the following key tradeoffs:

1. services offered versus security provided - Each service offered to users carries its own security risks. For some services the risk outweighs the benefit of the service and the administrator may choose to eliminate the service rather than try to secure it.
2. ease of use versus security - The easiest system to use would allow access to any user and require no passwords; that is, there would be no security. Requiring passwords makes the system a little less convenient, but more secure. Requiring a device-generated one-time password makes the system even more difficult to use, but much more secure.
3. cost of security versus risk of loss - There are many different costs to security: monetary (i.e., the cost of purchasing security hardware and software like firewalls and one-time password generators), performance (i.e., encryption and decryption take time), and ease of use (as mentioned above). There are also many levels of risk: loss of privacy (i.e., the reading of information by unauthorized individuals), loss of data (i.e., the corruption or erasure of information), and the loss of service (e.g., the filling of data storage space, usage of computational resources, and denial of network access). Each type of cost must be weighed against each type of loss.

Your goals should be communicated to all users, operations staff, and managers through a set of security rules, called a "security policy." We are using this term, rather than the narrower "computer security policy" since the scope includes all types of information technology and the information stored and manipulated by the technology.

Methods, Tools, and Techniques for Attacks

Attacks = motive + method + vulnerability.

The method in this formula exploits the organization's vulnerability in order to launch an attack as shown in Figure 2. Malicious attackers can gain access or deny services in numerous ways. Here are some of them:

- ?? *Viruses.* Attackers can develop harmful code known as viruses. Using hacking techniques, they can break into systems and plant viruses. Viruses in general are a threat to any environment. They come in different forms and although not always malicious, they always take up time. Viruses can also be spread via e-mail and disks.
- ?? *Trojan horses.* These are malicious programs or software code hidden inside what looks like a normal program. When a user runs the normal program, the hidden code runs as well. It can then start deleting files and causing other damage to the computer. Trojan horses are normally spread by e-mail attachments. The Melissa virus that caused denial-of-service attacks throughout the world in 1999 was a type of Trojan horse.
- ?? *Worms.* These are programs that run independently and travel from computer to computer across network connections. Worms may have portions of themselves running on many different computers. Worms do not change other programs, although they may carry other code that does.
- ?? *Password cracking.* This is a technique attackers use to surreptitiously gain system access through another user's account. This is possible because users often select weak passwords. The two major problems with passwords are when they are easy to guess based on knowledge of the user (for example, wife's maiden name) and when they are susceptible to dictionary attacks (that is, using a dictionary as the source of guesses).
- ?? *Denial-of-service attacks.* This attack exploits the need to have a service available. It is a growing trend on the Internet because Web sites in general are open doors ready for abuse. People can easily flood the Web server with communication in order to keep it busy. Therefore, companies connected to the Internet should prepare for (DoS) attacks. They also are difficult to trace and allow other types of attacks to be subdued.
- ?? *E-mail hacking.* Electronic mail is one of the most popular features of the Internet. With access to Internet e-mail, someone can potentially correspond with any one of millions of people worldwide. Some of the threats associated with e-mail are:
- ?? *Impersonation.* The sender address on Internet e-mail cannot be trusted because the sender can create a false return address. Someone could have modified the header in transit, or the sender could have connected directly to the Simple Mail Transfer Protocol (SMTP) port on the target computer to enter the e-mail.
- ?? *Eavesdropping.* E-mail headers and contents are transmitted in the clear text if no encryption is used. As a result, the contents of a message can be read or altered in transit. The header can be modified to hide or change the sender, or to redirect the message.
- ?? *Packet replay.* This refers to the recording and retransmission of message packets in the network. Packet replay is a significant threat for programs that require authentication sequences, because an intruder could replay legitimate authentication sequence messages

- to gain access to a system. Packet replay is frequently undetectable, but can be prevented by using packet time stamping and packet sequence counting.
- ?? *Packet modification.* This involves one system intercepting and modifying a packet destined for another system. Packet information may not only be modified, it could also be destroyed.
 - ?? *Eavesdropping.* This allows a cracker (hacker) to make a complete copy of network activity. As a result, a cracker can obtain sensitive information such as passwords, data, and procedures for performing functions. It is possible for a cracker to eavesdrop by wiretapping, using radio, or using auxiliary ports on terminals. It is also possible to eavesdrop using software that monitors packets sent over the network. In most cases, it is difficult to detect eavesdropping.
 - ?? *Social engineering.* This is a common form of cracking. It can be used by outsiders and by people within an organization. Social engineering is a hacker term for tricking people into revealing their password or some form of security information.
 - ?? *Intrusion attacks.* In these attacks, a hacker uses various hacking tools to gain access to systems. These can range from password-cracking tools to protocol hacking and manipulation tools. Intrusion detection tools often can help to detect changes and variants that take place within systems and networks.
 - ?? *Network spoofing.* In network spoofing, a system presents itself to the network as though it were a different system (computer A impersonates computer B by sending B's address instead of its own). The reason for doing this is that systems tend to operate within a group of other trusted systems. Trust is imparted in a one-to-one fashion; computer A trusts computer B (this does not imply that system B trusts system A). Implied with this trust is that the system administrator of the trusted system is performing the job properly and maintaining an appropriate level of security for the system. Network spoofing occurs in the following manner: if computer A trusts computer B and computer C spoofs (impersonates) computer B, then computer C can gain otherwise-denied access to computer A.

Security Vulnerabilities

As explained previously, a malicious attacker uses a *method* to exploit *vulnerabilities* in order to achieve a *goal*. Vulnerabilities are weak points or loopholes in security that an attacker exploits in order to gain access to the network or to resources on the network (see Figure 2). Remember that the vulnerability is not the attack, but rather the weak point that is exploited. Some weak points are:

- ?? *Passwords.* Password selection will be a contentious point as long as users have to select one. The problem usually is remembering the correct password from among the multitude of passwords a user needs to remember. Users end up selecting commonly used passwords because they are easy to remember. Anything from birthdays to the names of loved ones. This is a vulnerability because it gives others a good chance to guess the correct password.
- ?? *Protocol design.* Communication protocols sometimes have weak points. Attackers use these to gain information and eventually gain access to systems. Some known issues are:
 - *TCP/IP.* The TCP/IP protocol stack has some weak points that allow:
 - IP address spoofing
 - TCP connection request (SYN) attacks

- ?? *Telnet protocol.* Telnet can be used to administer systems running Microsoft® Windows® 2000 and Unix. When using the telnet client to connect from a Microsoft system to UNIX system and vice versa, user names and passwords are transmitted in clear text.
- ?? *File Transfer Protocol (FTP).* As with Telnet, if the FTP service is running and users need to send or retrieve information from a secure location then user names and passwords are transmitted in clear text.
- ?? *Commands revealing user information.* It is not uncommon to find interoperability between Microsoft products and various versions of UNIX. Commands that reveal user and system information pose a threat because crackers can use that information to break into a system. Here are some ways:
- ?? *Finger.* The finger client utility on Microsoft Windows NT® and Windows 2000 can be used to connect to a finger daemon service running on a UNIX-based computer to display information about users. When the finger program is run with no arguments, information for every user currently logged on to the system is displayed.
- ?? *Rexec.* The rexec utility is provided as a client on Microsoft Windows NT and Windows 2000. The rexec client utility allows remote execution on UNIX-based systems running the rexecd service. A client transmits a message specifying the user name, the password, and the name of a command to execute. The rexecd program is susceptible to abuse because it can be used to probe a system for the names of valid accounts. In addition, passwords are transmitted unencrypted over the network.
- ?? *Asynchronous transfer mode (ATM).* Security can be compromised by what is referred to as "manhole manipulation"—direct access to network cables and connections in underground parking garages and elevator shafts.
- ?? *Frame relay.* Similar to the ATM problem.
- ?? *Device administration.* Switches and routers are easily managed by an HTTP interface or through a command line interface. Coupled to the use of weak passwords (for example, public passwords), it allows anybody with some technical knowledge to take control of the device.

Modems. Modems have become standard features on many desktop computers. Any unauthorized modem is a serious security concern. People use them not just to connect to the Internet, but also to connect to their office so they can work from home. The problem is that a modem is a means of bypassing the "firewall" that protects a network from outside intruders. A hacker using a "war dialer" tool to identify the modem telephone number and a "password cracker" tool to break a weak password can gain access to the system. Due to the nature of computer networking, once a hacker connects to that one computer, the hacker can often connect to any other computer in the network.

Appendix B – Incident Response

We spend a lot of time trying to prevent security incidents from occurring. What sometimes gets lost in all of this preparation is plans for dealing with an incident should it actually occur. This document is a synopsis of several incident-handling guides that provides a high-level framework for dealing with either the realization that a system has been compromised *or* the recognition that a system is under active attack. The **SANS Reading Room** has a large set of papers about incident response at http://www.sans.org/infosecFAQ/incident/incident_list.htm.

1 - Remain calm

Or, in the words of the Hitchhiker, "Don't panic!" (with apologies to Douglas Adams, author of *The Hitchhiker's Guide to the Galaxy*). To successfully handle any perceived emergency situation, you must **remain calm** so that you can assess what's going on around you and react in a methodical manner. A compromised system/network or an attacker on the loose demands well-thought out action; and frankly, the bad guys have probably been in your computer for days or maybe even longer, and another few minutes won't make much difference. And you're probably going to have to rebuild your compromised servers anyway...

2 - Notify your organization's management and activate your response plan to get help

Your security policies should identify the pecking order of who gets called when if there's a security event. Individuals with particular responsibility for the affected server(s) and/or network(s) should be notified, as well as any information security personnel. The severity of the incident (and your own policies) will dictate who else is brought in: your ISP, department head, corporate officers, the press, law enforcement, consultants, response centers, etc. Notify whoever is *necessary* to assess the situation and get it under control — but it is generally best to maintain a "need to know" stance and communicate, at least initially, with only the necessary parties.

Whenever possible, use telephones and faxes during a computer security incident. If the attackers have full access to your computer, then they can possibly (probably?) read your mail. If you use your computer, this allows them to know when you report the incident and what response you got. There is a real possibility that other systems at your site have also been compromised and one or more packet sniffers are running on your network. So, if you absolutely must use a computer to communicate and you are fairly certain it can't be intercepted, then use a different system and/or dial-up ISP access if possible.

3 - Take good notes

This cannot be stressed enough — *document, document, document!!* Maintain a log of everything you see and do, everyone you speak with, and the team working with you. This will not only help you in criminal cases (and in remembering the events at a trial that might take place a year or two down the road), but also helps in the investigation/forensics process, post-event analysis, and as an educational/intelligence gathering vehicle for others in the infosec community. Notes should be detailed, organized, and complete, and should reflect the basic "who, what, where, when, and how" ("why" *might* be left for later on). Keep copies of any altered files before restoring your system!!

4 - Contain the problem

Take any necessary steps to keep the problem contained and prevent it from spreading to other systems and/or networks. This may well involve disconnecting the compromised system from your network and/or disconnecting your network from the Internet. Containment may require a physical disconnect or might be accomplished while you clean up and recover; circumstances, including whether you are dealing with an active attack or the aftermath of an attack, will dictate what is a prudent action. Note that the latter approach (containing the problem while still on-line) might well leave you vulnerable to additional attacks.

5 - Gather evidence and make backups

For purposes of learning what happened and to have evidence for future analysis (and possible prosecution), make backups of operating system and file system information, as well as any state and network information (e.g., output from *netstat* or *route*). Keep a detailed history of this activity if you have even the least suspicion that this information will be used in a criminal or civil trial; digital signatures and file timestamps are part of the procedures you should follow to maintain the custody chain. If possible, coordinate your evidence gathering with that of a second source, such as an ISP or another network (if you detect another network that is involved in this incident). Finally, as you make the backups, consider where they are going and who will be using them; if possible, make multiple copies and secure one for historical purposes while analyzing/sharing the other(s).

6 - Get rid of the problem and maintain your business

This step might be easier said than done. If your server has been compromised, you should totally rebuild it from scratch unless you are 100% certain what the entire problem is. Before you can eliminate the problem, you need to be sure that you understand the cause of the incident. What vulnerability did the intruder use to gain access and what have you done to prevent another attack? You should rebuild the server and applications from original media.

The next issue, of course, is that of re-installing content. Note that some files that were exploited might have been on your system for some time already and, therefore, might have been backed up as part of your regular operations. So, although you've rebuilt the operating system and applications software, you might very well reinstall files that can be exploited over and over. Again, it is imperative that we understand how this incident happened to avoid this eventuality. Finally, business continuity is a major issue. Get rid of the problem and get your server back on line as soon as possible.

7 - Do a post mortem

Once the situation is resolved and you're back in operation, get all relevant parties together to review the incident and the response. Review your security policies and operational procedures to see what changes, if any, are required. To the extent possible, contact appropriate incident response agencies, such as CERT/CC or incidents.org, and share your knowledge with them.

Hold this meeting a day or two after the incident is deemed "over," when everyone is rested and has had time to reflect on what happened and why, and what went well and what didn't go so well. Don't do it immediately while people are still tired and don't wait weeks when people will forget — and will have moved onto other things.

Appendix C - The Ten Immutable Laws of Security- Microsoft

Law #1: If a bad guy can persuade you to run his program on your computer, it's not your computer anymore.

It's an unfortunate fact of computer science: when a computer program runs, it will do what it's programmed to do, even if it's programmed to be harmful. When you choose to run a program, you are making a decision to turn over control of your computer to it. Once a program is running, it can do anything, up to the limits of what you yourself can do on the machine. It could monitor your keystrokes and send them to a web site. It could open every document on the machine, and change the word "will" to "won't" in all of them. It could send rude emails to all your friends. It could install a virus. It could create a "back door" that lets someone remotely control your machine. It could dial up an ISP in Katmandu. Or it could just reformat your hard drive.

That's why it's important to never run, or even download, a program from an untrusted source – and by "source", I mean the person who wrote it, not the person who gave it to you. There's a nice analogy between running a program and eating a sandwich. If a stranger walked up to you and handed you a sandwich, would you eat it? Probably not. What if your best friend gave you a sandwich? Maybe you would, maybe you wouldn't – it depends on whether she made it or found it lying in the street. Apply the same critical thought to a program that you would to a sandwich, and you'll usually be safe.

Law #2: If a bad guy can alter the operating system on your computer, it's not your computer anymore.

In the end, an operating system is just a series of ones and zeroes that, when interpreted by the processor, cause the machine to do certain things. Change the ones and zeroes, and it will do something different. Where are the ones and zeroes stored? Why, on the machine, right along with everything else! They're just files, and if other people who use the machine are permitted to change those files, it's "game over".

To understand why, consider that operating system files are among the most trusted ones on the computer, and they generally run with system-level privileges. That is, they can do absolutely anything. Among other things, they're trusted to manage user accounts, handle password changes, and enforce the rules governing who can do what on the computer. If a bad guy can change them, the now-untrustworthy files will do his bidding, and there's no limit to what he can do. He can steal passwords, make himself an administrator on the machine, or add entirely new functions to the operating system. To prevent this type of attack, make sure that the system files (and the registry, for that matter) are well protected.

Law #3: If a bad guy has unrestricted physical access to your computer, it's not your computer anymore.

Oh, the things a bad guy can do if he can lay his hands on your computer! Here's a sampling, going from Stone Age to Space Age:

- ?? He could mount the ultimate low-tech denial of service attack, and smash your computer with a sledgehammer.
- ?? He could unplug the computer, haul it out of your building, and hold it for ransom.
- ?? He could boot the computer from a floppy disk, and reformat your hard drive. But wait, you say, I've configured the BIOS on my computer to prompt for a password when I turn the power on. No problem – if he can open the case and get his hands on the system hardware, he could just replace the BIOS chips. (Actually, there are even easier ways).
- ?? He could remove the hard drive from your computer, install it into his computer, and read it.
- ?? He could make a duplicate of your hard drive and take it back his lair. Once there, he'd have all the time in the world to conduct brute-force attacks, such as trying every possible logon password. Programs are available to automate this and, given enough time, it's almost certain that he would succeed. Once that happens, Laws #1 and #2 above apply
- ?? He could replace your keyboard with one that contains a radio transmitter. He could then monitor everything you type, including your password.

Always make sure that a computer is physically protected in a way that's consistent with its value – and remember that the value of a machine includes not only the value of the hardware itself, but also the value of the data on it, **and** the value of the access to your network that a bad guy could gain. At a minimum, business-critical machines like domain controllers, database servers, and print/file servers should always be in a locked room that only people charged with administration and maintenance can access. But you may want to consider protecting other machines as well, and potentially using additional protective measures.

If you travel with a laptop, it's absolutely critical that you protect it. The same features that make laptops great to travel with – small size, lightweight, and so forth – also make them easy to steal. There are a variety of locks and alarms available for laptops, and some models let you remove the hard drive and carry it with you. You also can use features like the Encrypting File System in Windows 2000 to mitigate the damage if someone succeeded in stealing the computer. But the only way you can know with 100% certainty that your data is safe and the hardware hasn't been tampered with is to keep the laptop on your person at all times while traveling.

Law #4: If you allow a bad guy to upload programs to your web site, it's not your web site any more.

This is basically Law #1 in reverse. In that scenario, the bad guy tricks his victim into downloading a harmful program onto his machine and running it. In this one, the bad guy uploads a harmful program to a machine and runs it himself. Although this scenario is a danger anytime you allow strangers to connect to your machine, web sites are involved in the overwhelming majority of these cases. Many people who operate web sites are too hospitable for their own good, and allow visitors to upload programs to the site and run them. As we've seen above, unpleasant things can happen if a bad guy's program can run on your machine.

If you run a web site, you need to limit what visitors can do. You should only allow a program on your site if you wrote it yourself, or if you trust the developer who wrote it. But that may not be enough. If your web site is one of several hosted on a shared server, you need to be extra careful. If a bad guy can compromise one of the other sites on the server, it's possible he could extend his

control to the server itself, in which he could control all of the sites on it – including yours. If you're on a shared server, it's important to find out what the server administrator's policies are.

Law #5: Weak passwords trump strong security.

The purpose of having a logon process is to establish who you are. Once the operating system knows who you are, it can grant or deny requests for system resources appropriately. If a bad guy learns your password, he can log on as you. In fact, as far as the operating system is concerned, he **is** you. Whatever you can do on the system, he can do as well, because he's you. Maybe he wants to read sensitive information you've stored on your computer, like your email. Maybe you have more privileges on the network than he does, and being you will let him do things he normally couldn't. Or maybe he just wants to do something malicious and blame it on you. In any case, it's worth protecting your credentials.

Always use a password – it's amazing how many accounts have blank passwords. And choose a complex one. Don't use your dog's name, your anniversary date, or the name of the local football team. And don't use the word "password"! Pick a password that has a mix of upper- and lower-case letters, number, punctuation marks, and so forth. Make it as long as possible. And change it often. Once you've picked a strong password, handle it appropriately. Don't write it down. If you absolutely must write it down, at the very least keep it in a safe or a locked drawer – the first thing a bad guy who's hunting for passwords will do is check for a yellow sticky note on the side of your screen, or in the top desk drawer. Don't tell anyone what your password is. Remember what Ben Franklin said: two people can keep a secret, but only if one of them is dead.

Finally, consider using something stronger than passwords to identify yourself to the system. Windows 2000, for instance, supports the use of smart cards, which significantly strengthens the identity checking the system can perform. You may also want to consider biometric products like fingerprint and retina scanners.

Law #6: A machine is only as secure as the administrator is trustworthy.

Every computer must have an administrator: someone who can install software, configure the operating system, add and manage user accounts, establish security policies, and handle all the other management tasks associated with keeping a computer up and running. By definition, these tasks require that he have control over the machine. This puts the administrator in a position of unequalled power. An untrustworthy administrator can negate every other security measure you've taken. He can change the permissions on the machine, modify the system security policies, install malicious software, add bogus users, or do any of a million other things. He can subvert virtually any protective measure in the operating system, because he controls it. Worst of all, he can cover his tracks. If you have an untrustworthy administrator, you have absolutely no security.

When hiring a system administrator, recognize the position of trust that administrators occupy, and only hire people who warrant that trust. Call his references, and ask them about his previous work record, especially with regard to any security incidents at previous employers. If appropriate for your organization, you may also consider taking a step that banks and other security-conscious companies do, and require that your administrators pass a complete background check at hiring time, and at periodic intervals afterward. Whatever criteria you select, apply them across the board. Don't give anyone administrative privileges on your network unless they've been vetted – and this includes temporary employees and contractors, too.

Next, take steps to help keep honest people honest. Use sign-in/sign-out sheets to track who's been in the server room. (You do have a server room with a locked door, right? If not, re-read Law #3). Implement a "two person" rule when installing or upgrading software. Diversify management tasks as much as possible, as a way of minimizing how much power any one administrator has. Also, don't use the Administrator account – instead, give each administrator a separate account with administrative privileges, so you can tell who's doing what. Finally, consider taking steps to make it more difficult for a rogue administrator to cover his tracks. For instance, store audit data on write-only media, or house System A's audit data on System B, and make sure that the two systems have different administrators. The more accountable your administrators are, the less likely you are to have problems.

Law #7: Encrypted data is only as secure as the decryption key.

Suppose you installed the biggest, strongest, most secure lock in the world on your front door, but you put the key under the front door mat. It wouldn't really matter how strong the lock is, would it? The critical factor would be the poor way the key was protected, because if a burglar could find it, he'd have everything he needed to open the lock. Encrypted data works the same way – no matter how strong the crypto algorithm is, the data is only as safe as the key that can decrypt it.

Many operating systems and cryptographic software products give you an option to store cryptographic keys on the computer. The advantage is convenience – you don't have to handle the key – but it comes at the cost of security. The keys are usually obfuscated (that is, hidden), and some of the obfuscation methods are quite good. But in the end, no matter how well hidden the key is, if it's on the machine it can be found. It has to be – after all, the software can find it, so a sufficiently motivated bad guy could find it, too. Whenever possible, use offline storage for keys. If the key is a word or phrase, memorize it. If not, export it to a floppy disk, make a backup copy, and store the copies in separate, secure locations.

Law #8: An out of date virus scanner is only marginally better than no virus scanner at all.

Virus scanners work by comparing the data on your computer against a collection of virus "signatures". Each signature is characteristic of a particular virus, and when the scanner finds data in a file, email, or elsewhere that matches the signature, it concludes that it's found a virus. However, a virus scanner can only scan for the viruses it knows about. It's vital that you keep your virus scanner's signature file up to date, as new viruses are created every day.

The problem actually goes a bit deeper than this, though. Typically, a new virus will do the greatest amount of damage during the early stages of its life, precisely because few people will be able to detect it. Once word gets around that a new virus is on the loose and people update their virus signatures, the spread of the virus falls off drastically. The key is to get ahead of the curve, and have updated signature files on your machine before the virus hits.

Virtually every maker of anti-virus software provides a way to get free updated signature files from their web site. In fact, many have "push" services, in which they'll send notification every time a new signature file is released. Use these services. Also, keep the virus scanner itself – that is, the scanning software – updated as well. Virus writers periodically develop new techniques that require that the scanners change how they do their work.

Law #9: Absolute anonymity isn't practical, in real life or on the web.

All human interaction involves exchanging data of some kind. If someone weaves enough of that data together, they can identify you. Think about all the information that a person can glean in just a short conversation with you. In one glance, they can gauge your height, weight, and approximate age. Your accent will probably tell them what country you're from, and may even tell them what region of the country. If you talk about anything other than the weather, you'll probably tell them something about your family, your interests, where you live, and what you do for a living. It doesn't take long for someone to collect enough information to figure out who you are. If you crave absolute anonymity, your best bet is to live in a cave and shun all human contact.

The same thing is true of the Internet. If you visit a web site, the owner can, if he's sufficiently motivated, find out who you are. After all, the ones and zeroes that make up the web session have to be able to find their way to the right place, and that place is your computer. There are a lot of measures you can take to disguise the bits, and the more of them you use, the more thoroughly the bits will be disguised. For instance, you could use network address translation to mask your actual IP address, subscribe to an anonymizing service that launders the bits by relaying them from one end of the ether to the other, use a different ISP account for different purposes, surf certain sites only from public kiosks, and so on. All of these make it more difficult to determine who you are, but none of them make it impossible. Do you know for certain who operates the anonymizing service? Maybe it's the same person who owns the web site you just visited! Or what about that innocuous web site you visited yesterday, that offered to mail you a free \$10 off coupon? Maybe the owner is willing to share information with other web site owners. If so, the second web site owner may be able to correlate the information from the two sites and determine who you are.

Does this mean that privacy on the web is a lost cause? Not at all. What it means is that the best way to protect your privacy on the Internet is the same as the way you protect your privacy in normal life - through your behavior. Read the privacy statements on the web sites you visit, and only do business with ones whose practices you agree with. If you're worried about cookies, disable them. Most importantly, avoid indiscriminate web surfing - recognize that just as most cities have a bad side of town that's best avoided, the Internet does too. But if it's complete and total anonymity you want, better start looking for that cave.

Law #10: Technology is not a panacea.

Technology can do some amazing things. Recent years have seen the development of ever-cheaper and more powerful hardware, software that harnesses the hardware to open new vistas for computer users, as well as advancements in cryptography and other sciences. It's tempting to believe that technology can deliver a risk-free world, if we just work hard enough. However, this is simply not realistic.

Perfect security requires a level of perfection that simply doesn't exist, and in fact isn't likely to ever exist. This is true for software as well as virtually all fields of human interest. Software development is an imperfect science, and all software has bugs. Some of them can be exploited to cause security breaches. That's just a fact of life. But even if software could be made perfect, it wouldn't solve the problem entirely. Most attacks involve, to one degree or another, some manipulation of human nature - this is usually referred to as social engineering. Raise the cost and difficulty of attacking security technology, and bad guys will respond by shifting their focus

away from the technology and toward the human being at the console. It's vital that you understand your role in maintaining solid security, or you could become the chink in your own systems' armor.

The solution is to recognize two essential points. First, security consists of both technology and policy – that is, it's the combination of the technology and how it's used that ultimately determines how secure your systems are. Second, security is journey, not a destination – it isn't a problem that can be "solved" once and for all; it's a constant series of moves and countermoves between the good guys and the bad guys. The key is to ensure that you have good security awareness and exercise sound judgment. There are resources available to help you do this. The Microsoft Security web site, for instance, has hundreds of white papers, best practices guides, checklists and tools, and we're developing more all the time. Combine great technology with sound judgment, and you'll have rock-solid security.

Acknowledgements

This paper is based in part on resources including, but not limited to, those listed here.

The Computer Emergency Response Team/Coordinator Center (CERT/CC) www.cert.org

Internet Engineering Task Force (IETF) recommendations, particularly RFC 2196, but including several other RFCs. www.ietf.org

Federal Bureau of Investigation Security Recommendations www.fbi.gov

The Federal Computer Incident Response Center (FedCIRC) www.fedcirc.gov

The National Infrastructure Protection Center (NIPC) www.nipc.gov

The National Institute of Standards and Technology (NIST) www.nist.gov

The National Security Agency (NSA) www.nsa.gov

Papers readily available from Microsoft www.microsoft.com

The SANS Institute for System Administration, Networking and Security www.sans.org

The Vermont InfraGard (in conjunction with the NIPC) www.vtinfragard.org