55934

Building A Successful Security Infrastructure
What You Want vs. What You Need vs. What You Can Afford

Michele D. Crabb (crabb@nas.nasa.gov)
- Sterling Software/NASA Ames Research Center

ABSTRACT

With the fast growing popularity of the Internet, many organizations are racing to get onto the on-ramp to the Information Superhighway. However, with frequent headlines such as "Hackers' break in at General Electric raises questions about the Net's Security", "Internet Security Imperiled - Hackers steal data that could threaten computers world-wide" and "Stanford Computer system infiltrated; Security fears grow", organizations find themselves rethinking their approach to the on-ramp. Is the Internet safe? What do I need to do to protect my organization? Will hackers try to break into my systems? These are questions many organizations are asking themselves today.

In order to safely travel along the Information Superhighway, organizations need a strong security framework. Developing such a framework for a computer site, whether it be just a few dozen hosts or several thousand hosts is not an easy task. The security infrastructure for a site is often developed piece-by-piece in response to security incidents which have affected that site over time. Or worse yet, no coordinated effort has been dedicated toward security. The end result is that many sites are still poorly prepared to handle the security dangers of the Internet.

This paper presents guidelines for building a successful security infrastructure. The problem is addressed in a cookbook style method. First is a discussion on how to identify your assets and evaluate the threats to those assets; next are suggestions and tips for identifying the weak areas in your security armor. Armed with this information we can begin to think about what you really need for your site and what you can afford. In this stage of the process we examine the different categories of security tools and products that are available and then present some tips for deciding what is best for your site.

Where Do You Start?

Building a successful security infrastructure requires having the right tools and the right knowledge. Whether you are starting from scratch, or are re-evaluating the security needs of your site, creating the successful security framework in today's complex UNIX environments can be technically and politically challenging. The people tasked with this chore are faced with a multitude of decisions. Much of the time they are required to walk a fine line between what the organization needs versus what the organization wants, or even what they can afford. Often what a organization wants may be based on recent events on the Internet or good publicity about popular security products, and may not meet the actual needs of the company. At other times, what the company can afford may be the driving factor.

Before we begin discussing how to build the right security framework, we need to discuss what is meant by a "security infrastructure". Obviously, this is defined on a site-by-site basis. For my analysis, I consider the "security infrastructure" to include all aspects of security, from the site's philosophy on how computer and physical security should be approached to the tools and procedures used to make it all work. Basically, everything that is used and done to maintain security at a site. However, this paper will focus only on issues concerning how to implement computer security, also referred to as Automated Information Security (AIS).

Although there is no one-complete source for providing all the answers on how to build a successful security infrastructure, there is an abundant source of helpful information and tools freely available on the Internet (See Appendix A). This paper provides a cookbook method for developing your own successful security framework. First I will begin with a discussion on computer security philosophy, which provides the foundation for a security framework. Next, I present an overview on conducting a risk analysis. Although a risk analysis can be tedious and time-consuming, it is an effective tool for helping an organization identify their

assets, discover the weak areas in their current security framework and decide which solutions will best suit their needs. As part of the risk analysis, you create a list of weakness and possible solutions to correct those weaknesses. Armed with this information, you can begin to examine the organization's needs versus its wants.

The security "needs" of a organization are often driven by the threats that exist, while the "wants" of the organization are driven by customer perception, publicity on recent security events on the Internet and, all too often, "knee-jerk" reactions of management. The resulting security infrastructure is often a gentle balance between the needs and wants of a company, and what they can actually afford. In the final section of the paper I present some tips on how to best determine what an organization needs.

What is the Current Philosophy?

The best place to start in building a security infrastructure or re-evaluating the current one, is to examine the organization's current philosophy on security. A site's philosophy on security will most likely include a statement of the importance of security and the goals for security. For example, computer security at NASA Ames is taken seriously. The NAS security philosophy is "to provide an adequate level of computer security support such that NAS computing resources are protected from disruption; information stored on NAS computers is protected from modification and disclosure; the NAS Facility can quickly recover from disruptions, and NAS clients are not adversely affected by computer security measures." Our approach has been to be re-active rather than pro-active. By taking the re-active stance, we do not overly burden our clients by any security measures.

If your sites does not have an official AIS philosophy or you are unsure of what that philosophy might be, there are some questions you can ask:

- How much intellectual property is available on-line?
- How sensitive is this information?
- Is there "non-disclosure" information on-line?
- Are our internal users to be trusted?
- Are most of our users local or remote?
- Are we required to meet any quidelines from higher organizations?
- What do our clients and users expect in the way of system security?
- Is there a possibility that we will lose users and clients if we take security to seriously or not serious enough?
- How much down-time or monetary loss has occurred due to security incidents in the past?
- How much negative publicity has our organization suffered due to a poor security framework?
- Are we concerned about such negative publicity?

It might be the case that at this stage of the process you cannot answer all of those questions. Building the right security infrastructure can often be a "chicken and an egg" process in that you are unsure where to start.

What Are the Assets and Threats?

Before you can begin to think about what you need in the way of a security infrastructure, you must first be knowledgeable of what it is you have you need to protect and what are the threats associated with those assets. One tool for accomplishing this task is called a risk analysis. The purpose of a risk analysis is to identify all of the assets (e.g., hardware, software, intellectual property), the possible threats to these assets, identify areas of weakness and then make recommendations for improving or resolving the weaknesses. The main goals of a risk analysis are to balance the risks with the cost to protect your assets and to outline which risks you should prevent, limit or accept. Obviously, an organization would not gain anything from spending more on security protection than their total assets are worth. Also, it is not cost effective to spend $10K/year trying to prevent a threat with an estimated damage of $5K/year.

In a risk analysis, a dollar value is assigned to all assets. After identifying the possible threats and the frequency of their occurrence, a dollar loss amount is given to each threat and cost amount is given to each "improvement". Given these three cost factors, an organization can determine which controls are appropriate for the level of security they desire. The level of the risk analysis an organization chooses to perform will largely depend on the size of the organization and the types of products or services they provide. The larger the company, the greater the assets; hence, a more complex risk analysis is needed.

The first stage of a risk analysis involves identifying all of your assets. This can be done in a very formal fashion where each and every item of the organization is listed with its associated cost. The dollar associated with each asset should be either the cost to replace the asset, or in the case of software or data, the cost to re-install the software or re-create the information. The

replacement cost for any asset should take into account inflation and not just be the original cost of the item. For a government agency such as NASA Ames, the risk analysis process is very formal and includes every possible AIS asset. At the last risk analysis for the NAS facility, the assets were broken down into hardware, software, contract personnel, storage media assets, and facility building costs. However, for smaller organizations, it might be more appropriate to just include major assets (computer hardware, software, data/information, etc). Regardless of the level of formality you choose for your risk analysis, you should consider all intellectual property in your asset inventory. Intellectual property includes items such as program codes, input data, system and program documentation, databases. Probably one of the fastest growing areas of intellectual property today are World Wide Web servers and home pages. How much time and effort would be required to recreate your Web server or home page if it were destroyed?

The threat and vulnerability identification is the second stage of the risk analysis. Threat assessment involves the identification of all possible threats, the frequency of their occurrence and the estimated dollar loss if the threat were to occur. For the purpose of a risk analysis, a threat is defined as any force or phenomenon that could degrade the availability, integrity or confidentiality of an Automated Data Processing (ADP) resource, system or network. Threats against ADP resources are either human or environmental threats. These two categories are further broken down into intentional and accidental for human threats, and natural or fabricated for environmental threats. Some examples of each of these threats are listed below in Table 1.

Human Threats - Intentional
- Password sniffing
- I/P Spoofing
- Bomb threat
- Arson

Human Threats -Accidental
- Operational Errors
- System programming errors
- System configuration errors
- Data destruction/discloser

Environmental Threats - Natural
- Flood
- Earthquake

Environment Threats - Fabricated
- Accidental fire
- Water leakage
- System hardware failure

Table 1: Example Threats

A formal risk analysis will include an extensive list of all possible threats; however, those choosing to do a more casual risk analysis may choose to include only threats with a high annual frequency rate.

Once the threats have been identified, each threat will be given an Annual Frequency Estimate (AFE) which is the probability of the event happening in a one year time frame. The AFE is derived by analyzing national, local or site-specific data. For example, if the analysis of a site log showed that on average there were ten power failures a year, the AFE for a power failure would be calculated at 10/1 or 10. Since the AFEs may vary in numbers from .0001 to > 100, it is best to use a Calculated AFE Index so all numbers are in wholes.[1]

Along with establishing an AFE for each threat, you need to establish an estimated annual loss, sometimes referred to as the Annual Loss Expectancy (ALE), due to the threat. Again, let us use the threat of a power failure as an example. Let us say that on average, a power failure will cause two hours of down time for all computer systems. For this example, let us assume that one hour of computer time for all computers at a site is worth $10K. Then each power failure results in a loss of $20K. If there are ten power failures a year, the ALE for power failure is a whopping $200K.

For those organizations who are just approaching the on-ramp to the Information Superhighway, you may not be aware of what threats exists or how often they are expected to occur. Unfortunately there is no one-official source for threat analysis. However, there is some information on the Internet and in the bookstores. Some good resources for helping you identify threats are:

- "Coping with the Threat of Computer Security Incidents, A Primer from Prevention through Recovery", By Russell Brand
- The UNIX Security FAQ
- "Security in Computing", by C. Pfleeger
- "Control and Security of Computer Information Systems", by M. Fites and P. Kratz

The next stage of the risk analysis is to identify all of the current security controls (or safeguards), hence the current state of security of your site. If you are just starting out, there will not be much to analyze; however, if you are re-evaluating the security of your site, this stage may be quite involved. The analysis of the current state of security of an organization should include: the philosophy of computer security, all current security related policies and procedures, all security tools and methods in use, and any security awareness training that is provided. These security safeguards are then divided into three categories: physical, administrative and technical.

Physical safeguards include such items as building entrance access controls, locked equipment rooms, motion detectors and cameras. Administrative safeguards include policies, procedures and site philosophy related to computer security. Technical safeguards include any security tools or methods used to protect or audit the systems. Table 2 list some example safeguards from each group.

Physical Security Safeguards
- External and internal building access controls
- Fire warning and protection equipment
- CCT monitoring exit doors
- Computer equipment is locked down
- Computer equipment is tagged and documented

Technical Security Safeguards
- System activity is logged and archived
- File system auditing is performed regularly
- User account installation is done automatically and consistently
- Root and other special access passwords are changed frequently
- Network connections are monitored and logged

Administrative Access Safeguards
- An account usage/request policy exists
- Security awareness training is provided annually
- An off-site storage and backup procedure exists
- Login warning banners are on all systems
- All new major software packages and programs are audited for security problems before installation on the open network.
- Employees are required to wear badges at all times

Table 2: Example Safeguards

What are the Areas of Concern?

Armed with the knowledge of the current threats and security safeguards, you should be able to create a list of known weaknesses. If your site is just beginning to develop its security framework and does not have any, or has very few security safeguards in place, you may find the list of weakness to be alarmingly long. However, if you already have a security framework in place and you are re-evaluating your security needs, the list should be much shorter.

During this phase of the risk analysis, you should to focus on all aspects of site security. There are a number of simple exercises that can be performed to test your security armor. If you are not really sure where or what weak links there are in your security armor, the best way to learn is to try to "break the rules". That is, you want to think like an intruder or cyberpunk and try to break into your own systems, or try to violate known physical security controls.

A good way to test physical security is to have an outside friend come to your site and wander around. This person should try to get into various rooms where the doors are closed and locked. If the person can wander around for any length of time without people asking questions, then you have a possible physical security problem. If this person is able to get into a locked room with restricted access (e.g., they pose as a vendor field engineer), then you have a problem. People at a computing site need to be security conscious. They should stop strangers and ask to see their employee badge. They should always question strangers walking around, especially when they are walking out the door with computer equipment. Some questions that should be answered during this exercise are:

- Are building access controls sufficient to keep a stranger from entering a restricted access?
- Do employees stop strangers and ask to see verification of their employment?
- Are restricted areas and rooms locked at all times?
- Do employee where their badges in plain sight?

A method to discover weak areas in your procedural safeguards is to stage a mock security incident. You could report that you received a phone call from Site X saying that a copy of a password file from one of your local hosts was found on their system, along with a file containing cracked passwords. Staging a mock incident is an excellent way to determine how well prepared a site is for handling a security incident. Some questions that should be answered during this exercise are:

- Does our site have an incident response team?
- Does our site have a incident response procedure?
- Does the incident response team understand their roles?
- Do the front line support people know who to call when a problem occurs?
- Does the incident response team know where to search for the clues?
- Does the incident response team know what information should be logged/traced?
- Does the incident response team understand what information can be released to outside groups (e.g., the press, law enforcement)?

Another method to test procedural safeguards is to have someone try various social-engineering schemes. You can have a friend call the support center and pose as an employee who forgot their password. Or have someone call and pose as a manager and then give approval for an account on a local system. In both cases, proper procedures and training should prevent a social engineer

from being successful.

One of the best methods to test your technical security controls is to try to break into your own site or to have someone else do it for you. The paper "Improving The Security of Your Site by Breaking Into It", by Dan Farmer and Wietse Venema provides a nice tutorial on how to use the most common methods to break into systems. You can also use any reputable list of common security problems (e.g., the UNIX Security FAQ) and try each item.

It has been my experience that many sites suffer from the same security flaws and weaknesses. Even with the frequent publicity regarding Internet security, many sites still do not devote enough resources to maintaining a secure site. Below is a list of what I consider to be the ten most frequent weak links for a given site.

1) Inadequate resources dedicated to improving and maintaining security.

   Even with the growing concern about security on the Internet, I still hear stories about how site security is still a "side" responsibility of other people.

2) Reusable passwords and passwords transmitted in clear-text over the network.

   Password sniffing has become a popular method of intrusion. Many passwords can be captured in a single sweep operation.

3) Unrestricted and unmonitored network access. This includes lack of network filtering and monitoring, as well as running unnecessary services from *inetd*.

   Not having the knowledge of where packets are coming from or what services are being targeted, is like facing an invisible death squad. You know they are out there, but you have no idea where they are or what weapons they are going to use to carry out the deed.

4) Systems left in a default and insecure configuration.

   Many vendors still ship systems with a default configuration which is very open. This includes: */etc/hosts.equiv* files with "+" entries, extra services turned on in *inetd*, no logging set up via *syslog.conf*, default accounts with no passwords, the decode alias defined in */usr/lib/aliases* and world exportable file systems. Every system installed on your local network should undergo a minimum security configuration audit.

5) Vendor patches are not installed for known security problems.

   There are many widely publicized security holes in vendor software, along with vendor or O/S independent patches. There is no excuse not to install a readily available security patch, yet many sites choose not to.

6) User accounts are not installed in a consistent and secure manner.

   Inconsistent installation of new accounts can lead to exploitation of new user accounts via world-writable files and directories or poor initial passwords.

7) System logging is not configured consistently across all platforms. Log files are not archived at all, or are not kept for a sufficient period of time.

   System log files are a crucial element in tracing a security indent. In a large environment, you need to have logging configured in a similar manner and archived consistently on each host, to reduce the complexity of tracing events. Log files should be archived and kept for a minimum of eight weeks.

8) No procedures or controls for installing new hosts on the network.

   Without policies or procedures governing how new hosts are added to the network, any user at a site can bring in their own machine and attach it to the network. This can lead to hidden or unknown vulnerabilities.

9) Account activity is not monitored. Accounts are not disabled and removed after a user leaves. Accounts may be dormant for months at a time.

   Dormant accounts are one popular method of system abuse and are an easy target for intruders.

10) Weak controls on root and other special privileges.

    Root passwords should be changed on a frequent basis and within a few days of when ever someone with root access leaves the company. All to often I have heard someone say, "Yeah, I left the company three months ago and they still have not changed the root password."

What Corrective Measures are Needed?

Once you have identified all of the areas of concern

regarding your security safeguards, you need to begin thinking about what corrective measures are needed to reduce the current level threat to your assets. This can be the most difficult part of the entire process. Often, there are multiple ways to correct an insufficient safeguard or lack of a safeguard. The people responsible for deciding what is best for the organization may not agree on the best solution. For example, one very heavily debated topic at our facility is how to move away from reusable passwords or sending passwords clear-text over the network. Some people argue that Kerberos is the best solution, while others argue that one-time password mechanisms such a S/Key or Smartcards are the best way go. For the purpose of the risk analysis, it would be best to list several solutions to a problem, if they exist. In the end, the cost analysis may determine the final outcome.

Some of the corrective solutions will be quite obvious, while others may require additional investigation and research. For example, if during the safeguard appraisal, you discovered a breakdown on proper procedures for adding an account, this problem could most likely be resolved by additional training or by implementing a procedure where all accounts are audited for proper installation. This solution was very obvious. On the other hand, if during the safeguard appraisal you determined that lack of network filtering to be a large concern, the solution is not so obvious. You might choose to implement host-base filtering, router-based filtering or implement a firewall. The people involved in the decision making process may have differing opinions on what is the best solution.

Ideally corrective measures should reduce or alleviate the risk all together. Solutions which only minimally reduce a risk should not be considered. You should attempt make recommendations for improvements for all documented weakness. In some cases, the recommended corrective measure will help alleviate or reduce more than one threat. Most of the common security problems plaguing the Internet today can be resolved using freely available software and a little extra people power. Below are some suggested solutions for the top ten security problems discussed in the previous section.

Problem: Inadequate resources dedicated to improving and maintaining security

Solution: Allocate additional resources. Hire a person to perform full-time security, or distribute site security tasks among a group of people.

Problem: Reusable passwords and transmission of clear-text passwords across the net.

Solution: There are several ways to approach this problem. What might work best for one site would fail miserably for another. Many sites are using S/Key as a one-time password mechanism while other sites are using Kerberos to encrypt passwords. The tools are available to solve this problem.

Problem: Unrestricted and unmonitored network access.

Solution: Install the tcp_wrapper package. The software is free and very simple to install and configure. System overhead is minimal and logging capabilities are excellent. The package is very widely used throughout the Internet today.

Problem: Systems left in a default and insecure configuration.

Solution: All systems installed on the local network should under-go a site configuration audit. A standardize configuration should be developed for each vendor platform. All newly installed systems are audited against a system security checklist and the standard configuration.

Problem: Vendor patches are not installed for known security problems.

Solution: A local mail alias should be created for all people responsible for site security. This alias should be added to all vendor security alert mailing lists. Establish a site policy which states vendor security patches must be installed within a certain number of days from when they are published.

Problem: User accounts are not installed in a consistent and secure manner.

Solution: Develop or procure an automatic account installation package, or use the routines provided by the vendor. Sanitized and secure default environment files should be established for each hardware platform. All new accounts should be populated with the same default environment files. Default permissions on new home directories and environment files should allow access only by the owner of the account. Unique and strong passwords should be given to each new account.

Problem: System logging is not consistent and log files are not archived for a sufficient time.

Solution: Establish a default configuration for all syslog.conf files. Ideally, different facilities should be logged to different files. If possible, log all important messages (e.g., authentication messages) to a second host as well as the local host. If possible, use consistent log file names for the same facilities on the different platforms. Establish a log file archiving procedure (one is available from NAS). Archive log files either on a daily or weekly basis, depending on how fast the file grows. Keep archived log files for a minimum of eight

weeks or longer, if file space is available.

Problem: No procedures or controls for installing new hosts on the network.

Solution: Establish a site policy for adding new hosts to the local network. Funnel all IP address request to a hostmaster alias and have a central approving authority. Establish minimum security configurations for all platforms at a given site. Require system owners to act as the security point of contact (POC) for their system, if applicable, and have them be responsible for maintaining security on that system. For strange or new architectures, you can use host-based filtering to deny any network connections from a specific host.

Problem: Account activity is not monitored. Accounts are not disabled and removed after a user leaves.

Solution: Establish a maximum amount of time an account can be dormant (30, 60 or 90 days are good choices). Develop a program which runs out of cron and reports on dormant accounts. Dormant accounts should be disabled and archived. Establish an employee check-out procedure which requires notification of the accounts staff when person leaves. Have the account staff disable the account within X number of days after employee leaves.

Problem: Weak controls on root and other special accounts.

Solution: Establish a policy governing root (special) access which provides ground rules for who can have special access. Root and other special access passwords should be changed on a frequent basis (e.g., monthly) or whenever someone with special access leaves. Distribution of special access passwords should be from a single source and require an audit trail (e.g., signature upon pick-up of passwords).

What are the Essentials?

Before we discuss the security needs, lets take a look at the essential elements of a security infrastructure. In the beginning we discussed the importance of the security philosophy of a site. This is the first essential element. The next essential element is a collection of security tools, or "magic bag of tricks" as I like to call it. Every organization which is currently connected, or intends to connect to the Internet needs a magic bag of tricks for security. But what should be in this bag? My experience has always been the more tools and the greater the variety, the easier it is to do the job.

Security tools can be classified into four different categories, each of equal importance. The Internet is loaded with freely available security tools and programs, many of which have special mail or news groups which discuss problems and configuration issues. For brevity sake, only the names of the packages will be listed here. See Appendix B for a short description and location of some common security packages.

The first group of tools are those which scan or test a system for vulnerabilities which are exploitable from the Internet. For example, various problems in early versions of *sendmail* would allow anyone on the Internet to gain unauthorized access to a system running a buggy version of *sendmail*. A tool in this category would be able to scan all systems on the network for the existence of a particular *sendmail* bug. Tools in this category are used by Intruders as frequently or more so than system administrators or security analyst. Some examples of tools which fall into this category are: Internet Security Scanner (ISS), Securscan and SATAN (Security Analysis Tools for Auditing Networks). These tools allow you to check all hosts on your local network from a single host. Many sites run these types of tools on a regular basis. At any one given time, a large number of Internet intruders or "wanna-be's" are also running the same tools trying to discover a weak link they can exploit.

The second class of tools are those programs which scan the local host for configuration errors and other problems which lead to security vulnerabilities. Examples of configuration errors include world-writable files and directories, poor passwords, unnecessary entries in the */etc/inetd.conf* file, world exportable files, etc. One of the most popular tools in this category is COPS (Computer Oracle and Password Scanner). Some other tools which have become quite popular in recent years are: Tripwire, Tiger, Crack and TAMU.

The third class of tools are those programs which help you or your users perform functions in a more secure manner. An example would be a */bin/passwd* replacement program which enforces stricter password construction or an encryption package which allows you to encrypt email. Some examples of popular packages in this category are: npasswd, S/Key, Kerberos, tcp_wrapper, log_daemon, sudo, and firewall toolkits.

The fourth and final class of tools are those programs which help you during and after a security incident. If you discovered someone broke into your system, you would need tools to help you analyze what the intruder did (e.g., install a sniffer, replace binaries, etc). Tools in this category would help you determine all open files, scan log files for inconsistencies or determine if your network interface is running in promiscuous mode. More and more tools of this nature are being written today to help combat the growing number of security incidents. Some examples of tools in this category are: LSoF, naiad, SLIC, CMP, and prob_ports.

Aside from having the right security tools and a well stated security philosophy, a successful security framework must also contain a set of well-defined security policies and procedures. Policies should define the "rules" as well as the penalties for breaking the rules. Sometimes the policies will set the stage in terms of what security tools are needed to enforce the selected policies. There are several key policies and procedures every organization should have. These polices are briefly discussed below along with some questions which will help guide you in writing policies for your own site. For a more detailed discussion of the vital security policies needed for a site, refer to RFC 1244 (The Site Security Handbook).

The first vital policy is the Acceptable Use Policy. This policy discusses and defines the proper use of the computing resources. Some questions that you need to answer when writing an Acceptable Use Policy are:

- Are users allow to use password cracking programs?
- Are users allowed to access files/programs that are not owned by them, but open to them?
- Are users allowed to make copies of system configuration files for personal use?
- Are users allowed to download and run security tools which report on weaknesses in the system?
- Are users allowed to share accounts?
- Are users allowed to play games installed on the systems or read any of the available newsgroups?
- Are users allowed to use .rhosts files or .netrc files?

Fortunately, you don't have to write an Acceptable Use Policy from scratch. There are numerous examples floating around the Internet. One of the best places I have found for examples is the Computers and Academic Freedom Archive at Electronic Frontier Foundation (EFF). The Uniform Resource Locator (URL) is *http://www.eff.org:80/CAF*. This archive contains dozens of sample computer policies along with critiques from the EFF staff. There is also a FAQ on which policies are the best. Although most these policies are geared toward the academic industry, they are very helpful. Before I wrote the NAS Acceptable Use Policy, I read through most of the sample policies available on CAF at the time, and used what I thought were the best elements of the policies. The NAS Acceptable Use Policy (and other information relative to this paper) is available at:

http://www.nas.nasa.gov/NAS/RelatedPapers/SANS95.

Another vital policy is the User Account Policy. This policy outlines the requirements for requesting and maintaining an account on the site's resources. For some organizations this may seem unnecessary. However, for large sites such as universities or Internet providers, this policy should be a critical element of the security infrastructure. Some questions that you need to answer when writing the User Account Policy are:

- Who has the authority to approve or grant accounts?
- Who is allowed to use the resources?
- Must users reside in the United States (or local country)?
- How long are accounts allowed to be inactive before they are disabled or archived?
- Are users allowed to share accounts?
- What are requirements for password construction and aging?
- Is there a limit to the number of accounts a user may have at a site?
- What are the user's rights and responsibilities?

Some examples of User Account Policies are also available in the CAF archive.

Two essential procedures every site should have are a Security Incident Handling Procedure and a Backup and Off-site Storage Procedure. The Incident Handling Procedure should outline the steps to follow in response to the different type of security incidents. The procedure should also outline the areas of responsibilities for the support staff, list general procedures to follow and provide detailed instructions on how to respond to different type of incidents (e.g., where to look for clues, the type of information which should be logged, the appropriate people to contact).

The Backup and Off-site Storage Procedure should define what information is to be archived, how frequent the backups are performed, how long the data is stored and, if appropriate, how data is stored off-site.

The final element of a successful security infrastructure is a security awareness training program. The purpose of a security awareness training program is to make the users, support staff and management more aware of the roles they play in the success of security at their site. The security awareness program should also inform people of any policies that effect them and any monitoring activity that is performed at the site. Security awareness can be conveyed through a variety of methods (live training classes, videotaped class and on-line reading materials such as a Web page).

What Do You Need?

By this stage of the framework building process, you should have a good idea of what your assets are, their total value, what threats exist and how often they might occur, what weakness exist in your current security armor and some possible solutions to correct those weaknesses. You should also have a good understanding of what are the key elements of a successful security framework. The difficult part now is to determine how to put it all together. Hence, what do you really need.

What a site really needs in terms of a security infrastructure is largely dependant on the importance of security at that site. Even if during the risk analysis process you isolated a multitude of weaknesses, if your management does not believe in security, you don't really need to do anything. On the other hand, if security is an important topic for your organization, your needs will vary depending on what you discovered during the risk analysis. My philosophy is that all sites connected to the Internet should maintain some minimal security configuration, even if only for the "good neighbor" factor.

In the previous section I discussed that various elements of a security framework. A successful security infrastructure should contain healthy portions of each element. Just like an unbalanced diet can lead to health problems, not having a balanced security framework can lead to unwanted or hidden security problems. There is no golden rule that states you must have "this" and you must have "that." However, there is a general guideline that states the cost and effort put into building and maintaining a security framework should be some fraction of the total cost of the assets.

If you have some elements of a security framework already in place, then perhaps all you need to do to complete your framework is to implement the recommended solutions to reduce your risk of threat. Deciding on which solutions to implement can be accomplished by looking at the estimated cost to implement the solution, versus the cost benefit of the solution. For example, say one of your suggested solutions was to implement system accounting on all hardware platforms as a method to trace intruder incidents and prevent account misuse. Since the software is readily available on most UNIX systems, there is really no one-time cost for this solution, only the time required to implement it. Lets say the ALE for account misuse and intruder incidents is estimated at $3.5K and if we implement system accounting this loss will be reduced to $1.5K, which is a 42 percent reduction in estimated annual loss due to the threat. Since there is really no one-time cost for implementing system accounting and no real recurring cost, this is a cost effective solution.

If you are just starting to build your security framework, then you have lots of decisions to make. If at this stage of the process, you don't have a security philosophy, then that is were you need to start. Your philosophy on "how much" and "what type" of security will drive your decisions on what policies to implement and what tools to stock in your magic bag of tricks. Your security philosophy and policies form the foundation of your security infrastructure, while the security tools form the walls and the roof.

With a philosophy firmly developed, you can begin to develop the policies. Four vital policies were discussed in the previous section. At a minimum, every sites should have these four policies. Unfortunately, policy implementation can be the most difficult part of the security framework building task due to their controversial nature. From my experience at NAS, it would not be unusual for a large organization to spend 3-6 months finalizing policies and getting everyone to agree. A sad, but true fact.

Deciding on what tools to stock in your magic bag will depend to some extent on your policies, as some tools may be needed specifically to help enforce the policies. Ideally, you want to have tools from all four categories. It might be best to start with tools that will help further identify your areas of weaknesses (e.g., system configuration errors). These tools can also be used on a daily or weekly basis to audit systems for any new problems. The next group of tools to acquire would be those that help you perform functions in a more secure manner. For example, if your biggest security concern is re-usable passwords, then you might want to look at alternate password and authentication systems such as S/Key or SmartCards. Finally, you will need to acquire tools which will help you in the event of a security incident. It is best to have these tools in place and be familiar with their function before an incident occurs. Otherwise, you'll be fumbling around in a high state of stress trying to find these tools and implement them in the midst of a security incident investigation.

With the wide variety of tools and information available on the Internet, you can put together a reasonably good security infrastructure for just the price of human resources and the time to implement the changes. This is not to say that building a successful security infrastructure is a quick and easy task. Depending on the size and complexity of your organization, it could take one or more years to complete the work. As an example, let us look at the NAS facility at NASA Ames Research Center. Security at NAS is considered to be fairly good. Last year, two other people and I re-evaluated the security

needs of the NAS facility against the current security problems plaguing the Internet. Based on our findings and our recommendations, it was estimated to take three full-time employees seven months to implement all of the changes.

Conclusion

Building a successful security infrastructure is not a simple task. The "builders" are frequently faced with a multitude of decisions which require them to walk a fine line between what the organization needs versus what it wants. This paper has attempted to present a cookbook style method for analyzing the security needs of a site and determining the best materials from which to build the framework. The risk analysis was presented as a tool to accomplish most of the work during the decision phase. By performing a risk analysis an organization can identify their assets, the risks to those assets, areas of weakness and then possible remedies for those weaknesses. This knowledge then provides a basis from which an organization can begin to implement their security infrastructure.

Author Information

Michele Crabb has been the primary computer security analyst for the NAS Facility at NASA Ames Research Center for over five years. During her ten years at Ames, Michele has worked in several divisions, in a variety of positions ranging from applications programming to UNIX system support. Prior to becoming the NAS security analyst, she was actively involved in providing system administration support for the large number of workstations at the NAS facility. Michele can be reached via electronic mail at *crabb@nas.nasa.gov*, or via US Mail at NASA Ames Research Center, Mail Stop 258-6, Moffett Field, CA 94035-1000.

References

1. U.S. Department of Commerce, "Federal Information Processing Standard Guideline for Automatic Data Processing Risk Analysis"

APPENDIX A - Useful Security References

WEB Sites

Most of these WEB sites have a variety of security tools and information. Some sites also have pointers to other security WEB sites.

http://first.org/first/

http://nasirc.nasa.gov/NASIRC_home.html

http://www.cs.purdue.edu/coast/coast.html

http://www.alw.nih.gov/Security/security.html

http://www.alw.nih.gov/Security/first-papers.html

http://www.tansu.com.au/Info/security.html

http://www.cs.cmu.edu:8001/afs/cs.cmu.edu/user/bsy/www/sec.html

http://www.ibd.nrc.ca/~roberson/elaw/security_acts.html

Mailing Lists

bugtraq - a full disclosure list for the discussion of security bugs, how to exploit them and how to resolve them. To subscribe send a message containing the words subscribe bugtraq in the message body (not the subject header) to *bugtraq-request@fc.net.*

8lgm Security Team - Security advisories, discussion of vulnerabilities. To be added, send any message to *8lgm-request@bagpuss.demon.co.uk* and the address you mail from will automatically be added to the list.

Security Related Books

"Firewalls and Internet Security", by William R. Cheswick and Steven M. Bellovin

"Practical Unix Security", By Simson Garfinkle and Gene Spafford

Papers and Articles

Most of these papers are available at the WEB site: http://www.alw.nih.gov/Security/first-papers.html

The UNIX Security FAQ - Maintained Christopher Klaus. Is distributed via many security news groups. Email address is *iss@iss.net.*

RFC 1244 - Site Security Handbook

"Thinking About Firewalls", by Marcus Ranum

"Network (In)Security Through IP Packet Filtering", by Brent Chapman

"Compromise: What if Your Machines are Compromised by an Intruder", A FAQ maintained by Christopher Klaus

"Coping with the Threat of Computer Security Incidents: A Primer from Prevention through Recovery", by Russell Brand.

"Security Problems in the TCP/IP Protocol Suite", by Steven M. Bellovin

APPENDIX C - Vendors with Security Products

Below are a list of some of the vendors who supply security products. This list is by no means complete, and I am in no way endorsing any of these products, nor do I claim to know a lot about them. This information is merely provided as a helpful resource for those readers who are interested in pursuing vendor solutions. The information below was taken from personal experience and the UniForum 95 Preview magazine.

Axent Technologies, Raxco Inc. (301) 258-2620
Client/server security products and services that secure and protect information assets.

Barranca, Inc. (505) 662-3744
Provides consulting, training and software for asset protection, vulnerability assessment, and risk management.

Baseline Software, Inc. (800) 289-9555
A information security software, publishing and consulting firm. They offer a policy construction kit which contains over 600 already written information security policies.

C.A.S. Solutions (415) 346-4131
Network security products.

CheckPoint Software Technologies, Ltd.(800) 429-4391
Internet security products, including a firewall kit.

Cheyenne Software (516) 484-5110
UNIX and OS/2 security products.

CyberSAFE Corp. (206) 883-8721
Kerberos Security solutions, security consulting services, and security tutorials.

Digitial Equipment Corporation (800) DIGITAL
Various hardware/software security products for OpenVMS, DEC OSF/1, Ultrix, NetWare, SunOS, Solaris, AIX and HP-UX.

Freedman Sharp and Associates, Inc. (403) 264-4822 or *info@fsa.ca*
A comprehensive software package called PowerBroker which partitions root functionality and creates an audit trail of all actions. PowerBroker can also be used in conjunction with a firewall machine to control Internet access.

Hughes Aircraft Co. (714) 732-5352
A software-only security product called "NetLock" which provides network level security protection.

Los Altos Technologies, Inc. (415) 988-4848
Various UNIX security tools for auditing, authentication, user identification and other functions.

Memco Software, Inc. (800) 862-2602
Open Systems security products which help meet the security needs of heterogenous sites (UNIX and non-UNIX systems).

RSA Data Security, Inc. (415) 595-8782
Various security products which include a cyptographers toolkit and a privacy enhanced mail system toolkit.

Secure Computing Corp. (800) 692-LOCK
High-level computer security solutions including dial-in and Internet access protection at critical server connections.

Security Dynamics (617) 547-7820
SecurID Cards.

Symark Software (818) 865-6121
A software package which allows system administrators to delegate activities requiring root access without giving away the root password.

Trusted Information Systems (301) 854-6889
Firewall toolkits, Trusted Mail (TM), and other security products.